Research Article

Optimized Packing Clusters of Objects in a Rectangular Container

T. Romanova,1 A. Pankratov,1 I. Litvinchev,2 Yu. Pankratova,3 and I. Urniaieva3

1Department of Mathematical Modeling and Optimal Design, Institute for Mechanical Engineering Problems of the National Academy of Sciences of Ukraine, Kharkiv, 61046, Ukraine
2Graduate Program in Systems Engineering, Nuevo Leon State University (UANL), Mexico
3Department of System Engineering, Kharkiv National University of Radio Electronics, Kharkiv, 61166, Ukraine

Correspondence should be addressed to T. Romanova; sherom@kharkov.ua

Received 30 October 2018; Accepted 6 January 2019; Published 5 February 2019

Academic Editor: Hong-Yu Wu

Copyright © 2019 T. Romanova et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A packing (layout) problem for a number of clusters (groups) composed of convex objects (e.g., circles, ellipses, or convex polygons) is considered. The clusters have to be packed into a given rectangular container subject to nonoverlapping between objects within a cluster. Each cluster is represented by the convex hull of objects that form the cluster. Two clusters are said to be nonoverlapping if their convex hulls do not overlap. A cluster is said to be entirely in the container if so is its convex hull. All objects in the cluster have the same shape (different sizes are allowed) and can be continuously translated and rotated. The objective of optimized packing is constructing a maximum sparse layout for clusters subject to nonoverlapping and containment conditions for clusters and objects. Here the term sparse means that clusters are sufficiently distant one from another. New quasi-phi-functions and phi-functions to describe analytically nonoverlapping, containment and distance constraints for clusters are introduced. The layout problem is then formulated as a nonlinear nonconvex continuous problem. A novel algorithm to search for locally optimal solutions is developed. Computational results are provided to demonstrate the efficiency of our approach. This research is motivated by a container-loading problem; however similar problems arise naturally in many other packing/cutting/clustering issues.

1. Introduction

A two-dimensional packing problem of objects in a container is NP-hard [1]. This problem is typical in logistics (transporting rolls of wallpaper, pipes, boxes, paint buckets, etc.) and also has important applications in computer science, industrial engineering, manufacturing and production processes, healthcare, project portfolio selection, nanophysics, and agriculture (see, e.g., [2] and the references therein).

Various shapes of objects were studied. In particular, cutting and packing problems for ellipses are presented in [3–6], circular packing problems are investigated in [7–13], and packing problems for convex polygons are considered in [11–16]. Papers [17–21] are devoted to irregular packing involving arbitrary shaped objects.

In many cases the objects are not independent and have to be grouped in a number of certain clusters of nonoverlapping objects. This is typical, for example, for a container loading problem [22], where the objects in a large maritime container may form various clusters according to a type of objects (similar shapes, parts of the same machine). Similarly, clusters can be formed according to a supplier or a client (final destination) to facilitate loading/unloading the container.

While the composition of the cluster (number of objects and their shapes) is typically predefined, the overall shape of the cluster is frequently not specified. Bearing in mind a cluster as a number of objects placed in a flexible sack we define the shape of a cluster as a convex hull of the objects in the cluster. Note that the objects are nonoverlapping and the shape of the cluster (convex hull) depends on the layout of the objects in the cluster.

Throughout this paper it is assumed homogeneous clusters composed of the same shapes (different sizes are allowed). The number of clusters as well as the number of objects and their shapes and sizes is given. The shape of the cluster is represented by the convex hull of the objects
in the cluster. A feasible clusters layout has to meet the following conditions: (a) objects in the cluster are mutually nonoverlapping and (b) clusters are mutually nonoverlapping and nonoverlapping with the complement of the rectangular container.

Concerning optimized packing of clusters, different objectives can be used. For example, we may look for the “densest” layout fixing one dimension of the rectangular container and minimizing the other subject to feasibility of the clusters layout. On the contrary, we may fix both dimensions of the container and look for a “sparsest” layout maximizing a certain “distance” between the clusters. This objective is motivated by the need of more space between clusters to facilitate access for their loading/unloading. In this paper the latter approach is used.

To the best of our knowledge the problem of optimized packing for clusters of objects was never considered before. The main contributions of the paper are (1) a new formulation of the layout problem for clusters of objects; (2) new mathematical modeling tools for representing nonoverlapping and containment of clusters; (3) novel NLP model for optimized packing of clusters; and (4) new algorithm to get initial feasible solution to accelerate a local optimization procedure.

The paper is organized as follows. In Section 2 a layout problem for clusters of objects is formulated. Section 3 provides nonoverlapping and containment conditions for clusters obtained by phi-function technique. Mathematical model for the optimized layout problem is stated in Section 4. Solution algorithm and numerical experiments are presented in Sections 5 and 6, respectively. Conclusions in Section 7 complete this paper.

2. Problem Formulation

Let \( \Omega = \{(x, y) \in \mathbb{R}^2 : 0 \leq x \leq l, 0 \leq y \leq w\} \) be a rectangular domain (a container) and \( T_i, i \in \{1, 2, \ldots, n\} = I_n \), be an ordered collection of convex objects. Each object \( T_i \) is defined by its geometrical characteristics. In particular, an ellipse \( E_i \) is defined by its semiaxes \( a_i \) and \( b_i \), a circle \( C_i \) is defined by its radius \( r_i \), and a convex polygon \( K_i \) is defined by its vertices \( p_{r_k} = (p_{ik}, p_{jik}) \), \( k = 1, \ldots, j \).

The position of the object \( T_i \) is characterized by the vector of variable placement parameters \( u_i = (v_i, \theta_i) \), where \( v_j = (x_j, y_j) \) is a translation vector and \( \theta_i \) is a rotation angle. The center of the object coincides with the origin of its local coordinate system. Rotated by an angle \( \theta_i \) and translated by a vector \( v_j \) an object \( T_j \) is defined as \( T_j(u_i) = \{p \in \mathbb{R}^2 : p = v_j + M(\theta_i) \cdot p, \forall p \in T_j^0\} \), where \( T_j^0 \) denotes nontranslated and nonrotated object \( T_j \) and \( M(\theta_i) = (\begin{pmatrix} \cos \theta_i & \sin \theta_i \\ -\sin \theta_i & \cos \theta_i \end{pmatrix}) \) is a standard rotation matrix.

**Remark 1.** The position of a circle \( C_i \) is characterized by the motion vector \( u_i = (v_i, 0) = (x_i, y_i, 0) \).

In this paper we define a cluster of objects (or simply a cluster) as a group of objects having the same shape. The ordered collection of objects \( T_i, i = 1, \ldots, n \), is divided into \( N \) clusters \( \Lambda_1 = \{T_1, \ldots, T_{n_1}\}, \Lambda_2 = \{T_{n_1+1}, \ldots, T_{n_2}\}, \ldots, \Lambda_N = \{T_{n_{N-1}+1}, \ldots, T_n\} \) with respect to index sets \( \Xi_1 = \{1, \ldots, n_1\}, \ldots, \Xi_N = \{n_{N-1} + 1, \ldots, n\} \), and \( \Xi_2 = \{n_1 + 1, \ldots, n_2\}, \ldots, \Xi_N = \{n_{N-1} + 1, \ldots, n\} \).

We introduce vectors of placement parameters of clusters \( \Lambda_1, \Lambda_2, \ldots, \Lambda_N \) in the form \( z_i = (u_1, \ldots, u_{n_i}) \), \( z_2 = (u_{n_1+1}, \ldots, u_{n_2}), \ldots, z_N = (u_{n_{N-1}+1}, \ldots, u_n) \) and use notation \( \Lambda_q(z_q) \) for a cluster \( \Lambda_q \) that involves moving objects \( T_i(u_i) \), \( i \in \Xi_q \).

With each cluster \( \Lambda_q(z_q) = \bigcup_{i \in \Xi_q} T_i(u_i) \) we associate the convex hull of objects \( T_i(u_i) \), \( i \in \Xi_q \), denoted by \( \tilde{\Lambda}_q(z_q) = conv(\Lambda_q(z_q)), q \in I_N \).

Through this paper the following definitions are used: (1) two clusters \( \Lambda_q(z_q) \) and \( \Lambda_q(z_q') \) do not overlap each other if their convex hulls \( \tilde{\Lambda}_q(z_q) \) and \( \tilde{\Lambda}_q(z_q') \) do not overlap. \( q \in \Omega; (2) \) a cluster \( \Lambda_q(z_q) \) belongs to a container \( \Omega \) if its convex hull \( \tilde{\Lambda}_q(z_q) \) belongs to a container \( \Omega, q \in I_N \).

**Cluster Layout Problem CLP.** Arrange the collection of clusters \( \Lambda_q(z_q), q \in I_N \), sufficiently distant one from another into a rectangular container \( \Omega \), such that

\[
\text{int } T_i(u_i) \cap \text{int } T_j(u_j) = \emptyset, \quad (1)
\]

for \( i > j, (i, j) \in \Sigma_q \times \Sigma_q, q \in I_N \),

\[
\text{int } \tilde{\Lambda}_q(z_q) \cap \text{int } \tilde{\Lambda}_g(z_g) = \emptyset, \quad q > g \in I_N, \quad (2)
\]

\[
\tilde{\Lambda}_q(z_q) \subset \Omega \quad \text{for each } q \in I_N, \quad (3)
\]

The first constraint assures nonoverlapping of objects within the same cluster and the second guarantees nonoverlapping of clusters, while the third relation presents containment conditions for clusters into \( \Omega \).

3. Tools of Mathematical Modeling

In this study, we use phi-function technique as a constructive tool for mathematical modeling of placement constraints within the field of Packing & Cutting (see, e.g., [3, 23–25]). For the reader’s convenience we provide definitions of a phi-function (normalized phi-function) and a quasi-phi-function (normalized quasi-phi-function) in the appendix. In this section we introduce new tools to describe placement constraints (2) and (3) analytically.

3.1. Nonoverlapping of Objects within a Cluster. To describe relation (1) we use phi-functions and quasi-phi-functions for a pair of different shapes of convex objects.

Let us consider two objects \( T_i(u_i) \) and \( T_j(u_j) \) that belong to a cluster \( \Lambda_q(z_q), i > j, (i, j) \in \Sigma_q \times \Sigma_q, q \in I_N \).

According to the definition of a phi-function [23], we have that \( \Phi^{T_i T_j}(u_i, u_j) \geq 0 \) if and only if \( \text{int } T_i(u_i) \cap \text{int } T_j(u_j) = \emptyset \).

From the main property of a quasi-phi-function [3] it follows that if \( \Phi^{T_i T_j}(u_i, u_j) \geq 0 \) for some \( u_j \), then \( \text{int } T_i(u_i) \cap \text{int } T_j(u_j) = \emptyset \), where \( u_j \) is a vector of extra variables.
In particular, we provide nonoverlapping tools for a pair of ellipses, circles, and convex polygons.

**Quasi-Phi-Function for Two Ellipses.** Let \( E_i(u_i) \) and \( E_j(u_j) \) be ellipses given by their semiaxes \((a_i, b_i)\) and \((a_j, b_j)\), with placement parameters \( u_i = (x_i, y_i, \theta_i) \) and \( u_j = (x_j, y_j, \theta_j) \).

A quasi-phi-function for \( E_i(u_i) \) and \( E_j(u_j) \) can be defined as

\[
\Phi^{E_iE_j}(u_i, u_j, u_{ij} = \phi_{ij}) = x'_{ij} (v_i, \phi_{ij}) - x'_{ji} (v_j, \phi_{ij})
\]

\[
= \left( d_{ij} (\theta_i, \phi_i) + d_{ji} (\theta_j, \phi_j) \right),
\]

where

\[
x'_{ij} (v_i, \phi_{ij}) = x_i \cos \phi_{ij} - y_i \sin \phi_{ij},
\]

\[
x'_{ji} (v_j, \phi_{ij}) = x_j \cos \phi_{ij} - y_j \sin \phi_{ij},
\]

\[
d_{ij} (\theta_i, \phi_i) = \sqrt{b_i^2 + (a_i^2 - b_i^2)} \cos^2 (\theta_i - \phi_i),
\]

\[
d_{ji} (\theta_j, \phi_j) = \sqrt{b_j^2 + (a_j^2 - b_j^2)} \cos^2 (\theta_j - \phi_j).
\]

\(\phi_{ij} \in \mathbb{R}^1\) is an auxiliary variable (an angle between a straight line passing through the origin of the global coordinate system \(OXY\) and axis \(OX\)).

**Phi-Function for Two Circles.** Let \( C_i(v_i) \) and \( C_j(v_j) \) be circles given by their radii \( r_i \) and \( r_j \) with placement parameters \( v_i = (x_i, y_i) \) and \( v_j = (x_j, y_j) \).

A phi-function for \( C_i(v_i) \) and \( C_j(v_j) \) has the form

\[
\Phi^{C_iC_j}(v_i, v_j) = (x_i - x_j)^2 + (y_i - y_j)^2 - (r_i + r_j)^2.
\]

**Quasi-Phi-Function for Two Convex Polygons.** Let convex polygons \( K_i(u_i) \) and \( K_j(u_j) \) be defined by their vertices \( p_k = (r_k^x, r_k^y) \), \( k = 1, \ldots, l_i \), \( p_k = (r_k^x, r_k^y) \), \( k = 1, \ldots, l_j \), with placement parameters \( u_i = (x_i, y_i, \theta_i) \) and \( u_j = (x_j, y_j, \theta_j) \) and let \( P(u_{ij}) = \{(x, y) : \mu_{ij} = \cos \phi_{ij} \cdot x + \sin \phi_{ij} \cdot y + y_i \leq 0\} \) be a half plane.

Based on [26] a quasi-phi-function for \( K_i(u_i) \) and \( K_j(u_j) \) can be derived in the form

\[
\Phi^{K_iK_j}(u_i, u_j, u_{ij}) = \min \{\Phi^{K_iP}(u_i, u_{ij}), \Phi^{K_jP'}(u_j, u_{ij})\},
\]

where \( u_{ij} = (\phi_{ij}, y_{ij}) \in \mathbb{R}^2 \) is a vector of auxiliary variables,

\[
\Phi^{K_iP}(u_i, u_{ij}) = \min_{1 \leq k \leq l_i} \{\mu_{ij}(p_k)\},
\]

is a phi-function for \( K_i(u_i) \) and \( P(u_{ij}) \) and

\[
\Phi^{K_jP'}(u_j, u_{ij}) = \min_{1 \leq k \leq l_j} \{-\mu_{ij}(p_k)\}
\]

is a phi-function for \( K_j(u_j) \) and \( P'(u_{ij}) = \mathbb{R}^2 \setminus \text{int } P(u_{ij}) \).

**3.2. Nonoverlapping of Clusters.** Let us describe analytically the placement constraint (2). To this aim we introduce a quasi-phi-function for two objects \( \Lambda_q(z_q) \) and \( \Lambda_g(z_g) \).

Now we consider an extension of the phi-function technique to convex hulls of clusters of objects. Note that convex hull of objects in a cluster has variable shape and variable metrical characteristics that depend on variable placement parameters of objects in the cluster.

Let \( P_{ag} = \{(x, y) : \mu_{ag}(x, y) = \cos \phi_{ag} \cdot x + \sin \phi_{ag} \cdot y + y_{ag} \geq 0\} \) be a half plane with variable placement parameters \( (\phi_{ag}, y_{ag}) \).

**Proposition 2.** A continuous and everywhere defined function

\[
\Phi^{\tilde{\Lambda}_q\tilde{\Lambda}_g}(z_q, z_g, \phi_{ag}, y_{ag})
\]

is a quasi-phi-function for \( \tilde{\Lambda}_q(z_q) \) and \( \tilde{\Lambda}_g(z_g) \), where \( \Phi^{\tilde{\Lambda}_q\tilde{\Lambda}_g}(z_q, \phi_{ag}, y_{ag}) \) is a phi-function of \( \tilde{\Lambda}_q(z_q) \) and a half plane \( P_{ag} \) and \( \Phi^{\tilde{\Lambda}_q\tilde{\Lambda}_g}(z_q, \phi_{ag}, y_{ag}) \) is a phi-function of \( \tilde{\Lambda}_g(z_g) \) and the half plane \( P_{ag}^g = \mathbb{R}^2 \setminus \text{int } P_{ag} \).

**Proof.** Based on the properties of a quasi-phi-function [3] defined for two convex objects we can conclude that

\[
\Phi^{\tilde{\Lambda}_q\tilde{\Lambda}_g}(z_q, z_g, \phi_{ag}, y_{ag}) \geq 0 \quad \text{if and only if there exists at least one separated line } L_{ag}(x, y) = \mu_{ag}(x, y) = 0 \text{ for some } \phi_{ag}, \quad y_{ag} \text{ such that } \Lambda_q(z_q) \subset P_{ag} \text{ and } \Lambda_g(z_g) \subset P_{ag}^g \text{ and therefore } \tilde{\Lambda}_g(z_q) \cap \tilde{\Lambda}_g(z_g) = \emptyset \text{ for } q > g \in J_N.
\]

Next we define a phi-function of \( \tilde{\Lambda}_g(z_g) \) and a half plane \( P_{ag} \) in (10) in the form

\[
\Phi^{\tilde{\Lambda}_gP_{ag}}(z_g, \phi_{ag}, y_{ag})
\]

\[
= \min \{\Phi^{\tilde{T}_gP_{ag}}(u_i, \phi_{ag}, y_{ag}), i \in \Xi_k\},
\]

where \( \Phi^{\tilde{T}_gP_{ag}}(u_i, \phi_{ag}, y_{ag}) \) is a phi-function of an object \( T_i(u_i) \) and half plane \( P_{ag} \).

It follows from the properties of the convex hull of \( \Lambda_q(z_q) = \bigcup_{i \in \Xi_k} T_i(u_i) \) and characteristics of a separating line between two convex objects that

\[
(1) \Lambda_q(z_q) \cap P_{ag} = \emptyset \iff \Lambda_q(z_q) \cap P_{ag} = \emptyset 
\]

and therefore \( T_i(u_i) \cap P_{ag} = \emptyset \) for all \( i \in \Xi_k \). It implies that \( \min \{\Phi^{\tilde{T}_gP_{ag}}(u_i, \phi_{ag}, y_{ag}), i \in \Xi_k\} > 0 \).

(2) \( \text{int } \Lambda_q(z_q) \cap \text{int } P_{ag} \neq \emptyset \iff \text{int } \Lambda_q(z_q) \cap \text{int } P_{ag} \neq \emptyset \) and therefore there exists at least one object \( T_i(u_i) \), such that \( \text{int } T_i(u_i) \cap \text{int } P_{ag} \neq \emptyset \), \( i \in \Xi_k \). It implies that \( \min \{\Phi^{\tilde{T}_gP_{ag}}(u_i, \phi_{ag}, y_{ag}), i \in \Xi_k\} < 0 \).

(3) \( \text{int } \Lambda_q(z_q) \cap \text{int } P_{ag} = \emptyset \wedge (\text{fr } \Lambda_q(z_q) \cap \text{fr } P_{ag} \neq \emptyset) \iff \text{int } \Lambda_q(z_q) \cap \text{int } P_{ag} = \emptyset \wedge (\text{fr } \Lambda_q(z_q) \cap \text{fr } P_{ag} \neq \emptyset) \) and therefore \( \text{int } T_i(u_i) \cap \text{int } P_{ag} = \emptyset \), for all \( i \in \Xi_k \) and there exists at least one object \( T_i(u_i) \), such that \( \text{fr } T_i(u_i) \cap \text{fr } P_{ag} \neq \emptyset \), \( i \in \Xi_k \). It implies that \( \min \{\Phi^{\tilde{T}_gP_{ag}}(u_i, \phi_{ag}, y_{ag}), i \in \Xi_k\} = 0 \).

Thus, the function \( \Phi^{\tilde{T}_gP_{ag}}(u_i, \phi_{ag}, y_{ag}) \) defined in (11) is a phi-function for \( \tilde{\Lambda}_q \) and a half plane \( P_{ag} \).
Similarly a phi-function of $\bar{\Omega}_g(z_g)$ and a half plane $P_{gg}^*$ is given by:

$$\Phi^{P^*_{gg}}(z_g, \varphi_{gg}, y_{gg}) = \min \left\{ \Phi^{P^*_{gg}}(u_j, \varphi_{gg}, y_{gg}) \mid j \in \mathbb{Z}_{g} \right\},$$

where $\Phi^{P^*_{gg}}(u_j, \varphi_{gg}, y_{gg})$ is a phi-function of object $T_j(u_j)$, $j \in \mathbb{Z}_g$, and the half plane $P_{gg}^*$.

Thus, $\Phi^{P^*_{gg}}(z_g, \varphi_{gg}, y_{gg})$ is a phi-function for $\bar{\Omega}_g(z_g)$ and $\bar{\Omega}_g(z_g)$ is a normalized quasi-phi-function for $P_{gg}$.

Further we provide nonoverlapping tools (phi-functions) for a half plane $P_{gg} = \{(x, y) : \mu_{gg}(x, y) = \cos \varphi_{gg}x + \sin \varphi_{gg}y + y_{gg} \geq 0\}$ and the following types of objects $T_i(u_i)$, $i \in \mathbb{Z}_g$: an ellipse $E_i(u_i)$; a circle $C_i(v_i)$; and a convex polygon $K_i(u_i)$.

**Phi-Function for an Ellipse and Half Plane $P_{gg}$**. A phi-function for an ellipse $E_i(u_i)$ given by its semiaxes $(a_i, b_i)$ and half plane $P_{gg}$ can be defined in the form:

$$\Phi^{E_iP_{gg}}(u_i, \varphi_{gg}, y_{gg}) = x_i \cos \varphi_{gg} + y_i \sin \varphi_{gg} + y_{gg}$$

$$- \sqrt{(a_i^2 - b_i^2) \cdot \cos^2(\beta_i + \varphi_{gg}) + b_i^2}.$$ (14)

**Phi-Function for a Circle and Half Plane $P_{gg}$**. The phi-function for a circle $C_i(v_i)$ given by its radius $r_i$ and half plane $P_{gg}$ is defined in [24] and has the form:

$$\Phi^{C_iP_{gg}}(v_i, \varphi_{gg}, y_{gg}) = x_i \cos \varphi_{gg} + y_i \sin \varphi_{gg} + y_{gg} - r_i.$$ (15)

**Phi-Function for a Convex Polygon and Half Plane $P_{gg}$**. The phi-function for a convex polygon $K_i(u_i)$ given by its vertices $P_k = (p^x_k, p^y_k)$, $k = 1, \ldots, l$, and half plane $P_{gg}$ is defined in [24] and has the form:

$$\Phi^{K_iP_{gg}}(u_i, \varphi_{gg}, y_{gg}) = \min_{k=1-\ldots-l} \left\{ P_k^x \cos \varphi_{gg} + P_k^y \sin \varphi_{gg} + y_{gg} \right\}.$$ (16)

### 3.3. Containment of Clusters into a Container

Let us derive a phi-function representing analytically the containment constraint (3), $\bar{\Lambda}_g(z_g) \subset \Omega \iff \text{int} \bar{\Lambda}_g(z_g) \cap \Omega^* = \emptyset$ for each $q \in \mathbb{Z}_q$, where $\Omega^* = R^2 \setminus \text{int} \Omega$.

Let $\Phi^{\Omega^*}(u_i)$ be a phi-function of objects $T_i(u_i)$ and $\Omega^*$, $i \in \mathbb{Z}_q$.

**Proposition 4.** A continuous and everywhere defined function

$$\Phi^{\Lambda_0^*}(z_g) = \min \left\{ \Phi^{\Omega^*}(u_i), i \in \mathbb{Z}_q \right\}$$

is a phi-function of $\bar{\Lambda}_g(z_g)$ and the object $\Omega^*$.

**Proof.** We show that if $\bar{\Lambda}_g(z_g) \subset \Omega$ then $T_i \subset \Omega \iff \text{int} \bar{\Lambda}_g(z_g) \cap \Omega^* = \emptyset, i \in \mathbb{Z}_q$.

We assume that $\Omega^* = \Omega_1^* \cup \Omega_2^* \cup \Omega_3^* \cup \Omega_4^*$, where $\Omega_i^* = \{ (x, y) \mid g_i(x, y) \leq 0 \}$ is a half plane, $i = 1, 2, 3, 4$, $g_1(x, y) = x_1$, $g_2(x, y) = y_2$, $g_3(x, y) = -x + t$, and $g_4(x, y) = -y + w$.

For each half plane $\Omega_i^*$ we can conclude that

1. $\bar{\Lambda}_g(z_g) \cap \Omega_i^* = \emptyset \iff \Lambda_i(z_g) \cap \Omega_i^* = \emptyset$ and therefore $\text{int} \bar{\Lambda}_g(z_g) \cap \Omega_i^* = \emptyset$ for all $i \in \mathbb{Z}_q$;

2. $\text{int} \bar{\Lambda}_g \cap \text{int} \Omega_i^* = \emptyset \iff \text{int} \Lambda_i(z_g) \cap \text{int} \Omega_i^* = \emptyset$ and therefore there exists at least one object $T_i(u_i)$, such that $\text{int} T_i(u_i) \cap \text{int} \Omega_i^* = \emptyset, i \in \mathbb{Z}_q$.

3. $\text{int} \bar{\Lambda}_g(z_g) \cap \text{int} \Omega_i^* = \emptyset \iff (\text{fr} \bar{\Lambda}_g(z_g) \cap \text{fr} \Omega_i^* = \emptyset) \iff (\text{int} \Lambda_i(z_g) \cap \text{int} \Omega_i^* = \emptyset) \iff (\text{fr} \Lambda_i(z_g) \cap \text{fr} \Omega_i^* = \emptyset)$ and therefore $\text{int} T_i(u_i) \cap \text{int} \Omega_i^* = \emptyset$, for all $i \in \mathbb{Z}_q$, and there exists at least one object $T_i(u_i)$, such that $\text{fr} T_i(u_i) \cap \text{fr} \Omega_i^* = \emptyset, i \in \mathbb{Z}_q$.

Thus, $\bar{\Lambda}_g(z_g) \subset \Omega \iff \text{int} T_i(u_i) \cap \Omega_i^* = \emptyset, i \in \mathbb{Z}_q$. It means that if $\Phi^{\Lambda_0^*}(z_g) \geq 0$ if and only if $\Phi^{\Omega^*}(u_i) \geq 0$ for each $i \in \mathbb{Z}_q$.

Now we state by phi-functions containment conditions for the object $\Omega^*$ and the following types of objects $T_i(u_i)$, $i \in \mathbb{Z}_q$: an ellipse $E_i(u_i)$; a circle $C_i(v_i)$; and a convex polygon $K_i(u_i)$.

**Phi-Function for an Ellipse Containment**. A phi-function for an ellipse $E_i(u_i)$ given by its semiaxes $(a_i, b_i)$ and the object $\Omega^*$ can be defined in the form:

$$\Phi^{E_i\Omega^*}(u_i) = \min_{s=1-\ldots-l} g_i(u_s),$$

$$g_i(u_s) = x_i - \sqrt{b_i^2 + (a_i^2 - b_i^2) \cos^2 \theta_s},$$

$$g_{i2}(u_s) = y_i - \sqrt{b_i^2 + (a_i^2 - b_i^2) \sin^2 \theta_s}.$$
where \( \rho \) and \( \Phi \).

\[
\begin{align*}
g_3(u_i) &= l - x_i - \sqrt{b_i^2 + (a_i^2 - b_i^2) \cos^2 \theta_i}, \\
g_4(u_i) &= w - y_i - \sqrt{b_i^2 + (a_i^2 - b_i^2) \sin^2 \theta_i}.
\end{align*}
\tag{18}
\]

The inequality \( \Phi^E(u_i) \geq 0 \) guarantees that \( E_i(u_i) \in \Omega \). We define the phi-function based on the idea of nonoverlapping of an ellipse and half plane introduced in [5].

**Phi-Function for a Circle Containment.** A phi-function for a circle \( C(u_i) \) given by its radius \( r_i \) and the object \( \Omega^* \) has the form

\[
\Phi_{C,\Omega^*}(u_i) = \min_{s = 1, \ldots, 4} f_s(u_i),
\]

\[
\begin{align*}
g_1(u_i) &= x_i - r_i, \\
g_2(u_i) &= y_i - r_i, \\
g_3(u_i) &= -x_i + l - r_i, \\
g_4(u_i) &= -y_i + w - r_i.
\end{align*}
\tag{19}
\]

The inequality \( \Phi_{C,\Omega^*}(u_i) \geq 0 \) guarantees that \( C_i(u_i) \in \Omega \).

**Phi-Function for a Convex Polygon Containment.** A phi-function for a convex polygon \( K(u_i) \) given by its vertices \( p_{ik} = (x_{ik}, y_{ik}), k = 1, \ldots, N_i \), and the object \( \Omega^* \) has the form

\[
\Phi_{K,\Omega^*}(u_i) = \min_{s = 1, \ldots, 4} g_s^c(p_{ik}, p_{jk}^c),
\]

\[
\begin{align*}
g_1^c(u_i) &= x_i - r_i, \\
g_2^c(u_i) &= y_i - r_i, \\
g_3^c(u_i) &= -x_i + l - r_i, \\
g_4^c(u_i) &= -y_i + w - r_i.
\end{align*}
\tag{20}
\]

The inequality \( \Phi_{K,\Omega^*}(u_i) \geq 0 \) guarantees that \( K_i(u_i) \in \Omega \).

\[
\square
\]

### 4. Mathematical Model

A mathematical model for layout clusters of objects in a rectangular domain \( \Omega \) can be formulated in the form

\[
\max \sum_{(u_i, u_j, u_p)\in W} \rho,
\]

\[
W = \left\{ (u, \phi, \psi, \rho) \in \mathbb{R}^2 : \Phi_{C_i,\Omega^*}(z_{q}, z_{qr}, q_{gr}, y_{qr}) \right. \\
-0.5\rho \geq 0, q > g \in I_{N_i}, \rho \geq 0, \Phi^{T,T_i}(u_i, u_j, u_{ij}) \geq 0, i, j, (i, j) \in \Xi \times \Xi, q \in I_{N_i}, \Phi^{T,\Omega^*}(u_i) \geq 0, i \in I_n \}
\tag{21}
\]

where \( \rho \) is considered as a distance between convex hulls \( \overline{A}_q(z_q) \) and \( \overline{A}_q(z_q) \), \( u = (u_1, u_2, \ldots, u_n) \) is a vector of placement parameters, \( u_i = (v_i, \theta_i), v_i = (x_j, y_j), \psi = (q_{gr}, q_{qr}, q > g \in I_{N_i}) \) is a vector of auxiliary variables in a quasi-phi-function of convex hulls \( \overline{A}_q(z_q) \) and \( \overline{A}_q(z_q), \phi = (u_{ij}, (i, j) \in \Xi \times \Xi, q \in I_{N_i}) \) is vector of auxiliary variables in a quasi-phi-function of objects \( T_i(u_i) \) and \( T_j(u_j) \).

The feasible region \( W \) given by (22) is a phi-function of objects \( T_i(u_i) \) and \( T_j(u_j) \).

**Remark 5.** In the case of two circles \( C_i(u_i) \) and \( C_j(u_i) \) we use a phi-function \( \Phi^{C,\Omega^*}(u_i, u_j) \) instead of a quasi-phi-function \( \Phi^{T,\Omega^*}(u_i, u_j) \).

The feasible region \( W \) given by (22) is a phi-function of objects \( T_i(u_i) \) and \( T_j(u_j) \).

**Remark 6.** The inequality \( \Phi^{C,\Omega^*}(u_i) \geq 0 \) guarantees that \( C_i(u_i) \in \Omega \).

### 5. Solution Algorithm

Our solution strategy is based on the multistart algorithm and consists of three major stages: (1) generate a set of feasible starting points for the problems (21)-(22); (2) search for a set of local maxima for the problems (21)-(22) starting from each feasible point obtained at Stage (1); (3) choose the best local maxima from those found at Stage (2).

To generate feasible starting points of the problems (21)-(22) we develop a special algorithm CSPA (Cluster Starting Points Algorithm) that involves the following Steps.

**Step 1.** With each cluster \( A_q(z_q) \) we associate a circular region \( \lambda C_q \) with variable center point \( v_q = (x_q, y_q) \) and variable radius \( \lambda R_q \), where \( \lambda \) is a scaling parameter and \( R_q = \sqrt{\sum_{i \in I_q} S_i} \), \( S_i \) is the area of object \( T_i, i \in I_q, q \in I_N \).

**Step 2.** Then we grow up all circular regions \( \lambda C_q, q \in I_N \), within a rectangular container \( \Omega \) as much as possible, solving the following NLP subproblem:

\[
\max \lambda,
\]

\[
V = \left\{ (v, \lambda) \in \mathbb{R}^{2N+1} : \Phi_{C,\Omega^*}(v_q, \lambda) \geq 0, q > g \in I_{N}, \Phi^{T,\Omega^*}(v_q, \lambda) \geq 0, q \in I_{N}, \lambda \geq 0 \right\}
\tag{23}
\]

where \( v = (v_1, v_2, \ldots, v_N) \) is a vector of variable placement parameters of circular areas \( \lambda C_q, q \in I_N \);
\[ \Phi^{C \cup \Omega} (v_q, v_g, \lambda) \] is a phi-function of a circular area \( \lambda C_q \) and \( \lambda C_g \) with center points \( v_q = (x_q, y_q) \) and \( v_g = (x_g, y_g) \) and appropriate radii \( R_q \) and \( R_g, q > g \in I_N. \)

\[
\Phi^{C \cup \Omega} (v_q, v_g, \lambda) = \min_{x_q, y_q} f_q (v_q, \lambda),
\]

\[
f_q (v_q, \lambda) = x_q - \lambda R_q,
\]

\[
f_q (v_q, \lambda) = y_q - \lambda R_q,
\]

\[
f_q (v_q, \lambda) = l - x_q - \lambda R_q,
\]

\[
f_q (v_q, \lambda) = w - y_q - \lambda R_q.
\]

We search for a local maximum of problems (23)-(24) starting from a feasible point \((v^0 = (v^0_1, v^0_2, \ldots, v^0_N), \lambda^0 = 0)\), where \(v^0_i \in \Omega \) and \( q \in I_N \) are randomly generated points. Denote a local maximum point of problems (23)-(24) by \((v^*, \lambda^*).\)

**Step 3.** Next we derive starting values of variables \( \phi^0_q, \psi^0_g, \eta^0_g \) (obtained by trivial geometrical calculations); for each quasi-phi-function \( \Phi^{\lambda \times \lambda} (z_q, z_g, \phi_q, \psi_g, \eta_g, \beta) \), \( q > g \in I_N \) in the problems (21)-(22). These variables are considered as parameters of separating lines between each pair of circular areas \( \lambda^* C_q \) and \( \lambda^* C_g \) of radii \( \lambda^* R_q \) and \( \lambda^* R_g \) with center points \( v_q^* \) and \( v_g^*, q > g \in I_N. \)

**Step 4.** Then we search for feasible points of the problems (21)-(22) solving the following NLP subproblem:

\[
G = \{(u, \phi, \psi, \beta) \in R^4 : \Phi^{\lambda \times \lambda} (z_q, z_g, \phi_q, \psi_g, \eta_g, \beta) \\
\geq 0, q > g \in I_N, 0 \leq \beta \leq 1, \Phi^{IT} (u, i, u_i, i_g, \beta) \geq 0 \}
\]

\[

where \( u = (u_1, u_2, \ldots, u_n), u_i = (v_i, \theta_i, v_i = (x_i, y_i), \) \( \psi = (\phi_g, \eta_g, \gamma_q > g \in I_N); \beta \) is a scaling parameter of our objects; \( \Phi^{IT} (u, i, j) \in \Xi_q \times \Xi_q \) is a vector of auxiliary variables in a quasi-phi-function of objects \( \beta T_i(u_i) \) and \( \beta T_j(u_j); \Phi^{IT} (u, i, u_i, i_g, \beta) \) is a quasi-phi-function of objects \( \beta T_i(u_i) \) and \( \beta T_j(u_i); \Phi^{IT} (u, i) \) is a phi-function of auxiliary variables.

**Remark 6.** Application of the decomposition algorithm described in [3] is recommended to solve the problem (21)-(22), when \( m_q \geq 11 \) for clusters \( \Lambda_q(x_q), q \in I_N, \) in order to reduce the computational costs (time and memory).

**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++, Windows 7. For the local optimisation we use the IPOPT code (https://projects.coin-or.org/Ipopt), developed in [27]. Default options were used for running this software.

We present new instances for the layout problem in a rectangular container \( \Omega = \{(x, y) \in R^2 : 0 \leq x \leq 1, 0 \leq y \leq 15\}. \) We run our program 100 times for each example.

**Example 1.** An ordered collection of \( n = 30 \) ellipses is given. The collection of ellipses is divided into \( N = 3 \) clusters \( \Lambda_q(z_q), q = 1, 2, 3 : m_1 = 10, m_2 = 10, \) and \( m_3 = 10. \)

All ellipses are defined by their sizes \( \{a_i, b_i, i = 1, \ldots, 30\} = \{(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), (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), (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), \}

**Our Result**

**Placement Parameters of Ellipses.** \( \{(x_1, y_1, \theta_1, \ldots, x_3, y_3, \theta_3, \ldots, x_{30}, y_{30}, \theta_{30}\} = \{(10.469994, 12.267537, 2.244218), (7.279733, 12.352075, 2.063582), (2.471444, 12.548774, 13.249926), (8.464111, 9.691723, 0.145847), (12.920038, 11.433712, 5.530417), (4.909393, 12.890715, 1.640578), (4.716355, 10.894489, 0.195532), (13.092110, 13.353890, 6.393843), (10.806375, 9.993924, -0.401328), (6.267221, 10.149879, 6.032813), (0.813462, 5.582920. \)
Figure 1: Local optimal arrangement of clusters in Example 1.

Figure 2: Local optimal arrangement of clusters in Example 2.

1.582319, (2.129866 4.016161 2.043699), (4.516627 2.106510 5.713416), (2.014535 8.617626 4.606157), (5.875381 7.080877 1.710554), (5.246435 4.663067 0.181324), (3.712908 6.439680 3.845887), (1.613133 1.313706 6.313737), (4.250763 8.301450 4.942235), (1.768064 6.435263 0.065954), (13.527926 1.206001 2.906999), (10.087641 4.308896 2.501728), (12.390656 5.699737 2.716022), (8.409176 2.472279 4.931800), (9.003824 5.818192 1.638294), (13.004545 2.937130 5.336177), (13.095024 8.392657 -0.572537), (10.944860 1.726727 0.720864), (10.969680 7.572605 5.373848), (10.898378 3.513018 -0.101262).

The value of objective function is $\rho^*=2.109291$, while the computational time is 7131.446 sec. Corresponding optimized layout is presented in Figure 1.

Example 2. An ordered collection of $n=36$ circles is given. The collection of circles is divided into $N=4$ clusters $\Lambda_i(z_q)$, $q=1,2,3,4$: $m_1=10$, $m_2=10$, $m_3=8$, and $m_4=8$.

All circles are defined by their radii:

\[
\{(r_i), i = 1, \ldots, 36 \} = \{(1.495), (1.500), (1.501), (1.206), (1.009), (0.912), (0.815), (0.765), (0.621), (0.524), (0.336), (1.550), (1.533), (1.512), (1.239), (1.042), (0.945), (0.848), (0.7925), (0.654), (0.557), (0.380), (1.605), (1.566), (1.523), (1.272), (1.075), (0.978), (0.881), (0.820), (0.687), (0.590), (0.424), (1.660), (1.599), (1.534)\}
\]

Our Result

Placement Parameters of Circles. $\{(x_i, y_i), i = 1, \ldots, 36\} = \{(13.089089, 1.910911), (9.282866, 1.915911), (11.652953, 4.556297), (13.378097, 7.152616), (9.147148, 4.421237), (11.269926, 6.948703), (9.688888, 6.188978), (13.819089, 5.231585), (13.954597, 3.846912), (11.189529, 2.595038), (3.381819, 14.237068), (4.209528, 11.030014), (6.318116, 8.780847), (1.927911, 13.072089), (2.506044, 8.821697), (6.003824, 12.979437), (1.360911, 10.681408), (6.604792, 11.144527), (4.222026, 13.460604), (4.427002, 8.073882), (3.849810, 0.972911), (4.971098, 4.766312), (5.740826, 2.020911), (1.981911, 1.981911), (1.938911, 5.681867), (6.544523, 5.269738), (3.814234, 3.883877), (4.418702, 6.006885), (13.509999, 9.753854), (11.287589, 11.489670), (11.993830, 10.158402), (13.994089, 11.147396), (11.771968, 9.049173), (9.264512, 12.924089), (9.777002, 9.600308), (13.050089, 13.050089)\}

The value of objective function is $\rho^*=0.415912$, while the computational time is 484.461 sec. Corresponding optimized layout is presented in Figure 2.

Example 3. An ordered collection of $n=36$ objects is given. The collection of ellipses is divided into $N=4$ clusters $\Lambda_i(z_q)$, $q=1,2,3,4$: $m_1=10$ circles, $m_2=10$ circles, $m_3=8$ ellipses, and $m_4=8$ ellipses.

All circles are defined by their radii:

\[
\{(r_i), i = 1, \ldots, 20 \} = \{(1.5), (1.5), (1.5), (1.2), (1.0), (0.9), (0.8), (0.75), (0.6), (0.5), (0.3), (1.5), (1.5), (1.5), (1.5), (1.0), (0.9), (0.8), (0.75), (0.6)\}
\]

All ellipses are defined by their semi axes:

\[
\{(a_i, b_i), i = 21, \ldots, 36 \} = \{(0.8, 0.5), (0.7, 0.3), (2.0, 1.5), (1.8, 1.5), (1.6, 1.5), (1.5, 1.2), (1.3, 1.0), (1.2, 0.9), (1.1, 0.8), (1.0, 0.75), (0.9, 0.6), (0.8, 0.5), (0.7, 0.3), (2.0, 1.5), (1.8, 1.5), (1.6, 1.5)\}
\]

Our Result

Placement Parameters of Circles. $\{(x_i, y_i), i = 1, \ldots, 36\} = \{(5.969333, 5.831559), (1.910221, 4.910224), (1.910221, 1.910223), (4.593503, 1.601221), (3.633171, 6.721701), (3.920715, 3.599532), (5.620213, 3.558210), (1.160222, 7.031545), (4.001869, 5.097335), (6.120123, 2.358173), (11.629726, 8.901063), (13.089779, 13.089779), (13.089779, 10.089779), (8.950916, 11.15733), (10.406497, 13.389779), (7.836995, 8.063426), (9.688425, 8.490281), (10.812773, 9.765367), (11.294872, 11.568861), (7.474932, 9.621922)\}$.
Our Result

Placement Parameters of Circles. \[(x_i, y_i) = 1, \ldots, 20\] = \{(5.025933 1.876324), (1.826324 1.826324), (3.974219 6.518053), (6.563760 7.583078), (3.350845 3.993889), (1.451364 5.929808), (6.228908 4.078766), (1.109501 4.136463), (6.051897 5.654399), (4.881146 4.620133), (2.547234 11.99963), (1.876324 13.109992), (1.826324 9.360637), (7.011083 11.353004), (4.211329 11.390181), (8.183466 13.673676), (4.390190 13.773676), (6.234106 13.623755), (4.213121 9.302682), (5.668491 9.610177)\}.

Example 5. An ordered collection of \(n = 36\) objects is given. The collection of objects is divided into \(N = 4\) clusters \(\Lambda_i(x_i), i = 1, 2, 3, 4\): \(m_1 = 10\) convex polygons, \(m_2 = 10\) circles, \(m_3 = 8\) ellipses, and \(m_4 = 8\) ellipses.

All polygons are defined by their vertices:
\[
(P_{0,k}^{x}, P_{0,k}^{y}, k = 1, \ldots, 6, i = 1, \ldots, 10) = \{(1.975377 0.234652, 0.716736 1.400371, -1.285641 1.657199, -1.975377 -0.234652, -0.716736 -1.400371, 1.285641 -1.657199), (1.711902 0.436525, 0.374241 1.467221), (-1.337661 1.003696, -1.711902 -0.436525, -0.374241 -1.467221), (1.337661 -1.003696), (0.425610 0.680986, 0.083738 1.497944, -1.341873 0.816959, -0.680986, -0.083738 -1.497944, 1.341873 -0.816959), (0.213252 0.705342, -0.156793 1.193426, -1.370318 0.488084, -1.213523 -0.705342, 0.156793 -1.193426, 1.370318 -0.488084, (0.919239 0.707107, -0.336465 0.965926, -1.255704 0.258819, -0.919239 -0.707107, 0.336465 -0.965926, 1.255704 -0.258819), (0.705342 -0.488084 0.822191, -1.193426 0.094076, -0.705342 -0.728115, 0.488084 -0.822191, 1.193426 -0.094076), (0.499390 0.712805, -0.599103 0.670935, -0.109892 -0.014869, 0.499390 -0.712805, 0.599103 -0.670935, 1.098492 0.014869), (0.309017 0.713922, -0.669131 0.557359, -0.978148 -0.155934, -0.309017 -0.713922, 0.669131 -0.557359, 0.978148 0.155934), (0.140791 0.592613, -0.699431 0.377592, -0.840222 -0.215021, (0.840222 0.215021), (0.062767 0.498459, -0.283203, (0.062767 -0.498459, -0.659301 0.283203), (0.722068 0.215256), 0.722068 0.215256\}.

All circles are defined by their radii:
\[
(r_i, i = 11, \ldots, 20) = \{(0.336, 1.55, 1.533, 1.512, 1.239, 1.042, 0.945, 0.848, 0.7925, 0.654\}.

All ellipses are defined by their semi-axes:
\[(a_i, b_i) = 21, \ldots, 36\] = \{(0.8 0.5), (0.7 0.3), (2.0 1.5), (1.8 1.5), (1.6 1.5), (1.5 1.2), (1.2 0.9), (1.1 0.8), (1.0 0.75), (0.9 0.6), (0.8 0.5), (0.7 0.3), (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), (2.0 1.5), (1.8 1.5), (1.6 1.5)\}.
Our Result

**Placement Parameters of Polygons.** \( \{(x_i, y_i, \theta_i), i = 1, \ldots, 10\} = \{(3.447806, 9.597127, 2.606907), (4.632828, 12.154203, 6.566083), (6.375060, 2.896635, 0.292333), (4.062287, 14.055842, 3.636614), (7.862083, 9.559287, 6.671758)\} \)

**Placement Parameters of Circles.** \( \{(x_i, y_i), i = 11, \ldots, 20\} = \{(4.242145, 4.607830), (11.278663, 1.905800), (8.391616, 6.969211), (9.039026, 3.993831), (11.103865, 5.862139), (13.546072, 3.321421), (13.699200, 1.300800), (8.895716, 1.203801), (12.126975, 5.194957), (11.978464, 4.076973)\} \)

**Placement Parameters of Ellipses.** \( \{(x_i, y_i, \theta_i), i = 21, \ldots, 36\} = \{(3.64227, 1.15546, 1.33947), (1.038618, 8.531725, 0.246199), (2.256484, 6.107240, 0.489860), (5.709960, 1.865204, 0.435077), (3.536840, 4.958834, 1.597596), (1.798943, 6.23608, 0.581350), (1.423676, 4.10226, 7.386076), (3.423022, 3.445736, 3.374109), (9.031251, 9.996600, 5.101874), (11.636070, 11.136792, 2.665959), (13.383614, 11.23564, 2.678929), (13.844824, 7.219741, 6.338144), (14.343049, 10.348285, 1.605276), (12.459639, 8.974386, 0.003111), (9.963431, 12.848701, 1.699000), (13.051879, 13.13605, 10.285045)\} \)

The value of objective function is \( \rho^* = 0.355801 \), while the computational time is 4625.15 sec.

**Example 6.** An ordered collection of \( n = 30 \) objects is given.

The collection of objects is divided into \( N = 3 \) clusters \( \Lambda_q(z_q), q = 1, 2, 3 : m_1 = 10 \) convex polygons, \( m_2 = 10 \) circles, and \( m_3 = 10 \) ellipses.

All polygons are defined by their vertices:

\[
\{((p_{x_{1k}}, p_{y_{1k}}), k = 1, \ldots, 6), i = 1, \ldots, 10\} = \{(2.000000, 0.000000), (1.000000, 1.732051), (-1.000000, 1.732051), (-2.000000, 0.000000), (-1.000000, -1.732051), (1.000000, -1.732051)\},
\{(1.800000, 0.000000), (0.900000, 1.558846), (-0.900000, 1.558846), (-1.800000, 0.000000), (-0.900000, -1.558846), (0.900000, -1.558846)\},
\{(1.700000, 0.000000), (0.850000, 1.472243), (-0.850000, 1.472243), (-1.700000, 0.000000), (-0.850000, -1.472243), (0.850000, -1.472243)\},
\{(1.600000, 0.000000), (0.800000, 1.385641), (-0.800000, -1.385641), (0.800000, 1.385641), (1.400000, 0.000000), (0.700000, 1.212436), (-0.700000, 1.212436), (-1.400000, 0.000000), (-0.700000, -1.212436), (0.700000, -1.212436), (1.300000, 0.000000), (0.650000, 1.125833), (-0.650000, 1.125833), (-1.300000, 0.000000), (-0.650000, -1.125833), (0.650000, -1.125833), (1.200000, 0.000000), (0.600000, 1.039230), (-0.600000, 1.039230), (-1.200000, 0.000000), (-0.600000, -1.039230), (0.600000, -1.039230), (1.100000, 0.000000), (0.550000, 0.952628), (-0.550000, 0.952628), (-1.100000, 0.000000), (-0.550000, -0.952628), (0.550000, -0.952628), (1.000000, 0.000000), (0.500000, 0.866025), (-0.500000, 0.866025), (0.500000, -0.866025), (0.000000, 0.000000), (0.450000, 0.779423), (-0.450000, 0.779423), (-0.900000, 0.000000), (-0.450000, -0.779423), (0.450000, -0.779423)\}\}

All circles are defined by their radii:

\[
\{(r_i), i = 11, \ldots, 20\} = \{0.336, 1.55, 1.533, 1.524, 1.513, 1.512, 1.242, 1.23, 1.048, 0.951, 0.854\} \)

All ellipses are defined by their semi axes:

\[
\{(a_i, b_i), i = 21, \ldots, 30\} = \{(0.8 0.5), (0.7 0.3), (2.0 1.5), (1.9 1.6), (1.8 1.5), (1.6 1.5), (1.5 1.2), (1.3 1.0), (1.2 0.9), (1.1 0.8)\} \)

Our Result

**Placement Parameters of Polygons.** \( \{(x_i, y_i, \theta_i), i = 1, \ldots, 10\} = \{(13.066728, 7.204303, 2.567940), (7.917914, 11.613054)\} \)
Placement Parameters of Ellipses.

Placement Parameters of Circles.

Placement Parameters of Ellipses.

Figure 6: Local optimal arrangement of clusters in Example 6.

The value of objective function is \( \rho^* = 0.213358 \), while the computational time is 7986.627 sec.

7. Conclusions

We consider a new packing (layout) problem for a number of clusters of convex objects. This research is motivated by a container-loading problem; however, we think that similar problems arise naturally in many other packing/cutting/clustering issues. The layout is considered for a rectangular container of a given size subject to nonoverlapping of (continuously translated and rotated) objects within a cluster. The objects are specified by their sizes and have the same shape for the appropriate cluster. Each cluster is represented by the convex hull of objects that form this cluster. It is assumed that two clusters do not overlap each other if so are their convex hulls and a cluster belongs to a rectangular container if the corresponding convex hull does. New tools of mathematical modeling of nonoverlapping and containment for clusters are provided. An extension of the phi-function technique to the cluster convex hulls is presented taking into account variability of their shapes and metrical characteristics. New quasi-phi-functions and phi-functions are introduced. These functions avoid constructing convex hull for each cluster. A novel mathematical model for optimized layout of clusters is considered and stated as a nonlinear continuous problem. A novel algorithm for searching for initial feasible solutions to accelerate a local optimization procedure is developed. Computational results are provided to demonstrate the efficiency of our approach. The computational results are presented for new instances of the layout problem for clusters involving ellipses, circles, and convex polygons.

Appendix

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 \( u_A = (x_A, \theta_A) \), where \( x_A = (x_A, y_A) \) is a translation vector and \( \theta_A \) is a rotation angle. The object \( A \), rotated by an angle \( \theta_A \) and translated by a vector \( v_A \), is denoted by \( A(u_A) \).

Definition [see [23]]: 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

\[
\begin{align*}
\Phi^{AB}(u_A, u_B) < 0, & \quad \text{if } \int A(u_A) \cap \int B(u_B) \neq \emptyset; \\
\Phi^{AB}(u_A, u_B) = 0, & \quad \text{if } \int A(u_A) \cap \int B(u_B) = \emptyset \quad \text{and} \quad \mathfrak{fr}A(u_A) \cap \mathfrak{fr}B(u_B) \neq \emptyset; \\
\Phi^{AB}(u_A, u_B) > 0, & \quad \text{if } A(u_A) \cap B(u_B) = \emptyset.
\end{align*}
\]

(1.1)

Phi-functions allow us to distinguish the following three cases: \( A(u_A) \) and \( B(u_B) \) intersect 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 tangent; i.e., \( A(u_A) \) and \( B(u_B) \) have only common frontier points.

The inequality \( \Phi^{AB}(u_A, u_B) \geq 0 \) describes the nonoverlapping constraint, i.e., \( \int A(u_A) \cap \int B(u_B) = \emptyset \), and the inequality \( \Phi^{AB}(u_A, u_B) \leq 0 \) describes the containment constraint \( A(u_A) \subset B(u_B) \), i.e., \( \int A(u_A) \cap \mathfrak{fr}B(u_B) = \emptyset \), where \( B^* = R^2 \setminus \int B \).

Definition [see [23]]: A phi-function \( \overline{\Phi}^{AB}(u_A, u_B) \) is called a normalized phi-function of \( A(u_A) \) and \( B(u_B) \) if its values coincide with the Euclidian distance between the objects \( A(u_A) \) and \( B(u_B) \), provided that \( \int A(u_A) \cap \int B(u_B) = \emptyset \).

Let \( \rho \) be the minimum allowable distance between objects \( A \) and \( B \), then \( \overline{\Phi}^{AB} \geq \rho \iff \text{dist}(A, B) \geq \rho \). Here, \( \text{dist}(A, B) = \min_{a \in A, b \in B} d(a, b) \) and \( d(a, b) \) is the Euclidean distance between two points \( a \) and \( b \) in \( R^2 \).
**Definition [see [3]].** A continuous and everywhere defined function \( \Phi^{AB}(u_A, u_B, u') \) is called a quasi-phi-function for the objects in a rectangular container, if \( \Phi^{AB}(u_A, u_B, u') \geq 0 \) for some \( u' \), then \( A(u_A) \cap B(u_B) = \emptyset \).

**Definition [see [3]].** Function \( \Phi^{AB}_{\text{normalized}}(u_A, u_B, u') \) is called a normalized quasi-phi-function for the objects in a rectangular container, if function \( \max_{u'} \Phi^{AB}_{\text{normalized}}(u_A, u_B, u') \) is a normalized phi-function for the objects.

**Data Availability**

The data [input data for each example: the number of clusters, sizes, and numbers of objects that form each cluster; output data for each example: the value of objective function, computational time, and placement parameters of objects] used to support the findings of this study are included within the article.

**Conflicts of Interest**

The authors declare that there are no conflicts of interest regarding the publication of this paper.

**Acknowledgments**

Research of the third author was partially supported by CONACYT (Mexico) [167019,293403].

**References**

[1] B. Chazelle, H. Edelsbrunner, and L. J. Guibas, “The complexity of cutting complexes,” *Discrete & Computational Geometry*, vol. 4, no. 2, pp. 139–181, 1989.

[2] I. Litvinchey, L. Infante, and L. Ozuna, “Approximate packing: Integer programming models, valid inequalities and nesting,” in *Optimized Packings with Applications*, G. Fasano and J. D. Pinter, Eds., vol. 105 of *Springer Optimization and Its Applications*, pp. 187–205, Springer, 2015.

[3] Y. Stoyan, A. Pankratov, and T. Romanova, “Quasi-phi-functions and optimal packing of ellipses,”* Journal of Global Optimization*, vol. 65, no. 2, pp. 283–307, 2016.

[4] A. Pankratov, T. Romanova, and I. Litvinchey, “Packing ellipses in an optimized rectangular container,” *Wireless Networks*, pp. 1–11, 2018.

[5] J. Kalilrath and S. Rebennack, “Cutting ellipses from area-minimizing rectangles,” *Journal of Global Optimization*, vol. 59, no. 2, pp. 405–437, 2014.

[6] E. G. Birgin, R. D. Lobato, and J. M. Martinez, “Packing ellipsoids by nonlinear optimization,” *Journal of Global Optimization*, vol. 65, no. 4, pp. 709–743, 2016.

[7] P. I. Stetsyuk, T. E. Romanova, and G. Scheithauer, “On the global minimum in a balanced circular packing problem,” *Optimization Letters*, vol. 10, no. 6, pp. 1347–1360, 2016.

[8] Z. Fu, W. Huang, and Z. Lü, “Iterated tabu search for the circular open dimension problem,” *European Journal of Operational Research*, vol. 225, no. 2, pp. 236–243, 2013.

[9] Y. Stoyan and G. Yaskov, “Packing equal circles into a circle with circular prohibited areas,” *International Journal of Computer Mathematics*, vol. 89, no. 10, pp. 1355–1369, 2012.

[10] H. Akeb, M. Hifi, and S. Negre, “An augmented beam search-based algorithm for the circular open dimension problem,” *Computers & Industrial Engineering*, vol. 61, no. 2, pp. 373–381, 2011.

[11] I. Litvinchey, L. Infante, and L. Ozuna, “Packing circular-like objects in a rectangular container,” *Journal of Computer and Systems Sciences International*, vol. 54, no. 2, pp. 259–267, 2015.

[12] J. Peralta, M. Andretta, and J. F. Oliveira, “Packing Circles and Irregular Polygons using Separation Lines,” in *Proceedings of the International Conference on Operations Research and Enterprise Systems (ICORES)*, pp. 71–77, Funchal, Madeira, Portugal, 2018.

[13] J. Kalilrath, “Cutting circles and polygons from area-minimizing rectangles,” *Journal of Global Optimization*, vol. 43, no. 2-3, pp. 299–328, 2009.

[14] H. Alt and F. Hurtado, “Packing convex polygons into rectangular boxes,” in *Proceedings of the Japanese Conference on Discrete and Computational Geometry (JDCG)*, vol. 2098 of *Lecture Notes in Computer Science*, pp. 67–80, Springer, Tokyo, Japan, 2000.

[15] H. Alt, M. de Berg, and C. Knauer, “Approximating minimum-area rectangular and convex containers for packing convex polygons,” *Journal of Computational Geometry*, vol. 8, no. 1, pp. 1–10, 2017.

[16] K. Tang, C. C. Wang, and D. Z. Chen, “Minimum area convex packing of two convex polygons,” *International Journal of Computational Geometry & Applications*, vol. 16, no. 1, pp. 41–74, 2006.

[17] D. R. Jones, “A fully general, exact algorithm for nesting irregular shapes,” *Journal of Global Optimization*, vol. 59, no. 2-3, pp. 367–404, 2014.

[18] F. M. B. Toledo, M. A. Caravilla, C. Ribeiro, J. F. Oliveira, and A. M. Gomes, “The dotted-board model: A new MIP model for nesting irregular shapes,” *International Journal of Production Economics*, vol. 145, no. 2, pp. 478–487, 2013.

[19] Y. Stoyan, A. Pankratov, and T. Romanova, “Cutting and packing problems for irregular objects with continuous rotations: Mathematical modelling and non-linear optimization,” *Journal of the Operational Research Society*, vol. 67, no. 5, pp. 786–800, 2016.

[20] Y. Stoyan, A. Pankratov, and T. Romanova, “Placement problems for irregular objects: Mathematical modeling, optimization and applications,” in *Chapter in book Optimization Methods and Applications*, Butenko, Sergiy, Pardalos, M. Panos, Shylo, and Volodymyr, Eds., vol. 180 of *Springer Optimization and Its Applications Series*, p. 612, Springer, 2017.

[21] J. Bennell, G. Scheithauer, Y. Stoyan, T. Romanova, and A. Pankratov, “Optimal clustering of a pair of irregular objects,” *Journal of Global Optimization*, vol. 61, no. 3, pp. 497–524, 2015.

[22] A. Bortfeldt and G. Wäscher, “Constraints in container loading—a state-of-the-art review,” *European Journal of Operational Research*, vol. 229, no. 1, pp. 1–20, 2013.
[23] Y. Stoyan and T. Romanova, “Mathematical models of placement optimization: Two- and three-dimensional problems and applications,” in *Modeling and Optimization in Space Engineering*, G. Fasano and J. D. Pinter, Eds., vol. 73 of *Springer Optimization and Its Applications*, p. 404, Springer, 2012.

[24] N. Chernov, Yu. Stoyan, T. Romanova, and A. Pankratov, "Phi-functions for 2D objects formed by line segments and circular arcs,” *Advances in Operations Research*, vol. 2012, Article ID 346358, 26 pages, 2012.

[25] Y. Stoyan, T. Romanova, A. Pankratov, and A. Chugay, "Optimized object packings using quasi-phi-functions,” in *Optimized Packings and Their Applications*, G. Fasano and J. D. Pinter, Eds., vol. 105 of *Springer Optimization and Its Applications*, pp. 265–293, Springer, 2015.

[26] T. Romanova, J. Bennell, Y. Stoyan, and A. Pankratov, "Packing of concave polyhedra with continuous rotations using nonlinear optimisation,” *European Journal of Operational Research*, vol. 268, no. 1, pp. 37–53, 2018.

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