SPARSE PACKING AND ITS APPLICATION

Romanova Tatiana

tarom27@yahoo.com

A.Pidgorny Institute of Mechanical Engineering Problems
of the National Academy of Sciences of Ukraine

Department of Mathematical Modeling and Optimal Design

Motivation

In most of optimized packing approaches, the objective is to find the dense packing. That is, the packing resulting in the *smallest unused space* of the container (KP) or giving the *minimal volume container* in the case of an open dimension problem (ODP).

However, there are many modern industrial technologies where the concept of the dense packing is not suitable.

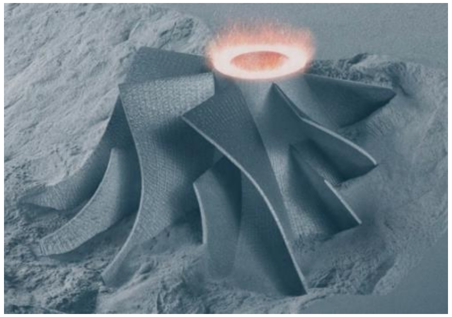
Motivation

Nonstandard packing problems arises, e.g., in additive manufacturing (3D-printing) for topological optimization of 3D parts (generating void structures), in logistics for loading problems (clustering problems), in thermal deburring – energy saving technology of removing burrs from 3D parts by detonating gas mixtures.

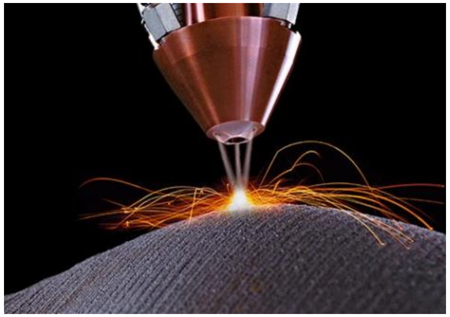
Motivation

Sparse packing is motivated by the Thermal Energy Method – state-of-the-art technology of cleaning complex shaped parts produced by additive manufacturing (3D printing) from particles of non-sintered powder.

SLS



SLM



Finishing after 3D printing

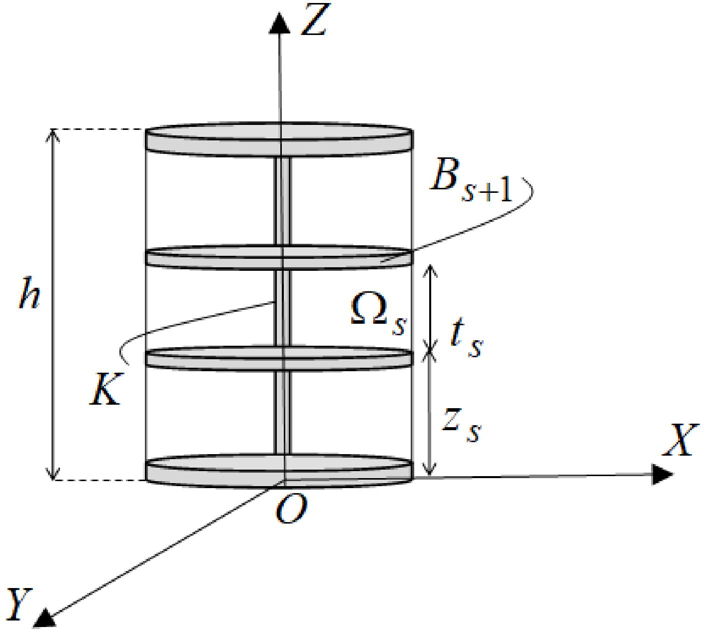
Motivation

To achieve a stable processing quality and the most “uniform” distribution of thermal and power effects, the parts have to be placed sufficiently distant one from another, as well as from the lateral cylindrical surface of the container.

The gas mixture is ignited by a spark and the temperature of the subsequent combustion ranges from 2,500 to 3,300 °C. The burr reaches its ignition temperature and reacts with the excess oxygen inside the deburring chamber. This leads to a complete combustion of the burr in a few milliseconds. The overall cycle time of the thermal deburring process is less than two minutes.

Problem formulation: Container

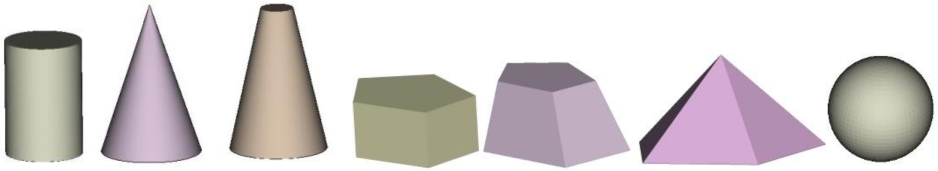
The cylindrical container Ω is divided into sub-containers Ω_s, $s = 1, \ldots, m-1$, by the horizontal cylindrical shelves B_s fixed on a thin cylindrical rod K passing through the center of the rack.



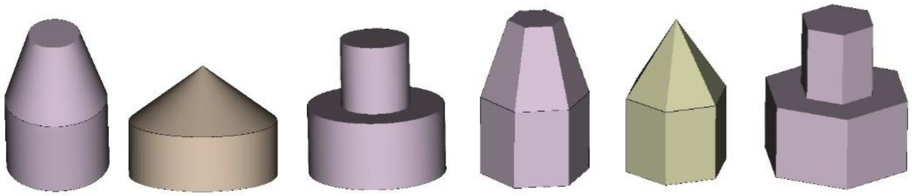
Problem formulation: Placement objects

$G = \{T_q \subset \mathbb{R}^3,\ q = 1,\ 2,\ \ldots,\ N\}$ is the set of objects

Each object $T_q,\ \ q \in I_N = \{1,\ 2,\ \ldots,\ N\}$ can be presented by a union of 3D basic convex objects $T_i^q, i = 1,\ \ldots,\ n_q$



(a)



(b)

Placement objects: (a) basic convex; (b) composed

Problem formulation: Placement objects

$T_q(u_q) = \{\tilde{p} \in \mathbb{R}^3 : \tilde{p} = v_q + A(\theta_q) \cdot (p)^T, \ \forall p \in T_q\}$ is the object T_q rotated by an angle θ_q and translated by a vector $v_q = (x_q, y_q, z_q)$ on the shelf B_s

$u_q = (v_q, \theta_q)$ is a vector of variable placement parameters

$$A(\theta_q) = \begin{pmatrix} \cos\theta_q & -\sin\theta_q & 0 \\ \sin\theta_q & \cos\theta_q & 0 \\ 0 & 0 & 1 \end{pmatrix}$$ is a rotation matrix

w_q is the weight of the object T_q, $q \in I_N$

For each object $T_q \in G_s$ only sliding movements (translation and rotation) on the corresponding shelf B_s are allowed.

Problem formulation: Placement objects

The set G is divided into m-1 disjoint subsets $G_s,\ \ s \in J_{m-1}$:

$T_q \in G_s$ if T_q has to be assigned to the sub-container $\Omega_s,\ \ q \in I_N$ (to the shelf B_s). Correspondingly, for $\mathbf{I}^s = \{q \mid \ T_q \in \Omega_s\},\ \ \bigcup_{s=1}^{m-1} \mathbf{I}^s = I_N$

$\mathbf{I}^1 = \{1,\ \dots,\ k_1\},\quad \mathbf{I}^2 = \{k_1+1,\ \dots,\ k_2\},..,\quad \mathbf{I}^{m-1} = \{k_{m-2}+1,\ \dots,\ N\}.$

Problem formulation: Placement constraints

Let ρ_s denote the minimum of Euclidean distances between objects $T_q(u_q)$ and $T_g(u_g)$, $(q, g) \in \Xi^s$, as well as, between each object $T_q(u_q)$ and the object $\mathfrak{I}^*$ for $q \in \mathbf{I}^s$, $s \in J_{m-1}$, i.e.

$$\rho_s = \min\{dist(T_q(u_q), T_g(u_g)), (q, g) \in \Xi^s, \; dist(T_q(u_q), \mathfrak{I}^*), q \in \mathbf{I}^s\},$$

$$dist(T_q(u_q), T_g(u_g)) = \min_{t_q \in T_q(u_q), t_g \in T_g(u_g)} \| t_q - t_g \|,$$

$$dist(T_q(u_q), \mathfrak{I}^*) = \min_{t_q \in T_q(u_q), t \in \mathfrak{I}^*} \| t_q - t \|.$$

Problem formulation: Placement constraints

$$dist(T_q(u_q),\ T_g(u_g)) \geq \rho_s \text{ for } (q,\ g) \in \Xi^s,\ s \in J_{m-1}$$

The distance between the objects T_q and T_g has to be at least ρ_s

$$dist(T_q(u_q),\ \Im^*) \geq \rho_s \text{ for } q \in \mathbf{I}^s,\ s \in J_{m-1}$$

The distance between the objects T_q and $\Im^*$ has to be at least ρ_s

$$dist(T_q(u_q),\ K_\Omega) \geq \Delta \text{ for } q \in I_N$$

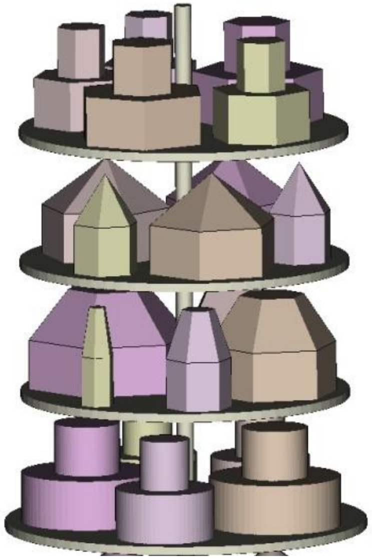
The distance between the object T_q and the “rod” K_Ω has to be at least Δ

$$z_q = z_s \text{ for } q \in \mathbf{I}^s,\ s \in J_{m-1}$$

For each object $T_q \in G_s$ only sliding movements on the shelf B_s are allowed

Problem formulation: Balancing conditions

The set Λ denotes the container Ω filled by the objects $T_q(u_q), q \in I_N$



$$M = \sum_{q=0}^{N} w_q$$ is the weight of Λ

The coordinate deviation of the gravity center $O_c = (x_c,\ y_c,\ z_c)$ from the center point $(0,\ 0,\ z_0)$ of the container has to be within a certain given threshold $\hat{\delta} \geq 0$:

$$-\hat{\delta} \leq x_c \leq \hat{\delta},\ -\hat{\delta} \leq y_c \leq \hat{\delta},\ 0 \leq z_c \leq h,$$

$$x_c(x) = \sum_{q=1}^{N} w_q x_q / M,\ y_c(y) = \sum_{q=1}^{N} w_q y_q / M$$

$$z_c(z) = \sum_{q=0}^{N} w_q (z_q + h_{zq}) / M$$

$$x = (x_1,\ \ldots,\ x_N),\ y = (y_1,\ \ldots,\ y_N),$$

$$z = (z_1,\ \ldots,\ z_N)$$

Sparse balance packing problem

Packing objects is referred to as the *sparse* if it *maximizes*

$$\sum_{s=1}^{m-1} \rho_s \,,$$

Sparse balance packing. Find the *sparse* arrangement of the objects $T_q, \; q \in \mathbf{I}^s$, inside sub-containers $\Omega_s, \; s \in J_{m-1}$, taking into account the *placement* constraints and *balancing* conditions.

The Sparse packing is aimed to place the 3D objects as distant as possible, freely sliding and rotating on the shelves subject to balancing conditions.

Tools of mathematical modelling:
The phi-function technique

To describe analytically the sparse placement constraints the phi-function technique is used.

Yu. Stoyan is the founder of the phi-function technique

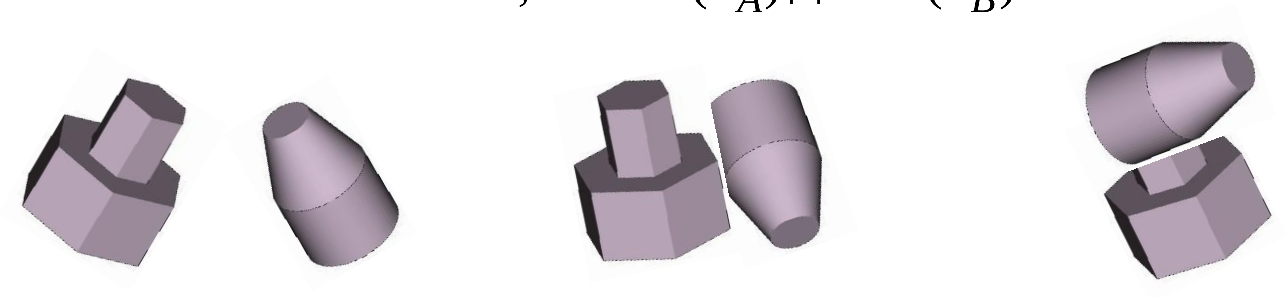
Phi-function

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

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

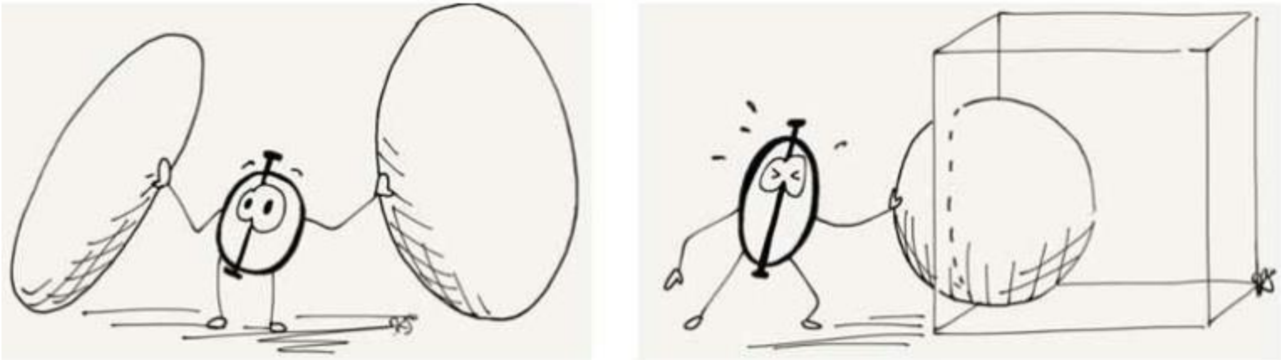
$$\Phi^{AB} = 0, \text{ if } \operatorname{int} A(u_A) \cap \operatorname{int} B(u_B) = \varnothing \text{ and } frA(u_A) \cap frB(u_B) \neq \varnothing;$$

$$\Phi^{AB} < 0, \text{ if } \operatorname{int} A(u_A) \cap \operatorname{int} B(u_B) \neq \varnothing.$$



[1]Chernov N., Stoyan Yu., Romanova T. Mathematical model and efficient algorithms for object packing problem. **Computational Geometry: Theory and Applications**. 2010, 43(5), P. 535–553.

Phi-functions



Phi-functions in their **positive mood** serve two purposes:
Keeping objects apart (*left*) and keeping objects into a target container (*right*).

The picture is produced by Diana Kallrath for chapter
Yu.Stoyan & T. Romanova
“Cutting & Packing beyond and within Mathematical Programming” in book
Kallarch, J.: Business Optimisation Using Mathematical Programming, 2nd Edition, 2021, Springer http://www.springer.com

Quasi-Phi-function

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

The main property of the quasi-phi-function for two objects $A(u_A)$ and $B(u_B)$ is :

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

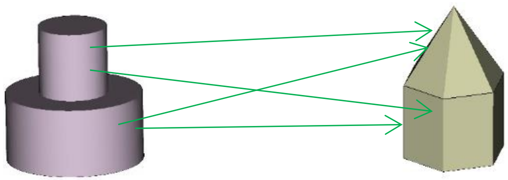
[1]Stoyan, Y., Pankratov, A., Romanova, T. Quasi-phi-functions and optimal packing of ellipses (2016) **Journal of Global Optimization**, 65 (2), pp. 283-307. DOI: 10.1007/s10898-015-0331-2

Quasi-phi-function for composed objects

$$T_q(u_q) = \bigcup_{i=1}^{n_q} T_i^q(u_q) \text{ and } T_g(u_g) = \bigcup_{j=1}^{n_g} T_j^g(u_g)$$

$$\Phi'_{qg}(u_q, u_g, \tau_{qg}) = \min_{i=1,..,n_q,\, j=1,...,n_g} \Phi'^{qg}_{ij}(u_q, u_g, \tau^{qg}_{ij}),$$

Φ'^{qg}_{ij} is a quasi-phi-function for basic convex objects $T_i^q(u_q)$ and $T_j^g(u_g)$



$$\Phi'_{qg} \geq 0 \Leftrightarrow \{\Phi'^{qg}_{ij} \geq 0,\ i=1,\ ...,\ n_q,\ \ j=1,\ ....,\ n_g\}$$

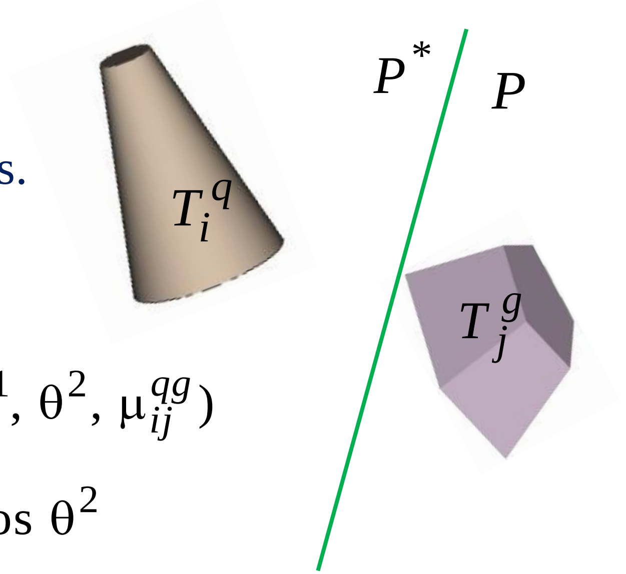
$$\Phi'_{qg} \geq 0 \text{ implies } \operatorname{int} T^q(u_q) \cap \operatorname{int} T^g(u_g) = \varnothing$$

Quasi-phi-function for convex objects

$$\Phi'^{qg}_{ij}(u_q, u_g, \tau^{qg}_{ij}) = \min\{\Phi^q_i(u_q, \tau^{qg}_{ij}), \Phi^{*g}_j(u_g, \tau^{qg}_{ij})\}$$

$\Phi^q_i(u_q, \tau^{qg}_{ij})$ is a *quasi*-phi-function for $T^q_i(u_q)$ and the half space $P = P^{qg}_{ij}(u_q, \tau^{qg}_{ij})$, $\Phi^{*g}_j(u_g, \tau^{qg}_{ij})$ is a *quasi*-phi-function for $T^g_j(u_g)$ and the half space $P^* = P^{*qg}_{ij}(u_q, \tau^{qg}_{ij}) = \mathbb{R}^3 \setminus \text{int } P^{qg}_{ij}$,

$\tau^{qg}_{ij} = (\theta^{1qg}_{ij}, \theta^{2qg}_{ij}, \mu^{qg}_{ij})$ is the vector of auxiliary variables.



$$P^{qg}_{ij}(p, \tau^{qg}_{ij}) = \{p = (x, y, z) : \psi^{qg}_{ij}(p, \tau^{qg}_{ij}) \le 0\}$$

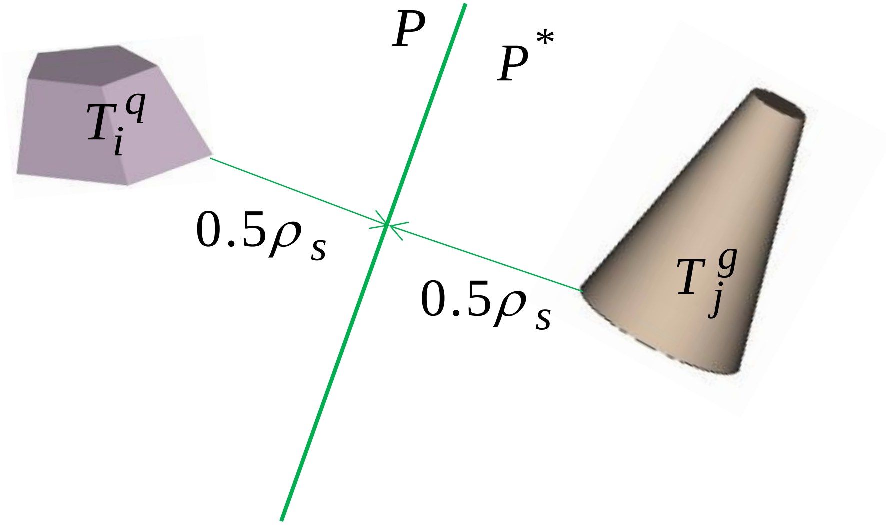
$$\psi^{qg}_{ij}(p, \tau^{qg}_{ij}) = \alpha^{qg}_{ij} \cdot x + \beta^{qg}_{ij} \cdot y + \gamma^{qg}_{ij} \cdot z + \mu^{qg}_{ij}, \ \tau^{qg}_{ij} = (\theta^1, \theta^2, \mu^{qg}_{ij})$$

$$\alpha^{qg}_{ij} = \cos\theta^1 \cdot \cos\theta^2, \ \beta^{qg}_{ij} = -\sin\theta^2, \ \gamma^{qg}_{ij} = \sin\theta^1 \cdot \cos\theta^2$$

An adjusted quasi-phi-function for basic objects

$$\widehat{\Phi}'^{qg}_{ij}(u_q, u_g, \tau^{qg}_{ij}, \rho_s) = 2\Phi'^{qg}_{ij}(u_q, u_g, \tau^{qg}_{ij}) - \rho_s,$$

$$\Phi'^{qg}_{ij}(u_q, u_g, \tau^{qg}_{ij}) = \min\{\Phi^{q}_{i}(u_q, \tau^{qg}_{ij}), \Phi^{*g}_{j}(u_g, \tau^{qg}_{ij})\}$$



$\widehat{\Phi}'^{qg}_{ij} \geq 0$ implies $dist(T^q(u_q), T^g(u_g)) \geq \rho_s$

An adjusted quasi-phi-function for basic objects

$$\Phi'_{AB}(u_A, u_B, \tau_{AB}) = \min\{\Phi'_A(u_q, \tau_{AB}), \Phi'^{*}_B(u_g, \tau_{AB})\},$$

If the object $A(u_A)$ is the sphere, then

$$\Phi'_A(u_A, \tau_{AB}) = \cos\theta^1_{AB} \cdot \cos\theta^2_{AB} \cdot (x_A + x_A \cos\theta_A - y_A \sin\theta_A)$$
$$- \sin\theta^2_{AB} \cdot (y_A + x_A \sin\theta_A + y_A \cos\theta_A) + \sin\theta^1_{AB} \cdot \cos\theta^2_{AB} \cdot z_A + \mu_{AB} - r_A.$$

If the object $A(u_A)$ is the polytope, then

$$\Phi'_A(u_A, \tau_{AB}) = \min\{\cos\theta^1_{AB} \cdot \cos\theta^2_{AB} \cdot (x_{Ak} + x_{Ak} \cos\theta_A - y_{Ak} \sin\theta_A)$$
$$- \sin\theta^2_{AB}(y_{Ak} + x_{Ak} \sin\theta_A + y_{Ak} \cos\theta_A) + \sin\theta^1_{AB} \cdot \cos\theta^2_{AB} \cdot z_{Ak} + \mu_{AB},\ k = 1, \ldots, m_A\}$$

An adjusted quasi-phi-function for basic objects

If the object $A(u_A)$ belongs to the family $\mathfrak{I}$, then

$$\Phi'_A(u_A, \tau_{AB}) = \min\{ f_1(u_A, \tau_{AB}),\ f_2(u_A, \tau_{AB})\}$$

$$f_1(u_A, \tau_{AB}) = \cos\theta^1_{AB} \cdot \cos\theta^2_{AB} \cdot (x_{A1} + x_{A1}\cos\theta_A - y_{A1}\sin\theta_A)$$
$$-\sin\theta^2_{AB} \cdot (y_{A1} + x_{A1}\sin\theta_A + y_{A1}\cos\theta_A) +$$
$$\sin\theta^1_{AB} \cdot \cos\theta^2_{AB} \cdot z_{A1} + \mu_{AB} - r_{A1}\sqrt{1 - (\mathbf{n}_{AB} \cdot \mathbf{n}_A)^2}$$

$$f_2(u_A, \tau_{AB}) = \cos\theta^1_{AB} \cdot \cos\theta^2_{AB} \cdot (x_{A2} + x_{A2}\cos\theta_A - y_{A2}\sin\theta_A)$$
$$-\sin\theta^2_{AB} \cdot (y_{A2} + x_{A2}\sin\theta_A + y_{A2}\cos\theta_A) +$$
$$\sin\theta^1_{AB} \cdot \cos\theta^2_{AB} \cdot z_{A2} + \mu_{AB} - r_{A2}\sqrt{1 - (\mathbf{n}_{AB} \cdot \mathbf{n}_A)^2}$$

Mathematical model

The *sparse* packing problem with *balancing* conditions can be formulated as the following nonlinear optimization problem:

$$\max \sum_{s=1}^{m-1} \rho_s \text{ s.t. } (u,\ \tau,\ \rho) \in W, \quad (1)$$

$$W = \{(u,\ \tau,\ \rho): \widehat{\Phi}'_{qg}(u_q,\ u_g,\ \tau_{qg},\ \rho_s) \geq 0,\ \ (q,\ g) \in \Xi^s,\ \widehat{\Phi}_q(u_q,\ \rho_s) \geq 0, \quad (2)$$

$$\widehat{\Phi}^{T_q K}(u_q) \geq 0,\ q \in \mathbf{I}^s,\ \rho_s > 0,\ s \in J_{m-1},\ \Upsilon(v) \geq 0\},$$

where $u = (u_1,\ \ldots,\ u_N),\ \rho = (\rho_1,\ \ldots,\ \rho_{m-1}),$
$\tau = (\tau^1,\ \ldots,\ \tau^{m-1}),\ \tau^s = (\tau_{qg},\ (q,\ g) \in \Xi^s),\ s \in J_{m-1}$

Mathematical model

In the model (1)-(2):

τ_{qg} is a vector of the auxiliary variables,

$\widehat{\Phi}'_{qg}(u_q, u_g, \tau_{qg}, \rho_s)$ is the adjusted quasi-phi-function for the objects $T_q(u_q)$ and $T_g(u_g)$, $(q, g) \in \Xi^s$, $s \in J_{m-1}$,

$\widehat{\Phi}_q(u_q, \rho_s)$ is the adjusted phi-function for the objects $T_q(u_q)$ and $\Im^*$, $q \in \mathbf{I}^s$, $s \in J_{m-1}$, $\mathbf{I}^s = \{q \mid T_q \subset \Omega^s\}$,

$\widehat{\Phi}^{T_q K}(u_q)$ is the adjusted phi-function for the objects $T_q(u_q)$ and “rod” K_Ω, $q \in \mathbf{I}^s$,

$\Upsilon(v) \geq 0$ states the balancing conditions.

Mathematical model

The feasible region W given by (1)–(2) is defined by a system of non-smooth inequalities that can be reduced to a system of inequalities with differentiable functions.

The model (1)–(2) is a non-convex and continuous nonlinear programming problem. This is an exact formulation in the sense that it gives all optimal solutions to the *sparse balance* packing problem.

Mathematical model

The number of the problem variables is $\sigma = m - 1 + 3N + 3\sum_{s=1}^{m-1} \sum_{(q,g)\in\Sigma_s} n_q n_g$.

The model (1)–(2) involves $O(N^2)$ nonlinear inequalities and $O(N^2)$ variables due to the auxiliary variables in the quasi-phi-functions.

Solution strategy

The following multistart strategy is used to solve the problem (1)–(2).

Stage 1. A number of feasible starting points are generated based on the homothetic transformations of objects subject to balancing conditions.

Stage 2. A local maximum of the problem (1)–(2) is obtained starting from each feasible point generated at the first stage. For local optimisation the IPOPT code (https://github.com/coin-or/Ipopt).

Stage 3. The best solution is selected from those obtained at the second stage. This result is considered as the approximate solution of the problem (1)–(2).

Algorithm for generating feasible starting points

Step 1. Form a set of points $v_q^0 = (x_q^0,\ y_q^0,\ z_q^0)$ within the sub-container Ω_s for $q \in \mathbf{I}^s, s \in J_{m-1}$, where $x_q^0,\ y_q^0$ are randomly chosen, $(x_q^0,\ y_q^0) \in B_s$, and $z_q^0 = z_s$. Form the point $v^{(0)} = (v_1^{(0)}, \dots, v_N^{(0)})$. The result of this step is a collection of points placed inside the sub-containers.

Algorithm for generating feasible starting points

Step 2. Start from the point (v^0, δ^0), where $\delta^0 = r$. Search for a balance packing of N weighted points (degenerated objects of the given weights) within the container Ω, solving the following nonlinear programming problem:

$$\min \delta \text{ s.t. } (\delta, v) \in V \subset R^{3N+1},$$

$$V = \{(\delta, v) : \omega(x_q, y_q, z_q) \geq 0,\ z_q = z_q^s,\ q \in \mathbf{I}^s,\ s \in J_{m-1},$$

$$-\delta \leq x_c \leq \delta,\ -\delta \leq y_c \leq \delta\},$$

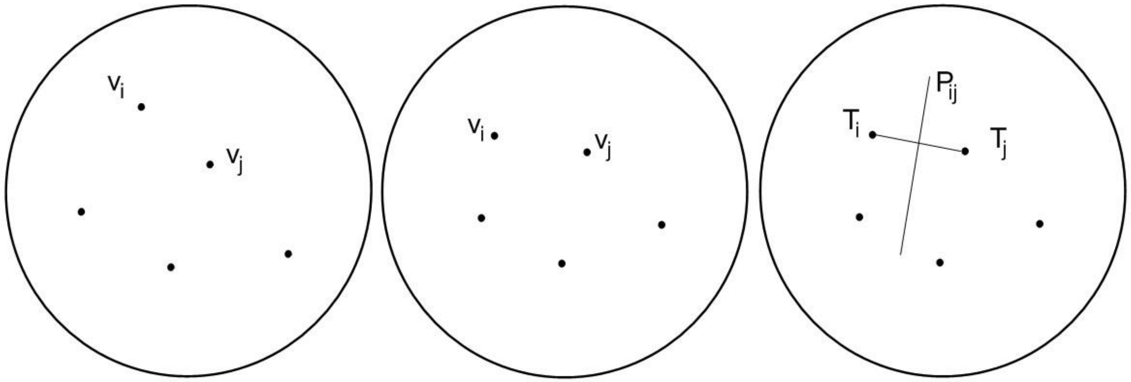
where $\omega(x_q, y_q, z_q) = \min\{r^2 - x_q^2 - y_q^2,\ x_q^2 + y_q^2 - \Delta^2\}$.

Find the point $(\delta^{(1)}, v^{(1)})$ of a local minimum of the problem, where $v^{(1)} = (v_1^{(1)}, \ldots, v_n^{(1)})$. The result of this step is a collection of weighed points providing the balance conditions.

Algorithm for generating feasible starting points

Step 3. Derive starting values of variable vector τ_{qg} for each quasi-phi-function $\Phi'_{qg}(u_q, u_g, \tau_{qg}, \rho_s) \geq 0, \ (q, g) \in \Xi^s$.

These variables are considered as parameters of separating planes between each pair of the points $v_q^{(1)}$ and $v_g^{(1)}, q \in \mathbf{I}^s, s \in J_{m-1}$. Find $\tau^{(1)} = (\tau_{qg}^{(1)}, q \in \mathbf{I}^s, s \in J_{m-1})$.



Illustrations to Steps 1-3

Algorithm for generating feasible starting points

Step 4. Form the point $(u^{(1)}, \tau^{(1)})$, where $u^{(1)} = (v^{(1)}, \theta^{(1)})$, $\theta^{(1)} = (\theta_1^{(1)}, \ldots, \theta_N^{(1)})$ is the vector of randomly generated rotation parameters.

The point $(u^{(1)}, \tau^{(1)})$ obtained at Steps 1-4 is used as a starting point for the nonlinear programming problem defined at the next step.

Algorithm for generating feasible starting points

Step 5. Extend the weighted points $v_q^{(1)}$ inside the appropriate sub-containers Ω_s, $q \in \mathbf{I}^s$, $s \in J_{m-1}$, to the original objects of the given shapes and sizes, solving the following nonlinear programming problem:

$$\max \lambda \text{ s.t. } (u, \tau, \lambda) \in W_\lambda, \tag{3}$$

$$W_\lambda = \{(u, \tau, \lambda) : \widehat{\Phi}'_{qg}(u_q, u_g, \tau_{qg}, \lambda) \geq 0, \ (q, g) \in \Xi^s, \tag{4}$$

$$\widehat{\Phi}_q(u_q, \lambda) \geq 0, \ \widehat{\Phi}^{T_q K}(u_q, \lambda) \geq 0, \ q \in \mathbf{I}^s, \ s \in J_{m-1}, \ \Upsilon(u) \geq 0, \ 0 \leq \lambda \leq 1\},$$

starting from the point $(u^{(1)}, \tau^{(1)}, \lambda = 0)$.

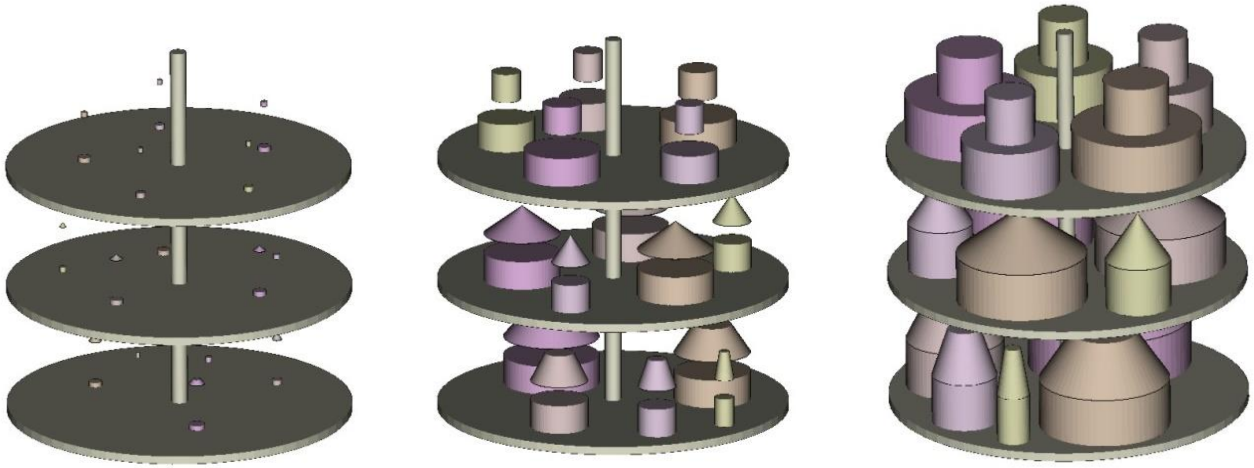
The constraints of the problem (3)–(4) assure non-overlapping, containment and balancing of freely rotated scaled objects for variable scaling parameter $0 \leq \lambda \leq 1$

Algorithm for generating feasible starting points

Find a point $(u^{(2)}, \tau^{(2)}, \lambda^{(2)})$ providing an optimal solution to the problem (3)–(4). If $\lambda^{(2)} = 1$, then go to Step 6, else ($\lambda^{(2)} < 1$) return to Step 1.

The case $\lambda^{(2)} = 1$ corresponds to non-overlapping balanced original sized objects feasibly placed in the sub-containers. If $\lambda^{(2)} < 1$, then the feasible balance packing for the original sized objects can't be obtained from the collection of random points used at Step 1, and a new collection has to be generated.

Algorithm for generating feasible starting points



(a) (b) (c)

Illustration to Step 5 : a) the intermediate feasible arrangement of 3D objects for $\lambda = 0.1$; b) the intermediate feasible arrangement of objects for $\lambda = 0.6$; c) feasible arrangement of original sized objects ($\lambda = 1$).

Algorithm for generating feasible starting points

Step 6. Calculating the minimal distance between all pairs of objects in each sub-container.

Define

$$\rho_s^{(2)} = \min\{\widehat{\Phi}'_{qg}(u_q^{(2)}, u_g^{(2)}, \tau_{qg}^{(2)}),\ (q, g) \in \Xi^s,\ \widehat{\Phi}_q(u_q^{(2)}),\ q \in \mathbf{I}^s\} \text{ for } s \in J_{m-1}$$

as a starting value for the variable ρ_s in the problem (1)–(2).

Algorithm for generating feasible starting points

After completing Step 6, a point $(u^{(2)}, \tau^{(2)}, \rho^{(2)})$ feasible to the problem (1)–(2) is generated.

Step 7. Solve the problem (1)–(2), starting from the point $(u^{(2)}, \tau^{(2)}, \rho^{(2)})$.

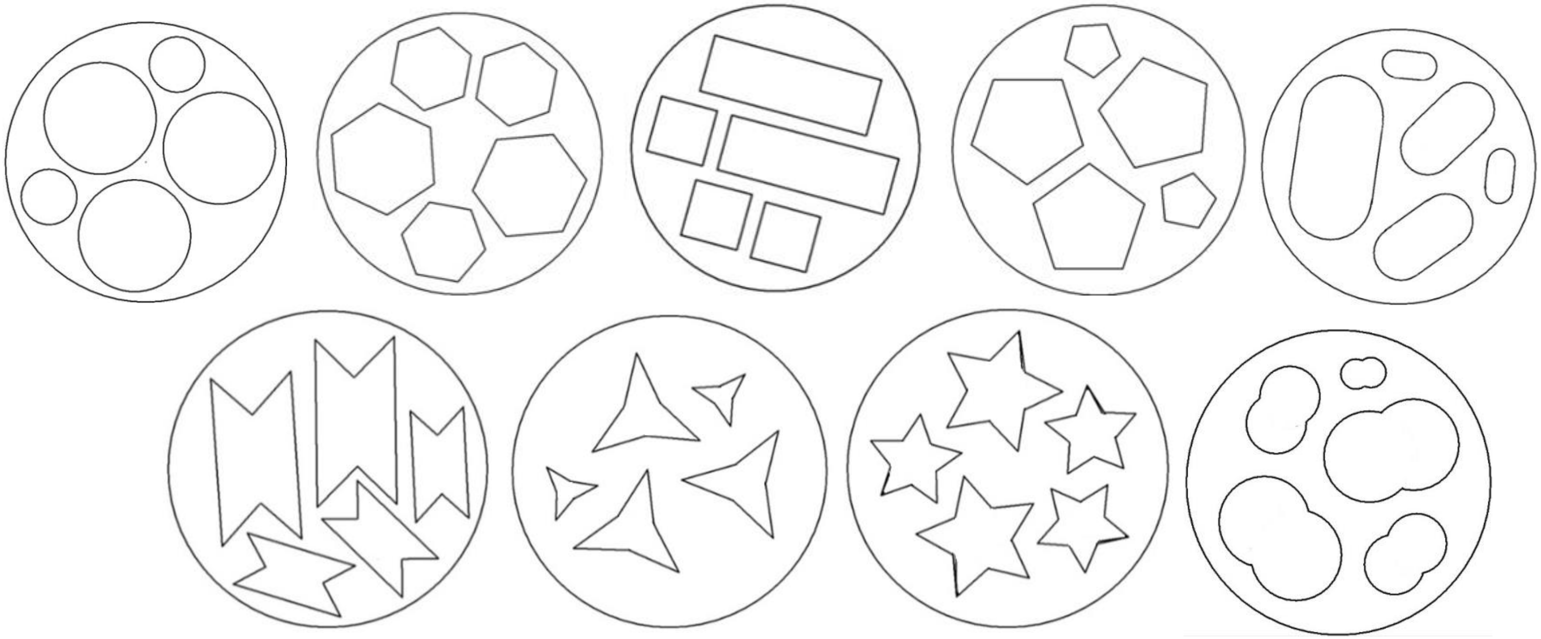
Computational results

Benchmark instances are provided to illustrate the proposed methodology.

Instances with real industry data are also considered.

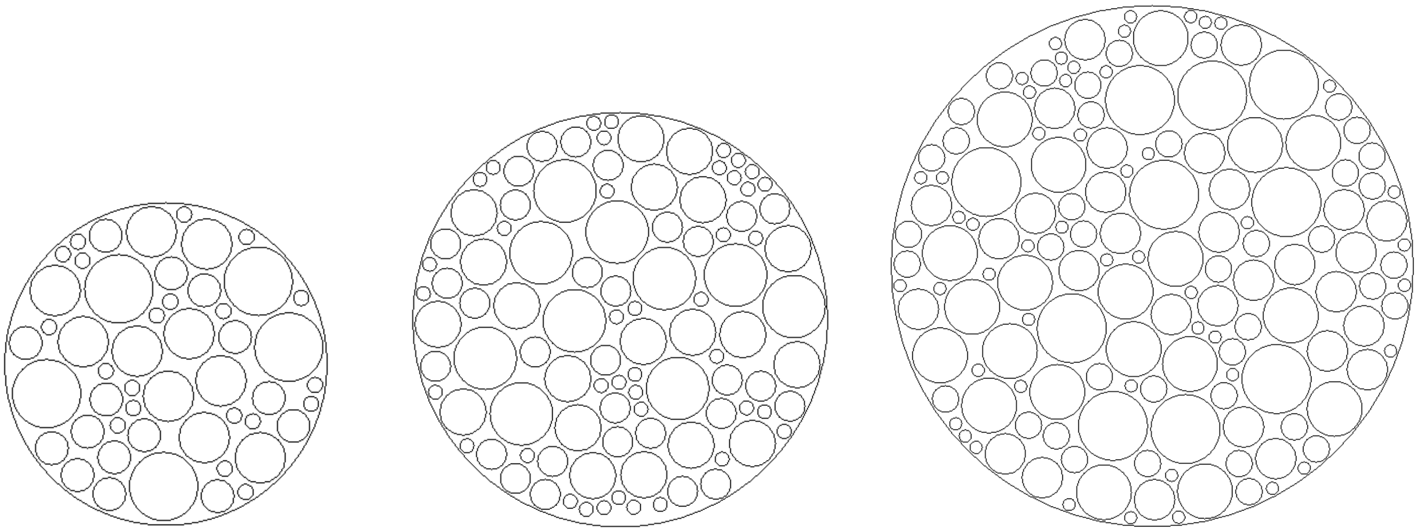
All experiments were running on an AMD FX(tm)-6100, 3.30 GHz computer, Programming Language C++, Windows 7. For local optimisation the IPOPT code (https://github.com/coin-or/Ipopt) reported in (Wächter & Biegler, 2006) was used under default options.

Examples of local optimal sparse packings of 2D objects in a circular container[1]



[1]Romanova T., Pankratov A., Litvinchev et al. Sparsest packing of two-dimensional objects. **International Journal of Production Research**. 2021. 59(13), 3900-3915, doi.org/10.1080/00207543.2020.1755471

Examples of local optimal sparse packings of circles in a circular container[1]



[1]T. Romanova, A. Pankratov, I. Litvinchev, P. Stetsyuk, A. Likhovid, et al, Balanced Circular Packing Problems with Distance Constraints. In: V. Kharchenko, G.-W. Weber, J. Thomas, P. Vasant (Eds), **Artificial Intelligence Approaches for Renewable Energy and Agro Engineering**, River Publishers, 2021 (to appear)

Computational results: 3D objects

It is assumed that: $h = 24.5$, $r = 10$, $\delta_0 = 0$, $w_0 = 0$, $\bar{h} = 0.5$ and $\tilde{r} = 0.5$, $h_q = 6$, $h_{zq} = 3$ for $q = 1, \ldots, N$, $t_s = 7.5$ for $s = 1, .., m-1$, $m = 4$ for Examples 1–5.

For each problem instance **100 starting points** were generated.

Then **100 corresponding local minima** were obtained by our algorithm.

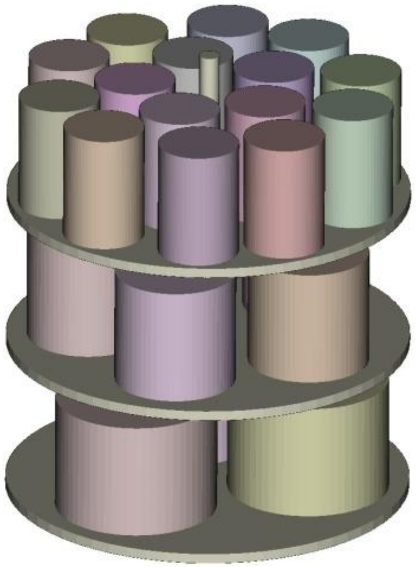
The best local mimimum was selected as an approximate solution to the problem (1)–(2).

The CPU time indicated for each problem instance is the total time for all 100 runs.

Computational results

Example 1. The sparse packing of $N = 23$ cylinders.

The best solution obtained by the algorithm for 69.43 sec. is $\rho^* = 2.246959$, the corresponding optimized distances for Ω_s, $s = 1, 2, 3$ are $\rho_1^* = 0.875644$, $\rho_2^* = 1.024558$, $\rho_3^* = 0.346757$.

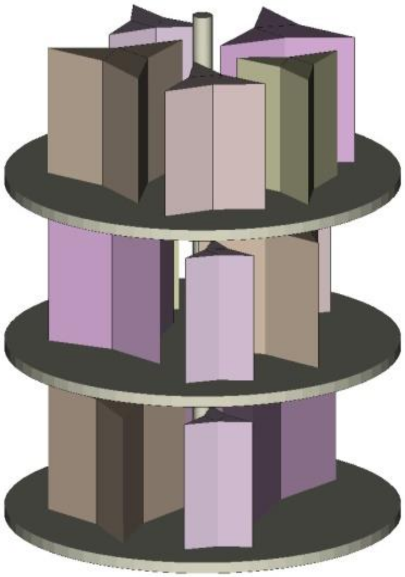


Sparse balance packing of cylinders

Computational results

Example 2. The sparse packing of $N = 15$ non-convex prisms.

The best solution obtained by the algorithm for 555.05 sec. is $\rho^* = 6.551790$, the corresponding optimized distances for Ω_s, $s = 1, 2, 3$ are $\rho_1^* = 2.162019$, $\rho_2^* = 2.208503$, $\rho_3^* = 2.181267$.



Sparse balance packing of non-convex prisms

Computational results

Example 3. The sparse packing $N = 15$ convex truncated pyramids.

The best solution obtained by the algorithm for 198.75 sec. is $\rho^* = 2.813018$, the corresponding optimized distances for Ω_s, $s = 1, 2, 3$ are $\rho_1^* = 0.454915$, $\rho_2^* = 1.296247$, $\rho_3^* = 1.061855$.

Sparse balance packing of convex truncated pyramids

Computational results

Example 4. The sparse packing of $N = 15$ non-convex composed objects.

The best solution obtained by the algorithm for 298.58 sec. is $\rho^* = 2.493670$, the corresponding optimized distances for Ω_s, $s = 1, 2, 3$ are $\rho_1^* = 0.504725$, $\rho_2^* = 1.089519$, $\rho_3^* = 0.899425$.

Sparse balance packing of composed objects

Computational results

Example 5. The sparse packing of $N = 15$ basic convex and composed non-convex objects.

The best solution obtained by the algorithm for 52.88 sec. is $\rho^* = 2.741667$, the corresponding optimized distances for Ω_s, $s = 1, 2, 3$ are $\rho_1^* = 0.5$, $\rho_2^* = 1.259309$, $\rho_3^* = 0.982356$.

Sparse balance packing of basic and composed objects

Computational results

The following instances are based on real-world industrial data corresponding to Sparse balance packing of parts processed in a thermal deburring machine.

Example 6. The iTEM400/600 thermal deburring machine (ATL Anlagentechnik Luhden, 2019) used for processing the parts has the following geometric parameters. The circular deburring chamber has 400 mm diameter and 600 mm height, while the other parameters are as follows: $h=580$, $r=200$, $\delta_0=0$, $w_0=0$, and $\tilde{r}=10$, $h_q=125$ for $q=1,\ldots,12$, $\bar{h}=10, t_s=132.5$. The shapes of the objects in this example are similar to the shapes of the parts used in hydraulic units of **aircraft engines**.

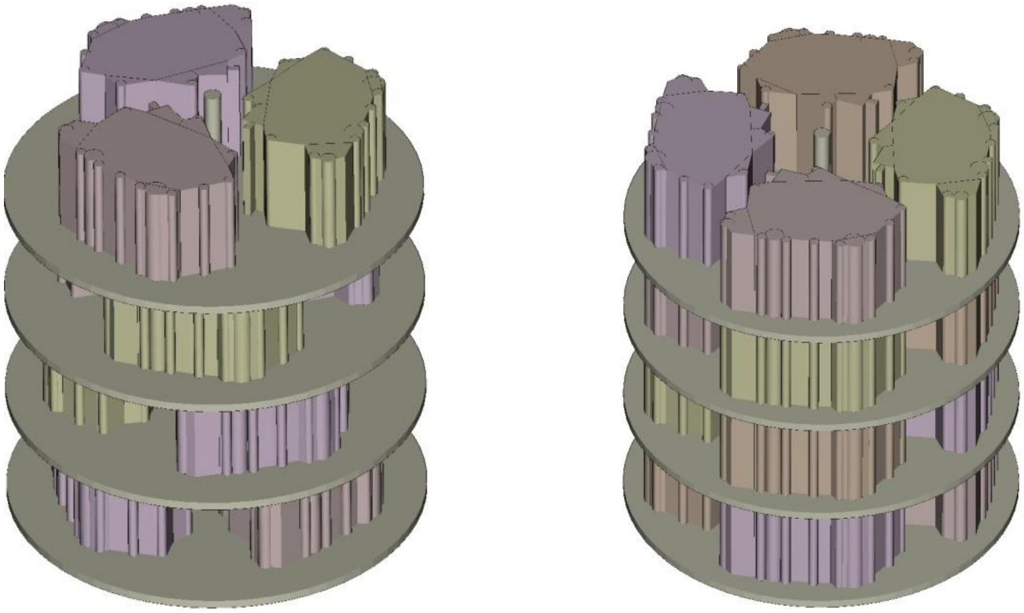
Computational results

The best solutions obtained by the algorithm for these two cases are:

a) $\rho^* = 135.0881$, the corresponding optimized distances for Ω_s, $s = 1, \ldots, 4$ are $\rho_1^* = \rho_2^* = \rho_3^* = \rho_4^* = 33.772$ mm, the CPU time is 3636.07 sec;

b) $\rho^* = 107.0438$, the corresponding optimized distances for Ω_s, $s = 1, \ldots, 4$ are $\rho_1^* = \rho_2^* = \rho_3^* = \rho_4^* = 17.8406$ mm, the CPU time is 8351.9 sec.

Computational results



(a) (b)

Sparse balance packings of objects (units of aircraft engine) in Example 6:

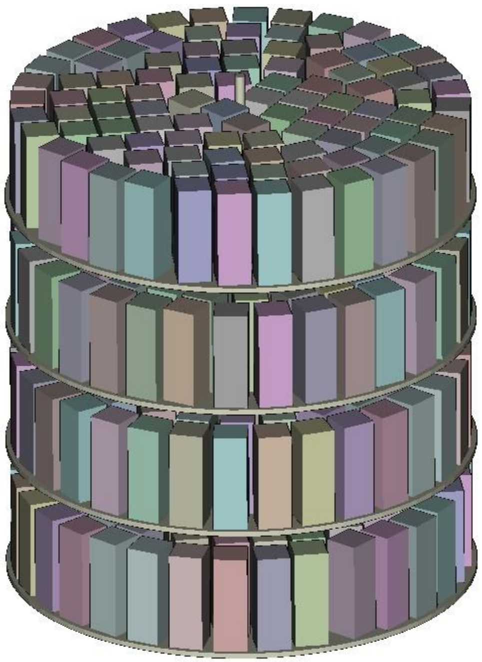
a) three objects in each sub-container; b) four objects in each sub-container

Computational results

For problem instances 7–9 we used $h=31.5$, $r=14$, $\bar{w}=96, \delta_0=0$, $w_0=0$, $\bar{h}=0.3$ and $\tilde{r}=0.3$, $h_q=6$, $h_{zq}=3$ for $q=1, \ldots, 100$, $t_s=7.5$ for $s=1, .., m-1$, $m=5$. The shapes of the objects are similar to those presented in, e.g., Custom Design Components (2016); Benseler (2020).

Computational results

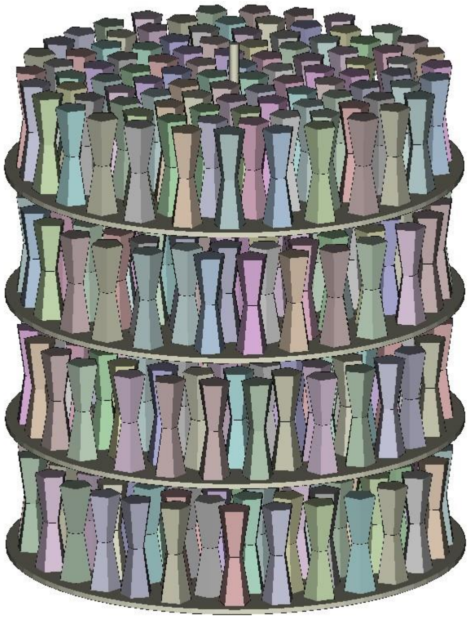
Example 7. The Sparse packing of $N = 400$ cuboids. The best solution obtained by the algorithm for 24983.78 sec is $\rho^* = 0.28768$.



Sparse packing cuboids

Computational results

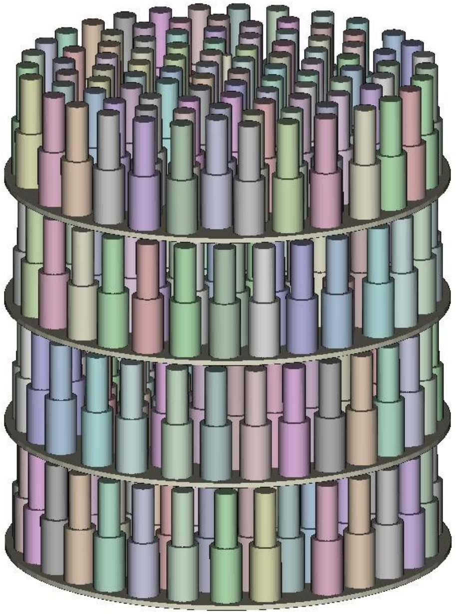
Example 8. The Sparse packing of $N = 400$ non-convex objects composed by the right prisms. The best solution obtained by the algorithm for 45256.73 sec is $\rho^* = 0.723218$.



Sparse packing of non-convex composed objects

Computational results

Example 9. The Sparse packing of $N = 400$ non-convex objects composed by circular cylinders. The best solution obtained by the algorithm for 3855.611 sec. is $\rho^* = 0.481760$.



Sparse packing of non-convex composed objects

Related papers

Stoyan, Y.G., Romanova, T.E., Pankratov, O.V., Stetsyuk, P.I., Stoian, Y.E. Sparse Balanced Layout of Spherical Voids in Three-Dimensional Domains (2021) **Cybernetics and Systems Analysis**, 57(4), 542–551.

Romanova T., Stoyan Y., Pankratov A., Plankovskyy, S., Tsegelnyk, Y., Shypul, O: Sparsest balanced packing of irregular 3D objects in a cylindrical container. **European Journal of Operational Research**. 2021. 291(1), 84–100.

Pankratov, A., Romanova, T., Litvinchev, I.: Packing Oblique 3D Objects. **Mathematics**, 8(7), 1130, (2020).

Romanova, T., Litvinchev, I., Pankratov, A.: Packing ellipsoids in an optimized cylinder. **European Journal of Operational Research**, 285(2), 429-443, (2020).

Romanova, T., Bennell, J., Stoyan, Y., Pankratov, A.: Packing of concave polyhedra with continuous rotations using nonlinear optimization. **European Journal of Operational Research**, 268(1), 37–53, (2018).

Stoyan, Y., Romanova, T., Pankratov, A., Kovalenko, A., Stetsyuk, P. Balance layout problems: Mathematical modeling and nonlinear optimization (2016) **Springer Optimization and Its Applications**, 114, pp. 369-400. DOI: 10.1007/978-3-319-41508-6_14

Stetsyuk, P.I., Romanova, T.E., Scheithauer, G. On the global minimum in a balanced circular packing problem (2016) **Optimization Letters**, 10 (6), pp. 1347-1360. DOI: 10.1007/s11590-015-0937-9

Acknowledgements

It is my pleasure to thank my colleagues who have contributed
to the success of this research

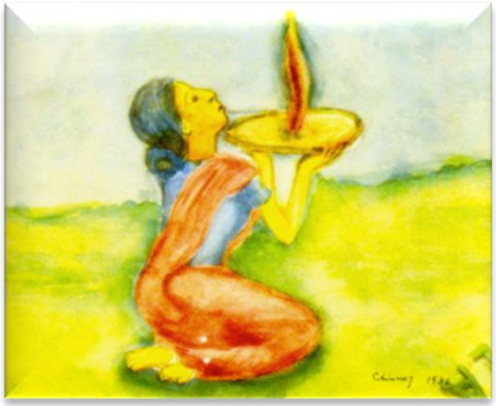
Stoyan, Y., Pankratov, A. (*A.Pidgorny Institute of Mechanical Engineering Problems of the National Academy of Sciences of Ukraine*)

Litvinchev, I. (*Nuevo Leon State University, San Nicolas de los Garza, Mexico*)

Stetsyuk, P., A. Likhovid (*V.M. Glushkov of the National Academy of Sciences of Ukraine*)

Plankovskyy, S., Tsegelnyk, Y., Shypul, O. (*O.M.Beketov National University of Urban Economy in Kharkiv*, *National Aerospace University "Kharkiv Aviation Institute"*)

This research patricianly supported by Volkswagen Foundation (grant 97 775)



Thank you for your attention!