

Optimized Packings in Space Engineering Applications: Part II



Yu. Stoyan, I. Grebennik, T. Romanova, and A. Kovalenko

Abstract This chapter, dedicated to a specific packing optimization scenario of considerable interest in space engineering and logistics, follows a previous one appearing in this volume [1]. Although it is presented as the second part of the whole topical discussion proposed, it can be read independently.

The layout optimization, with balancing conditions, of a given set of 3D-objects, in a container partitioned by horizontal planes into subcontainers, is considered.

We define special combinatorial configurations describing the specific structure of the problem. A mathematical model, based on the combination of the phi-function technique and the introduced configurations, is provided. The model takes into account not only the placement constraints (i.e., nonoverlapping, containment) and the mechanical characteristics of the system but also the combinatorial features relevant to the partitions of the set of objects placed inside the subcontainers. The solution strategy is proposed and the results of numerical experiments are presented.

1 Introduction

Layout instances with balancing conditions belong to the class of NP-hard problems and are a subject of study in computational geometry and operations research [2, 3], where the methods adopted for their solution represent quite a recent branch.

The essence of the problem lies in the search for the optimal placement of a given set of 3D-objects in a container, ensuring the balancing of the system under consideration.

Y. Stoyan · T. Romanova (✉)

Department of Mathematical Modeling and Optimal Design, Institute for Mechanical Engineering Problems of the National Academy of Sciences of Ukraine, Kharkov, Ukraine
e-mail: tarom27@yahoo.com

I. Grebennik · A. Kovalenko

Department of System Engineering, Kharkov National University of Radio Electronics, Kharkov, Ukraine

© Springer Nature Switzerland AG 2019

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_16

439

The necessity of taking into account the assigned balancing constraints in optimization layout problems arises in various applied science fields and technologies, such as, for example, in engineering concerning the design of aircraft, ships, submarines, devices, and components, in logistics problems (when packing goods for transportation or storage) [4, 5]. Rocket design and space technology represent an area of particular interest in this class of problems.

At the initial stages of the layout definition of a spacecraft, it is necessary to take into account special constraints on static and dynamic characteristics (center of mass, axial and centrifugal moments of inertia), as described in [6].

In [7–10], the problems of the layout of cylinders in a cylindrical container with balancing constraints are considered. These publications provide mathematical models with different objective functions. To solve these problems, heuristic algorithms, based on the specific features of each mathematical model, are proposed. Papers [11–13] consider mathematical models and methods for solving the layout problem of a given set of objects, with balancing conditions.

In [3], the authors study (NP-hard) placement optimization problems, which cover a wide spectrum of industrial applications, including space engineering. The present chapter considers mathematical modelling tools and a solution strategy for placement problems. A class of 2D/3D geometric objects, called *phi*-objects, is introduced and considered as mathematical models of real items. The concept of *phi*-functions is used to describe nonoverlapping and containment constraints. A mathematical model of a basic placement issue is constructed as a constrained optimization problem that takes into account allowable distances between objects. A solution strategy is proposed. As an example, a placement optimization problem with balancing conditions arising in space engineering is considered. This consists in the placement of cylinders and cuboids of given weights and sizes in a parabolic container, divided by parallel axial circles minimizing the deviation of the system center of gravity from a given point. The chapter also provides a number of computational results for 2D and 3D applications.

Paper [13] studies the layout optimization problem, called BLP, of 3D-objects (solid spheres, straight circular cylinders, spherocylinders, straight regular prisms, cuboids, and tori) in a container (with cylindrical, parabolic, or truncated conical shapes) partitioned into sectors by parallel axial circles. The problem takes into account given minimum and maximum allowable distances between objects, as well as balancing conditions in terms of equilibrium, moments of inertia, and stability constraints. A continuous nonlinear programming (NLP) model of the problem is developed using the *phi*-function technique. The abovementioned paper also considers several BLP variants, providing appropriate mathematical models and solution algorithms, based on nonlinear programming and nonsmooth optimization methods, as well as the relevant computational analysis. In this work, however, the assignment of objects to the container sectors is assumed to be established a priori.

The innovative contribution of this chapter relates to the following points:

1. We extend the formulation of the layout optimization problem discussed in [13]. Our new formulation, called CBLP, takes into account not only placement

and balancing constraints of the system but also combinatorial features of the problem. These consist, in particular, in the assignment (no longer established a priori) of the given items to the system sectors.

2. We investigate the concept of combinatorial configurations to handle the discrete structure of the CBLP problem.
3. We define mathematical modelling tools for placement constraints, with both continuous and discrete variables, called D-phi-function and quasi-D-phi-function.
4. We provide a mathematical model formulation of the CBLP problem that involves both continuous and discrete variables.
5. We propose a solution strategy that uses the novel algorithm for the combinatorial configuration generation.

2 Problem Formulation

Let Ω be a container of height H that can take the form of a cuboid, cylinder, paraboloid of rotation, or truncated cone. The container Ω is defined in the fixed coordinate system $Oxyz$, where Oz is the longitudinal axis of symmetry. We assume that container Ω is divided by horizontal planes into subcontainers Ω^j , $j \in J_m = \{1, \dots, m\}$. We denote distances between circles S_j and S_{j+1} by t_j , $j \in J_m$, $\sum_{j=1}^m t_j = H$. The center of the base of container Ω is located in the origin of the coordinate system $Oxyz$.

Let $A = \{\mathbb{T}_i, i \in J_n\}$ be a set of homogeneous 3D-objects given by their metrical characteristics. Each object \mathbb{T}_i of height h_i and mass m_i is defined in its local coordinate system $O_i x_i y_i z_i$, $i \in J_n$. The location of object \mathbb{T}_i inside container Ω is defined by vector $u_i = (v_i, z_i, \theta_i)$, where (v_i, z_i) is a translation vector in the coordinate system $Oxyz$, θ_i is a rotation angle of object \mathbb{T}_i in the plane $O_i x_i y_i$, where $v_i = (x_i, y_i)$, and the value of z_i , $i \in J_n$, is uniquely defined by subcontainer Ω^j , $j \in J_m$, in which object \mathbb{T}_i will be placed.

In the BLP problem, the requirement for placing objects in specific subcontainers Ω^j , $j \in J_m$, is known a priori. In this study, the issue of the balanced layout of objects is formulated, considering the generation and selection of a partition of the set A into nonempty subsets A^j , $j \in J_m$. Here, A^j is a subset of objects which have to be placed on circle S_j inside subcontainer Ω^j .

Regarding the placement of object \mathbb{T}_i , $i \in J_n$, inside subcontainer Ω^j the following constraints are imposed:

$$z_i = \sum_{l=1}^j t_{l-1} + h_i, \tag{1}$$

where $j \in J_m$. We consider that $t_0 = 0$ and $\forall i \in J_n$ there exists $j^* \in J_m$: $h_i \leq t_{j^*}$.

Let $J_n^j \subseteq J_n$ be a set of indexes of objects which are placed in subcontainer Ω^j , $j \in J_m$:

$$\bigcup_{j=1}^m J_n^j = J_n, \quad J_n^i \cap J_n^j = \emptyset, \quad i \neq j \in J_m; \tag{2}$$

$k_j = |A^j|$ is the number of objects which are placed in subcontainer Ω^j , $k_j > 0$, $j \in J_m$:

$$\sum_{j=1}^m k_j = n. \tag{3}$$

In addition, the following placement constraints have to be taken into account:

$$\text{int}\mathbb{T}_{i_1} \cap \text{int}\mathbb{T}_{i_2} = \emptyset, \quad i_1 < i_2 \in J_n^j, \quad j \in J_m, \tag{4}$$

$$\mathbb{T}_i \subseteq \Omega^j, \quad i \in J_n^j, \quad j \in J_m, \tag{5}$$

$$h^j \leq t_j, \quad h^j = \max \left\{ h_i^j, i \in J_n^j \right\}, \quad j \in J_m. \tag{6}$$

We designate a system, formed as a result of the placement of objects \mathbb{T}_i of the set A in container Ω by Ω_A , and a reference frame of Ω_A by O_sXYZ , where $O_s = (x_s(v), y_s(v), z_s(v))$ is the mass center of Ω_A :

$$x_s(v) = \frac{\sum_{i=1}^n m_i x_i}{M}, \quad y_s(v) = \frac{\sum_{i=1}^n m_i y_i}{M}, \quad z_s = \frac{\sum_{i=1}^n m_i z_i}{M}, \tag{7}$$

$M = \sum_{i=1}^n m_i$ is the mass of system Ω_A and $O_sX \parallel Ox$, $O_sY \parallel Oy$, $O_sZ \parallel Oz$.

We consider the deviation of the center of mass O_s of system Ω_A from the given point (x_0, y_0, z_0) as the objective function.

Combinatorial Balanced Layout Problem (CBLP) Define a partition of set A into nonempty subsets A^j , $j \in J_m$, and the corresponding placement parameters $u_i = (v_i, z_i, \theta_i)$ of objects \mathbb{T}_i , $i \in J_n$, such that the objective function is minimized, taking into account relations (2)–(6).

We assume that the problem has at least one feasible solution.

N.B Restrictions on the axial and centrifugal moments of the system and allowable distances between objects may also be given.

3 Mathematical Model

Now, we define special combinatorial configurations describing the discrete structure of the CBLP problem. Some basic approaches for mathematical modelling of optimization problems on combinatorial configurations are described, for instance, in [14–16].

The possibilities of partitioning set A into nonempty subsets $A^j, j \in J_m$, are determined by both the number of elements in each subset and the order of the subsets.

Let us consider the subcontainers and the assumed corresponding sets of objects $A^j, j \in J_m$. Then, the tuple of natural numbers (k_1, k_2, \dots, k_m) , such that $\sum_{j=1}^m k_j = n$, denotes the number k_j of objects associated with each subcontainer Ω^j .

The number of all such tuples is equal to the number of compositions of the number n of length m [17], which is $\frac{(n-1)!}{(m-1)!(n-m)!}$.

We shall derive how it is possible to partition a set A of n objects into m subcontainers $\Omega^j, j \in J_m$, containing k_1, k_2, \dots, k_m items, respectively, with no ordering condition within each Ω^j . We denote subsets of objects that are placed inside corresponding subcontainers Ω^j by $A^j, j \in J_m$.

Without loss of generality, we will distinguish the objects with the same values of metrical characteristics, height h_i and mass m_i (for example, providing them with different identification numbers).

We order the elements of set A . We assign to each object the number of the subcontainer into which it is expected to be placed. We get a tuple consisting of n elements that form a permutation with repetitions from m numbers $1, 2, \dots, m$, in which the first element is repeated k_1 times, the second element is repeated k_2 times, ..., the last element is repeated k_m times.

The total number of permutations is equal to:

$$P(n, k_1, k_2, \dots, k_m) = \frac{n!}{k_1! \cdot k_2! \cdot \dots \cdot k_m!}. \tag{8}$$

Therefore, the total number of partitions of n objects into m subcontainers Ω^j , provided that each Ω^j contains at least one object and the order in which objects are placed inside Ω^j is not considered, is equal to:

$$\sum_{k_1+k_2+\dots+k_m=n} P(n, k_1, k_2, \dots, k_m) = \sum_{k_1+k_2+\dots+k_m=n} \frac{n!}{k_1! \cdot k_2! \cdot \dots \cdot k_m!} \tag{9}$$

Note that the number of summands in (9) is equal to $N = \left| C_{n-1}^{m-1} \right| = \frac{(n-1)!}{(m-1)!(n-m)!}$.

To generate subsets A^j , $j \in J_m$, we introduce a special combinatorial configuration [18].

Rather complex combinatorial configurations can formally be described and generated using compositional κ -images of combinatorial sets (κ -sets) introduced in [19]. A combinatorial set is a set of tuples that are constructed from a finite set of arbitrary elements (so-called generating elements), according to certain rules. Permutations, combinations, and binary sequences are examples of classical combinatorial sets. We describe the concept of κ -sets.

The basic idea of κ -sets generation is introduced in [19]. However, the problem of generating κ -sets of more complicated combinatorial structure remains an open problem. One of these special cases is studied in [20].

The problem of generating κ -sets is based on special techniques of generating basic combinatorial sets. The basic sets can be defined as combinatorial sets with the given features, i.e., both classical combinatorial sets (e.g., permutations, combinations, compositions, partitions, and n -tuples) and nonclassical combinatorial sets (e.g., permutations of tuples, compositions of permutations, and permutations with a prescribed number of cycles). Generation algorithms for basic combinatorial sets are described, e.g., in [21–25].

A generation strategy for the compositional κ -images of combinatorial sets (κ -sets).

We denote as $\mathbb{C}_{\mathbb{P}}(n, m)$ the set of compositions of the number n of length m (which corresponds to the partition of different objects from set A into m subcontainers Ω^j , $j \in J_m$), provided that each subcontainer contains at least one object and the order of objects inside the subcontainer is not considered, where, $|\mathbb{C}_{\mathbb{P}}(n, m)| = N = \left| C_{n-1}^{m-1} \right|$.

$$\text{Let } \mathbb{k} = (k_1, k_2, \dots, k_m) \in \mathbb{C}_{\mathbb{P}}(n, m), \sum_{j=1}^m k_j = n, k_j \geq 1, j \in J_m.$$

We introduce a combinatorial set $\mathbb{Q}(\mathbb{k})$ that is a composition image of combinatorial sets (κ -set) $\mathbb{C}_{\mathbb{P}}(n, m)$; $C_n^{k_1}, C_{n_1}^{k_2}, C_{n_2}^{k_3}, \dots, C_{n_{m-1}}^{k_m}$, generated by sets $I_{n_0}, I_{n_1}, I_{n_2}, \dots, I_{n_{m-1}}$, where $n_i = n - k_1 - \dots - k_i$, $i \in J_{m-1}$,

$$I_{n_0} = J_n,$$

$$I_{n_1} = I_{n_0} \setminus \left\{ j_1^{n_0}, j_2^{n_0}, \dots, j_{k_1}^{n_0} \right\}, \left(j_1^{n_0}, j_2^{n_0}, \dots, j_{k_1}^{n_0} \right) \in C_n^{k_1},$$

$$I_{n_2} = I_{n_1} \setminus \left\{ j_1^{n_1}, j_2^{n_1}, \dots, j_{k_2}^{n_1} \right\}, \left(j_1^{n_1}, j_2^{n_1}, \dots, j_{k_2}^{n_1} \right) \in C_{n_1}^{k_2},$$

...

$$I_{n_{m-1}} = I_{n_{m-2}} \setminus \left\{ j_1^{n_{m-2}}, j_2^{n_{m-2}}, \dots, j_{k_{m-1}}^{n_{m-2}} \right\}, \left(j_1^{n_{m-2}}, j_2^{n_{m-2}}, \dots, j_{k_{m-1}}^{n_{m-2}} \right) \in C_{n_{m-2}}^{k_{m-1}},$$

$$I_{n_{m-1}} = \left\{ j_1^{n_{m-1}}, j_2^{n_{m-1}}, \dots, j_{k_m}^{n_{m-1}} \right\}, \left(j_1^{n_{m-1}}, j_2^{n_{m-1}}, \dots, j_{k_m}^{n_{m-1}} \right) \in C_{n_{m-1}}^{k_m}.$$

Note that

$$I_{n_0} \cup I_{n_1} \cup \dots \cup I_{n_{m-1}} = J_n = \{1, 2, \dots, n\},$$

$$I_{n_s} \cap I_{n_t} = \emptyset, s \neq t \in J_{m-1}^0 = \{0, 1, \dots, m-1\}.$$

Each element $q(\mathbb{k}) \in \mathbb{Q}(\mathbb{k})$ can be described in the form:

$$q(\mathbb{k}) = (q_1, \dots, q_{k_1} \mid q_{k_1+1}, \dots, q_{k_1+k_2} \mid, \dots, \mid q_{k_1+\dots+k_{m-1}}, \dots, q_{k_{m-1}+k_m}),$$

where $(q_1, \dots, q_{k_1}) = (j_1^{n_0}, j_2^{n_0}, \dots, j_{k_1}^{n_0}) \in C_{n_0}^{k_1}$,

$$(q_{k_1+1}, \dots, q_{k_1+k_2}) = (j_1^{n_1}, j_2^{n_1}, \dots, j_{k_2}^{n_1}) \in C_{n_1}^{k_2},$$

...

$$(q_{k_1+\dots+k_{m-1}}, \dots, q_{k_{m-1}+k_m}) = (j_1^{n_{m-1}}, j_2^{n_{m-1}}, \dots, j_{k_m}^{n_{m-1}}) \in C_{n_{m-1}}^{k_m}.$$

The cardinality of set $\mathbb{Q}(\mathbb{k})$ is derived by (9).

An element $q(\mathbb{k})$ of the set $\mathbb{Q}(\mathbb{k})$ is said to be a *tuple of partition* of the set A into subsets $A^j, j \in J_m$.

Now, we define the vector of the basic variables of the problem CBLP: $u = (v, z, \theta)$, where $v = (v_1, \dots, v_n) \in \mathbf{R}^{2n}$, $\theta = (\theta_1, \dots, \theta_n) \in \mathbf{R}^n$, $v_i = (x_i, y_i) \in \mathbf{R}^2, x_i, y_i, \theta_i$ are continuous variables, and $z = (z_1, \dots, z_n) \in \mathbf{R}^n, z_i, i \in J_n$, are discrete variables defined by (1).

The values of variables $z_i, i \in J_n$, are determined in the order given by elements $q(\mathbb{k})$ of combinatorial set $\mathbb{Q}(\mathbb{k})$:

$$z_{q_i} = \sum_{l=1}^s t_{l-1} + h_{q_i}, \tag{10}$$

where:

$$s = \begin{cases} 1, & \text{if } i \leq k_1, \\ 2, & \text{if } k_1 < i \leq k_1 + k_2, \\ \dots & \\ m, & \text{if } k_1 + k_2 + \dots + k_{m-1} < i \leq k_1 + k_2 + \dots + k_m, \end{cases}$$

$i = 1, 2, \dots, n, q_i \in \{1, 2, \dots, n\}, q(\mathbb{k}) \in \mathbb{Q}(\mathbb{k})$.

Let us formalize placement constraints (4)–(6), using the phi-function technique.

We consider two objects \mathbb{T}_1 and \mathbb{T}_2 with variable parameters $u_1 = (v_1, z_1, \theta_1) \in \mathbf{R}^3, u_2 = (v_2, z_2, \theta_2) \in \mathbf{R}^3$, where $v_1 = (x_1, y_1), v_2 = (x_2, y_2), x_1, y_1, \theta_1, x_2, y_2, \theta_2$ are continuous variables and z_1, z_2 are discrete variables.

By definition [2, 3], a phi-function is a continuous function, therefore we extend the concept to discrete variables z_1, z_2 .

Definition 1 Function $\Upsilon_{12}(u_1, u_2)$ is called a *D-phi-function* of 3D-objects \mathbb{T}_1 and \mathbb{T}_2 if function $\Upsilon_{12}(v_1, z_1^0, \theta_1, v_2, z_2^0, \theta_2)$ is a *phi-function* $\Phi_{12}(v_1, z_1^0, \theta_1, v_2, z_2^0, \theta_2)$ of objects \mathbb{T}_1 and \mathbb{T}_2 for fixed values $z_1 = z_1^0$ and $z_2 = z_2^0$.

Definition 2 Function $\Upsilon'_{12}(u_1, u_2, u_{12})$ is called a *quasi-D-phi-function* of 3D-objects, \mathbb{T}_1 and \mathbb{T}_2 if function $\Upsilon'_{12}(v_1, z_1^0, \theta_1, v_2, z_2^0, \theta_2, u_{12})$ is a *quasi-phi-function* $\Phi'_{12}(v_1, z_1^0, \theta_1, v_2, z_2^0, \theta_2, u_{12})$ of objects \mathbb{T}_1 and \mathbb{T}_2 for fixed values $z_1 = z_1^0$ and $z_2 = z_2^0$.

Here, u_{12} is the vector of auxiliary continuous variables that is used to construct a quasi-*phi-function* of objects \mathbb{T}_1 and \mathbb{T}_2 .

The placement constraints (4)–(6) are described by the system of inequalities $\Upsilon_1(u, \tau) \geq 0, \Upsilon_2^*(u) \geq 0$, where the inequality $\Upsilon_1(u, \tau) \geq 0$ describes the nonoverlapping constraints and the inequality $\Upsilon_2^*(u) \geq 0$ describes the containment constraints:

$$\Upsilon_1(u, \tau) = \min \left\{ \Upsilon_1^j(u, \tau), j \in J_m \right\},$$

$$\Upsilon_1^j(u, \tau) = \min \left\{ \Upsilon_{q_1 q_2}^j(u_{q_1}, u_{q_2}, u_{q_1 q_2}), q_1 < q_2 \in J_n^j \right\}, \tag{11}$$

$$\tau = \left(u_{q_1 q_2}, q_1 < q_2 \in J_n^j \right),$$

$$\Upsilon_2^*(u) = \min \left\{ \Upsilon_2^{*j}(u), j \in J_m \right\}, \Upsilon_2^{*j}(u) = \min \left\{ \Upsilon_{q_i}^*(u_{q_i}), q_i \in J_n^j \right\}, \tag{12}$$

$\Upsilon_{q_1 q_2}^j(u_{q_1}, u_{q_2}, u_{q_1 q_2})$ is the function that describes the nonoverlapping condition between objects \mathbb{T}_{q_1} and \mathbb{T}_{q_2} , and $u_{q_1} = (x_{q_1}, y_{q_1}, z_{q_1}, \theta_{q_1}), u_{q_2} = (x_{q_2}, y_{q_2}, z_{q_2}, \theta_{q_2}), \Upsilon_{q_i}^*(u_{q_i})$ is the function that describes the nonoverlapping condition between objects \mathbb{T}_{q_i} and $\Omega^{*j} = \mathbf{R}^3 / \text{int } \Omega^j$.

Thus, in expressions (11) and (12) for fixed values z_{q_1} and z_{q_2} , we have: $\Upsilon_{q_1 q_2}^j(u_{q_1}, u_{q_2})$ is a *phi*-function [26] $\Phi_{q_1 q_2}^{\mathbb{T}\mathbb{T}}(u_{q_1}, u_{q_2})$ for objects \mathbb{T}_{q_1} and \mathbb{T}_{q_2} or a quasi-*phi*-function [27] $\Phi'_{q_1 q_2}(u_{q_1}, u_{q_2}, u_{q_1 q_2})$ for objects \mathbb{T}_{q_1} and \mathbb{T}_{q_2} ; $\Upsilon_{q_i}^*(u_{q_i})$ is a *phi*-function $\Phi_{q_i}^{\mathbb{T}\Omega^*j}(u_{q_i})$ for objects \mathbb{T}_{q_i} and Ω^*j .

If a minimum allowable distance condition between objects is given, adjusted *phi*-functions (quasi-*phi*-functions) are used for the corresponding pairs of objects [26, 27].

The mathematical model of the CBLP problem can be defined as follows:

$$F(u^*, \tau^*) = \min F(u, \tau) \quad \text{s.t. } (u, \tau) \in W, \quad (13)$$

$$W = \{(u, \tau) \in \mathbf{R}^\sigma : \Upsilon_1(u, \tau) \geq 0, \Upsilon_2^*(u) \geq 0, \mu(u) \geq 0\}, \quad (14)$$

where:

$$F(u) = d = (x_s(v, z) - x_0)^2 + (y_s(v, z) - y_0)^2 + (z_s - z_0)^2$$

$u = (v, z, \theta)$, $v = (v_1, \dots, v_n)$, $\theta = (\theta_1, \dots, \theta_n)$, $v_i = (x_i, y_i)$, $i \in J_n$, $v = (v_1, \dots, v_n)$, $\theta = (\theta_1, \dots, \theta_n)$, $v_i = (x_i, y_i)$, $i \in J_n$, $z = (z_1, \dots, z_n)$, function $\Upsilon_1(u, \tau)$ is described by (11) with $\Xi = \bigcup_{j=1}^m \Xi^j$, $\Xi^j = \{(q_1, q_2) : q_1 < q_2 \in J_n^j\}$, $\tau = (\tau_1, \dots, \tau_s) = (u_{q_1 q_2}, q_1 < q_2 \in J_n^j)$ is a vector of auxiliary variables for quasi-*phi*-functions, $s = |\Xi|$, function $\Upsilon_2^*(u)$ is defined by (12), elements of vector z are given by (10), and $\mu(u) \geq 0$ describes the given balancing constraints.

For example, problem (13) and (14) for the layout of cylinders in a cylindrical container takes the form:

$$\min d, \text{ s.t. } u = (v, z) \in W, \quad (15)$$

where:

$$v = (x_1, y_1, \dots, x_n, y_n), z = (z_1, \dots, z_n),$$

$$d = \left[\sum_{i=1}^n m'_i x_i \right]^2 + \left[\sum_{i=1}^n m'_i y_i \right]^2 + \left[\sum_{i=1}^n m'_i z_i - z_0 \right]^2,$$

and the feasible region W is described by the inequality system:

$$\begin{cases} (x_{q_2} - x_{q_1})^2 + (y_{q_2} - y_{q_1})^2 - (r_{q_2} + r_{q_1})^2 \geq 0, \\ q_1, q_2 \in \Xi^j, j \in J_m, \\ -x_{q_i}^2 - y_{q_i}^2 + (R_{q_i}^z - r_{q_i})^2 \geq 0, \\ q_i \in \Xi^j, j \in J_m. \end{cases} \quad (16)$$

Note, that $m'_i = \frac{m_i}{M} = \text{const}$, $M = \sum_{i=1}^n m_i = \text{const}$.

The problem of packing cylinders into cylindrical containers, with balancing conditions, is considered, for instance, in [11, 13].

The CBLP problem can be represented as a mixed integer programming (MIP) problem, using Boolean variables. However, unlike (13) and (14), this approach increases the number of discrete variables and therefore increases the dimension of the CBLP problem.

4 Solution Strategy

The following strategy is used to solve CBLP problems:

1. Generate a subset $\{q(\mathbb{k})\}_\chi \subset \mathbb{Q}(\mathbb{k})$ using the concept of the Nested Combinatorial κ -sets.
2. Construct a subset $\{q'(\mathbb{k})\}_{\chi'} \subseteq \{q(\mathbb{k})\}_\chi$ of tuples that satisfy (6). If $\{q'(\mathbb{k})\}_{\chi'} = \emptyset$, then go to Step 1.
3. Construct a set of feasible starting points $\{u'_0\}$ for each tuple from the set $\{q'(\mathbb{k})\}_{\chi'}$, using the algorithm presented in [13].
4. Search for a local extremum of problem (13) and (14) for each starting point $u'_0 \in W$ with respect to $q'(\mathbb{k}) \in \{q'(\mathbb{k})\}_{\chi'}$.
5. Choose the best of the local extrema found for all tuples of the set $\{q'(\mathbb{k})\}_{\chi'}$ and feasible starting points $\{u'_0\}$ as a local optimal solution of problem (13) and (14).

To solve nonlinear programming problems, IPOPT is used, being available as an open noncommercial resource (<https://projects.coin-or.org/Ipopt>). IPOPT is based on the internal point method described in [28].

In order to generate a subset $\{q(\mathbb{k})\}_\chi \subset \mathbb{Q}(\mathbb{k})$, we use the concept of the Nested Combinatorial κ -sets.

Concept of the Nested Combinatorial κ -Sets To define a structure of the nested combinatorial κ -set, we use the κ -level tree.

Let $i \in J_{\kappa}^0 = \{0, 1, \dots, \kappa\}$, where κ is the number of levels of the tree. At each level of the tree, we have η_i nodes.

And, let

$$\begin{aligned}
 & Y_{01} \\
 & Y_{11}, Y_{12}, \dots, Y_{1\eta_1} \\
 & Y_{21}, Y_{22}, \dots, Y_{2\eta_2} \\
 & \dots \\
 & Y_{\kappa 1}, Y_{\kappa 2}, \dots, Y_{\kappa \eta_{\kappa}}
 \end{aligned} \tag{17}$$

be combinatorial sets that correspond to the nodes of the i -th level of the κ -level tree, $i = 0, 1, \dots, \kappa$.

Each combinatorial set Y_{ij} of the tree is defined by a finite set β_{ij} of generative elements of Y_{ij} , $i = 0, 1, \dots, \kappa, j = 1, \dots, \eta_i$. We denote the number of elements of β_{ij} by n_{ij} , therefore:

$$\eta_0 = 1, \eta_1 = n, \eta_2 = \sum_{j=1}^{\eta_1} n_{2j}, \eta_3 = \sum_{j=1}^{\eta_2} n_{3j}, \dots, \eta_i = \sum_{j=1}^{\eta_{i-1}} n_{ij}, \eta_{\kappa} = \sum_{j=1}^{\eta_{\kappa-1}} n_{\kappa j}.$$

The core idea of the *Nested Combinatorial κ -set* is based on relationships between generative elements β_{ij} of each combinatorial set Y_{ij} of i -th level and elements of combinatorial sets of $(i + 1)$ -th level, using the following correspondence:

$$\beta \in \beta_{ij} \leftrightarrow (y_1, y_2, \dots, y_{n_l}) \in Y_{i+1,l}, l \in \{1, \dots, \eta_{i+1}\}, i \in \{0, 1, \dots, \kappa - 1\}. \tag{18}$$

A nested combinatorial κ -set is a composed combinatorial set produced by means of the tree structure (17) and correspondence (18) (see [19] for formal definition).

Let us consider an example to make clear the concept. We denote a nested combinatorial κ -set by T_{κ} .

Example 1 Let T_{κ} have a two-level structure (17), where Y_{01} is a permutation set $P(a,b)$ generated by $\beta_{01} = \{a, b\}$; Y_{11} is a combination set $C_3^2(c, d, e)$ generated by $\beta_{11} = \{c, d, e\}$; and Y_{12} is a permutation set $P(g,h)$ generated by $\beta_{12} = \{g, h\}$ (see Figure 1). Therefore, $n_{01} = 2, n_{11} = 3, n_{12} = 2, \eta_0 = 1, \eta_1 = 2$.

In order to produce the nested combinatorial set T_{κ} , we replace the generative elements a, b in each element of $P(a,b)$ by each element of the combinatorial sets

Figure 1 Two-level structure (17) of the nested combinatorial κ -set T_κ

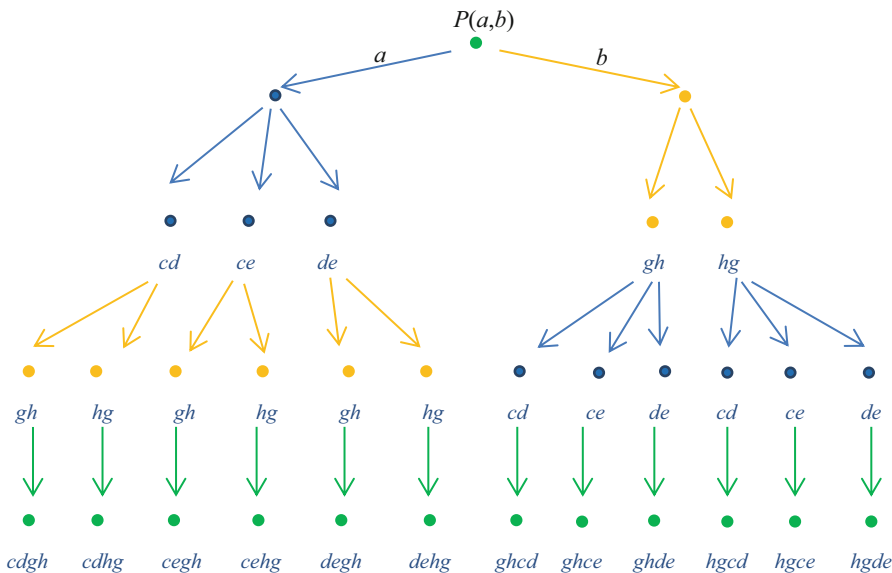
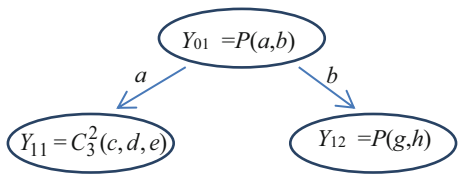


Figure 2 Elements of the combinatorial set T_κ

$C_3^2(c, d, e)$ and $P(g,h)$ consequently, using correspondence (18) (see Figure 2). Terminal nodes of the tree correspond to elements of the combinatorial set T_κ : $(cdgh)$, $(cdhg)$, $(cegh)$, $(cehg)$, $(degh)$, $(dehg)$, $(ghcd)$, $(ghce)$, $(ghde)$, $(hgcd)$, $(hgce)$, $(hgde)$.

Details of algorithm for generating a nested combinatorial κ -set are represented in [25].

Now, we consider an example of the algorithm generating $q(\mathbb{k})$.

Example 2 Let the basic sets $\mathbb{C}_{\mathbb{P}}(5, 2)$, $C_5^{k_1}$, $C_{5-k_1}^{k_2}$ generated by elements $\{q_1, q_2, q_3, q_4, q_5\}$ be given, $k_1 + k_2 = n = 5$.

Since $\mathbb{C}_{\mathbb{P}}(5, 2) = \{(k_1^i, k_2^i)\} = \{(1, 4), (2, 3), (3, 2), (4, 1)\}$, then we have:

1. $k_1 = 1, k_2 = 4$
 $C_5^{k_1} = C_5^1 = \{(q_1), (q_2), (q_3), (q_4), (q_5)\}$
 $C_{5-k_1}^{k_2} = C_4^4 : \{(q_2, q_3, q_4, q_5)\}, \{(q_1, q_3, q_4, q_5)\}, \{(q_1, q_2, q_4, q_5)\},$
 $\{(q_1, q_2, q_3, q_5)\}, \{(q_1, q_2, q_3, q_4)\}$
2. $k_1 = 4, k_2 = 1$
 $C_5^{k_1} = C_5^4 = \{(q_1, q_2, q_3, q_4), (q_1, q_2, q_3, q_5), (q_1, q_2, q_4, q_5),$
 $(q_1, q_3, q_4, q_5), (q_2, q_3, q_4, q_5)\}$
 $C_{5-k_1}^{k_2} = C_1^1 : \{(q_5)\}, \{(q_4)\}, \{(q_3)\}, \{(q_2)\}, \{(q_1)\}$
3. $k_1 = 2, k_2 = 3$
 $C_5^{k_1} = C_5^2 = \{(q_1, q_2), (q_1, q_3), (q_1, q_4), (q_1, q_5), (q_2, q_3),$
 $(q_2, q_4), (q_2, q_5), (q_3, q_4), (q_3, q_5), (q_4, q_5)\}$
 $C_{5-k_1}^{k_2} = C_3^3 : \{(q_3, q_4, q_5)\}, \{(q_2, q_4, q_5)\}, \{(q_2, q_3, q_5)\},$
 $\{(q_2, q_3, q_4)\}, \{(q_1, q_4, q_5)\}, \{(q_1, q_3, q_5)\}, \{(q_1, q_3, q_4)\},$
 $\{(q_1, q_2, q_5)\}, \{(q_1, q_2, q_4)\}, \{(q_1, q_2, q_3)\}$
4. $k_1 = 3, k_2 = 2$
 $C_5^{k_1} = C_5^3 = \{(q_1, q_2, q_3), (q_1, q_2, q_4), (q_1, q_2, q_5), (q_1, q_3, q_4), (q_1, q_3, q_5),$
 $(q_1, q_4, q_5), (q_2, q_3, q_4), (q_2, q_3, q_5), (q_2, q_4, q_5), (q_3, q_4, q_5)\}$
 $C_{5-k_1}^{k_2} = C_2^2 : \{(q_4, q_5)\}, \{(q_3, q_5)\}, \{(q_3, q_4)\}, \{(q_2, q_5)\}, \{(q_2, q_4)\},$
 $\{(q_2, q_3)\}, \{(q_1, q_5)\}, \{(q_1, q_4)\}, \{(q_1, q_3)\}, \{(q_1, q_2)\}$

Example 3 We show here the κ -set of compositions of two combinations $C_5^{k_1}, C_{5-k_1}^{k_2}$ generated by elements $\{q_1, q_2, q_3, q_4, q_5\}$, $k_1 + k_2 = n = 5$, using the *Gen_κ-set* algorithm and the results of Example 1. Then, $\kappa=1$ and Y_0 is the set of compositions $\mathbb{C}_{\mathbb{P}}(5, 2)$, $Y_{11} = C_5^{k_1}$, $Y_{12} = C_{5-k_1}^{k_2}$. Let us present the structure of the κ -set constructed in Example 2.

In the set Y_{01} , the first generating element k_1 will be replaced with the unique element of the set Y_{11} , i.e., with the tuple $(q_{i_1}, q_{i_2}, \dots, q_{i_{k_1}}) \in C_5^{k_1}$, and element k_2 with the tuple $(q_{j_1}, q_{j_2}, \dots, q_{j_{k_2}}) \in C_5^{k_2}$.

We apply the algorithm presented in [25] to generate all elements of the set $\mathbb{Q}(\mathbb{k})$ (see Table 1). According to (9), the number of elements in the set $\mathbb{Q}(\mathbb{k})$ is equal to 30.

5 Computational Results

Example 4 We consider the problem (15) and (16) for cylinders \mathbb{C}_i , $i \in J_n$ that have to be placed into the cylindrical container Ω with one separation plane (circle) in order to minimize the deviation of the center of mass of Ω_A from the given point (x_0, y_0, z_0) . Characteristics of cylinders are given in Table 2.

Table 1 Data for Example 3 and appropriate elements of set $\mathbb{Q}(\mathbb{k})$

N_b	$q \in \mathbb{Q}$	\mathbb{k}	$q(\mathbb{k}) \in \mathbb{Q}(\mathbb{k})$
1	$(q_1, q_2, q_3, q_4, q_5)$	(1,4)	$(q_1, q_2, q_3, q_4, q_5)$
2	$(q_2, q_1, q_3, q_4, q_5)$	(1,4)	$(q_2, q_1, q_3, q_4, q_5)$
3	$(q_3, q_1, q_2, q_4, q_5)$	(1,4)	$(q_3, q_1, q_2, q_4, q_5)$
4	$(q_4, q_1, q_2, q_3, q_5)$	(1,4)	$(q_4, q_1, q_2, q_3, q_5)$
5	$(q_5, q_1, q_2, q_3, q_4)$	(1,4)	$(q_5, q_1, q_2, q_3, q_4)$
6	$(q_1, q_2, q_3, q_4, q_5)$	(4,1)	$(q_1, q_2, q_3, q_4, q_5)$
7	$(q_1, q_2, q_3, q_5, q_4)$	(4,1)	$(q_1, q_2, q_3, q_5, q_4)$
8	$(q_1, q_2, q_4, q_5, q_3)$	(4,1)	$(q_1, q_2, q_4, q_5, q_3)$
9	$(q_1, q_3, q_4, q_5, q_2)$	(4,1)	$(q_1, q_3, q_4, q_5, q_2)$
10	$(q_2, q_3, q_4, q_5, q_1)$	(4,1)	$(q_2, q_3, q_4, q_5, q_1)$
11	$(q_1, q_2, q_3, q_4, q_5)$	(2,3)	$(q_1, q_2, q_3, q_4, q_5)$
12	$(q_1, q_3, q_2, q_4, q_5)$	(2,3)	$(q_1, q_3, q_2, q_4, q_5)$
13	$(q_1, q_4, q_2, q_3, q_5)$	(2,3)	$(q_1, q_4, q_2, q_3, q_5)$
14	$(q_1, q_5, q_2, q_3, q_4)$	(2,3)	$(q_1, q_5, q_2, q_3, q_4)$
15	$(q_2, q_3, q_1, q_4, q_5)$	(2,3)	$(q_2, q_3, q_1, q_4, q_5)$
16	$(q_2, q_4, q_1, q_3, q_5)$	(2,3)	$(q_2, q_4, q_1, q_3, q_5)$
17	$(q_2, q_5, q_1, q_3, q_4)$	(2,3)	$(q_2, q_5, q_1, q_3, q_4)$
18	$(q_3, q_4, q_1, q_2, q_5)$	(2,3)	$(q_3, q_4, q_1, q_2, q_5)$
19	$(q_3, q_5, q_1, q_2, q_4)$	(2,3)	$(q_3, q_5, q_1, q_2, q_4)$
20	$(q_4, q_5, q_1, q_2, q_3)$	(2,3)	$(q_4, q_5, q_1, q_2, q_3)$
21	$(q_3, q_4, q_5, q_1, q_2)$	(3,2)	$(q_3, q_4, q_5, q_1, q_2)$
22	$(q_2, q_4, q_5, q_1, q_3)$	(3,2)	$(q_2, q_4, q_5, q_1, q_3)$
23	$(q_2, q_3, q_5, q_1, q_4)$	(3,2)	$(q_2, q_3, q_5, q_1, q_4)$
24	$(q_2, q_3, q_4, q_1, q_5)$	(3,2)	$(q_2, q_3, q_4, q_1, q_5)$
25	$(q_1, q_4, q_5, q_2, q_3)$	(3,2)	$(q_1, q_4, q_5, q_2, q_3)$
26	$(q_1, q_3, q_5, q_2, q_4)$	(3,2)	$(q_1, q_3, q_5, q_2, q_4)$
27	$(q_1, q_3, q_4, q_2, q_5)$	(3,2)	$(q_1, q_3, q_4, q_2, q_5)$
28	$(q_1, q_2, q_5, q_3, q_4)$	(3,2)	$(q_1, q_2, q_5, q_3, q_4)$
29	$(q_1, q_2, q_4, q_3, q_5)$	(3,2)	$(q_1, q_2, q_4, q_3, q_5)$
30	$(q_1, q_2, q_3, q_4, q_5)$	(3,2)	$(q_1, q_2, q_3, q_4, q_5)$

Table 2 Characteristics of cylinders in Example 4

	C_1	C_2	C_3	C_4	C_5
m_i	4	2	1	3	5
r_i	1	0.7	0.45	0.8	0.9
h_i	1.27	1.3	1.57	1.49	1.96

Let $m = 2, H = 6, R = 2.5, t_1 = 3$ be the parameters characterizing our cylindrical container and $(x_0, y_0, z_0) = (0, 0, 3)$.

The values of the objective function for all $n = 30$ tuples of the partition $q(\mathbb{k}) \in \mathbb{Q}(\mathbb{k})$ and appropriate compositions \mathbb{k} are presented in Table 3.

Figure 3 shows the local optimal placements of cylinders in the two subcontainers found by our algorithm that correspond to the tuples: (a) $q^1(\mathbb{k})$, (b) $q^{18}(\mathbb{k})$ in Example 4.

The best value of the objective function in Example 4 is 0.0003 that corresponds to two tuples $q^1(\mathbb{k}) = (1|2\ 3\ 4\ 5)$ and $q^{18}(\mathbb{k}) = (3\ 4|1\ 2\ 5)$.

Table 3 Output data in Example 4

i	$q(\mathbb{k}) \in \mathbb{Q}(\mathbb{k})$	\mathbb{k}	Objective function, d
1	(1 2 3 4 5)	(1,4)	0.0003
2	(2 1 3 4 5)	(1,4)	0.1475
3	(3 1 2 4 5)	(1,4)	0.3411
4	(4 1 2 3 5)	(1,4)	0.0339
5	(5 1 2 3 4)	(1,4)	0.0467
6	(1 2 3 4 5)	(4,1)	1.4787
7	(1 2 3 5 4)	(4,1)	2.6115
8	(1 2 4 5 3)	(4,1)	4.0643
9	(1 3 4 5 2)	(4,1)	3.2979
10	(2 3 4 5 1)	(4,1)	2.0051
11	(1 2 3 4 5)	(2,3)	0.1731
12	(1 3 2 4 5)	(2,3)	0.0467
13	(1 4 2 3 5)	(2,3)	0.3795
14	(1 5 2 3 4)	(2,3)	1.0323
15	(2 3 1 4 5)	(2,3)	0.0339
16	(2 4 1 3 5)	(2,3)	0.0467
17	(2 5 1 3 4)	(2,3)	0.3795
18	(3 4 1 2 5)	(2,3)	0.0003
19	(3 5 1 2 4)	(2,3)	0.1731
20	(4 5 1 2 3)	(2,3)	0.6659
21	(3 4 5 1 2)	(3,2)	1.0323
22	(2 4 5 1 3)	(3,2)	1.4787
23	(2 3 5 1 4)	(3,2)	0.6659
24	(2 3 4 1 5)	(3,2)	0.1731
25	(1 4 5 2 3)	(3,2)	2.6115
26	(1 3 5 2 4)	(3,2)	1.4787
27	(1 3 4 2 5)	(3,2)	0.6659
28	(1 2 5 3 4)	(3,2)	2.0051
29	(1 2 4 3 5)	(3,2)	1.0323
30	(1 2 3 4 5)	(3,2)	0.3795

Example 5 Let us consider the problem (13) and (14). Let Ω be a cylindrical container of height $H = 1$ and the basis radius $R = 0.55$. The container has two separation circles. We assume that $t_1 = t_2 = 0.35$. Let $m_0 = 500$ be the mass and $(x_0, y_0, z_0) = (0, 0, 0.5)$ be the center of mass of the cylindrical container Ω .

We consider the collection of 3D-objects of six shapes (Figure 4): $A = \{\mathbb{S}_i, i = 1, \dots, 6, \mathbb{C}_i, i = 7, \dots, 13, \mathbb{Q}_i, i = 14 \dots 17, \mathbb{S}_{\mathbb{C}i}, i = 18, \dots 21, \mathbb{P}_i, i = 22, 23, 24, \mathbb{K}_{25}\}$ with the following characteristics:

$\{m_i, i = 1, \dots, 25\} = \{20.944, 15.2681, 27.8764, 20.944, 20.944, 34.5575, 63.7115, 41.8146, 30.4106, 6.28319, 20.1062, 31.4159, 28.4245, 49.9649, 24.8714, 38.6888, 26.2637, 20.7764, 17.2159, 16.8756, 52.8, 52.8, 52.8, 23.1489\}$ are the given masses of the 3D-objects;

$r_1 = 0.1, r_2 = 0.09, r_3 = 0.11, r_4 = 0.11, r_5 = 0.1, r_6 = 0.1$ are the radii of spheres $\mathbb{S}_i, i = 1, \dots, 6$;

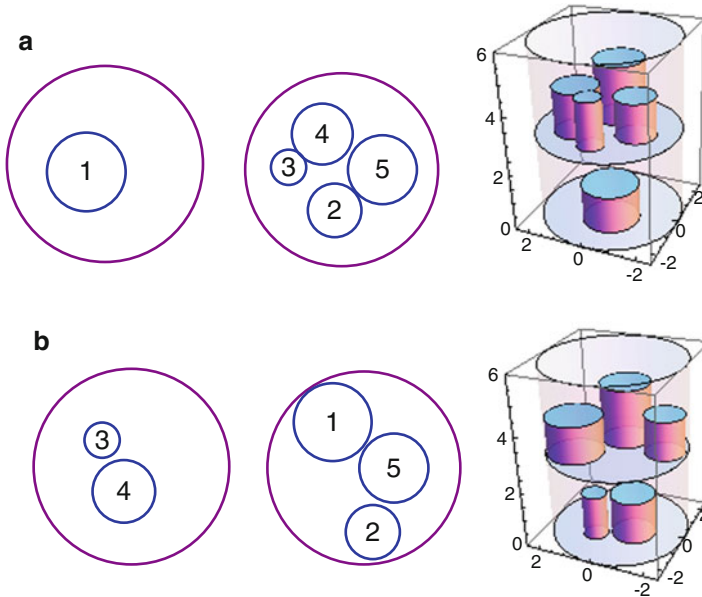


Figure 3 Balanced layout of cylinders and relevant item projections on the base of the container and separation circle corresponding to tuples: (a) $q^1(\mathbb{k})$, (b) $q^{18}(\mathbb{k})$ in Example 4

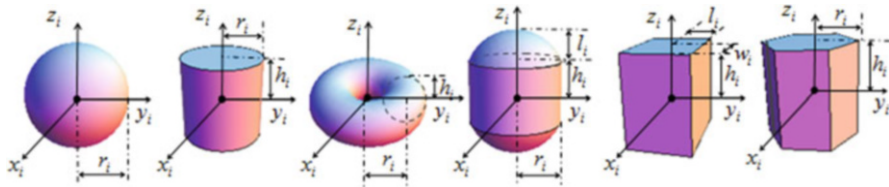


Figure 4 Shapes of objects in Example 5: $S_i, C_i, Q_i, S_{C_i}, P_i, K_i$

$r_7 = 0.1, h_7 = 0.11, r_8 = 0.13, h_8 = 0.12, r_9 = 0.11, h_9 = 0.11, r_{10} = 0.11, h_{10} = 0.08, r_{11} = 0.05, h_{11} = 0.08, r_{12} = 0.08, h_{12} = 0.1, r_{13} = 0.1, h_{13} = 0.1$ are the radii and half-heights of cylinders $C_i, i = 7, \dots, 13$;

$r_{14} = 0.08, h_{14} = 0.07, r_{15} = 0.09, h_{15} = 0.075, r_{16} = 0.07, h_{16} = 0.06, r_{17} = 0.08, h_{17} = 0.07$ are the radii of the generating circles and half-heights of tori $Q_i, i = 14 \dots 17$;

$r_{18} = 0.1, h_{18} = 0.05, l_{18} = 0.07, r_{19} = 0.05, h_{19} = 0.05, l_{19} = 0.08, r_{20} = 0.08, h_{20} = 0.05, l_{20} = 0.06, r_{21} = 0.08, h_{21} = 0.04, l_{21} = 0.07$ are the radii, half-heights of cylinders, and heights of spherical segments for spherocylinders $S_{C_i}, i = 18, \dots 21$;

$w_i = 0.11, l_i = 0.1, h_i = 0.12, i = 22, 23, 24$ are the half-widths, half-lengths, and half-heights of cuboids $P_i, i = 22, 23, 24$;

$r_{25} = 0.09, h_{25} = 0.11$ are the length of the basis side and half-height of the right hexagonal prism K_{25} .

Table 4 Output data in Example 5

i	$q(\mathbb{k}) \in \mathbb{Q}(\mathbb{k})$	\mathbb{k}	Objective function, d
1	(2,3,6,8,10,14,17,21,24 4,7,11,13,18,20,22,23 1,5,9,12,15,16,19,25)	(9,8,8)	0.002387
2	(3,5,6,7,10,13,17,21 2,4,8,12,14,18,19,22,25 1,9,11,15,16,20,23,24)	(8,9,8)	0.000269
3	(1,5,10,17,19,21,22,25 2,4,6,9,12,13,14,18 3,7,8,11,15,16,20,23,24)	(8,8,9)	0.943362×10^{-7}

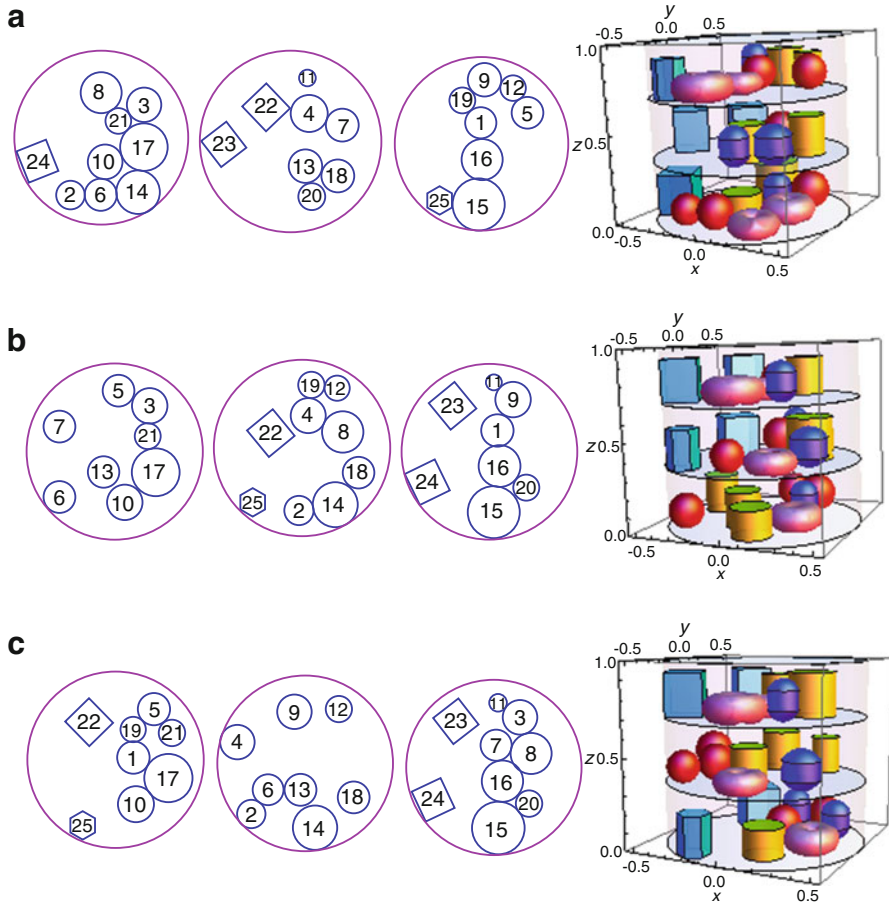


Figure 5 Local optimal placement of 3D-objects and the relative object projections onto the separation circles corresponding to tuples: (a) $q^1(\mathbb{k})$, (b) $q^2(\mathbb{k})$, (c) $q^3(\mathbb{k})$ in Example 5

The values of the objective function for three chosen tuples of the partition $q(\mathbb{k}) \in \mathbb{Q}(\mathbb{k})$ and appropriate compositions \mathbb{k} are presented in Table 4.

Figure 5 shows the local optimal placement of 3D-objects in subcontainers and the relative object projections onto the separation circles corresponding to tuples: (a) $q^1(\mathbb{k})$, (b) $q^2(\mathbb{k})$, (c) $q^3(\mathbb{k})$ given in Table 4.

The best values found for the objective function in Example 5 is 0.943362×10^{-7} that corresponds to tuple $q^3(\mathbb{k}) = (1,5,10,17,19,21,22,25|2,4,6,9,12,13,14,18|3,7,8,11,15,16,20,23,24)$.

6 Concluding Remarks

This chapter discusses the problem of placing 3D-objects into a container, partitioned into sectors by parallel separation planes, minimizing the distance of the overall center of mass from an assigned position.

The mathematical model formulated for the purpose, on the basis of the phi-function methodology, is illustrated in detail. It takes into account not only the geometrical and balancing constraints, but also the combinatorial features relevant to the assignment of items to sectors.

A solution strategy is provided, which includes the following procedures: generation of partition tuples, based on combinatorial configurations, construction of feasible starting points, and local optimization. This approach implements the multi-start strategy to search for “good” feasible solutions. The results of the numerical experiments show the efficiency of the proposed approach for the class of layout problems considered.

References

1. Stoyan, Y., Pankratov, A., Romanova, T., Fasano, G., Pintér, J., Stoian, Y., Chugay, A., Kovalenko, A.: Optimized packings in space engineering applications—part I. In: Fasano, G., Pintér, J.D. (eds.) *Modeling and Optimization in Space Engineering*. Springer, New York (2019)
2. Chernov, N., Stoyan, Y., Romanova, T.: Mathematical model and efficient algorithms for object packing problem. *Comput. Geom.* **43**(5), 535–553 (2010)
3. Stoyan, Y., Romanova, T.: Mathematical models of placement optimisation: Two- and three-dimensional problems and applications. In: Fasano, G., Pinter, J. (eds.) *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications, vol. 73, pp. 363–388. Springer, New York (2012)
4. Fasano, G.: *Solving Non-standard Packing Problems by Global Optimization and Heuristics*. Springer Briefs in Optimization. Springer, New York (2014)
5. Fasano, G., Pintér, J.D.: *Optimized Packings and Their Applications*, Springer Optimization and its Applications. Springer, New York (2015)
6. Fasano, G., Pinter, J.: *Modeling and Optimization in Space Engineering*. Springer, New York (2013)
7. Che, C., Wang, Y., Teng, H.: Test problems for quasi-satellite packing: Cylinders packing with behavior constraints and all the optimal solutions known. *Optimization Online* (2008.) http://www.optimizationonline.org/DB_HTML/2008/09/2093.html
8. Fasano, G., Pinter, J.: *Space engineering*. In: *Modeling and Optimization with Case Studies*, Springer Optimization and its Applications. Springer, New York (2016)

9. Sun, Z., Teng, H.: Optimal layout design of a satellite module. *Eng. Opt.* **35**(5), 513–530 (2003)
10. Lei, K.: Constrained layout optimization based on adaptive particle swarm optimizer. In: Zhihua, C., Zhenhua, L., Zhuo, K., Yong, L. (eds.) *Advances in Computation and Intelligence, Series 1*, pp. 434–442. Springer, Heidelberg (2009)
11. Kovalenko, A., Romanova, T., Stetsyuk, P.: Balance layout problem for 3D-objects: mathematical model and solution methods. *Cybern. Syst. Anal.* **51**(4), 556–565 (2015)
12. Stetsyuk, P., Romanova, T., Scheithauer, G.: On the global minimum in a balanced circular packing problem. *Opt. Lett.* **10**, 1347–1360 (2016)
13. Stoyan, Y., Romanova, T., Pankratov, A., Kovalenko, A., Stetsyuk, P.: Modeling and optimization of balance layout problems. In: Fasano, G., Pinter, J. (eds.) *Space Engineering. Modeling and Optimization with Case Studies, Springer Optimization and its Applications*, vol. 114, pp. 369–400. Springer, New York (2016)
14. Hulianytskyi, L., Riasna, I.: Formalization and classification of combinatorial optimization problems. In: Butenko, S., Pardalos, P., Shylo, V. (eds.) *Optimization Methods and Applications*, pp. 239–250. Springer, New York (2017)
15. Papadimitriou, C., Steiglitz, K.: *Combinatorial Optimization: Algorithms and Complexity*. Courier Corporation (1998)
16. Yakovlev, S., Pichugina, O.: Properties of combinatorial optimization problems over polyhedral-spherical sets. *Cybern. Syst. Anal.* **54**(1), 111–123 (2018)
17. Reingold, E., Nievergelt, J., Deo, N.: *Combinatorial Algorithms: Theory and Practice*. Pearson Education, North York, ON (1977)
18. Sachkov, V.: *Combinatorial Methods in Discrete Mathematics*, 1st edn. Cambridge University Press, Cambridge (1996)
19. Stoyan, Y., Grebennik, I.: Description and generation of combinatorial sets having special characteristics. *Int. J. Biomed. Soft Comput. Hum. Sci.* **18**(1), 83–88 (2013)
20. Grebennik, I.: Description and generation of permutations containing cycles. *Cybern. Syst. Anal.* **46**(6), 945–952 (2010)
21. Knuth, D.: *The Art of Computer Programming*, 4(2): Generating All Tuples and Permutations. Addison-Wesley, Boston (2005)
22. Kreher, D., Stinson, D.: *Combinatorial Algorithms: Generation, Enumeration and Search*. CRC Press, Boca Raton, FL (1999)
23. Ruskey, F.: *Combinatorial Generation*, Department of Computer Science, University of Victoria, Canada, Ij-CSC 425/20 (2003)
24. Grebennik, I., Kovalenko, A., Romanova, T., Urniaieva, I., Shekhovtsov, S.: Combinatorial configurations in balance layout optimization problems. *Cybern. Syst. Anal.* **54**(2), 55–67 (2018)
25. Grebennik, I., Lytvynenko, O.: Generating combinatorial sets with given properties. *Cybern. Syst. Anal.* **48**(6), 890–898 (2012)
26. Stoyan, Y., Pankratov, A., Romanova, T., Chugay, A.: Optimized object packings using quasi-phi-functions. In: Fasano, G., Pinter, J. (eds.) *Optimized Packings and Their Applications, Springer Optimization and its Applications*, vol. 105, pp. 265–291. Springer, New York (2015)
27. Stoyan, Y., Pankratov, A., Romanova, T.: Quasi phi-functions and optimal packing of ellipses. *J. Glob. Optim.* **65**(2), 283–307 (2016)
28. Wachter, A., Biegler, L.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* **106**(1), 25–57 (2006)