Journal of Global Optimization (2019) 75:495–522
https://doi.org/10.1007/s10898-019-00777-y

Packing ellipses in an optimized convex polygon

A. Pankratov[1] · T. Romanova[1] · I. Litvinchev[2]

Received: 17 May 2018 / Accepted: 17 April 2019 / Published online: 30 April 2019

Abstract

Packing ellipses with arbitrary orientation into a convex polygonal container which has a given shape is considered. The objective is to find a minimum scaling (homothetic) coefficient for the polygon still containing a given collection of ellipses. New phi-functions and quasi phi-functions to describe non-overlapping and containment constraints are introduced. The packing problem is then stated as a continuous nonlinear programming problem. A solution approach is proposed combining a new starting point algorithm and a new modification of the LOFRT procedure (J Glob Optim 65(2):283–307, 2016) to search for locally optimal solutions. Computational results are provided to demonstrate the efficiency of our approach. The computational results are presented for new problem instances, as well as for instances presented in the recent paper (http://www.optimization-online.org/DB_FILE/2016/03/5348.pdf, 2016).

Keywords Packing · Ellipses · Continuous rotations · Convex polygon · Phi-function technique · Nonlinear optimization

1 Introduction

Packing ellipses in a container is a benchmark problem for two-dimensional packing. The problem is NP-hard [3] and has multiple applications in molecular dynamics (structure and properties of liquid crystals, the structure of liquids and glasses, chromosome packing in cell nucleus mineral, powder metallurgy, mineral industries), logistics (packing rolls of wallpaper, transportation of the pipes, transportation of paint buckets), cutting of industrial materials (mirror manufacturers, furniture making, 3D modeling of granular, structures and substances), robotics (see, e.g. [4–8] and the references therein).

Many authors tackled placement problems for ellipses. In [6] the following problem is considered: a set of ellipses with given axes has to be cut from a rectangular design plate, while minimizing the area of the design rectangle. The design plate is subject to lower and

✉ T. Romanova
tarom27@yahoo.com

[1] Department of Mathematical Modelling and Optimal Design, Institute for Mechanical Engineering Problems of the National Academy of Sciences of Ukraine, Kharkiv, Ukraine

[2] Faculty of Mechanical and Electrical Engineering, Nuevo Leon State University (UANL), San Nicolas de los Garza, Mexico

upper bounds of its width and length while orientations of ellipses are free. A mathematical programming formulation for this ellipse cutting problem is presented. Separating lines are used to derive non-overlapping conditions. For a small number of ellipses the authors report globally optimal solutions. However, for more than 14 ellipses none of the local or global NLP solvers available in GAMS can even compute a feasible point. So called polylithic approaches are proposed where ellipses are added sequentially in a strip-packing fashion to the rectangle restricted in width, but unrestricted in length. The rectangle's area is minimized in each step in a greedy fashion.

The problem of placing a given collection of ellipses into a rectangular container of minimum area is studied in [1]. Quasi phi-functions for an analytical description of non-overlapping and containment constraints for ellipses under continuous rotation and translation are introduced. A corresponding nonlinear programming problem is stated and an efficient solution algorithm is proposed. Computational results of new instances and instances that compare favorably with those published elsewhere before are considered in the paper.

In [7] continuous and differentiable nonlinear programming models and algorithms for packing multidimensional ellipsoids in a container (a cuboid and a sphere) of minimum area are considered. Two different models for non-overlapping and containment conditions are presented. A simple multi-start strategy is combined with an intelligent choice of starting points and a nonlinear programming local solver. The number of variables (constraints) for non-overlapping models grows quadratically with the number of ellipsoids to be packed. Numerical experiments are suggested for instances with up to 100 ellipsoids to obtain acceptable solutions in a reasonable time.

In [8] the authors present a nonlinear programming model for packing ellipses and ellipsoids that contains a linear number of variables and constraints. The proposed model finds its basis in a transformation-based non-overlapping model introduced in [7]. Numerical experiments in [8] show the efficiency and effectiveness of the proposed model and methodology for solving large-sized instances.

Regular grids are used in [9, 10] to approximate a container resulting in corresponding linear integer programming models for optimal packing. Packing different shapes including so-called circular-like objects (ellipses, rectangles, rhombuses, octagons, etc.) is considered as well as nesting objects inside one another (recursive packing). However, it is assumed that all objects have the same orientation and no rotation is allowed.

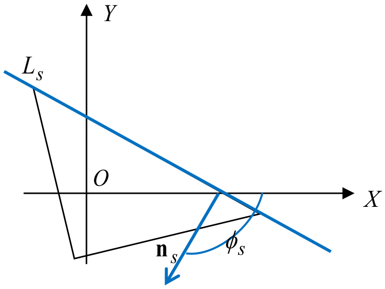
In the recent paper [2] the authors present a model and a numerical solution approach to packing ellipses into an optimized regular polygon. Specifically, the optimization strategy is based on the concept of embedded Lagrange multipliers. In this Lagrangian setting the apothem (and thereby the area) of a regular polygon is optimizing while preventing ellipses overlapping. They proceed simultaneously towards these objectives using the LGO solver system for global–local nonlinear optimization. The numerical results given in the paper demonstrate the applicability of the embedded Lagrange multipliers based modeling approach combined with global optimization to tackle a broad class of highly non-convex ellipse packing problems.

In this work a problem of packing a given collection of ellipses into a fixed shape convex polygonal container is considered. The ellipses can be continuously rotated, while the container can be homothetically scaled. The objective is to minimize the scaling coefficient for the container subject to non-overlapping and containment constraints.

The main contributions of this work are as follows:

- It is demonstrated that the original problem is closely related to packing homothetically scaling ellipses in a fixed container while maximizing the scaling coefficient for the ellipses.

Fig. 1 Illustration of the angle ϕ_s



- New tools for mathematical modelling of placement constraints are proposed: a quasi phi-function to describe non-overlapping of ellipses and a phi-function to describe the containment of an ellipse in a convex polygon analytically;
- An exact mathematical model for optimal packing ellipses under rotation is stated as a continuous nonlinear programming problem;
- An efficient algorithm for finding a good solution (local optimum) to the original packing problem is proposed. The method involves problem reformulation and employs two principal steps: a simple technique to find a starting point and a local optimization algorithm. The optimization algorithm results in a sequence of NLP subproblems of considerably smaller dimension.

2 Problem formulation

A set of ellipses $E_i, i \in \{1, 2, \ldots, n\} = I_n$, is given. Each ellipse E_i is defined by its semi-axes a_i and b_i.

The position of the ellipse E_i is characterized by the vector of variable object placement parameters $u_i = (v_i, \theta_i)$, where $v_i = (x_i, y_i)$ is a translation vector, and θ_i is a rotation angle. The center of the ellipse coincides with the origin of its own coordinate system. Rotated by angle θ_i and translated by vector v_i ellipse E_i is defined as $E_i(u_i) = \{p \in R^2 : p = v_i + M(\theta_i) \cdot p^0, \forall p^0 \in E_i^0\}$, where E_i^0 denotes non-translated and non-rotated ellipse E_i, $M(\theta_i)$ is a rotation matrix, $M(\theta_i) = \begin{pmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{pmatrix}$.

Let $\Omega(\lambda)$ be a convex m-polygonal domain given by its vertices $\{\lambda p_1, \lambda p_2, \ldots, \lambda p_m\}$ in the global coordinate system XOY. Here we define and use the variable homothetic coefficient $\lambda > 0, \lambda p_s = (\lambda x_s^p, \lambda y_s^p)$, $s = 1, 2, \ldots, m$, and it is assumed that the origin O is an interior point of $\Omega \equiv \Omega(\lambda)$.

In what follows we will also use the presentation of Ω in the form $\Omega = \bigcap_{s=1}^m P_s(\lambda)$, where $P_s(\lambda) = \{(x, y) : \mu_s(x, y, \lambda) = \cos\phi_s \cdot x + \sin\phi_s \cdot y + \lambda\gamma_s \geq 0\}$ is a half plane. Here the equation $\mu_s(x, y, \lambda) = 0$ corresponds to a straight line $L_s(\lambda) = \{(x, y) : \mu_s(x, y, \lambda) = 0\}$ passing through the *s-th* side of $\Omega(\lambda)$; ϕ_s is the angle between axis *OX* and the normal $\mathbf{n}_s$ to the straight line $L_s(\lambda)$ (Fig. 1), $\varphi_s = const$, $\gamma_s = const$, $s = 1, 2, \ldots, m$, and it is assumed that $\mu_s(O, \lambda) > 0$.

In what follows two related problems are considered: (1) packing the set of ellipses in a homothetically scaled convex polygonal container minimizing the scaling coefficient for the container; (2) packing a set of homothetically scaled ellipses in a fixed polygonal container maximizing the scaling coefficient for the ellipses. More specifically, these two problems are as follows.

Ellipse packing problem EPP_1. Pack the set of ellipses $E_i(u_i)$, $i \in I_n$ in a convex polygonal domain $\lambda\Omega$ minimizing the homothetic coefficient $\lambda \geq 0$.

In ***EPP_1*** the following ***placement constraints*** have to be satisfied:

- Non-overlapping constraints: $\operatorname{int} E_i(u_i) \cap \operatorname{int} E_j(u_j) = \emptyset$, for $i > j \in I_n$,
- Containment constraints: $E_i(u_i) \subset \Omega(\lambda)$ for each $i \in I_n$.

Let each ellipse $E_i(u_i)$ be defined by its semi-axes $\beta \cdot a_i$ and $\beta \cdot b_i$, where $\beta \geq 0$ is a homothetic coefficient of the ellipse $E_i(u_i)$, $i \in I_n$. In what follows the notation $\widehat{E}_i(u_i, \beta) = \beta \cdot E_i(u_i)$, $i \in I_n$ is used.

Ellipse packing problem EPP_2. Place the set of ellipses $\widehat{E}_i(u_i, \beta)$,$i \in I_n$ into a fixed convex polygonal domain $\Omega(\lambda = 1)$ maximizing the ellipse homothetic coefficient $\beta \geq 0$.

In ***EPP_2*** the following ***placement constraints*** have to be satisfied:

- Non-overlapping constraints: $\operatorname{int} \widehat{E}_i(u_i, \beta) \cap \operatorname{int} \widehat{E}_j(u_j, \beta) = \emptyset$, for $i > j \in I_n$,
- Containment constraints: $\widehat{E}_i(u_i, \beta) \subset \Omega$ for each $i \in I_n$.

It will be demonstrated in Sect. 5 that although these two problems are closely related, the second is more attractive from the computational point of view and will be used further in algorithmic constructions.

3 Mathematical modeling tools

In this study we use the phi-function technique which provides a powerful tool for mathematical modeling of placement constraints in the field of Packing& Cutting (see, e.g. [11–14]).

We introduce a new phi-function to describe containment constraints and a new quasi phi-function to describe non-overlapping constraints in the ellipse packing problem. These new tools allow us reducing considerably the number of auxiliary variables and using simpler mathematical model than in [1] proposed for the ellipse packing problem in a rectangular container.

For the reader's convenience we provide here definitions of a phi-function and a quasi phi-function (see [2, 11] for more details).

Let $A \subset R^2$ and $B \subset R^2$ be two objects. The position of object A is defined by a vector of *placement parameters* $u_A = (v_A, \theta_A)$, where: $v_A = (x_A, y_A)$ is a translation vector and θ_A is a rotation angle. The object A, rotated by angle θ_A and translated by vector v_A will be denoted by $A(u_A)$.

Phi-functions allow us to distinguish the following three cases: $A(u_A)$ and $B(u_B)$ are intersecting so that $A(u_A)$ and $B(u_B)$ have common interior points; $A(u_A)$ and $B(u_B)$ do not intersect, i.e. $A(u_A)$ and $B(u_B)$ do not have common points; $A(u_A)$ and $B(u_B)$ are in contact, i.e. $A(u_A)$ and $B(u_B)$ have only common frontier points. We denote by $frA(\cdot)$ the frontier of set $(\cdot)$.

Definition [11]. A continuous and everywhere defined function $\Phi^{AB}(u_A, u_B)$ is called a phi-function for objects $A(u_A)$ and $B(u_B)$ if

$$\Phi^{AB}(u_A, u_B) < 0, \quad \text{if int } A(u_A) \cap \text{int } B(u_B) \neq \emptyset;$$

$$\Phi^{AB}(u_A, u_B) = 0, \quad \text{if int } A(u_A) \cap \text{int } B(u_B) = \emptyset \text{ and } frA(u_A) \cap frB(u_B) \neq \emptyset;$$

$$\Phi^{AB}(u_A, u_B) > 0, \quad \text{if } A(u_A) \cap B(u_B) = \emptyset.$$

It should be noted that inequality $\Phi^{AB}(u_A, u_B) \geq 0$ provides the non-overlapping constraint, i.e., int $A(u_A) \cap$ int $B(u_B) = \emptyset$, and inequality $\Phi^{AB^*}(u_A, u_B) \geq 0$ provides the containment constraint $A(u_A) \subset B(u_B)$, i.e. int $A(u_A) \cap$ int $B^*(u_B) = \emptyset$, where $B^* = R^2 \backslash \text{int } B$.

Definition [1]. A continuous and everywhere defined function $\Phi'^{AB}(u_A, u_B, u')$ is called a quasi phi-function for two objects $A(u_A)$ and $B(u_B)$ if $\max_{u'} \Phi'^{AB}(u_A, u_B, u')$ is a phi-function for the objects.

Here u' is a vector of auxiliary continuous variables that belongs to Euclidean space.

Based on features of a quasi phi-function [1] the non-overlapping constraint can be describe in the form:

$$\text{if } \Phi'^{AB}\left(u_A, u_B, u'\right) \geq 0 \text{ for some } u', \text{ then int } A(u_A) \cap \text{int } B(u_B) = \emptyset.$$

3.1 Phi-function for containment constraints

Let us construct a phi-function for containment constraints for the problem EPP_1:

$$E_i \subset \Omega(\lambda) \Leftrightarrow \text{int } E_i(u_i) \cap \Omega^*(\lambda) = \emptyset, \text{ where } \Omega^*(\lambda) = R^2 \backslash int\ \Omega(\lambda).$$

Proposition 1 *A function defined by*

$$\Phi^{E_i \Omega^*}(u_i, \lambda) = \min\left\{\Phi^*_{is}(u_i, \lambda), s \in I_m\right\}, \tag{1}$$

is a phi-function of $E_i(u_i)$ *and* $\Omega^*(\lambda)$, *where* $\Phi^*_{is}(u_i, \lambda)$ *is a phi-function of ellipse* $E_i(u_i)$ *and half plane* $P^*_s(\lambda)$, $P^*_s(\lambda) = R^2 \backslash \text{int } P_s(\lambda)$.

Proof It is clear that ellipse $E_i(u_i)$ is arranged inside a convex polygon $\Omega(\lambda)$ if and only if ellipse $E_i(u_i)$ does not overlap with each half plane $P^*_s(\lambda)$, i.e. $E_i(u_i) \cap$ int $P^*_s(\lambda) = \emptyset$, for each $s \in I_m$. In terms of phi-functions it means that a phi-function of $E_i(u_i)$ and half plane $P^*_s(\lambda)$ should take nonnegative value for all $s \in I_m$. Q.E.D.

Therefore, according to (1), the inequality $\Phi^{E_i \Omega^*}(u_i, \lambda) \geq 0$ provides the containment constraint $E_i(u_i) \subset \Omega$, i.e. int $E_i(u_i) \cap$ int $\Omega^*(\lambda) = \emptyset$.

Now we define a phi-function $\Phi^*_{is}(u_i, \lambda)$ of ellipse $E_i(u_i)$ and half plane $P^*_s(\lambda)$.

Proposition 2 *A phi-function of* $E_i(u_i)$ *and half plane* $P^*_s(\lambda)$ *has the form*

$$\Phi^*_{is}(u_i, \lambda) = \delta_{is}(v_i, \lambda) - d_{is}(\theta_i), \tag{2}$$

where

$$\delta_{is}(v_i, \lambda) = x_i cos\phi_s + y_i \sin\phi_s + \lambda\gamma_s, d_{is}(\theta_i) = \sqrt{\Delta_i \cdot cos^2(\theta_i + \phi_s) + b_i^2}, \quad \Delta_i = \left(a_i^2 - b_i^2\right) \cdot$$

Proof A deviation of v_i to $L_s(\lambda)$ is derived as $\delta_{is}(v_i, \lambda) = x_i cos\phi_s + y_i \sin\phi_s + \lambda\gamma_s$ (Fig. 2a).

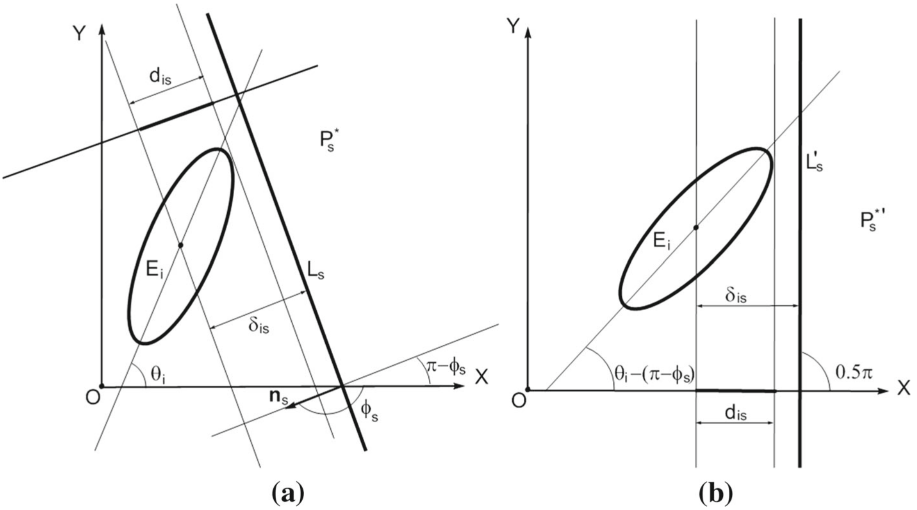


Fig. 2 Illustration for phi-function Φ^*_{is}: **a** the given position of ellipse $E_i(v_i, \theta_i)$ and $L_s(\lambda)$; **b** the position of the rotated ellipse E_i and the rotated straight line $L'_s(\lambda)$

Now let us get the expression for the semi length $d_{is}(\theta_i)$ of the orthogonal projection of $E_i(v_i, \theta_i)$ on the straight line $L_s^{\perp}(\lambda)$ that is perpendicular to $L_s(\lambda)$ (Fig. 2a). To this aim we rotate ellipse $E_i(v_i, \theta_i)$ and straight line $L_s(\lambda)$ by angle $\pi - \phi_s$. As a result we obtain $L'_s(\lambda) \perp OX$ (Fig. 2b).

Now, based on the idea proposed in [6], we can derive the semi length of the projection of $E_i(v'_i, \theta'_i)$ on *OX* in the form $d_{is}(\theta_i) = \sqrt{\Delta_i \cdot cos^2(\theta_i + \phi_s) + b_i^2}$.

Function (2) is everywhere defined and continuous. It addition, since $d_{is}(\theta_i)$ is always positive, then either $\delta_{is}(v_i, \lambda) > d_{is}(\theta_i)$ or $\delta_{is}(v_i, \lambda) = d_{is}(\theta_i)$, or $\delta_{is}(v_i, \lambda) < d_{is}(\theta_i)$. Thus: (1) $\Phi^*_{is}(u_i, \lambda) > 0$ if and only if $\delta_{is}(v_i, \lambda) - d_{is}(\theta_i) > 0$ (it means that $E_i(u_i) \cap P_s^*(\lambda) = \emptyset$); (2) $\Phi^*_{is}(u_i, \lambda) = 0$ if and only if $\delta_{is}(v_i, \lambda) - d_{is}(\theta_i) = 0$ (it means that int $E_i(u_i) \cap P_s^*(\lambda) = \emptyset$ and $frE_i(u_i) \cap L_s(\lambda) \neq \emptyset$); (3) $\Phi^*_{is}(u_i, \lambda) < 0$ if and only if $\delta_{is}(v_i, \lambda) - d_{is}(\theta_i) < 0$ (it means that int $E_i(u_i) \cap P_s^*(\lambda) \neq \emptyset$). Therefore function (1) is a phi-function of the ellipse $E_i(u_i)$ and the half plane $P_s^*(\lambda)$. Q.E.D.

The inequality $\Phi^*_{is}(u_i, \lambda) \geq 0$ guarantees the non-overlapping of $E_i(u_i)$ and $P_s^*(\lambda)$.

Now let us construct a phi-function for containment constraints for the problem EPP_2:

Based on (2) and assuming $\lambda = 1$ we define a phi-function for containment of $\widehat{E}_i(u_i, \beta)$ into Ω that we present as follows:

$$\Phi^{\widehat{E}_i \Omega^*}(u_i, \beta) = \min\left\{\Phi^*_{is}(u_i, \beta), s \in I_m\right\}, \tag{3}$$

$$\Phi^*_{is}(u_i, \beta) = x_i \cdot \cos\phi_s + y_i \cdot \sin\phi_s + \gamma_s - \beta\sqrt{\Delta_i cos^2(\theta_i + \phi_k) + b_i^2}. \tag{4}$$

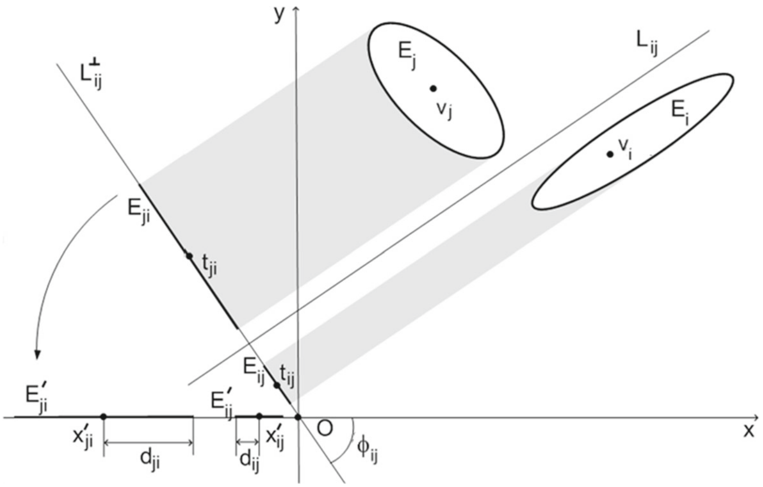


Fig. 3 Illustration for the construction of quasi-phi function $\Phi'^{E_i E_j}$

3.2 Quasi-phi-function for non-overlapping constraints

Next, we construct a quasi phi-function of two ellipses $E_i(u_i)$ and $E_j(u_j)$ for the problem **EPP_1,** that is considerably simpler than a quasi phi-function considered in [2].

The key idea is based on the following statement: if two ellipses do not intersect each other, then there exists a straight line, passing through the center of the coordinate system XOY, such that the projections of the ellipses on this line do not intersect each other. In fact, let two ellipses have no common interior points. Then, there exists a line that divides the plane into two half-planes in such a way that the two ellipses lie in different half-planes. Consequently, the projections of the ellipses on any straight line, that is perpendicular to the separating line (in particular, passing through the center of the coordinate system), do not have common interior points in R^1.

We denote the straight line, that is perpendicular to a separating line L_{ij} and passing through the origin O, by $L_{ij}^{\perp}$. Let ϕ_{ij} be the angle between the line $L_{ij}^{\perp}$ and axis OX, while E_{ij} and E_{ji} be projections (line segments with the appropriate centers $t_{ij} = (x_{ij}, y_{ij})$ and $t_{ji} = (x_{ji}, y_{ji})$) of ellipses $E_i(u_i)$ and $E_j(u_j)$ (Fig. 3).

Rotating the straight line $L_{ij}^{\perp}$ around the origin O by the angle $-\phi_{ij}$, we obtain two segments $E'_{ij} \subset OX$ and $E'_{ji} \subset OX$ with centers $x'_{ij}(v_i, \phi_{ij})$ and $x'_{ji}(v_j, \phi_{ij})$ (Fig. 3) and semi lengths $d_{ij}(\theta_i, \phi_{ij})$ and $d_{ji}(\theta_i, \phi_{ij})$, where

$$x'_{ij}\left(v_i, \phi_{ij}\right) = x_i cos\phi_{ij} - y_i sin\phi_{ij}, x'_{ji}\left(v_j, \phi_{ij}\right) = x_j cos\phi_{ij} - y_j sin\phi_{ij}, \tag{5}$$

$$d_{ij}\left(\theta_i, \phi_{ij}\right) = \sqrt{b_i^2 + \Delta_i cos^2\left(\theta_i - \phi_{ij}\right)}, d_{ji}\left(\theta_j, \phi_{ij}\right) = \sqrt{b_j^2 + \Delta_j cos^2\left(\theta_j - \phi_{ij}\right)}, \tag{6}$$

$$\Delta_i = \left(a_i^2 - b_i^2\right), \Delta_j = \left(a_j^2 - b_j^2\right).$$

Proposition 3 *A function defined by*

$$\Phi'^{E_i E_j}\left(u_i, u_j, \phi_{ij}\right) = x'_{ij}\left(v_i, \phi_{ij}\right) - x'_{ji}\left(v_j, \phi_{ij}\right) - \left(d_{ij}\left(\theta_i, \phi_{ij}\right) + d_{ji}\left(\theta_j, \phi_{ij}\right)\right) \tag{7}$$

is a quasi phi-function of ellipses $E_i(u_i)$ and $E_j(u_j)$.

Proof To show that function $\Phi'^{E_i E_j}(u_i, u_j, \phi_{ij})$ is a quasi phi-function of ellipses $E_i(u_i)$ and $E_j(u_j)$ we need to prove that $\max_{\phi_{ij}\in[0,2\pi]} \Phi'^{E_i E_j}(u_i, u_j, \phi_{ij})$ is a phi-function for ellipses $E_i(u_i)$ and $E_j(u_j)$ (see definitions of a phi-function and a quasi phi-function at the beginning of this section).

Further we show that the function possesses the following characteristics:

(1) $\max_{\phi_{ij}\in[0,2\pi]} \Phi'^{E_i E_j}(u_i, u_j, \phi_{ij}) > 0$ if $E_i(u_i) \cap E_j(u_j) = \emptyset$;
(2) $\max_{\phi_{ij}\in[0,2\pi]} \Phi'^{E_i E_j}(u_i, u_j, \phi_{ij}) = 0$ if $\text{int } E_i(u_i) \cap \text{int } E_j(u_j) = \emptyset$ and $frE_i(u_i) \cap frE_j(u_j) \neq \emptyset$;
(3) $\max_{\phi_{ij}\in[0,2\pi]} \Phi'^{E_i E_j}(u_i, u_j, \phi_{ij}) < 0$ if $\text{int } E_i(u_i) \cap \text{int } E_j(u_j) \neq \emptyset$.

First we show that the following equation

$$\max_{\phi_{ij}\ \in[0,2\pi]} \Phi'^{E_i E_j}\left(u_i, u_j, \phi_{ij}\right)$$
$$= \max_{\phi_{ij}\in[0,\pi]} \left(|x'_{ij}\left(v_i, \phi_{ij}\right) - x'_{ji}\left(v_j, \phi_{ij}\right)| - \left(d_{ij}\left(\theta_i, \phi_{ij}\right) + d_{ji}\left(\theta_j, \phi_{ij}\right)\right)\right). \tag{8}$$

is valid.

Based on (5) we have

$$x'_{ij}\left(v_i, \phi_{ij} + \pi\right) - x'_{ji}\left(v_j, \phi_{ij} + \pi\right) = x'_{ij}\left(v_i, \phi_{ij}\right) - x'_{ji}\left(v_j, \phi_{ij}\right), \text{ since}$$

$$x'_{ij}\left(v_i, \phi_{ij} + \pi\right) = x_i cos\left(\phi_{ij} + \pi\right) - y_i sin\left(\phi_{ij} + \pi\right)$$
$$= -x_i cos\phi_{ij} + y_i sin\phi_{ij} = -x'_{ij}\left(v_i, \phi_{ij}\right).$$

Similarly we get

$$x'_{ji}\left(v_j, \phi_{ij} + \pi\right) = -x_j cos\phi_{ij} + y_j sin\phi_{ij} = -x'_{ji}\left(v_j, \phi_{ij}\right).$$

Based on (6) we have

$$d_{ij}\left(\theta_i, \phi_{ij} + \pi\right) + d_{ji}\left(\theta_j, \phi_{ij} + \pi\right) = d_{ij}\left(\theta_i, \phi_{ij}\right) + d_{ji}\left(\theta_j, \phi_{ij}\right)$$

Then we consider three cases.

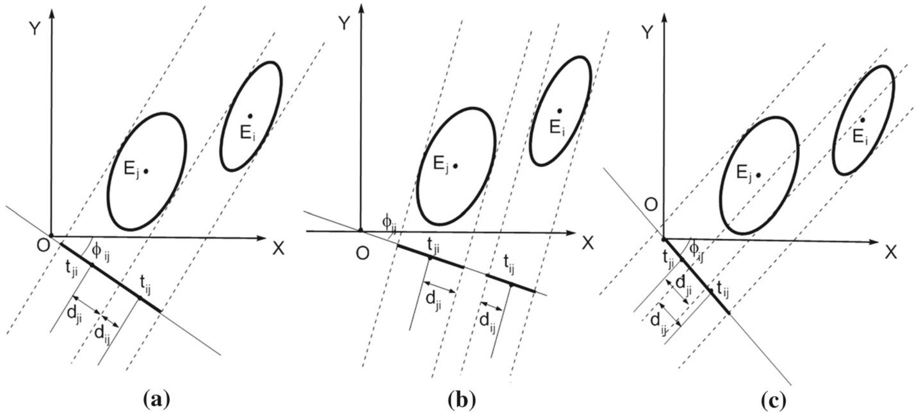
1. Let $E_i(u_i) \cap E_j(u_j) = \emptyset$. It is known that if two ellipses $E_i(u_i)$ and $E_j(u_j)$ do not overlap each other then there exists a straight line such that orthogonal projections E_{ij} and E_{ji} of the ellipses on the straight line have no common interior points. It means that there exists a separating line L_{ij} and therefore the straight line $L_{ij}^{\perp}$, that is perpendicular to L_{ij} and passing through the origin O with the angle ϕ_{ij} between the line $L_{ij}^{\perp}$ and axis OX (Fig. 4a).

Based on (8) the non-overlapping constraint, $E_i(u_i) \cap E_j(u_j) = \emptyset$, can be described by the following inequality:

$$\left|x'_{ij}\left(v_i, \phi_{ij}\right) - x'_{ji}\left(v_j, \phi_{ij}\right)\right| > d_{ij}\left(\theta_i, \phi_{ij}\right) + d_{ji}\left(\theta_j, \phi_{ij}\right).$$

Therefore $\max_{\phi_{ij}\in[0,2\pi]} \Phi'^{E_i E_j}(u_i, u_j, \phi_{ij}) > 0$.

It should be noted that for some values of the angle ϕ_{ij} quasi phi-function (7) can take zero value (Fig. 4b) or be negative (Fig. 4c).



(a) **(b)** **(c)**

Fig. 4 Illustration to case 1: **a** $\Phi'^{E_iE_j} = 0$, **b** $\Phi'^{E_iE_j} > 0$, **c** $\Phi'^{E_iE_j} < 0$

2. Let int $E_i(u_i) \cap$ int $E_j(u_j) = \emptyset$ and $frE_i(u_i) \cap frE_j(u_j) \neq \emptyset$. Then either (a)int $E'_{ij}(x_{ij}, 0) \cap$ int $E'_{ji}(x_{ji}, 0) \neq \emptyset$ or (b) int $E'_{ij}(x_{ij}, 0) \cap$ int $E'_{ji}(x_{ji}, 0) = \emptyset$ and $frE'_{ij}(x_{ij}, 0) \cap frE'_{ji}(x_{ji}, 0) \neq \emptyset$. Therefore, $\left|x'_{ij}(v_i, \phi_{ij}) - x'_{ji}(v_j, \phi_{ij})\right| < d_{ij}(\theta_i, \phi_{ij}) + d_{ji}(\theta_j, \phi_{ij})$ and $\Phi'^{E_iE_j}(u_i, u_j, \phi_{ij}) < 0$ for case (a) or $\left|x'_{ij}(v_i, \phi_{ij}) - x'_{ji}(v_j, \phi_{ij})\right| = d_{ij}(\theta_i, \phi_{ij}) + d_{ji}(\theta_j, \phi_{ij})$ and $\Phi'^{E_iE_j}(u_i, u_j, \phi_{ij}) = 0$ for case (b). Thus, there is always exists ϕ_{ij} with $\max_{\phi_{ij}\in[0,2\pi]} \Phi'^{E_iE_j}(u_i, u_j, \phi_{ij}) = 0$.
3. Let int $E_i(u_i) \cap$ int $E_j(u_j) \neq \emptyset$. Then also int $E'_{ij}(x_{ij}, 0) \cap$ int $E'_{ji}(x_{ji}, 0) \neq \emptyset$. It means that $\left|x'_{ij}(v_i, \phi_{ij}) - x'_{ji}(v_j, \phi_{ij})\right| < d_{ij}(\theta_i, \phi_{ij}) + d_{ji}(\theta_j, \phi_{ij})$ for any ϕ_{ij}, i.e. $\max_{\phi_{ij}\in[0,2\pi]} \Phi'^{E_iE_j}\left(u_i, u_j, \phi_{ij}\right) < 0$. Q.E.D.

Thus, according to the main property of a quasi phi-function [1], we can conclude that, if $\Phi'^{E_iE_j}(u_i, u_j, \phi_{ij}) \geq 0$ for some ϕ_{ij}, then int $E_i(u_i) \cap$ int $E_j(u_j) = \emptyset$.

Now based on (7), we define a quasi phi-function to describe non-overlapping of $\widehat{E}_i(u_i, \beta)$ and $\widehat{E}_j(u_j, \beta)$ in problem **EPP_2**

$$\Phi'^{\widehat{E}_i\widehat{E}_j}\left(u_i, u_j, \phi_{ij}, \beta\right) = \left(x_i - x_j\right)cos\phi_{ij} + \left(y_j - y_i\right)sin\phi_{ij} - \beta\left(\sqrt{b_i^2 + \Delta_i cos^2\left(\theta_i - \phi_{ij}\right)} + \sqrt{b_j^2 + \Delta_j cos^2\left(\theta_j - \phi_{ij}\right)}\right), \tag{9}$$

4 Mathematical models for ellipse packings

Let us state a mathematical model of the problem ***EPP_1.*** A mathematical model for packing rotatable ellipses in a convex polygonal domain Ω of minimum homothetic coefficient can be formulated in the form

$$\min_{(t,\phi)\in V\subset R^{\sigma}} \lambda, \tag{10}$$

$V=\{(t,\phi)\in R^{\sigma}:\Phi'^{E_iE_j}(u_i,u_j,\phi_{ij})\geq 0,(i,j)\in\Xi,\Phi^{E_i\Omega^*}(u_i,\lambda)\geq 0,i\in I_n,\lambda\geq 0\}$, where $t=(v,\theta,\lambda)$ is a vector of variables, $v=(v_1,v_2,\ldots,v_n)$, $\theta=(\theta_1,\theta_2,\ldots,\theta_n)$, $\phi=(\phi_{ij},(i,j)\in\Xi)$, $\Xi=\{(i,j):i<j\in I_n\}$, $\Phi'^{E_iE_j}(u_i,u_j,\phi_{ij})$ is defined in (7), $\Phi^{E_i\Omega^*}(u_i,\lambda)$ is defined in (1), $u_i=(v_i,\theta_i)$, $v_i=(x_i,y_i)$, $\sigma=3n+0.5\cdot n(n-1)+1$ is the number of the problem variables.

Based on (1) the inequality $\Phi^{E_i\Omega^*}(u_i,\lambda)\geq 0$ can be substituted by m inequalities $\Phi^*_{is}(u_i,\lambda)\geq 0, s\in I_m$, where $\Phi^*_{is}(u_i,\lambda)$ is defined in (2). Therefore we can rewrite the feasible region V in the form

$$V=\left\{(t,\phi)\in R^{\sigma}:\Phi'^{E_iE_j}\left(u_i,u_j,\phi_{ij}\right)\geq 0,\quad (i,j)\in\Xi,\Phi^*_{is}(u_i,\lambda)\geq 0,s\in I_m,i\in I_n,\lambda\geq 0\right\}, \tag{11}$$

The number of inequalities that describes feasible region (11) is $N=0.5n(n-1)+nm$.

Feasible region V given by (11) is defined by a system of inequalities with differentiable functions. Our model (10)–(11) is a non-convex and continuous nonlinear programming problem. This is an exact formulation in the sense that it gives all optimal solutions to the ellipse packing problem. It is possible, at least in theory, using a global solver for this nonlinear programming problem to get an optimal packing.

However in practice, the model has a large number of variables and inequalities. The model (10)–(11) involves O(n^2) nonlinear inequalities and O(n^2) variables due to the auxiliary variables in quasi phi-functions. As a result, even finding a locally optimal solution becomes a complicated procedure for the available state of the art NLP-solvers employed directly to model (10)–(11).

Therefore developing an efficient algorithm combining a fast and simple (feasible) starting point selection technique and local optimization procedure (linear to the number of ellipses) is of very important.

Then we state a mathematical model of ***EPP_2*** in the form:

$$\max_{(u,\phi)\in W\subset R^{\sigma}}\beta, \tag{12}$$

$$W=\left\{(u,\phi)\in R^{\sigma}:\Phi'^{\widehat{E}_i\widehat{E}_j}\left(u_i,u_j,\beta,\phi_{ij}\right)\geq 0,\quad (i,j)\in\Xi,\Phi^*_{is}(u_i,\beta)\geq 0,s\in I_m,i\in I_n,\beta\geq 0\right\}, \tag{13}$$

where, $u=(v,\theta,\beta)$, $v=(v_1,v_2,\ldots,v_n)$, $\theta=(\theta_1,\theta_2,\ldots,\theta_n)$,$\phi=(\phi_{ij},(i,j)\in\Xi)$,$\Xi=\{(i,j):i<j\in I_n\}$, $u_i=(v_i,\theta_i)$, $v_i=(x_i,y_i)$,$\Phi'^{\widehat{E}_i\widehat{E}_j}(u_i,u_j,\beta,\phi_{ij})$ is defined in (9), $\Phi^*_{is}(u_i,\beta)$ is defined in (4), $\sigma=3n+0.5\cdot n(n-1)+1$ is the number of the problem variables. The number of inequalities that describe the feasible region (13) is $N=0.5n(n-1)+nm$.

It is not hard to verify the following relation between the models (10)–(11) and (12)–(13). The problem (10)–(11) is equivalent to the problem (12)–(13) for $\beta>0$ in the sense that there exists the following one-to-one correspondence between the appropriate feasible solutions of these problems:

$$\left(\frac{v}{\beta},\theta,\lambda=\frac{1}{\beta},\phi\right)\leftrightarrow(v,\theta,\beta,\phi),\quad\forall(t,\phi)\in V,\forall(u,\phi)\in W,\beta>0. \tag{14}$$

Further we refer to (14) as a scaling transformation.

To find "good" locally optimal solutions of the problem (10)–(11) within a reasonable computational time we propose in Sect. 5 an efficient solution algorithm for the problem (12)–(13). We focus on the problem (12)–(13) and provide a fast and simple procedure to generate a good initial feasible solution. In most cases the approach reduces our problem with O(n^2) variables and nonlinear constraints to a sequence of nonlinear programming subproblems of considerably smaller dimension, with O(n) variables and nonlinear constraints.

5 Solution algorithm

Our solution strategy consists of four stages:

Stage 1. Generate a set of vectors of feasible placement parameters of ellipses in problem (12)–(13) provided that $\beta = 0$ (see Sect. 5.1).
Stage 2. Search for a set of local maxima of problem (12)–(13) starting from each starting point obtained at Stage 1 (see Sect. 5.2).
Stage 3. Derive local minima of problem (10)–(11) based on local maxima found at Stage 2, using scaling transformation (14).
Stage 4. Choose the best local minimum from those found at Stage 3.

We note that this is a heuristic local solution strategy which produces good results as shown below.

5.1 Starting feasible parameters algorithm (SFP)

Let us consider problem (12)–(13). Let Ω given by its vertices $p_s = (x_s^p, y_s^p), s \in I_m$. We assume $\beta = 0$.

Step 1. Generate within Ω a set of n randomly chosen centers $v_i^1 = (x_i^1, y_i^1)$, of ellipses E_i,$i \in I_n$, using the formula $v_i^0 = \sum_{s=1}^m \alpha_{is} p_s$,$\sum_{s=1}^m \alpha_{is} = 1, 0 \le \alpha_{is} \le 1, s \in I_m$. To find $\alpha_{is}, s \in I_m$ we randomly generate m positive numbers $n_{is}, s \in I_m$, and derive $\alpha_{is} = \frac{n_{is}}{\sum_{s=1}^m n_{is}}, s \in I_m$.
Step 2. Generate random rotation parameters $\theta_i^1 \in [0, \pi)$ of ellipses E_i,$i \in I_n$ (Fig. 5a).
Step 3. Form a vector $u^1 = (v^1, \theta^1, \beta^1 = 0)$.
Step 4. Form a vector ϕ^1 of auxiliary variables based on the elementary geometrical calculations. Each element of the vector ϕ^1 is defined as an angle ϕ_{ij}^1 between axis OX and a straight line paralleled to $L_{ij}^{\perp}$ and passing through points v_i^1 and v_j^1 (Fig. 5b).
Step 5. Return a starting point (u^1, ϕ^1) for a subsequent search for a local maximum of the problem (12)–(13).

Note that the starting feasible point obtained by the algorithm for the problem (12)–(13) can not be transformed to a feasible point of the problem (10)–(11) using relation (14) for $\beta = 0$. This is the reason for considering the problem EPP_2 instead of the problem EPP_1.

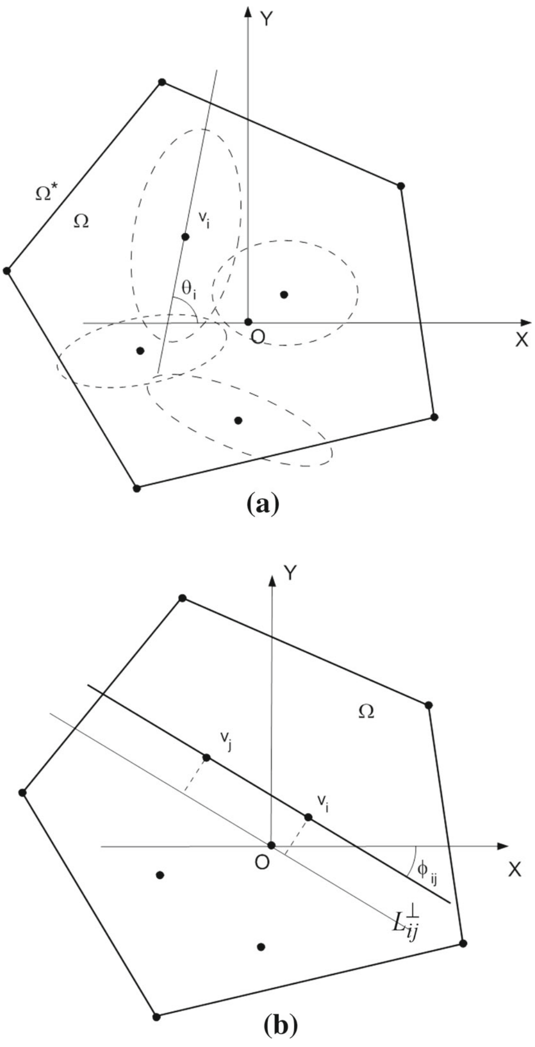
5.2 Local optimization algorithm (LOFRT_P)

We propose a decomposition algorithm (LOFRT_P) to search for local maxima of the problem (12)–(13) with O(n^2) variables and O(n^2) nonlinear constraints.

The iterative algorithm reduces the large scale problem to a sequence of nonlinear programming subproblems of smaller dimension (O(n) variables and O(n) nonlinear constraints). It is based on a modification of the LOFRT procedure introduced in [1] for the packing problem of ellipses of given sizes in a rectangular domain of minimum area.

The key idea of the LOFRT_P algorithm is as follows. At each iteration a fixed individual rectangular ε-container centered at the feasible staring point is constructed for each ellipse.

Fig. 5 Illustrations to the ***SFP*** algorithm: **a** Steps 1–3; **b** Step 4



Then each ellipse can be moved within its individual ε-container. The motion of each ellipse is described by a system of four ε-inequalities. Then a subregion of the feasible region W is formed in the following way: we add $4n$ ε-inequalities to constraints (13) and then delete O(n^2) phi-inequalities corresponding to pairs of ellipses with non-overlapping individual containers. Some redundant containment constraints are also deleted.

While deleting quasi phi-functions for some pairs of ellipses we also delete O(n^2) corresponding auxiliary variables. This results in reducing the number of variables in our subproblem. Then we search for a local maximum for the subproblem with O(n) variables and nonlinear constraints. This local maximum is then used as a starting point for the next iteration. On the last iteration of our algorithm we find a local maximum of the problem (12)–(13).

The LOFRT_P algorithm comparing to the LOFRT procedure deals with an arbitrary convex polygon (vs. a rectangular container in LOFRT); does not use circular approximation of

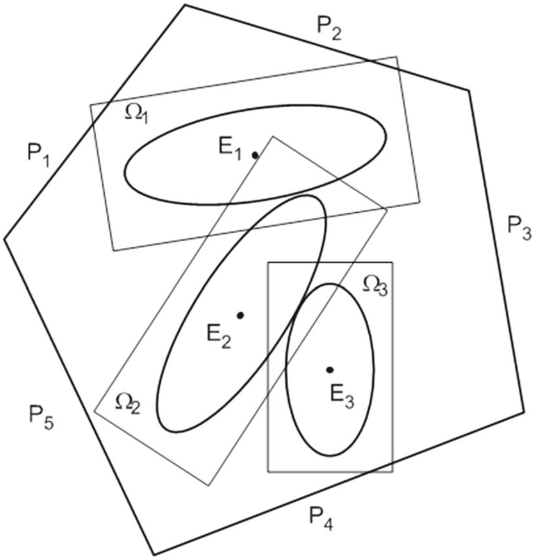


Fig. 6 Illustration to construction of individual containers of ellipses for Steps 2–6

ellipses to generate ε-inequalities (vs. circumscribed circles around ellipses of given sizes); takes into account variable scaling parameters of ellipses while generating individual containers (vs. given sizes of ellipses); orientation of the individual container is associated with the ellipse orientation (vs. the fixed orientation of the individual container); maximizes the scaling parameter for ellipses (vs. minimizing area of the rectangular container).

Let us consider the algorithm in details. Let u^1 be one of the points found by the ***SFP*** algorithm. Now we describe the LOFRT_P algorithm which is an iterative decomposition procedure.

We denote the value of the decomposition step of the algorithm by ε and assume that $\varepsilon = \frac{\gamma}{2n} \cdot \sum_{i=1}^{n} b_i$, where $\gamma = \frac{S_\Omega}{\sum_{i=1}^{n} S_i}$, S_Ω is the area of container $\Omega(\lambda = 1), S_i = \pi \cdot a_i \cdot b_i$ is the area of ellipse E_i.

Step 1. Let $k = 1$.

Step 2. Construct a fixed “individual” rectangular container $\Omega_i^k \supset \widehat{E}_i(u_i^k, \beta^k)$ of sizes $2(\beta^k \cdot a_i + \varepsilon)$ and $2(\beta^k \cdot b_i + \varepsilon)$ with the center point v_i^k for each $i \in I_n$ (Fig. 6).

Step 3. Create a system of auxiliary “artificial” inequality constraints on the vector (u_i, β) of each ellipse E_i in the form: $\Phi^{\widehat{E}_i \Omega_i^{k*}}(u_i, \beta) \geq 0$, $i \in I_n$, where $\Phi^{\widehat{E}_i \Omega_i^{k*}}(u_i, \beta)$ is defined based on (3)

$$\Phi^{\widehat{E}_i \Omega_i^{k*}}(u_i, \beta) = \min\left\{ f_{i1}^k(u_i, \beta),\ f_{i2}^k(u_i, \beta),\ f_{i3}^k(u_i, \beta),\ f_{i4}^k(u_i, \beta) \right\}.$$

The inequality $\Phi^{\widehat{E}_i \Omega_i^{k*}}(u_i, \beta) \geq 0$ is equivalent to the system of four inequalities

$$f_{i1}^k(u_i, \beta) \geq 0,\ f_{i2}^k(u_i, \beta) \geq 0,\ f_{i3}^k(u_i, \beta) \geq 0,\ f_{i4}^k(u_i, \beta) \geq 0,$$

where

$$f_{i1}^k(u_i, \beta) = \left(x_i - x_i^k\right) \cdot \cos\theta_i^k + \left(y_i - y_i^k\right) \cdot \sin\theta_i^k + \beta^k \cdot a_i + \varepsilon - \beta\sqrt{\Delta_i cos^2\left(\theta_i + \theta_i^k\right) + b_i^2},$$

$$f_{i2}^k(u_i, \beta) = -\left(x_i - x_i^k\right) \cdot \cos\theta_i^k - \left(y_i - y_i^k\right) \cdot \sin\theta_i^k + \beta^k \cdot a_i + \varepsilon - \beta\sqrt{\Delta_i cos^2\left(\theta_i + \theta_i^k\right) + b_i^2},$$

$f_{i3}^k(u_i,\beta) = (x_i - x_i^k)\cdot\sin\theta_i^k - (y_i - y_i^k)\cdot\cos\theta_i^k + \beta^k\cdot b_i + \varepsilon - \beta\sqrt{\Delta_i\sin^2(\theta_i+\theta_i^k)+b_i^2}$, $f_{i4}^k(u_i,\beta) = -(x_i - x_i^k)\cdot\sin\theta_i^k + (y_i - y_i^k)\cdot\cos\theta_i^k + \beta^k\cdot b_i + \varepsilon - \beta\sqrt{\Delta_i\sin^2(\theta_i+\theta_i^k)+b_i^2}$.We note that $v^k,\theta^k,\varphi^k,\beta^k,a_i,b_i,\Delta_i$ are constant.

Step 4. Construct an index set

$$\varXi^k = \left\{(i,j) : \varPhi^{\varOmega_i^k\varOmega_j^k}\left(u_i^k,u_j^k,\beta^k\right) < 0,\quad i > j \in I_n\right\},$$

where $\varPhi^{\varOmega_i^k\varOmega_j^k}(u_i^k,u_j^k,\beta^k)$ is a phi-function for two polygons $\varOmega_i^k(u_i^k,\beta^k)$ and $\varOmega_j^k(u_j^k,\beta^k)$ [13].

It is clear that if two individual containers $\varOmega_i^k$ and $\varOmega_j^k$ do not have common interior points (i.e. $\varPhi^{\varOmega_i^k\varOmega_j^k}(u_i^k,u_j^k,\beta^k) \geq 0$), then we do not need to check the non-overlapping constraint for the corresponding pair of ellipses $\widehat{E}_i(u_i^k,\beta^k)$ and $\widehat{E}_j(u_j^k,\beta^k)$ (e.g., the ellipses E_1 and E_3 in Fig. 6. In the case $\varXi^k = \{(1,2),(2,3)\}$).

Step 5. Form an subset

$$\varPi_k^\varepsilon = \left\{\left(u,\phi_{w_k}\right) \in R^{\sigma-\sigma_k} : f_{i1}^k(u_i,\beta) \geq 0, f_{i2}^k(u_i,\beta) \geq 0, f_{i3}^k(u_i,\beta) \geq 0, f_{i4}^k(u_i,\beta) \geq 0,\quad i \in I_n\right\}, \tag{15}$$

where $\sigma_k = card(\varXi\backslash\varXi^k)$.

Step 6. Construct an index set

$$\varXi^{*k} = \left\{(i,s) : \varPhi^{\varOmega_i^k P_s^*}\left(u_i^k,\beta^k\right) < 0,\quad s \in I_m, i \in I_n\right\},$$

where $\varPhi^{\varOmega_i^k P_s^*}(u_i^k,\beta^k)$ is a phi-function for polygon $\varOmega_i^k(u_i^k,\beta^k)$ and a half plane P_s^* [13].

In other words, if individual container $\varOmega_i^k$ and half plane P_s^* have common interior points (i.e.$\varPhi^{\varOmega_i^k P_s^*}(u_i^k,\beta^k) < 0$), then we need to take into account the containment constraint for the corresponding pair of objects. For example, we need to monitor pairs E_1 and P_1^*, E_1 and P_2^*, E_3 and P_4^* shown in Fig. 6. In the case $\varXi^{*k} = \{(1,1),(1,2),(3,4)\}$).

Step 7. Construct a vector of starting values for auxiliary variables $\phi_{w_k}^k = (\phi_{ij}^k,(i,j) \in \varXi^k)$.

For each pair $(i,j) \in \varXi^k$ we search for the maximum value of auxiliary variable ϕ_{ij} (see Fig. 3), using the following non-constrained nonlinear optimization problem:

$$\phi_{ij}^k = \arg\max_{\phi_{ij}\in[0,2\pi]\subset R^1} \varPhi'^{\widehat{E}_i\widehat{E}_j}\left(u_i^k,u_j^k,\beta^k,\phi_{ij}\right),$$

where u_i^k,u_j^k,β^k are fixed parameters.

Step 8. Solve the k-th subproblem, starting from point $(u^k,\phi_{w_k}^k) = (v^k,\theta^k,\beta^k,\phi_{w_k}^k)$:

$$\max_{(u^k,\phi_{w_k}^k)\in W_k} \beta, \tag{16}$$

$$W_k = \left\{\left(u^k,\phi_{w_k}^k\right) \in R^{\sigma-\sigma_k} : \varPhi'^{\widehat{E}_i\widehat{E}_j}\left(u_i,u_j,\beta,\phi_{ij}\right) \geq 0,\quad (i,j) \in \varXi_k\right\}, \tag{17}$$

$$\varPhi_{is}^*(u_i,\beta) \geq 0, (i,s) \in \varXi_k^*, \varPhi^{\widehat{E}_i\varOmega_i^{k*}}(u_i,\beta) \geq 0,\quad i \in I_n\},$$

where $(u,\phi_{w_k}) = (v,\theta,\beta,\phi_{w_k})$,$\varPhi'^{\widehat{E}_i\widehat{E}_j}(u_i,u_j,\beta,\phi_{ij})$ is defined in (9), $\varPhi_{is}^*(u_i,\beta)$ is defined by (4), $\varPhi^{\widehat{E}_i\varOmega_i^{k*}}(u_i,\beta)$ is defined at Step 3, $\varXi_k$ is defined at Step 4, σ_k is defined at Step 5, $\varXi_k^*$ is defined at Step 6.

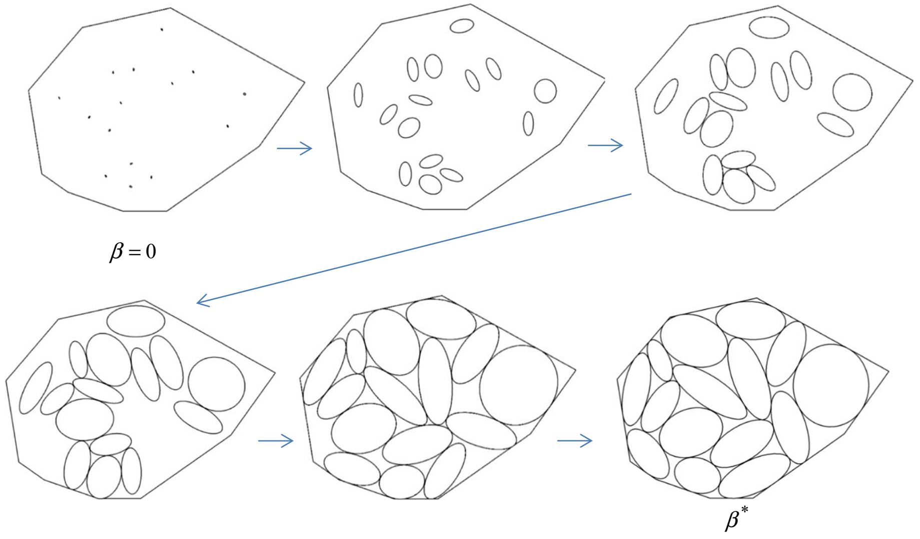


Fig. 7 Illustration to **LOFRT_P** algorithm

Step 9. If the point $(u^{k*}, \phi^*_{w_k})$ of local maximum of the k-th subproblem (16)–(17) belongs to the frontier of the subset Π^{ε}_{k} described by (15) then we set $u^{k+1} = u^{k*}$, take $k = k+1$ and go to Step 2, otherwise (i.e. $(u^{k*}, \phi^*_{w_k}) \in \text{int}\, \Pi^k_{\varepsilon}$) set $u^* = u^{k*}$ and go to Step 10.

Remark We do not need to redefine the deleted auxiliary variables at the last step of our algorithm based on the following reasons. Let $(u^*_{w_k}, \phi^*_{w_k}) \in R^{\sigma - \sigma_k}$ be the last point of our iterative procedure. We can construct vector ϕ^{k*} by means of redefining values of the previously deleted auxiliary variables for $(i, j) \in \Xi \backslash \Xi^k$ (see Step 7). The point $(u^*, \phi^*) = (u^{k*} = u^*_{w_k}, \phi^{k*}) \in R^{\sigma}$ is a point of local maximum of the problem (12)–(13). The assertion comes from the property of quasi phi-functions: any arrangement of each pair of non-overlapping ellipses $\widehat{E}_i$ and $\widehat{E}_i$ for $(i, j) \in \Xi \backslash \Xi^k$ guarantees that there always exists a vector ϕ^{k*} of auxiliary variables ϕ^*_{ij} such that $\Phi'^{\widehat{E}_i \widehat{E}_j}(u^*_i, u^*_j, \beta^*, \phi^*_{ij}) \geq 0, (i, j) \in \Xi \backslash \Xi^k$. Therefore the values of auxiliary variables of the vector ϕ^{k*} have no effect on the objective function value, i.e. $F(u^*_{w_k}, \phi^*_{w_k}) = F(u^{k*}, \phi^{k*})$.

Figure 7 shows a diagram for local optimization of problem **EPP_2:** maximization of β assuming $\lambda = 1$, starting from $\beta = 0$. We grow ellipses as much as possible in the fixed container $\Omega(\lambda = 1)$.

Step 10. Get a local minimum solution of the problem (10)–(11) using the local maximum of the k-th subproblem (16)–(17) and scaling transformation (14)

$$u^* = \left(v^*, \theta^*, \beta^*\right) \to t^* = \left(\frac{v^*}{\beta^*}, \theta^*, \lambda^* = \frac{1}{\beta^*}\right)$$

and stop our **LOFRT_P** procedure.

Figures 8 and 9 show diagrams for the Step 10 corresponding to $\beta^* > 1$ and $0 < \beta^* < 1$. Using the correspondence (14) between the problem **EPP_2** (maximizing the homothetic coefficient β of ellipses $\widehat{E}_i$) and the problem **EPP_1** (minimizing the homothetic coefficient λ of the convex polygon) we find $\lambda^* = \frac{1}{\beta^*}$.

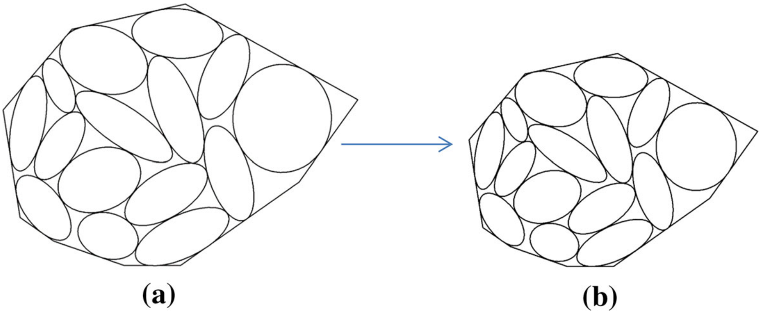


Fig. 8 Illustration to Step 10 for $\beta^* > 1$: **a** $\beta^* > 1, \lambda = 1$; **b** $\lambda^* = \frac{1}{\beta^*} < 1$

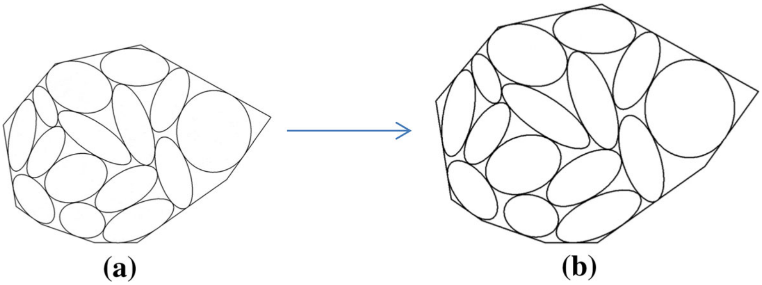


Fig. 9 Illustration to Step 10 for $\beta^* < 1$: **a** $\beta^* < 1, \lambda = 1$; **b** $\lambda^* = \frac{1}{\beta^*} > 1$

Note that if $\beta^* > 1$ (each ellipse size is larger than the original ellipse), then $\lambda^* < 1$ (Fig. 8), otherwise $0 < \beta^* \leq 1$ (each ellipse size is smaller than the original ellipse), then $\lambda^* \geq 1$ (Fig. 9).

Our algorithm can in most cases actively control only $O(n)$ pairs of ellipses from $O(n^2)$ since for each ellipse only its "ε-neighbors" have to be monitored. This depends on the sizes of ellipses and the value of ε [1]. The ε parameter provides a balance between the number of inequalities in each NLP sub-problem and the number of the subproblems (16)–(17) have to be generated and solved to get a locally optimal solution of the problem (12)–(13). The LOFTR_P procedure allows us to reduce considerably the computational time.

6 Computational results

Here we present a number of examples to demonstrate the efficiency of our methodology. We have run all experiments on an AMD FX(tm)-6100, 3.30 GHz computer, programming language C++, OS: Windows 7. For local optimisation we use the IPOPT code https://projects.coin-or.org/Ipopt), developed in [15], with default options.

First we present our new instances for the ellipse packing problem.

Example 1 A convex polygonal domain with $m = 5$ sides is given by its vertices $\{(x_s, y_s), s = 1,\ldots,5\} = \{(-7.266244, 1.593456), (-5.941336, -6.880334), (2.310916, -10.663272), (6.702019, -10.663272), (20.670273, 2.387873)\}$. Collections of ellipses are taken from [6]: (a) $n = 20$ (TC20), (b) $n = 30$ (TC30), c() $n = 50$ (TC50). Corresponding optimized packings are presented in Fig. 10a, b, c.

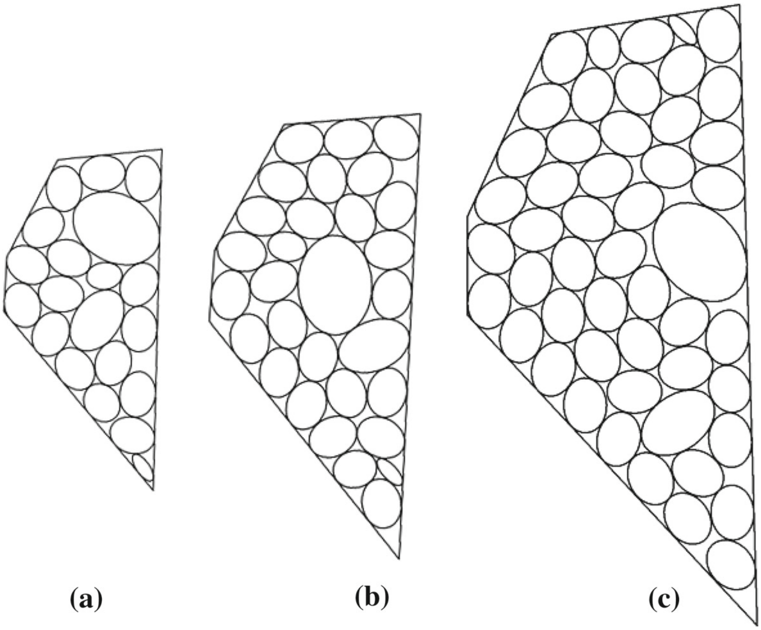


Fig. 10 Ellipse packings: **a** $n = 20$, **b** $n = 30$, **c** $n = 50$, in the optimized convex polygon with $m = 5$ in Example 1

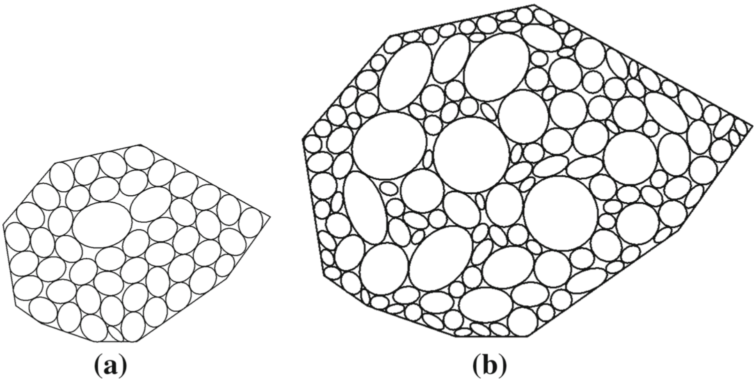


Fig. 11 Ellipse packings: **a** $n = 50$, **b** $n = 140$, in the optimized convex irregular polygon with $m = 9$ in Example 2

Example 2 A convex polygonal domain with $m = 9$ sides is given by its vertices $\{(x_s, y_s), s = 1,\ldots,9\} = \{(-7.266244, 1.593456), (-5.941336, -6.880334), (-3.291530, -8.733971), (2.310916, -10.663272), (6.702019, -10.663272), (16.052038, -4.080957), (20.670273, 2.387873), (7.118418, 9.915923), (-1.853066, 7.948795)\}$. We consider collections of ellipses: (c) $n = 50$ (TC50) is taken from [6], (d) $n = 20 \times 7 = 140$ is produced from [2]. Corresponding optimized packings are presented in Fig. 11a, b.

We used 100 runs for the LOFTR_P procedure in our multistart strategy for each instance in Examples 1 and 2.

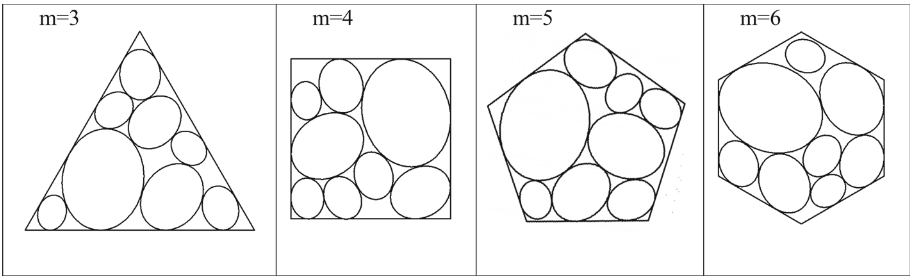


Fig. 12 Packings for $n = 8$ ellipses with parameter $c = 1.25$ in a regular polygon of $m = 3, 4, 5, 6$ sides

For these examples the optimized value of the homothetic coefficient λ^* and the corresponding CPU time are as follows: (a) $\lambda^* = 0.536303$, CPU = 654.96 s., (b) $\lambda^* = 0.640662$, CPU = 1222.99 s., (c) $\lambda^* = 0.810372$, CPU = 3764.04 s. in Example 1; (a) $\lambda^* = 0.6307$, CPU = 298.125 s., (b) $\lambda^* = 0.419$, CPU = 35840.98 s. in Example 2.

Further we apply our approach to the problem instances considered in [2]. The ellipse packing problem instances in [2] were designed as follows: $a_i = \frac{1}{\sqrt{i}}, b_i = \frac{a_i}{c}, i \in I_n$, for ellipses with eccentricity $e = \sqrt{1 - \frac{1}{c^2}}$ and a regular polygonal domain with m sides for the container.

Two groups of instances presented in [2] were running by our approach:

The first group of instances: for $c = 1.00, 1.05, 1.15, 1.35, 1.50, 1.75, 2.00$, $n = 3, 4, 5, 6, 8, 10$ and $m = 3, 4, 5, 6, 8$.

The second group of instances: for $c = 1.00, 1.05, 1.15, 1.35, 1.50, 1.75, 2.00$, $n = 10,15, 20$ and $m = 40$.

Comparative results for 210 instances of the first group are presented in Tables 1, 2, 3, 4, 5, 6 and 7, while Table 8 summarizes computational results for 21 instances of the second group. For each problem instance we present the area of the optimized container (line Area) and CPU time in sec. (line Time) corresponding to our algorithm (*column our result*). The area of the optimized container obtained in [2] is also presented (*column* [2]). Since our computer, computing platform and solver used for subproblems are different from those used in [2], we do not compare here CPU times for both techniques. The column "*impr. %*" corresponds to the area of the optimized container and gives the value (result of [2] minus our result)/(result of [2]) in %. Thus positive values in this column indicate the better performance of our approach, while zero or negative (highlighted in bold) show that results of [2] were not improved. We used 10 runs for the LOFTR_P procedure in our multistart strategy for each instance.

Figures 12 and 13 demonstrate our optimized packings for some problem instances of the first group. In [2] one can find configurations obtained for the same instances by their approach.

Figure 14 provides our optimized packings for some problem instances of the second group: packing $n = 20$ ellipses with six different values of parameter $c = \frac{a_i}{b_i}, i \in I_n$, in a regular polygon of $m = 40$ sides. In [2] one can find configurations for the same instances obtained by their approach.

Table 1 Comparison of results for c = 1

	m	3			4			5			6			8		
n		Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %
3	Area	9.31690	9.31690	**0.000**	9.11140	9.11140	**0.000**	9.04650	9.04650	**0.00**	9.12250	9.12260	**0.000**	9.28340	9.28340	**0.000**
	Time	1.09000			1.15000			1.26000			1.93000			1.76000		
4	Area	9.31690	9.31690	**0.000**	9.31370	9.31370	**0.00**	9.26470	9.26470	**0.000**	9.12250	9.56950	4.670	9.32100	9.32100	**0.000**
	Time	1.59000			1.70000			1.86000			2.22000			2.14000		
5	Area	10.2554	11.1001	7.610	9.56680	9.56680	**0.00**	9.78820	10.1249	3.330	9.39010	9.39010	**0.00**	9.54350	9.93320	3.920
	Time	2.06000			2.26000			2.70000			2.76000			3.01000		
6	Area	10.4243	11.3869	8.450	10.4850	10.6465	1.520	10.0855	10.8185	6.780	10.3551	10.8276	4.360	10.2485	10.4665	2.080
	Time	6.27000			3.25000			3.51000			3.45000			3.82000		
8	Area	11.5093	12.7463	9.700	11.1486	11.4082	2.280	10.8923	11.7270	7.120	10.8807	11.6258	6.410	10.8428	11.8728	8.680
	Time	4.65000			5.24000			5.85000			6.77000			7.32000		
10	Area	12.2783	13.1380	6.540	11.8141	12.4713	5.270	11.6402	12.4286	6.340	11.6815	12.6535	7.680	11.8202	12.9234	8.540
	Time	7.43000			12.1400			42.9300			11.1700			11.0600		

Table 2 Comparison of results for c = 1.05

	m	3			4			5			6			8		
n		Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %
3	Area	8.65040	8.65030	**0.00**	8.40930	8.40930	**0.00**	8.31090	8.31090	**0.00**	8.40940	8.40940	**0.00**	8.49900	8.49900	**0.00**
	Time	1.54000			1.81000			1.81000			2.12000			2.29000		
4	Area	8.65610	8.65610	**0.000**	8.57690	8.97460	4.430	8.48680	8.48680	**0.00**	8.47790	8.49950	0.250	8.59010	8.59010	**0.000**
	Time	2.51000			2.53000			2.87000			3.03000			3.15000		
5	Area	9.43560	9.43560	**0.000**	8.86910	8.94960	0.900	9.13320	9.17630	0.470	8.71010	9.51810	8.490	8.86590	8.87130	0.060
	Time	2.89000			3.54000			4.03000			3.87000			4.57000		
6	Area	9.74800	10.6288	8.290	9.69260	9.69260	**0.00**	9.29850	9.76040	4.730	9.64600	9.89180	2.480	9.59150	9.76100	1.740
	Time	3.87000			4.38000			5.02000			5.54000			6.24000		
8	Area	10.8726	10.9832	1.010	10.1786	10.6157	4.120	10.0584	10.4946	4.160	10.3333	11.3600	9.040	10.2461	11.1158	7.820
	Time	6.46000			7.96000			9.25000			38.3600			9.22000		
10	Area	11.4088	12.3612	7.700	11.1052	11.5162	3.570	10.8717	11.2872	3.680	11.0011	11.7032	6.000	10.9713	11.3016	2.920
	Time	11.5000			13.1500			15.4900			15.1600			15.6500		

Table 3 Comparison of results for c = 1.15

	m	3			4			5			6			8		
n		Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %
3	Area	7.55740	7.55740	**0.000**	7.21580	7.21580	**0.000**	7.10840	7.10840	**0.000**	7.24810	7.24810	**0.000**	7.19930	7.19930	**0.000**
	Time	1.67000			1.86000			2.12000			2.36000			2.18000		
4	Area	7.76380	7.76380	**0.000**	7.43030	7.43030	**0.000**	7.25170	7.25170	**0.000**	7.43700	7.43700	**0.000**	7.50550	7.68400	2.320
	Time	3.21000			2.64000			2.78000			2.86000			3.29000		
5	Area	8.15420	8.24400	1.090	7.84140	7.84140	**0.000**	8.11070	8.00130	− **1.37**	7.81060	7.87720	0.850	7.98450	8.01930	0.430
	Time	2.89000			3.35000			4.13000			3.99000			4.37000		
6	Area	8.67250	9.43380	8.070	8.38740	8.64160	2.940	8.38870	8.95280	6.300	8.49190	8.50110	0.110	8.53640	8.80570	3.060
	Time	4.52000			4.59000			5.04000			5.93000			6.12000		
8	Area	9.63970	10.0630	4.210	9.12680	9.35480	2.440	9.17530	9.63620	4.780	9.06150	9.22620	1.790	8.98110	9.61190	6.560
	Time	6.58000			7.99000			9.14000			9.61000			10.7200		
10	Area	9.90260	11.8797	16.64	9.80190	10.1791	3.710	9.83150	10.4341	5.780	9.61220	10.2820	6.510	9.77340	10.1904	4.090
	Time	10.9800			13.1800			15.5700			16.2600			17.6300		

Table 4 Comparison of results for c = 1.25

	m	3			4			5			6			8		
n		Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %
3	Area	6.09670	6.70410	9.060	5.73880	6.34010	9.480	5.64520	6.26120	9.840	5.65150	6.35280	11.04	5.46540	6.18140	11.58
	Time	1.94000			1.81000			2.03000			2.00000			2.36000		
4	Area	6.39790	7.03240	9.020	6.35350	6.87040	7.520	5.98010	6.57290	9.020	6.13860	6.81060	9.870	6.08220	6.84450	11.14
	Time	2.45000			2.87000			2.95000			3.20000			3.60000		
5	Area	6.80300	8.18730	16.91	6.61060	7.42040	10.91	6.53630	7.36280	11.23	6.54220	7.59520	13.86	6.65400	7.29900	8.840
	Time	3.20000			3.59000			3.99000			3.93000			4.76000		
6	Area	7.23670	8.17070	11.43	6.96490	7.94290	12.31	6.77860	7.77700	12.84	6.81370	7.78210	12.44	6.90110	7.70840	10.47
	Time	4.07000			4.74000			5.23000			5.69000			6.77000		
8	Area	7.77710	9.25430	15.96	7.61380	8.95050	14.94	7.49270	8.63590	13.24	7.53900	8.45910	10.88	7.50490	8.76420	14.37
	Time	6.99000			8.35000			9.52000			10.4500			11.2900		
10	Area	8.39920	9.60120	12.52	8.17580	9.10440	10.20	8.03490	8.95570	10.28	8.03600	9.05680	11.27	8.10270	9.11210	11.08
	Time	11.5100			34.9900			17.0500			17.5500			17.6000		

Table 5 Comparison of results for c = 1.50

	m	3			4			5			6			8		
n		Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %
3	Area	5.46860	5.46860	**0.000**	5.01030	5.01030	**0.00**	4.92580	4.92580	**0.00**	4.85290	4.85290	**0.00**	4.70860	4.70860	**0.00**
	Time	2.18000			1.98000			2.21000			2.11000			2.26000		
4	Area	5.78240	6.33790	8.760	5.66480	5.69420	0.520	5.27220	5.44830	3.230	5.37450	5.46260	1.610	5.33810	5.33810	**0.000**
	Time	2.34000			2.78000			2.81000			3.00000			3.51000		
5	Area	6.13060	6.73450	8.970	5.88030	6.02740	2.440	5.81010	5.87730	1.140	5.68670	5.83060	2.470	5.78880	5.96080	2.890
	Time	3.26000			3.73000			4.01000			4.18000			5.05000		
6	Area	6.53910	6.92540	5.580	6.24130	6.27300	0.510	6.21510	6.32140	1.680	6.05990	6.14750	1.430	6.12410	6.12830	0.070
	Time	4.15000			5.07000			5.62000			5.84000			6.71000		
8	Area	7.10550	7.35670	3.410	6.79720	7.08330	4.040	6.74200	7.04830	4.350	6.74600	6.84630	1.460	6.67240	6.89610	3.240
	Time	28.4700			8.49000			9.34000			10.2500			12.4300		
10	Area	7.54290	7.92680	4.840	7.35410	7.64210	3.770	7.25520	7.43590	2.430	7.19550	7.42440	3.080	7.14630	7.29270	2.010
	Time	11.7000			15.1800			16.0800			19.8300			18.7200		

Table 6 Comparison of results for c = 1.75

	m	3			4			5			6			8		
n		Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %
3	Area	4.76150	4.76150	**0.00**	4.16880	4.16880	**0.00**	4.14860	4.14860	**0.000**	4.01970	4.01970	**0.000**	4.01870	4.01870	**0.000**
	Time	2.00000			1.89000			1.95000			2.17000			2.50000		
4	Area	4.96210	4.96920	0.140	4.79460	4.79460	**0.000**	4.49720	4.49720	**0.000**	4.53260	4.53260	**0.000**	4.47800	4.47800	**0.000**
	Time	2.39000			2.65000			2.84000			3.00000			3.52000		
5	Area	5.23440	5.38490	2.800	5.13910	5.21970	1.540	4.85760	4.85210	− **0.11**	4.85850	4.92070	1.260	4.80770	4.80770	**0.000**
	Time	3.45000			3.84000			3.93000			4.48000			5.18000		
6	Area	5.53910	5.68480	2.560	5.39430	5.52500	2.370	5.22020	5.30730	1.640	5.17210	5.17500	0.060	5.14870	5.15660	0.150
	Time	4.40000			5.48000			5.48000			5.71000			6.83000		
8	Area	6.09910	6.95630	12.32	5.85570	5.91040	0.930	5.74570	6.01020	4.400	5.73390	5.88690	2.600	5.63120	5.86350	3.960
	Time	7.05000			9.92000			9.69000			10.2700			12.5700		
10	Area	6.38480	6.62410	3.610	6.25150	6.72520	7.040	6.17910	6.45370	4.250	6.10970	6.37590	4.180	6.06250	6.24000	2.850
	Time	48.1900			13.9600			51.1100			17.2400			18.5200		

Table 7 Comparison of results for c = 2

	m	3			4			5			6			8		
n		Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %
3	Area	4.26840	4.26840	**0.000**	3.72250	3.72250	**0.000**	3.72500	3.72500	**0.000**	3.50890	3.63640	3.510	3.56620	3.56620	**0.000**
	Time	1.69000			2.11000			2.11000			2.14000			2.65000		
4	Area	4.38870	4.51450	2.790	4.13910	4.13910	**0.000**	4.05810	4.17100	2.710	4.01470	4.10700	2.250	4.01400	4.01400	**0.00**
	Time	2.42000			2.78000			2.85000			3.09000			3.56000		
5	Area	4.56580	4.70590	2.980	4.49300	4.51520	0.490	4.36130	4.35600	− **0.12**	4.34770	4.30680	− **0.95**	4.24200	4.24230	0.010
	Time	3.31000			3.96000			4.20000			4.21000			5.23000		
6	Area	4.81850	4.88880	1.440	4.72900	4.86760	2.850	4.62250	4.66240	0.860	4.55660	4.61150	1.190	4.51940	4.57290	1.170
	Time	4.43000			5.04000			5.43000			5.43000			7.33000		
8	Area	5.28060	6.03080	12.44	5.16220	5.48220	5.840	5.06010	5.14150	1.580	5.02820	5.16040	2.560	4.95210	5.12500	3.380
	Time	7.50000			9.17000			9.72000			10.9500			12.2500		
10	Area	5.64990	6.28060	10.04	5.48790	5.77640	5.000	5.37660	5.77300	6.870	5.38880	5.60580	3.870	5.35620	5.56690	3.790
	Time	12.4000			15.1600			17.1800			17.3600			17.2500		

Table 8 Comparison results for m = 40

	n	10			15			20		
c		Our result	[2]	Impr %	Our result	[2]	Impr %	Our result	[2]	Impr %
1	Area	11.9841	12.7618	6.090	13.4129	13.9156	3.610	14.5358	14.8480	2.100
	Time	5.55000			9.38000			43.5900		
1.05	Area	11.0175	11.5416	4.540	12.4215	12.9380	3.990	13.4746	13.5649	0.670
	Time	6.47000			15.6800			55.8200		
1.15	Area	10.0881	10.1715	0.820	11.3100	11.4132	0.900	11.8785	12.2667	3.160
	Time	5.40000			21.6400			57.1000		
1.25	Area	8.25730	9.25550	10.78	9.09080	10.1671	10.59	9.79380	11.2711	13.11
	Time	5.32000			20.8100			70.1500		
1.5	Area	7.12860	7.24320	1.580	8.13970	8.38260	2.900	8.77000	9.11340	3.770
	Time	5.47000			21.9200			64.4800		
1.75	Area	6.14890	6.36260	3.360	6.79740	6.99610	2.840	7.48320	7.71640	3.020
	Time	8.13000			17.9100			70.2900		
2	Area	5.35890	5.50100	2.580	6.01160	6.46780	7.050	6.50510	6.57780	1.100
	Time	6.05000			21.5100			65.6800		

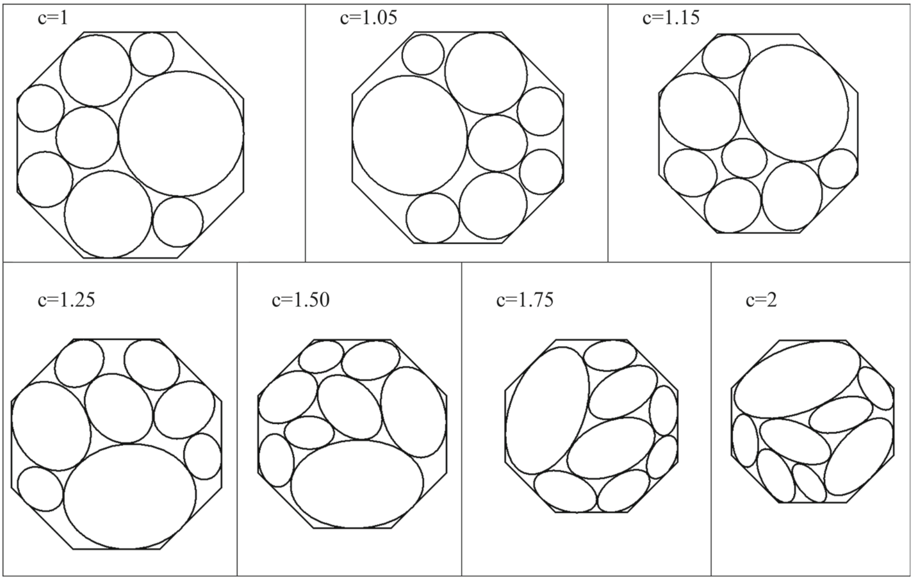


Fig. 13 Packings for $n = 8$ ellipses with parameter $c = 1.0, 1.05, 1.15, 1.25, 1.50, 1.75, 2.0$ in a regular polygon of $m = 8$ sides

7 Conclusions

In this work we study a packing problem of a set of n arbitrary ellipses into an optimized convex polygon. The packed ellipses are given by their semi-axes and can be continuously

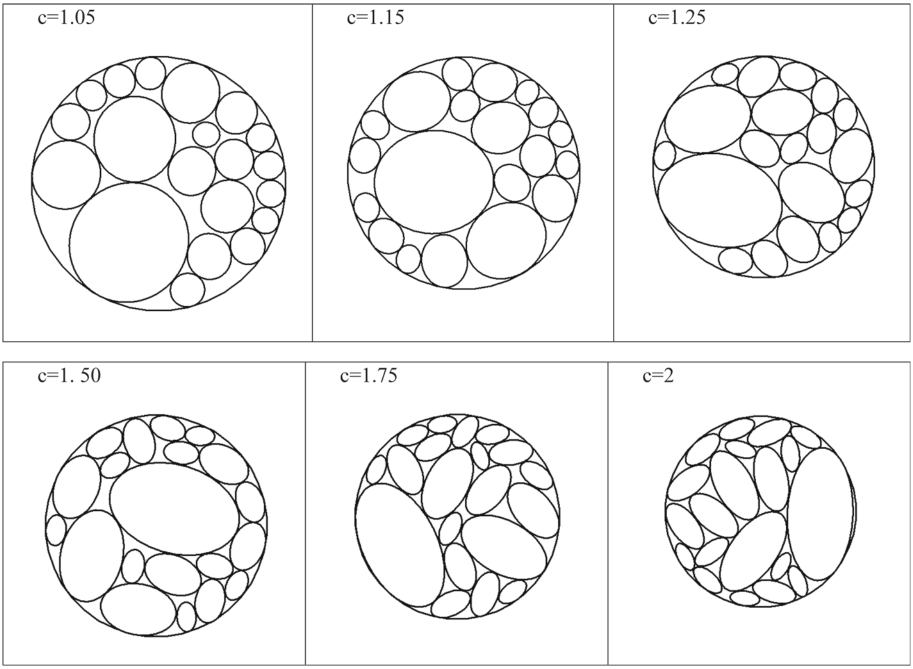


Fig. 14 Packings for $n = 20$ ellipses with parameter $c = 1.05, 1.15, 1.25, 1.50, 1.75, 2.0$, in a regular polygon of $m = 40$ sides

translated and rotated. We propose new tools of mathematical modelling of placement constraints: a quasi phi-function to describe non-overlapping ellipses and a phi-function for containment constraints. Our quasi phi-function is considerably simpler than the quasi phi-function presented in [1, 12]: it has only one auxiliary variable and does not involve the "min" operator. An exact mathematical formulation for the ellipse packing problem is stated as the large scale nonlinear programming problem with $O(n^2)$ continuous variables and $O(n^2)$ nonlinear inequalities. We propose a simple algorithm to construct feasible starting points and the LOFTR_P procedure to search for good local optimal solutions. The LOFTR_P typically reduces our problem to a sequence of nonlinear programming subproblems of smaller dimension with $O(n)$ variables and $O(n)$ constraints. Computational results are provided to demonstrate the efficiency of our approach. The experiments were performed by solving new problem instances, as well as for the instances presented in [2]. In this work an open source local solver IPOPT [15] was used for implementing the algorithms. However, open source or commercial global solvers (i.e., COUENNE [16] or LGO [17]) can be used directly for the exact formulations (NLP-models) of the problems EPP_1 or EPP_2. Using global solvers in combination with the proposed starting point algorithm is an interesting direction for the future research.

Acknowledgements Our interest in packing continuously rotating ellipses in an optimized convex polygonal container was largely motivated by the paper [2]. We would like to thank all authors of the paper and personally János Pintér for sharing with us their experience and for many fruitful discussions. The work of the third author was partially supported by the CONACYT Grants #167019, 293403.

References

1. Stoyan, Y., Pankratov, A., Romanova, T.: Quasi-phi-functions and optimal packing of ellipses. J. Glob. Optim. **65**(2), 283–307 (2016)
2. Kampas, F.J., Castillo, I., Pintér, J.D.: Optimized ellipse packings in regular polygons. Optim. Lett. (2019). https://doi.org/10.1007/s11590-019-01423-y
3. Chazelle, B., Edelsbrunner, H., Guibas, L.J.: The complexity of cutting complexes. Discrete Comput. Geom. **4**(2), 139–181 (1989)
4. Donev, A., Cisse, I., Sachs, D., Variano, E., Stillinger, F.H., Connelly, R., Torquato, S., Chaikin, P.M.: Improving the density of jammed disordered packings using ellipsoids. Science **303**(5660), 990–993 (2004)
5. Kallrath, J.: Packing ellipsoids into volume-minimizing rectangular boxes. J. Glob. Optim. **67**(1–2), 151–185 (2017)
6. Kallrath, J., Rebennack, S.: Cutting ellipses from area-minimizing rectangles. J. Glob. Optim. **59**(2), 405–437 (2014)
7. Birgin, E.G., Lobato, R.D., Martinez, J.M.: Packing ellipsoids by nonlinear optimization. J. Glob. Optim. **65**(4), 709–743 (2016)
8. Birgin, E.G., Lobato, R.D., Martínez, J.M.: A nonlinear programming model with implicit variables for packing ellipsoids. J. Glob. Optim. **68**(3), 467–499 (2017)
9. Litvinchev, I., Infante, L., Ozuna, L.: Packing circular like objects in a rectangular container. J. Comput. Syst. Sci. Int. **54**(2), 259–267 (2015)
10. Litvinchev, I., Infante, L., Ozuna, L.: Approximate packing: integer programming models, valid inequalities and nesting. In: Fasano, G., Pintér, J. (eds.) Optimized Packings and Their Applications. Springer Optimization and its Applications, vol. 105, p. 326. Springer, New York (2015)
11. Stoyan, Y., Romanova, T.: Mathematical models of placement optimization: two- and three-dimensional problems and applications. In: Fasano, G., Pintér, J. (eds.) Modeling and Optimization in Space Engineering. Springer Optimization and Its Applications, vol. 73, pp. 363–388. Springer, New York (2013)
12. Stoyan, Y., Pankratov, A., Romanova, T., Chugay, A.: Optimized object packings using quasi-phi-functions. In: Fasano, G., Pintér, J. (eds.) Optimized Packings and Their Applications. Springer Optimization and its Applications, vol. 105, pp. 265–291. Springer, New York (2015)
13. Chernov, N., Stoyan, Y., Romanova, T., Pankratov, A.: Phi-functions for 2D objects formed by line segments and circular arcs. Adv. Oper. Res. (2012). https://doi.org/10.1155/2012/346358
14. Stoyan, Yu., Pankratov, A., Romanova, T.: Cutting and packing problems for irregular objects with continuous rotations: mathematical modeling and nonlinear optimization. J. Oper. Res. Soc. **67**(5), 786–800 (2016)
15. Wachter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006). https://projects.coin-or.org/Ipopt
16. Belotti, P.: COUENNE: a user's manual (2016). https://www.coin-or.org/Couenne/
17. Pintér, J.: How difficult is nonlinear optimization? A practical solver tuning approach, with illustrative results. Ann. Oper. Res. **265**, 119–141 (2018)