Article:
Romanova, T, Bennell, J, Stoyan, Y et al. (2018) Packing of concave polyhedra with continuous rotations using nonlinear optimisation. European Journal of Operational Research, 268 (1). pp. 37-53. ISSN: 0377-2217

https://doi.org/10.1016/j.ejor.2018.01.025

PACKING OF CONCAVE POLYHEDRA WITH CONTINUOUS ROTATIONS USING NONLINEAR OPTIMISATION

T. Romanova[a*], J. Bennell[b] ,Y. Stoyan[a], A. Pankratov[a]

[a]*Department of Mathematical Modeling and Optimal Design, Institute for Mechanical Engineering Problems of the National Academy of Sciences of Ukraine, Pozharsky Str., 2/10, Kharkov, 61046, Ukraine*

[b]*Southampton Business School, University of Southampton, Highfield, Southampton SO17 1BJ, UK*

Abstract. We study the problem of packing a given collection of arbitrary, in general concave, polyhedra into a cuboid of minimal volume. Continuous rotations and translations of polyhedra are allowed. In addition, minimal allowable distances between polyhedra are taken into account. We derive an exact mathematical model using adjusted radical free quasi phi-functions for concave polyhedra to describe non-overlapping and distance constraints. The model is a nonlinear programming formulation. We develop an efficient solution algorithm, which employs a fast starting point algorithm and a new compaction procedure. The procedure reduces our problem to a sequence of nonlinear programming subproblems of considerably smaller dimension and a smaller number of nonlinear inequalities. The benefit of this approach is borne out by the computational results, which include a comparison with previously published instances and new instances.

Keywords: *packing; concave polyhedra; continuous rotations; mathematical modeling; nonlinear optimisation*

1. Introduction

Cutting and packing problems have a long history of being tackled by the Operational Research community. Where the objects have arbitrary shape, this research has a strong link with the field of computational geometry (see, e.g., [24], [1], [9]). These problems have a wide spectrum of applications, for example in modern biology, mineralogy, medicine, materials science, nanotechnology, robotics, pattern recognition systems, control systems, space apparatus control systems, as well as in the chemical industry, power engineering, mechanical engineering, shipbuilding, aircraft construction and civil engineering.

At present, the interest in finding effective solutions for packing problems is growing rapidly. This is due to a large number of applications and the development of new and sophisticated methods that can exploit the ever increasing speed of computer processing.

In this paper, we consider the practical problem of packing a collection of non-identical, and in general, concave polyhedra into a cuboid of minimal sizes (in particular volume). We will refer to the problem as the polyhedron packing problem.

An interesting example of applications of the polyhedron packing arises in engineering design. Optimal packing of electronic components and payload has always been a pivotal concern in vehicle engineering, in particular in applications where volume is at a premium, for example embedding avionics in aircraft. The aim is to design an external envelope and determine the configuration of the payload subject to a fixed volume constraint. Alternatively, the approach may be to design an envelope around a fixed packing of the payload and the avionics in order to minimize volume while satisfying a set of mechanical, technical and maneuverability constraints.

Another application arises in the recent advent of additive manufacturing (AM), often referred to as 3D printing. There are a variety of different AM technologies that build up objects by adding one very thin layer of material at a time, for example through material extrusion or sintering layers of powder material. This procedure is very slow and not appropriate for repetitive manufacturing but useful for individual items and prototyping. Combining objects into one compact print pattern can reduce the print time, improving capacity utilization, and reduce the need for extra supporting material that is often required as part of the printing process when objects are arranged in certain configurations.

The polyhedron problems are NP-hard [2] and, as a result, solution methodologies generally employ heuristics, for example see [3], [8], [11], [12], [15], [20], [21]. Some researchers develop approaches based on mathematical modeling and general optimisation procedures; for example see [5], [6], [22].

Egeblad et al [5] present an efficient solution method for packing polyhedra within the bounds of a container (a polyhedron). The central geometric operation of the method is an exact horizontal or vertical translation of a given polyhedron to a position, which minimizes its volume of overlap with all other polyhedra. The translation algorithm is embedded into a local search heuristic. Additional details are given for the three-dimensional case and appropriate results are reported for the problem of packing polyhedra into a rectangular parallelepiped. Utilization of container space is improved by an average of more than 14 percentage points compared to previous methods proposed in [18]. In the experiments the largest total volume of overlap allowed in a solution corresponds to 0.01% of the total volume of all polyhedra for the given problem.

Liu et al [13] propose a new constructive algorithm, called HAPE3D, which is a heuristic algorithm based on the principle of minimum total “potential energy” for the 3D irregular packing problem, involving packing a set of irregularly shaped polyhedrons into a box-shaped container with fixed width and length but unconstrained height. The objective is to allocate all the polyhedrons in the container, and thus minimize the waste or maximize profit. HAPE3D can deal with arbitrarily shaped polyhedrons, which can be rotated around each coordinate axis at different angles. The most outstanding merit is that HAPE3D does not need to calculate no-fit polyhedrons. HAPE3D can also be hybridized with a meta-heuristic algorithm such as simulated annealing. Two groups of computational experiments demonstrate the good performance of HAPE3D and prove that it can be hybridized with a meta-heuristic algorithm that further improves the packing quality.

Our approach is based on the mathematical modeling of relations between geometric objects

and allowing the packing problem to be formulated as a nonlinear programming problem. To this end we use the phi-function technique (see, [4]) to provide an analytic description of objects placed in a container taking into account their *continuous rotations* and *translations*. At present phi-functions for the simplest 3D-objects, such as parallelepipeds, convex polyhedra and spheres are considered in [16]. Phi-functions for 3D-objects, in particular polyhedra, can be highly complicated analytically, since they involve many radicals and maximum operators, and are therefore difficult for NLP-solvers to solve.

In this paper we apply the *quasi phi-functions* concept introduced in [19], which is based on the idea proposed by [10] to use a separating plane to model non-overlapping constraints for circles and convex polygons. The concept of *quasi phi-functions* extends the domain of *phi-functions* by including auxiliary variables. The new functions can be described by analytical formulas that are substantially simpler than those used for phi-functions, for some types of objects, in particular, for convex polyhedra.

The use of quasi phi-functions, instead of phi-functions, allows us to describe (or simplify) the non-overlapping constraints. While this makes our models easier to solve, this comes at a price, which is performing the optimisation over a larger set of parameters, including the extra (auxiliary) variables used by the quasi phi-functions. Our approach is capable of finding a good local optimal solution in reasonable computational time.

The phi- and quasi phi-functions have been widely and successfully used to model a variety of packing problems, as in ([4], [14], [17]-[19]). In the current manuscript, we consider packing problem of *concave* polyhedra. The contributions of the work presented in this manuscript are as follows.

- We construct radical free *quasi phi-functions* to describe analytically the non-overlapping constraints for *concave* polyhedra and *adjusted quasi phi-functions* to describe analytically the minimal allowable distances between *concave* polyhedra.
- We derive an *exact mathematical model* of the optimal packing problem of *concave* polyhedra as a *continuous nonlinear programming problem*. Our feasible region is described by a system of inequalities with infinitely differentiable functions.
- We develop an efficient solution algorithm, which employs a clear and simple starting point algorithm and a new and original optimisation procedure (called COMPOLY) for the compaction of concave polyhedra. The COMPOLY procedure reduces our problem to a sequence of NLP subproblems of considerably smaller dimension and a smaller number of nonlinear inequalities. The procedure allows us to search for local optimal solutions of the packing problem.
- Our approach allows us to apply state of the art NLP solvers to the optimal packing problem of *concave* polyhedra.

The paper is organized as follows: in Section 2 we formulate the polyhedron packing problem. In Section 3 we give definitions of a phi-function and a quasi phi-function, an adjusted phi-function and an adjusted quasi phi-function and derive related functions for an analytical description of non-overlapping, containment and distance constraints in the problem. In Section 4 we provide an exact mathematical model in the form of a nonlinear programming problem by means of the phi-function technique. In Section 5 we describe a solution algorithm, which involves a fast starting point and

efficient local optimisation procedures. In Section 6 we present our computational results for some new instances and several instances studied before. Finally, Section 7 concludes this paper with a brief summary and a discussion about our future research directions.

2. Problem formulation

We consider here the packing problem in the following setting**.** Let Ω denote a cuboid, $\Omega = \{(x, y, z) \in R^3 : 0 \le x \le l, 0 \le y \le w, 0 \le z \le h\}$. It should be noted that each of the three dimensions (l or w or h) can be variable. Let $\{1, 2, \ldots, N\} = J_N$ and a set of polyhedra $\mathbb{Q}_q$, $q \in J_N$ be given.

Each polyhedron $\mathbb{Q}_q$ can be concave or convex. With each polyhedron $\mathbb{Q}_q$ we associate its local coordinate system with origin denoted by v_q.

Assume that each concave polyhedron $\mathbb{Q}_q$ is presented as a union of convex polyhedra K_j^q, j=1,…,n_q. With each convex polyhedron K_j^q we associate the local coordinate system of the polyhedron $\mathbb{Q}_q$. Each convex polyhedron K_j^q is defined by its vertices p_s^{qj}, $s = 1, \ldots, m_j^q$, in the local coordinate system of $\mathbb{Q}_q$.

We give here input data that form a concave polyhedron $\mathbb{Q}_q$ by two lists:

- List_1 contains the vertex coordinates of all the convex polyhedra K_j^q, j=1,…,n_q, and
- List_2 contains the index sets J_j^q, j=1,…,n_q, of the numbers of vertices (with respect to List_1) that define appropriate convex polyhedra K_j^q, j=1,…,n_q.

We note that List_1 involves all the original vertices of the concave polyhedron and, in general, additional vertices that appear as a result of decomposing the concave polyhedron into convex polyhedra. See Appendix A for details.

For the purposes of this paper, we assume that $\mathbb{Q}_q = \bigcup_{j=1}^{n_q} K_j^q$ is known.

Without loss of generality, we assume that the origin v_q of a polyhedron $\mathbb{Q}_q$ coincides with the center point of its circumscribed sphere S_q of radius r_q. In order to circumscribe a sphere around a polyhedron we employ the algorithm described in [7], which computes the smallest enclosing sphere of a collection of points. We use the library function found at (https://github.com/hbf/miniball), which is sufficiently fast.

The location and orientation of each polyhedron $\mathbb{Q}$ is defined by a vector $u = (v, \theta)$ of its variable placement parameters. Here $v = (x, y, z)$ is a translation vector, $\theta = (\theta^1, \theta^2, \theta^3)$ is a vector of rotation parameters, where θ^1, θ^2, θ^3 are Euler angles.

A polyhedron rotated through angles θ^1, θ^2, θ^3 and translated by vector v is denoted as $\mathbb{Q}(u) = \{p \in R^3 : p = v + M(\theta) \cdot p^0, p^0 \in \mathbb{Q}^0\}$, where $u = (v, \theta)$, $\mathbb{Q}^0$ denotes the non-translated and non-rotated polyhedron $\mathbb{Q}$, $M(\theta) = M(\theta^1, \theta^2, \theta^3)$ is a rotation matrix of the form:

$$M(\theta) = \begin{pmatrix} \cos\theta^1 \cos\theta^3 - \sin\theta^1 \cos\theta^2 \sin\theta^3 & -\cos\theta^1 \sin\theta^3 - \sin\theta^1 \cos\theta^2 \cos\theta^3 & \sin\theta^1 \sin\theta^2 \\ \sin\theta^1 \cos\theta^3 + \cos\theta^1 \cos\theta^2 \sin\theta^3 & -\sin\theta^1 \sin\theta^3 + \cos\theta^1 \cos\theta^2 \cos\theta^3 & -\cos\theta^1 \sin\theta^2 \\ \sin\theta^2 \sin\theta^3 & \sin\theta^2 \cos\theta^3 & \cos\theta^2 \end{pmatrix}.$$

It is possible to define minimal allowable distances between each pair of polyhedra $\mathbb{Q}_q$ and $\mathbb{Q}_g$, $q < g \in J_N$, as well as, between a polyhedron $\mathbb{Q}_q$, $q \in I_N$, and the boundary of container Ω. It means that each polyhedron $\mathbb{Q}_q$ has to be located no closer to polyhedron $\mathbb{Q}_g$ than the given allowable distance and each polyhedron $\mathbb{Q}_q$ has to be located inside the container and no closer to the boundary of the container than the given allowable distance.

We note that the minimal allowable distance between each pair of convex polyhedra $K_j^q \subset Q_q$, $j=1,\ldots,n_q$, and $K_l^g \subset Q_g$, $l=1,\ldots,n_g$, $q < g \in J_N$, is equal to the given allowable distance between the original polyhedra $\mathbb{Q}_q$ and $\mathbb{Q}_g$. Moreover, the minimal allowable distance between each polyhedron K_j^q, $q \in I_N$, and the boundary of the container Ω is equal to the given allowable distance between the original polyhedron $\mathbb{Q}_q$, $q \in I_N$, and the boundary of container Ω.

The polyhedron packing problem can be formulated in the form:

Pack the set of polyhedra $\mathbb{Q}_q$, $q \in J_N$, within a cuboid container Ω of minimal volume $F = l \cdot w \cdot h$, taking into account the given minimal allowable distances.

We note that it is possible that just one of the metrical characteristics of Ω can be variable.

In this definition, the term “pack” assumes polyhedra do not overlap and are fully enclosed in the containing cuboid.

3 Mathematical modeling of placement constraints

In this section we describe our methodology for modeling the non-overlapping, containment and minimal distance constraints. Here we introduce phi-functions and quasi phi-functions.

3.1 Placement constraints

Let us consider placement constraints that are met in the polyhedron packing problem:

- *non-overlapping constraints* – two polyhedra Q_q and Q_g do not have common interior points but may touch, i.e.

$$\operatorname{int}\mathrm{Q}_q \cap \operatorname{int}\mathrm{Q}_g = \varnothing \text{ for each } q, g \in J_N \text{ with } q \neq g;$$

- *containment constraints* – each polyhedron $\mathbb{Q}_q$ has to be fully enclosed in the container, i.e.

$$Q_q \subset \Omega \Leftrightarrow \operatorname{int} Q_q \cap \Omega^* = \varnothing \text{ for each } q \in J_N \text{ , } \Omega^* = R^3 \setminus \operatorname{int} \Omega.$$

Distance constraints

Let $\rho_{qg} > 0$ denote the minimal allowable distance between two polyhedra $\mathbb{Q}_q$ and $\mathbb{Q}_g$ and $\rho_q > 0$ denote the minimal allowable distance between a polyhedron $\mathbb{Q}_q$ and the object Ω^*.

- *distance constraints for "non-overlapping"* – each polyhedron $\mathbb{Q}_q$ has to be located no closer to polyhedron $\mathbb{Q}_g$ than the given allowable distance ρ_{qg} , i.e.

$$\operatorname{dist}(\mathbb{Q}_q, \mathbb{Q}_g) \geq \rho_{qg} \quad \text{for each } q, g \in J_N \text{ with } q \neq g, \text{ where}$$
$$\operatorname{dist}(\mathbb{Q}_q, \mathbb{Q}_g) = \min_{a \in \mathbb{Q}_q, b \in \mathbb{Q}_g} d(a,b) \text{ ;}$$

- *distance constraints for "containment"* – each polyhedron $\mathbb{Q}_q$ has to be located inside the container no closer to the boundary of the container than the given allowable distance ρ_q, i.e.

$$\operatorname{dist}(\mathbb{Q}_q, \Omega^*) \geq \rho_q \quad \text{for each } q \in J_N \text{ , } \Omega^* = R^3 \setminus \operatorname{int} \Omega \text{, where}$$
$$\operatorname{dist}(\mathbb{Q}_q, \Omega^*) = \min_{a \in \mathbb{Q}_q, b \in \Omega^*} d(a,b),$$

$d(a,b)$ represents the Euclidean distance between two points $a, b \in R^3$.

In order to feasibly place two objects within a container, we need an analytical description of the relationships between a pair of objects *A* and *B* considered in the placement constraints. We employ the phi-function technique for this [4], [19].

3.2 Phi-functions

Phi-functions allow us to distinguish the following three cases: *A* and *B* are intersecting so that *A* and *B* have common interior points; *A* and *B* do not intersect, i. e. *A* and *B* do not have common points; *A* and *B* are in contact, i. e. *A* and *B* have only common frontier points.

Let $A \subset R^3$ and $B \subset R^3$ be two objects. Sizes of objects can change according to homothetic coefficients (*scaling parameters of objects*) $\lambda_A, \lambda_B > 0$. The position of object A is defined by a vector of *placement parameters* (v_A, θ_A), where: $v_A = (x_A, y_A, z_A)$ is a translation vector and $\theta_A = (\theta_A^1, \theta_A^2, \theta_A^3)$ is a vector of rotation angles. We denote the vector of variables for the object A by $u_A = (v_A, \theta_A, \lambda_A)$ and the vector of variables for the object B by $u_B = (v_B, \theta_B, \lambda_B)$. The object *A*, rotated by angles $\theta_A^1, \theta_A^2, \theta_A^3$, translated by vector v_A, and rescaled by homothetic coefficient λ_A, will be denoted by $A(u_A)$.

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

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

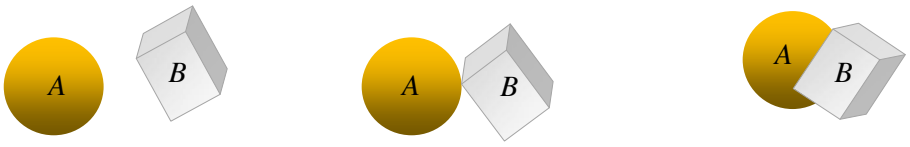
$$\Phi^{AB} = 0, \text{ if } \operatorname{int} A(u_A) \cap \operatorname{int} B(u_B) = \varnothing \text{ and } frA(u_A) \cap frB(u_B) \neq \varnothing \text{ ;}$$

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

provided that λ_A, λ_B are fixed.

Here *frA* means the boundary (frontier) and *intA* means the interior of object *A*.

Figure 1 illustrates three situations that a phi-function distinguishes.



(a) (b) (c)

Fig. 1 – Illustrations of definition 1: a) $\Phi^{AB} > 0$; b) $\Phi^{AB} = 0$; c) $\Phi^{AB} < 0$.

Thus, inequality $\Phi^{AB} \geq 0$ represents the *non-overlapping relationship* $\operatorname{int} A(u_A) \bigcap \operatorname{int} B(u_B) = \varnothing$, i.e. $\Phi^{AB} \geq 0 \Leftrightarrow \operatorname{int} A(u_A) \bigcap \operatorname{int} B(u_B) = \varnothing$.

We employ phi-functions for the description of the *containment relation* $A \subseteq B$ as follows: $\Phi^{AB^*} \geq 0$, where $B^* = R^3 \setminus \operatorname{int} B$.

We emphasize that according to Definition 1, the phi-function Φ^{AB} for a pair of objects *A* and *B* can be constructed by many different formulas [4], and we can choose the most convenient ones for our optimisation algorithms.

We can take into account *minimum allowable distance constraints* by replacing the phi-functions in the *non-overlapping* and *containment constraints* with adjusted phi-functions.

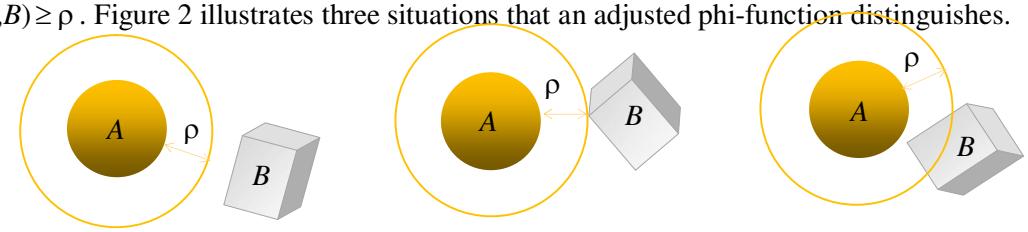
Let $\rho > 0$ be a given minimal allowable distance between objects $A(u_A)$ and $B(u_B)$.

Definition 2. A continuous and everywhere defined function $\tilde{\Phi}^{AB}(u_A, u_B)$ is called an adjusted phi-function for objects $A(u_A)$ and $B(u_B)$, if

$$\tilde{\Phi}^{AB} > 0, \text{ if } \operatorname{dist}(A, B) > \rho; \ \tilde{\Phi}^{AB} = 0, \text{ if } \operatorname{dist}(A, B) = \rho;$$

$$\tilde{\Phi}^{AB} < 0, \text{ if } \operatorname{dist}(A, B) < \rho.$$

We can describe the distance constraint for objects $A(u_A)$ and $B(u_B)$ in the form: $\tilde{\Phi}^{AB} \geq 0 \Leftrightarrow \operatorname{dist}(A,B) \geq \rho$. Figure 2 illustrates three situations that an adjusted phi-function distinguishes.



(a) (b) (c)

Fig. 2 – Illustrations of Definition 2: a) $\bar{\Phi}^{AB} > 0$; b) $\bar{\Phi}^{AB} = 0$; c) $\bar{\Phi}^{AB} < 0$.

The literature only contains the construction of phi-functions for concave polyhedra without rotation [17]. Constructing phi-functions for concave polyhedra with rotation is too complicated, therefore in this research we apply the concept of quasi phi-functions.

3.3 Quasi phi-functions

We introduce a function $\Phi'^{AB}(u_A, u_B, u')$ that must be defined for all values of u_A and u_B. In addition to the placement parameters of objects used with phi-functions, quasi phi-functions depend on auxiliary variables u'. These extra variables u' take values in some domain $U \subset R^{\eta}$. The number and the nature of variables u' depend on the shapes of objects $A(u_A)$ and $B(u_B)$, as well as on the restrictions of a packing problem. We define η for a quasi phi-function of a pair of polyhedra later.

Definition 3. A continuous and everywhere defined function $\Phi'^{AB}(u_A, u_B, u')$ is called *a quasi phi-function* for two objects $A(u_A)$ and $B(u_B)$ if $\max_{u' \in U} \Phi'^{AB}(u_A, u_B, u')$ is a phi-function for the objects.

The main property of a quasi phi-function is:

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

where $\Phi'^{AB}(u_A, u_B, u')$ is a quasi phi-function for two objects $A(u_A)$ and $B(u_B)$.

We note that the inverse proposition is not valid. It means that a quasi phi-function can take negative values while objects do not overlap, in contrast to a phi-function.

Let $\rho > 0$ be a given minimal allowable distance between objects $A(u_A)$ and $B(u_B)$.

Definition 4. Function $\bar{\Phi}'^{AB}(u_A, u_B, u')$ is called an adjusted quasi phi-function for objects $A(u_A)$ and $B(u_B)$, if function $\max_{u' \in U} \bar{\Phi}'^{AB}(u_A, u_B, u')$ is an adjusted phi-function for the objects.

We can define the distance constraint for objects $A(u_A)$ and $B(u_B)$ in the form: $\bar{\Phi}'^{AB} \geq 0$. The inequality implies $\text{dist}(A,B) \geq \rho$.

In order to describe the non-overlapping constraints in our polyhedron packing problem, we use quasi phi-functions, while for the containment constraints we use phi-functions. To formalise the distance constraints, we employ adjusted quasi phi-functions and adjusted phi-functions.

3.4 Construction of quasi phi-functions for non-overlapping and distance constraints

To construct a quasi phi-function and an adjusted quasi phi-function of two concave polyhedra we will use a quasi phi-function and an adjusted quasi phi-function for each pair of convex polyhedra that together form the original concave polyhedra.

First we consider a quasi phi-function for a pair of convex polyhedra.

Let $A(u_A)$ and $B(u_B)$ be two convex polyhedra given by their vertices p_s^A, $s=1,\ldots,m_A$, and p_s^B, $s=1,\ldots,m_B$.

A radical free quasi phi-function $\Phi'^{AB}(u_A,u_B,u'=u_P)$ for convex polyhedra $A(u_A)$ and $B(u_B)$ can be defined by the following formula:

$$\Phi'^{AB}(u_A,u_B,u'=u_P)=\min\{\Phi^{AP}(u_A,u_P),\Phi^{BP^*}(u_B,u_P)\}, \quad (1)$$

where $P(u_P)=\{(x,y,z):\psi_P=\alpha\cdot x+\beta\cdot y+\gamma\cdot z+\mu_P\le 0\}$ is a half-space,

$$\begin{pmatrix}\alpha\\ \beta\\ \gamma\end{pmatrix}=M(\theta_P^1,\theta_P^2,0)\cdot\begin{pmatrix}0\\ 0\\ 1\end{pmatrix}=\begin{pmatrix}\cos\theta_P^1 & -\sin\theta_P^1\cos\theta_P^2 & \sin\theta_P^1\sin\theta_P^2\\ \sin\theta_P^1 & \cos\theta_P^1\cos\theta_P^2 & -\cos\theta_P^1\sin\theta_P^2\\ 0 & \sin\theta_P^2 & \cos\theta_P^2\end{pmatrix}.\begin{pmatrix}0\\ 0\\ 1\end{pmatrix}=\begin{pmatrix}\sin\theta_P^1\sin\theta_P^2\\ -\cos\theta_P^1\sin\theta_P^2\\ \cos\theta_P^2\end{pmatrix},$$

θ_P^1 and θ_P^2 are appropriate (precession and nutation rotations) variable Euler angles (under intrinsic rotation $\theta_P^3=0$),

$u_P=(\theta_P^1,\theta_P^2,\mu_P)$ is a vector of variable parameters that define a plane $L_{AB}=\{(x,y,z):\psi_P=\alpha\cdot x+\beta\cdot y+\gamma\cdot z+\mu_P=0\}$ in three-dimensional Euclidean space (we assume $\alpha^2+\beta^2+\gamma^2=1$),

$\Phi^{AP}(u_A,u_P)$ is a phi-function of $A(u_A)$ and half plane $P(u_P)$,

$\Phi^{BP^*}(u_B,u_P)$ is a phi-function of $B(u_B)$ and half plane $P^*(u_P)$ (the complement to $P(u_P)$),

$$\Phi^{AP}(u_A,u_P)=\min_{1\le s\le m_A}\psi_P(p_s^A),\quad \Phi^{BP^*}(u_B,u_P)=\min_{1\le s\le m_B}(-\psi_P(p_s^B)).$$

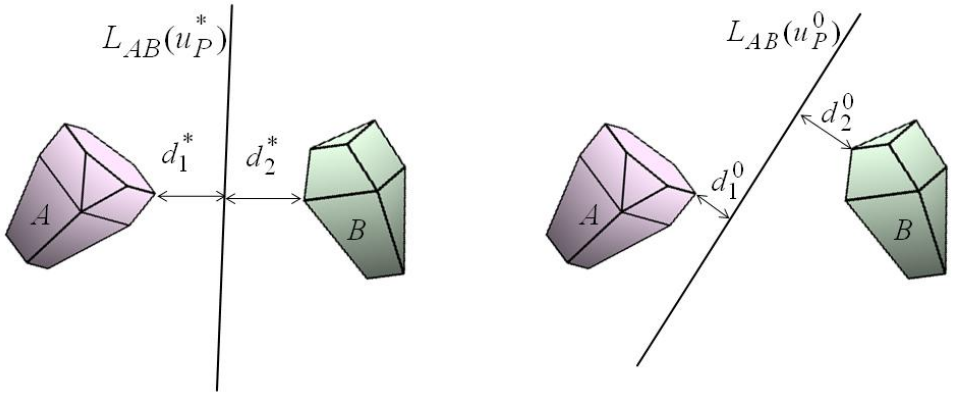
We note that $u_p\in U\equiv R^3$, $\eta=3$.

It is known that if two fixed convex objects A and B do not have common points then there exists at least one separating plane. Therefore there exists a vector u_P^* of parameters of a plane L_{AB} such that the distance $d_1=\Phi^{AP}(u_A,u_P^*)$ from A to L_{AB} equals to the distance $d_2=\Phi^{BP^*}(u_B,u_P^*)$ from B to L_{AB}. Thus function $\Phi'^{AB}(u_A,u_B,u_P)$ reaches its maximum when $(u_A,u_B,u_P^*)=(u_A,u_B,d^*,d^*)$, where $d^*=d_1=d_2$.

Figure 3 illustrates two cases when $\Phi'^{AB}>0$:

a) $\Phi'^{AB}(u_1,u_2,u_P^0)=\min\{d_1^0,d_2^0\}=d_1^0$;

b) $\max\limits_{u_P}\Phi'^{AB}(u_1,u_2,u_P)=\Phi'^{AB}(u_1,u_2,u_P^*)=\min\{d_1^*,d_2^*\}=d_1^*=d_2^*=d^*$.



(a) (b)

Fig.3 Separating planes for two fixed convex objects A and B : a) $\Phi'^{AB} = d^*$; b) $\Phi'^{AB} = d_1^0$.

Therefore always exists u_P such that $\max\limits_{u_P} \Phi'^{AB} > 0$ for two non-overlapping convex polyhedra and $\max\limits_{u_P} \Phi'^{AB} \geq 0 \Leftrightarrow \text{int } A(u_A) \bigcap \text{int } B(u_B) = \varnothing$.

We identify here the important characteristic of a *quasi phi-function*: if $\Phi'^{AB}(u_A, u_B, u_P) \geq 0$ for some u_P, then $\text{int } A(u_A) \bigcap \text{int } B(u_B) = \varnothing$ (see [19] for details).

Let the *minimal allowable distance* ρ_{AB} between two arbitrary convex polyhedra $A(u_A)$ and $B(u_B)$ be given. To describe a *distance* constraint, dist(A , B) $\geq \rho_{AB}$, we use an adjusted radical free quasi phi-function for convex polyhedra $A(u_A)$ and $B(u_B)$ derived by

$$\hat{\Phi}'^{AB}(u_A, u_B, u_P) = \Phi'^{AB}(u_A, u_B, u_P) - 0.5\rho_{AB}. \quad (2)$$

Since $\max\limits_{u_P} \hat{\Phi}'^{AB}(u_A, u_B, u_P) = \hat{\Phi}^{AB}(u_A, u_B)$ and $\hat{\Phi}^{AB}(u_A, u_B) \geq 0 \Leftrightarrow \text{dist}(A, B) \geq \rho_{AB}$, then $\max\limits_{u_P} \hat{\Phi}'^{AB}(u_A, u_B, u_P) \geq 0 \Leftrightarrow \text{dist}(A, B) \geq \rho_{AB}$. Based on the characteristic of a quasi phi-function, mentioned above, and formulas (1), (2), we can conclude that $\hat{\Phi}'^{AB}(u_A, u_B, u_P) \geq 0$ implies $\text{dist}(A, B) \geq \rho_{AB}$.

A quasi phi-function of two concave polyhedra is composed by quasi phi-functions for all pairs of convex polyhedra that together form the original concave polyhedra. By analogy an adjusted quasi

phi-function of two concave polyhedra is constructed.

Before we introduce a quasi phi-function and an adjusted quasi phi-function for a pair of concave polyhedra we present a given collection of convex polyhedra, K_j^q, j=1,…,n_q, $q \in J_N$, as a set of $n = \sum_{q=1}^{N} n_q$ convex polyhedra K_i, $i \in \{1, 2, \ldots, n\} = I_n$ using the following rule: $K_j^q \to K_i$, $i = \sum_{l=0}^{q-1} n_l + j$, j=1,…,n_q, $q \in J_N$, provided that $n_0 = 0$.

Now we introduce the "gluing" vector $\mathbf{a} = (a_1, , , , a_n)$, $a_i \in J_N$, where $a_i = q$, if K_i takes part in the composition of a polyhedron $\mathbb{Q}_q$, $q \in J_N$. Let $I_n = I^1 \bigcup I^2 \bigcup \ldots \bigcup I^N$ be an ordered partition of I_n, where $I^q = \{i \in I_n, a_i = q\}$, $\left|I^q\right| = n_q, q \in J_N$. For example, the "gluing" vector for polyhedra $\mathbb{Q}_1 = K_1 \bigcup K_2$, $\mathbb{Q}_2 = K_3$, $\mathbb{Q}_3 = K_4 \bigcup K_5 \bigcup K_6$ has the form $\mathbf{a} = (a_1, a_2, a_3, a_4, a_5, a_6) = (1, 1, 2, 3, 3, 3)$ (Fig.4). In the example N=3 and $n = \sum_{q=1}^{3} n_q = 2 + 1 + 3 = 6$.

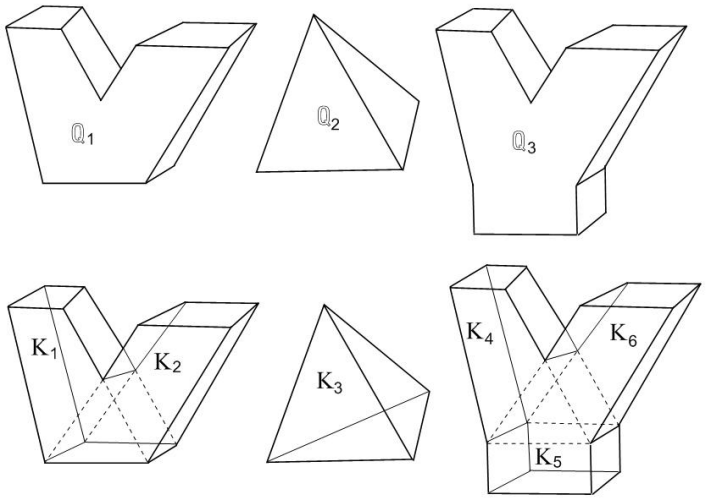


Fig.4 – Generation of the "gluing" vector for polyhedra $\mathbb{Q}_1$, $\mathbb{Q}_2$, $\mathbb{Q}_3$.

Let $\mathbb{Q}_q = \bigcup_{i \in I^q} K_i$ and $\mathbb{Q}_g = \bigcup_{j \in I^g} K_j$ be concave polyhedra and $q \neq g$.

We introduce the following function:

$$\widehat{\mathbf{\Phi}}'_{qg}(u_q, u_g, u_{qg}) = \min\{\widehat{\Phi}'_{ij}(u_q, u_g, u'_{ij}), i \in I^q, j \in I^g\}, \quad (3)$$

where $\widehat{\Phi}'_{ij}(u_q, u_g, u'_{ij})$ is the adjusted quasi phi-function and u'_{ij} is a vector of auxiliary variables for a pair of convex polyhedra $K_i(u_q)$ and $K_j(u_g)$, $i \in I^q$, $j \in I^g$, $u_{qg} = (u'_{ij}, i \in I^q, j \in I^g)$.

We note that $u_p \in U \equiv R^{\eta}$, $\eta = 3n_{qg}$, where $n_{qg} = n_q \cdot n_g$ is the number of all pairs of appropriate

convex polyhedra that form $\mathbb{Q}_q$ and $\mathbb{Q}_g$.

We show now that function (3) is an adjusted quasi phi-function $\hat{\boldsymbol{\Phi}}'_{qg}$ for concave polyhedra $\mathbb{Q}_q(u_q)$ and $\mathbb{Q}_g(u_g)$. In fact, we need to prove that $\max\limits_{u_{qg}} \hat{\boldsymbol{\Phi}}'_{qg}(u_q, u_g, u_{qg})$ is an adjusted phi-function for polyhedra $\mathbb{Q}_q(u_q)$ and $\mathbb{Q}_g(u_g)$.

Since each vector u'_{ij} of auxiliary variables is met in appropriate function $\hat{\Phi}'_{ij}(u_q, u_g, u'_{ij})$ only, then

$$\max_{u_{qg}} \hat{\boldsymbol{\Phi}}'_{qg}(u_q, u_g, u_{qg}) = \max_{u_{qg}} \min\{\hat{\Phi}'_{ij}(u_q, u_g, u'_{ij}), i \in I^q,\ j \in I^g\} =$$

$$\min\{\max_{u'_{ij}} \hat{\Phi}'_{ij}(u_q, u_g, u'_{ij}), i \in I^q,\ j \in I^g\} = \min\{\hat{\Phi}_{ij}(u_q, u_g), i \in I^q,\ j \in I^g\} = \hat{\boldsymbol{\Phi}}_{qg}(u_q, u_g),$$

where $\hat{\Phi}_{ij}(u_q, u_g)$ is the adjusted phi-function for convex polyhedra $K_i(u_q)$ and $K_j(u_g)$, $\hat{\boldsymbol{\Phi}}_{qg}(u_q, u_g)$ is an adjusted phi-function for concave polyhedra $\mathbb{Q}_q(u_q)$ and $\mathbb{Q}_g(u_g)$. It should be noted that function (3) is radical free.

From (3), a quasi phi-function for a pair of concave polyhedra, $\mathbb{Q}_q(u_q)$ and $\mathbb{Q}_g(u_g)$, can be defined in the form:

$$\boldsymbol{\Phi}'_{qg}(u_q, u_g, u_{qg}) = \min\{\Phi'_{ij}(u_q, u_g, u'_{ij}), i \in I^q,\ j \in I^g\},$$

where $\Phi'_{ij}(u_q, u_g, u'_{ij})$ is a quasi phi-function and u'_{ij} is a vector of auxiliary variables for convex polyhedra $K_i(u_q)$ and $K_j(u_g)$, $i \in I^q$, $j \in I^g$, $u_{qg} = (u'_{ij}, i \in I^q, j \in I^g)$.

Let us consider an example of a quasi phi-function for two polyhedra: $\mathrm{Q}_1(u_1) = K_1(u_1)$ and $\mathrm{Q}_2(u_2) = K_2(u_2) \cup K_3(u_2)$ (Fig. 5a).

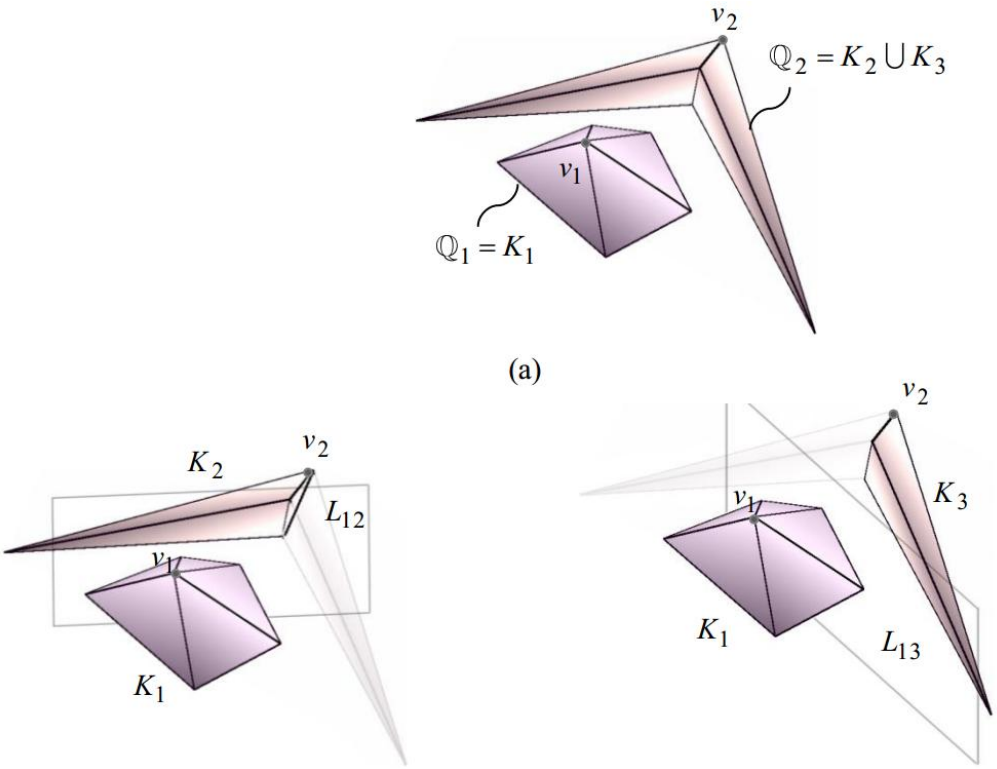


Fig.5 – a) polyhedra Q_1 and Q_2; b) separating planes L_{12} and L_{13} for two pairs of appropriate convex polyhedra K_1 and K_2; K_1 and K_3

A quasi phi-function for $\mathrm{Q}_1(u_1)$ and $\mathrm{Q}_2(u_2)$ can be defined in the following form:

$$\mathbf{\Phi}'_{12}(u_1, u_2, u_{12}) = \min\{\Phi'_{12}(u_1, u_2, u'_{12}), \Phi'_{13}(u_1, u_2, u'_{13})\},$$

where $u_{12} = (u'_{12}, u'_{13})$, $\Phi'_{12}(u_1, u_2, u'_{12})$ is a quasi phi-function and u'_{12} is a vector of auxiliary variables for a pair of convex polyhedra $K_1(u_1)$ and $K_2(u_2)$, $\Phi'_{13}(u_1, u_2, u'_{13})$ is a quasi phi-function and u'_{13} is a vector of auxiliary variables for a pair of convex polyhedra $K_1(u_1)$ and $K_3(u_2)$.

Figure 5b illustrates two separating planes L_{12} and L_{13} that provide $\Phi'_{12}(u_1, u_2, u'_{12}) > 0$ and $\Phi'_{13}(u_1, u_2, u'_{13}) > 0$ that implies $\mathbf{\Phi}'_{12}(u_1, u_2, u_{12}) > 0$. Here $L_{ij} = \{(x, y, z) : \psi_{ij} = 0\}$ is a separating plane for $K_i(u_q)$ and $K_j(u_g)$, where $\psi_{ij} = \alpha_{ij} \cdot x + \beta_{ij} \cdot y + \gamma_{ij} \cdot z + \mu_{ij}$, $\alpha_{ij} = \sin\theta^1_{ij} \sin\theta^2_{ij}$, $\beta_{ij} = -\cos\theta^1_{ij} \sin\theta^2_{ij}$, $\gamma_{ij} = \cos\theta^2_{ij}$ and $u'_{ij} = (\theta^1_{ij}, \theta^2_{ij}, \mu_{ij})$, $i = 1, j = 2, 3$, $q = 1, g = 2$.

3.5 Construction of phi-functions for containment - distance constraints

An adjusted phi-function for a concave polyhedron $\mathbb{Q}_q(u_q)$ and the object Ω^* can be defined in the form [4]

$$\bar{\mathbf{\Phi}}_q(u_q) = \min\{\bar{\Phi}_i(u_q), i \in I^q\}, \quad (4)$$

where $\bar{\Phi}_i(u_q)$ is an adjusted phi-function for a convex polyhedron $K_i(u_g)$ and Ω^*, $i \in I^q$. Replacing each adjusted phi-function $\bar{\Phi}_i(u_q)$ in (4) by a phi-function $\Phi_i(u_q)$ for $i \in I^q$, we can get a phi-function $\mathbf{\Phi}_q(u_q)$ for a polyhedron $\mathbb{Q}_q(u_q)$ and the object Ω^*.

To describe a containment constraint, $K_i(u_q) \subset \Omega \Leftrightarrow \operatorname{int} K_i(u_q) \bigcap \Omega^* = \varnothing$, we use a phi-function for a convex polyhedron $K_i(u_q)$ and the object Ω^* [4].

Let $K_i(u_q)$ be convex polyhedron, given in its local coordinate system by their vertices p_k^i, $k = 1, \ldots, m_i$, where $p_k^i = (p_{xk}^i, p_{yk}^i, p_{zk}^i)$. A radical free phi-function for a convex polyhedron $K_i(u_q)$ and the object Ω^* can be defined as

$$\Phi_i(u_q) = \min\{\min_{1 \le k \le m_i} \varphi_{kj}^i(u_q),\ j = 1, \ldots, 6\}, \tag{5}$$

$$\varphi_{k1}^i(u_q) = x_q + p_{xk}^i,\ \varphi_{k2}^i(u_q) = -(x_q + p_{xk}^i) + l,\ \varphi_{k3}^i(u_q) = y_q + p_{yk}^i,$$

$$\varphi_{k4}^i(u_q) = -(y_q + p_{yk}^i) + w,\ \varphi_{k5}^i(u_q) = z_q + p_{zk}^i,\ \varphi_{k6}^i(u_q) = -(z_q + p_{zk}^i) + h.$$

Let minimal allowable distance $\rho_q > 0$ between a convex polyhedron $K_i(u_q)$ and the object Ω^* be given. To describe distance constraint, dist(K_i, Ω^*) $\ge \rho_q$, we use an adjusted phi-function for a convex polyhedron $K_i(u_q)$ and the object Ω^* defined by

$$\bar{\Phi}_i(u_q) = \Phi_i(u_q) - \rho_q. \tag{6}$$

4. Mathematical model

The vector $u \in R^\sigma$ of all variables can be described as follows: $u = (\varsigma, \tau) \in R^\sigma$, where $\varsigma = (l, w, h, u_1, u_2, \ldots, u_N)$, (l, w, h) denote the variable dimensions (length, width and height) of the cuboid Ω and $u_{a_i} = (v_{a_i}, \theta_{a_i}) = (x_{a_i}, y_{a_i}, z_{a_i}, \theta_{a_i}^1, \theta_{a_i}^2, \theta_{a_i}^3)$ is the vector of placement parameters of K_i, $i \in I_n$, an index $a_i \in \{1, 2, \ldots, N\}$ is a component of the "gluing" vector $\mathbf{a}$, defined in Section 3. Here $\tau = (u_P^1, \ldots, u_P^m)$ denotes the vector of all auxiliary variables, where $u_P^s = (\theta_P^{1s}, \theta_P^{2s}, \mu_P^s)$ is a vector of auxiliary variables for the *s*-th pair of convex polyhedra defined in (1), $s = 1, \ldots, m$, $m = card(\Xi)$,

$$\Xi = \{(i, j), a_i \ne a_j, i < j = 1, \ldots, n\}. \tag{7}$$

The number of the problem variables is derived as $\sigma = 3 + 6N + 3m$.

Now a mathematical model of the polyhedron packing problem can be stated in the form

$$\min_{u \in W \subset R^{\sigma}} F(u), \tag{8}$$

$$W = \{u \in R^{\sigma} : \hat{\Phi}'_{ij}(u_{a_i}, u_{a_j}, u'_{a_i a_j}) \geq 0, (i, j) \in \Xi, \hat{\Phi}_i(u_{a_i}) \geq 0, i = 1, 2, \ldots, n\}, \tag{9}$$

where $F(u) = l \cdot w \cdot h$, $\hat{\Phi}'_{ij}(u_{a_i}, u_{a_j}, u'_{a_i a_j})$ is an adjusted quasi phi-function defined by (2), $a_i, a_j \in I_N$, under $(i, j) \in \Xi$, $u'_{a_i a_j} = u^s_P$, $s = 1, \ldots, m$, Ξ is given by (7), for the pair of polyhedra K_i and K_j, taking into account minimal allowable distance $\rho_{qg} > 0$, $\hat{\Phi}_i(u_{a_i})$ is an adjusted phi-function defined by (6) for a polyhedron K_i and the object Ω^*, taking into account minimal allowable distance $\rho_q > 0$. If $\rho_{qg} = 0$ and $\rho_q = 0$ then we replace the adjusted quasi phi-function $\hat{\Phi}'_{ij}(u_{a_i}, u_{a_j}, u'_{a_i a_j})$ by the quasi phi-function $\Phi'_{ij}(u_{a_i}, u_{a_j}, u'_{a_i a_j})$, defined by (1), to enforce the non-overlapping constraint and the adjusted phi-function $\hat{\Phi}_i(u_{a_i})$ by the phi-function $\Phi_i(u_{a_i})$, defined in (5), to enforce the *containment* constraint.

It should be noted that in order to avoid redundant inequalities in containment constraints one can use a collection of adjusted phi-functions $\hat{\boldsymbol{\Phi}}^h_q(u_q) \geq 0, q = 1, \ldots, N$, for the convex hull of concave polyhedra $\mathbb{Q}_q$, $q = 1, \ldots, N$, instead of the collection of adjusted phi-functions $\hat{\Phi}_i(u_{a_i}) \geq 0$, $i = 1, \ldots, n$, for convex polyhedra K_i, $i = 1, \ldots, n$.

Let us consider a mathematical model for a simple example of a packing problem for *N*=2 polyhedra: $Q_1(u_1) = K_1(u_1)$ and $Q_2(u_2) = K_2(u_2) \cup K_3(u_2)$ (Fig. 5a) in a cuboid $\Omega = \{(x, y, z) \in R^3 : 0 \leq x \leq l, 0 \leq y \leq w, 0 \leq z \leq h\}$. Here *n*=3 is the number of convex polyhedra, $\mathbf{a} = (a_1, a_2, a_3) = (1, 2, 2)$ is the gluing vector, $\Xi = \{(i, j), a_i \neq a_j, i < j = 1, 2, 3\} = \{(1, 2), (1, 3)\}$, $u_{a_i} = (v_{a_i}, \theta_{a_i}) = (x_{a_i}, y_{a_i}, z_{a_i}, \theta^1_{a_i}, \theta^2_{a_i}, \theta^3_{a_i})$ is the vector of placement parameters of K_i, $i = 1, 2, 3$. according to the gluing vector, *m*=2 is the number of pairs of convex polyhedra with respect to Ξ, $\tau = (u^1_P, u^2_P) = (u'_{12}, u'_{13})$ is the vector of auxiliary variables, $|\tau| = 3m = 6$, $u = (l, w, h, u_1, u_2, u'_{12}, u'_{13})$ is the vector of the problem variables. The number of the problem variables is $\sigma = 3 + 6N + 3m = 21$. Now mathematical model (8)-(9) for the packing problem takes the form

$$\min_{u \in W \subset R^{21}} F(u),$$

$$W = \{u \in R^{21} : \Phi'_{12}(u_1, u_2, u'_{12}) \geq 0, \Phi'_{13}(u_1, u_2, u'_{13}) \geq 0, \Phi_1(u_1) \geq 0, \Phi_2(u_2) \geq 0, \Phi_3(u_2) \geq 0\},$$

where

$\Phi'_{12}(u_1, u_2, u'_{12})$ is a quasi phi-function for $K_1(u_1)$ and $K_2(u_2)$,

$\Phi'_{13}(u_1, u_2, u'_{13})$ is a quasi phi-function for $K_1(u_1)$ and $K_3(u_2)$,

$\Phi_1(u_1)$ is a phi-function for $K_1(u_1)$ and the object Ω^*,

$\Phi_2(u_2)$ is a phi-function for $K_2(u_2)$ and the object Ω^*,

$\Phi_3(u_2)$ is a phi-function for $K_3(u_2)$ and the object Ω^*.

We note, that in the model we can use two phi-functions: phi-function $\Phi_1(u_1)$ and a phi-function for the convex hull of concave polyhedra $\mathbb{Q}_2$ and the object Ω^* instead of phi-functions $\widehat{\Phi}_i(u_{a_i}) \geq 0$, $i = 1, 2, 3,$ for convex polyhedra K_i, $i = 1, 2, 3$ and the object Ω^*.

Each quasi phi-function inequality in (9) is presented by a system of inequalities with infinitely differentiable functions. Our model (8)-(9) is a non-convex and continuous nonlinear programming problem and an exact formulation for the polyhedron packing problem. It contains all globally optimal solutions. It is possible, at least in theory, to use a global solver for the nonlinear programming problem and to obtain a solution, which is an optimal packing.

However in practice, the model contains a large number of variables and a huge number of inequalities. Specifically, the model (8)-(9) involves O(n^2) nonlinear inequalities and O(n^2) variables due to the auxiliary variables in quasi phi-functions, where n is the number of convex polyhedra. As a result, finding a locally optimal solution becomes an unrealistic task for the available state of the art NLP-solvers employed **directly** to model (8)-(9): for N >15 starting from a random point and for N >30 starting from a feasible point.

In order to search for a "good" locally optimal polyhedron packing within a reasonable computational time we propose here an efficient solution algorithm, which employs a fast starting point algorithm (FAPA) and a new compaction procedure. In most cases the procedure reduces our problem to a sequence of nonlinear programming subproblems of considerably smaller dimension (O(n)) and a smaller number of nonlinear inequalities (O(n)). We use NLP-solver (IPOPT) to solve each of the NLP subproblems starting from the feasible points found by the special procedures described in Section 5.

5. Solution algorithm

Our multi-start solution strategy involves the following steps:

1) Generate a set $\{\varsigma^0\}_\chi$ of vectors $\varsigma^0 = (l^0, w^0, h^0, u_1^0, u_2^0, \ldots, u_N^0)$ of feasible placement parameters $(u_1^0, u_2^0, \ldots, u_N^0)$ of polyhedra placed into the container Ω^0 of sizes (l^0, w^0, h^0) in the problem (8)-(9). Various algorithms exist for obtaining a feasible solution (for example [17]). We employ here the clear and fast algorithm, which is described in Subsection 5.1.
2) Search for a local minimum of the objective function $F(u)$ in problem (8)-(9), starting from each point from the set $\{\varsigma^0\}_\chi$ obtained at Step 1. To get a local minimum of problem (8)-(9) we develop a compaction algorithm for rotated polyhedra described in Subsection 5.2.

3) Choose the best local minimum from those found at Step 2 as the final solution of the problem (8)-(9).

The actual search for a local minimum in all optimization procedures (to realize steps 1-2) is performed by IPOPT [23], which is available at an open access noncommercial software depository (https://projects.coin-or.org/Ipopt) .

5.1 Feasible Placement Parameters Algorithm (FPPA)

In order to find a vector of starting feasible placement parameters of polyhedra we apply an algorithm, which is based on the homothetic (scaling) transformation of objects. The algorithm consists of the following steps.

Firstly we choose a sufficiently large starting length l^0, width w^0 and height h^0 for a container Ω^0 to allow for a placement of all spheres S_q^ρ, $q=1,2,\dots,N$, within the container Ω^0, where $S_q^\rho = S_q \oplus S^\rho$ is the Minkovski sum of a sphere S_q of radius r_q (Fig. 6) and a sphere S^ρ of radius $\rho = 0.5\max\{\max\limits_{q,g\in J_N} \rho_{qg}, \max\limits_{q\in J_N} \rho_q\}$, provided that S_q and S^ρ have the same center point. For example, we can set $l^0 = w^0 = h^0 = 2\sum\limits_{q=1}^{n} r_q + (n+1)\rho.$

Secondly we generate within the container Ω^0 a set of N randomly chosen center points (x_q^0, y_q^0, z_q^0) of S_q^ρ, $q=1,2,\dots,N$.

Fig. 6 – Concave polyhedra $\mathbb{Q}_q$ and appropriate spheres S_q.

Thirdly we grow the spheres S_q^ρ of radius $\lambda(r_q+\rho)$, $q=1,2,\dots,N,$ starting from $\lambda=0$ to the full size ($\lambda=1$) and the decision variables are: the centres of S_q^ρ and a homothetic coefficient (a scaling parameter) λ, where $0\le\lambda\le 1$ (Fig 7.). In order to realise this step we fix $l=l^0$, $w=w^0$, $h=h^0$, and,

starting from the point $v^0=(x_1^0, y_1^0, z_1^0,\dots, x_N^0, y_N^0, z_N^0, \lambda^0=0)$, solve the following NLP-subproblem:

$$\max_{v\in W_\lambda} \lambda, \tag{10}$$

$$W_\lambda=\{v\in \mathrm{R}^{3N+1}: \widehat{\Phi}^{S_qS_g}(v)\ge 0, \widehat{\Phi}^{S_q\Omega^*}(v)\ge 0, q<g=1,2,\dots,N, 1-\lambda\ge 0, \lambda\ge 0\}, \tag{11}$$

where $v=(x_1, y_1, z_1,\dots, x_N, y_N, z_N, \lambda)$,

$$\widehat{\Phi}^{S_qS_g}(v)=(x_q-x_g)^2+(y_q-y_g)^2+(z_q-z_g)^2-\lambda^2(r_q+2\rho+r_g)^2, \tag{12}$$

is an adjusted phi-function for a sphere S_q of radius λr_q and a sphere S_g of radius λr_g;

$$\widehat{\Phi}^{S_q\Omega^*}(v)=\min\{\varphi_{kq}(v), k=1,\dots,6\}, \tag{13}$$

is an adjusted phi-function for a sphere S_q of radius λr_q and the object Ω^*, where

$$\varphi_{1q}(v)=-x_q+l^0-\lambda(r_q+\rho_q),\ \varphi_{2q}(v)=x_q-\lambda(r_q+2\rho),$$

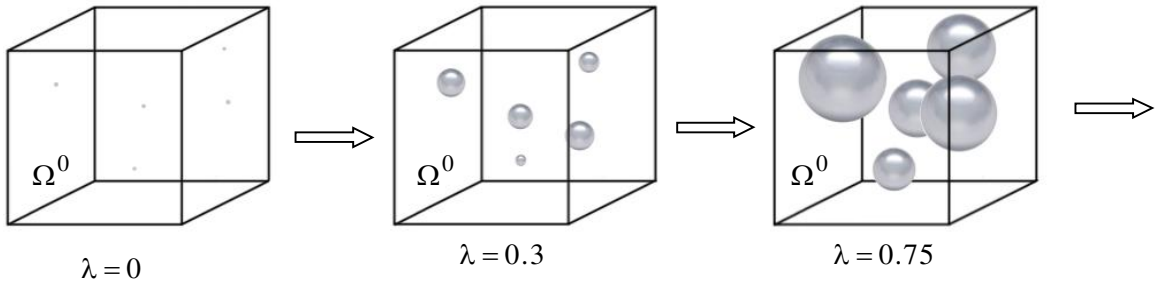
$$\varphi_{3q}(v)=-y_q+w^0-\lambda(r_q+\rho_q),\ \varphi_{4q}(v)=y_q-\lambda(r_q+2\rho),$$

$$\varphi_{5q}(v)=-z_q+h^0-\lambda(r_q+\rho_q),\ \varphi_{4q}(v)=z_q-\lambda(r_q+2\rho).$$

We denote a point of the global maximum of problem (10)-(11) by $v^*=(x_1^*, y_1^*, z_1^*,\dots, x_N^*, y_N^*, z_N^*, \lambda^*=1)$.

Finally we form a vector of feasible parameters $\varsigma^0=(l^0, w^0, h^0, u_1^0,\dots, u_q^0,\dots, u_N^0)$, assuming that $u_q^0=(x_q^0, y_q^0, z_q^0, \theta_q^0)$, $(x_q^0, y_q^0, z_q^0)=(x_q^*, y_q^*, z_q^*)$ and θ_q^0 is a vector of randomly generated rotation parameters of polyhedra $\mathbb{Q}_q$, $q=1,\dots,N$.

We note that the global solution of problem (10)-(11) always can be found (since the chosen starting sizes l^0, w^0 and h^0 at the first step are sufficiently large). The solution automatically respects all the non-overlapping, containment and distance constraints for the concave polyhedra.



$\lambda=0.97$ $\lambda=1$

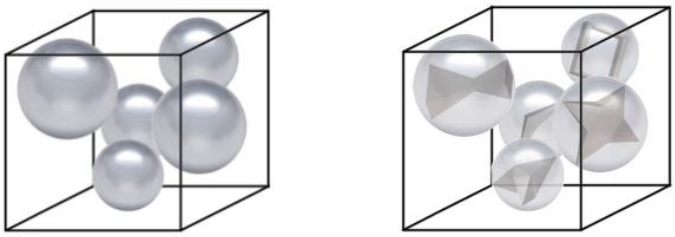


Fig. 7 – Illustration of the optimisation procedure FPPA to search for feasible placement parameters of polyhedra, using homothetic transformations

Our FPPA algorithm returns the vector ς^0 to generate a starting point $u^0 = (\varsigma^0, \tau^0)$ for a subsequent search for a local minimum of the problem (8)-(9). To search for vector τ^0 we apply special optimisation procedure, called Feasible Auxiliary Parameters Algorithm (FAPA), described below.

5.2 Compaction Algorithm (COMPOLY)

Since our problem (8)-(9) can not be solved for N >30 by direct use of state of the art NLP-solvers (starting from a feasible point), we propose an iterative compaction algorithm to search for local minima of the problem.

Our algorithm reduces the problem (8)-(9) that has a large number of inequalities and dimension O(n^2) of the feasible set W, described by (9), to a sequence of nonlinear programming subproblems that have a smaller number of nonlinear inequalities (O(n)) and dimension O(n). The key idea of the algorithm is as follows: For each vector of feasible placement parameters of our polyhedral, we construct fixed individual cubic containers of spheres that circumscribe the appropriate convex polyhedra. Then we move each sphere within the appropriate individual container. The motion of each sphere we describe by a system of six linear ε -inequalities. Then we form a subregion of feasible region W in the following way: we **add** O(n) ε -inequalities (for all spheres) to the inequality system (9), that allows us to **delete** O(n^2) phi-inequalities for such pairs of polyhedra whose individual containers do not overlap each other and **delete** some redundant containment constraints. Then we search for a local minimum on the subregion of dimension O(n). The subregion is described by O(n) nonlinear inequalities. Then we use this local minimum as a starting point for the next iteration. On the last iteration of our algorithm we find a local minimum of problem (8)-(9).

Let us consider the algorithm in details.

We assume here that spheres $S_q^0 \equiv S_q(0)$ of radius r_q and the center point $v_q = (x_q, y_q, z_q)$, circumscribed around each non-translated and non-rotated concave polyhedron $\mathbb{Q}_q$, $q = 1, \ldots, N$, as well as, spheres $S_i^0 \equiv S_i(0)$ of radius r_i and the center point $v_{ci} = (x_{ci}, y_{ci}, z_{ci})$ circumscribed around each non-translated and non-rotated convex polyhedron K_i^0 , $i = 1, \ldots, n$, are constructed.

The COMPOLY algorithm is an iterative procedure and involves the following steps.

Step 1. Let $k=1$. Take the vector $\varsigma^{k-1}=(l^{k-1},w^{k-1},h^{k-1},u_1^{k-1},...,u_N^{k-1})$ of feasible placement parameters of polyhedra $\mathbb{Q}_q$, $q=1,...,N$, within the container Ω^{k-1}.

Step 2. Derive the appropriate vector $(v_{c1}^{(k-1)},..,v_{cn}^{(k-1)})$ of center points of spheres $S_i(u_{a_i}^{(k-1)})$, $i=1,2,...,n$. With respect to the gluing vector **a,** the center point v_{ci} of $S_i \supset K_i$ after translation and rotation of initial convex polygon K_i^0 takes the form

$$v_{ci}^{(k-1)}=v_{ci}(u_{a_i}^{(k-1)})=v_{a_i}^{(k-1)}+M(\theta_{a_i}^{(k-1)})\cdot v_{ci}.$$

For the sake of simplicity, we provide some illustrations to the algorithm for the 2D case.

Figure 8 illustrates the concave polygon $\mathbb{Q}(0)=K_1(0)\cup K_2(0)$ with translation vector $(0,0)$ and rotation angle $\theta=0$. Circles $S_1(0)$ and $S_2(0)$, circumscribed around $K_1(0)$ and $K_2(0)$ have center points v_{c1} and v_{c2}. Polygon $\mathbb{Q}(0)$, translated by vector v^0 and rotated by angle θ^0, is denoted by $\mathbb{Q}(u^0)=K_1(u^0)\cup K_2(u^0)$, where $u^0=(v^0,\theta^0)$. Center points of circles $S_1(u^0)$ and $S_2(u^0)$ are denoted by v_{c1}^0 and v_{c2}^0.

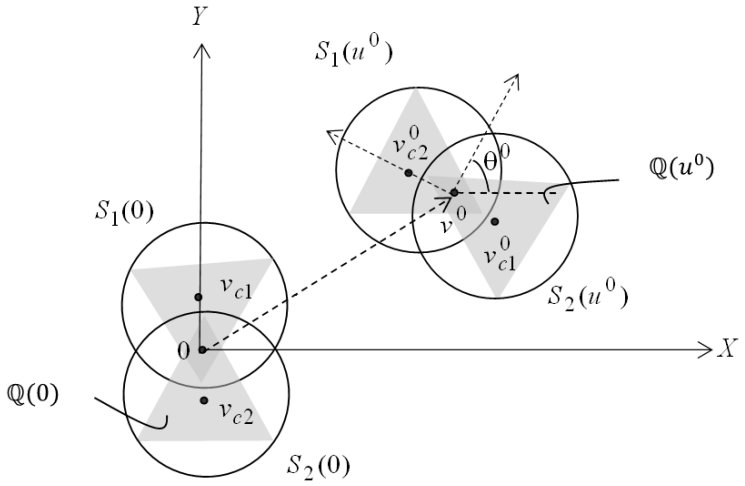


Fig. 8 – Translation and rotation parameters of $S_1(u^0)$ and $S_2(u^0)$ of concave polygon $\mathbb{Q}(0)=K_1(0)\cup K_2(0)$, translated by vector v^0 and rotated by angle θ^0.

Step 3. For each sphere $S_i(u_{a_i}^{(k-1)})$ we construct a fixed individual container $\Omega_i^k \supset S_i \supset K_i$ with equal half-sides of length $r_i+\varepsilon$, $i=1,...,n$, and the center of symmetry point $v_{ci}^{(k-1)}$, assuming $\varepsilon=\sum_{i=1}^{n} r_i/n$. Figure 9 illustrates individual containers $\Omega_1(u_{a_1}^0)$ and $\Omega_2(u_{a_2}^0)$ for circles $S_1(u_{a_1}^0)$ and $S_2(u_{a_2}^0)$ considered in the above example. Note, that here $a_1=a_2$.

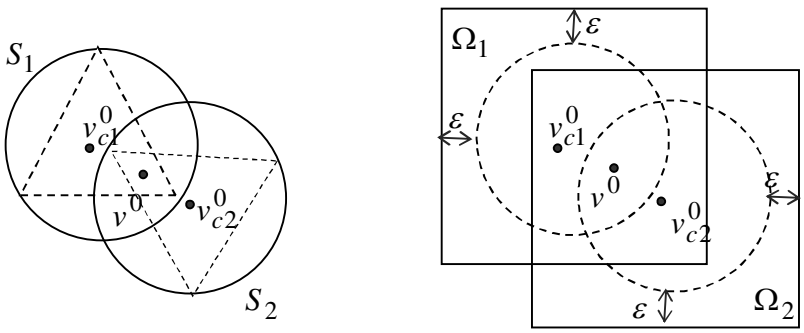


Fig. 9 – Individual containers $\Omega_1(u_{a_1}^0)$ and $\Omega_2(u_{a_2}^0)$ for circles $S_1(u_{a_1}^0)$ and $S_2(u_{a_2}^0)$.

Step 4. Move each sphere S_i, associated with the convex polyhedron K_i, within the appropriate fixed individual container Ω_i^k (found at Step 3). Hence, for each sphere S_i we construct a phi-function $\Phi^{S_i\Omega_i^*}$ for sphere S_i and $\Omega_i^* = R^3 \setminus \operatorname{int}\Omega_i$ in the form:

$$\Phi^{S_i\Omega_i^*}(v_{a_i}, v_{a_i}^{(k-1)}) = \min\{-x_{ci}(u_{a_i}^{(k-1)}) + x_{a_i} + \varepsilon, -y_{ci}(u_{a_i}^{(k-1)}) + y_{a_i} + \varepsilon, -z_{ci}(u_{a_i}^{(k-1)}) + z_{a_i} + \varepsilon,$$

$$x_{ci}(u_{a_i}^{(k-1)}) - x_{a_i} + \varepsilon,\ y_{ci}(u_{a_i}^{(k-1)}) - y_{a_i} + \varepsilon,\ z_{ci}(u_{a_i}^{(k-1)}) - z_{a_i} + \varepsilon\}.$$

The inequality $\Phi^{S_i\Omega_i^*}(v_{a_i}, v_{a_i}^{(k-1)}) \ge 0$ provides $S_i \subset \Omega_i^k$ and can be described by the following inequality system of six linear "ε-constraints":

$$-x_{ci}(u_{a_i}^{(k-1)}) + x_{a_i} + \varepsilon \ge 0, -y_{ci}(u_{a_i}^{(k-1)}) + y_{a_i} + \varepsilon \ge 0,\ -z_{ci}(u_{a_i}^{(k-1)}) + z_{a_i} + \varepsilon \ge 0,$$

$$x_{ci}(u_{a_i}^{(k-1)}) - x_{a_i} + \varepsilon \ge 0,\ \ y_{ci}(u_{a_i}^{(k-1)}) - y_{a_i} + \varepsilon \ge 0,\ z_{ci}(u_{a_i}^{(k-1)}) - z_{a_i} + \varepsilon \ge 0.$$

Now we introduce an auxiliary (artificial) subset Λ_k^ε of additional "ε-constraints" on the translation vectors $v_{a_i} = (x_{a_i}, y_{a_i}, z_{a_i})$, $i = 1, \ldots, n$, of convex polyhedra K_i, $i = 1, \ldots, n$:

$$\Lambda_k^\varepsilon = \{u \in R^\sigma : -x_{ci}(u_{a_i}^{(k-1)}) + x_{a_i} + \varepsilon \ge 0, -y_{ci}(u_{a_i}^{(k-1)}) + y_{a_i} + \varepsilon \ge 0,$$

$$-z_{ci}(u_{a_i}^{(k-1)}) + z_{a_i} + \varepsilon \ge 0,\ \ x_{ci}(u_{a_i}^{(k-1)}) - x_{a_i} + \varepsilon \ge 0,$$

$$y_{ci}(u_{a_i}^{(k-1)}) - y_{a_i} + \varepsilon \ge 0, z_{ci}(u_{a_i}^{(k-1)}) - z_{a_i} + \varepsilon \ge 0,\ i = 1, \ldots, n\}.$$

Then we add the inequality system of $6n$ additional linear "ε-constraints" that describe the subset Λ_k^ε to the inequality system that defines the feasible region W and obtain k-th subregion $W_k = W \bigcap \Lambda_k^\varepsilon$.

It should be noted that the inequality system that describes the feasible subregion W_k in most cases involves $O(n^2)$ redundant phi-inequalities.

Step 5. To avoid the redundant phi-inequalities that describe W_k we form special index sets Ξ_1^k and Ξ_2^k that involve indexes of all pairs of objects that are associated with non-redundant non-overlapping and containment constraints respectively.

To form index set Ξ_1^k we exclude from Ξ (7) indexes of all pairs of convex polyhedra where individual containers do not intersect each other (see Appendix B):

$$\Xi_1^k = \{(i,j) \in \Xi_1^{kS} : \varphi^{\Omega_i^k \Omega_j^k}(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) < 0\}, \text{ where } \Xi_1^{kS} = \{(i,j) \in \Xi : \hat{\Phi}^{S_{a_i} S_{a_j}}(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) < 0\}$$

$$\varphi^{\Omega_i^k \Omega_j^k}(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) = \max\{\varphi_{ij}^s(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}), s = 1,\dots,6\},$$

$$\varphi_{ij}^1(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) = (x_i^{(k-1)} - x_j^{(k-1)}) - R_{ij},\ \varphi_{ij}^2(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) = (y_i^{(k-1)} - y_j^{(k-1)}) - R_{ij},$$

$$\varphi_{ij}^3(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) = (z_i^{(k-1)} - z_j^{(k-1)}) - R_{ij},\ \varphi_{ij}^4(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) = -(x_i^{(k-1)} - x_j^{(k-1)}) - R_{ij},$$

$$\varphi_{ij}^5(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) = -(y_i^{(k-1)} - y_j^{(k-1)}) - R_{ij},\ \varphi_{ij}^6(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) = -(z_i^{(k-1)} - z_j^{(k-1)}) - R_{ij},$$

$$R_{ij} = (r_i + r_j) + \rho_{ij} + 2\varepsilon,$$

$\hat{\Phi}^{S_{a_i} S_{a_j}}(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)})$ is an adjusted phi-function (12) for a pair of spheres S_q and S_g ($a_i = q$, $a_j = g$), circumscribed around concave polyhedra $\mathbb{Q}_q(u_q^{(k-1)}) \supset K_i(u_q^{(k-1)})$ and $\mathbb{Q}_g\left(u_g^{(k-1)}\right) \supset K_j\left(u_g^{(k-1)}\right)$. We provide some illustrations to form index set Ξ_1^k in Appendix B.

We note that if $(i,j) \notin \Xi_1^k$, then we do not need to check the distance (or non-overlapping) constraint for the corresponding pair of polyhedra $K_i(u_{a_i}^{(k-1)})$ and $K_j(u_{a_j}^{(k-1)})$. If $\rho_{ij} = 0$ then function $\varphi^{\Omega_i^k \Omega_j^k}(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)})$ becomes a phi-function for two oriented parallelepipeds Ω_i and Ω_j.

To form index set Ξ_2^k we exclude from (8) all phi-inequalities for containment constraints of convex polyhedra where individual containers do not intersect the set $\Omega_\varepsilon^{k*} = R^3 \setminus \operatorname{int} \Omega_\varepsilon^k$, such that

$$\Omega_\varepsilon^k = \{(x,y,z) : \varepsilon \le x \le l^{(k-1)} - \varepsilon, \varepsilon \le y \le w^{(k-1)} - \varepsilon, \varepsilon \le z \le h^{(k-1)} - \varepsilon\}.$$

Thus, $\Xi_2^k = \{i \in \Xi_2^{kS} : \hat{\Phi}^{\Omega_i^k \Omega_\varepsilon^*}(v_{a_i}^{(k-1)}) < 0\}$, where $\hat{\Phi}^{\Omega_i^k \Omega_\varepsilon^*}(v_{a_i}^{(k-1)})$ is an adjusted phi-function for a polyhedron $K_i(u_{a_i}^{(k-1)})$ and the object Ω_ε^{k*}, $\Xi_2^{kS} = \{i \in I_n : \hat{\Phi}^{S_{a_i} \Omega_\varepsilon^{k*}}(v_{a_i}^{(k-1)}) < 0\}$, $\hat{\Phi}^{S_{a_i} \Omega_\varepsilon^{k*}}$ is an adjusted phi-function (13) for a sphere S_q, associated with concave polyhedron $\mathbb{Q}_q(u_q^{(k-1)}) \supset K_i(u_q^{(k-1)})$, and the object Ω_ε^*, $a_i = q$,

$$\widehat{\Phi}^{\Omega_i^k \Omega_\varepsilon^{k*}}(v_{a_i}^{(k-1)}) = \min\{\psi_i^s(v_{a_i}^{(k-1)}), s = 1,\dots,6\},$$

$$\psi_i^1(v_{a_i}^{(k-1)}) = x_i^{(k-1)} - R_i,\ \psi_i^2(v_{a_i}^{(k-1)}) = y_i^{(k-1)} - R_i,\ \psi_i^3(v_{a_i}^{(k-1)}) = z_i^{(k-1)} - R_i,$$

$$\psi_i^4(v_{a_i}^{(k-1)}) = -x_i^{(k-1)} + l^{(k-1)} - R_i,\ \psi_i^5(v_{a_i}^{(k-1)}) = -y_i^{(k-1)} + w^{(k-1)} - R_i,$$

$$\psi_i^6(v_{a_i}^{(k-1)}) = -z_i^{(k-1)} + h^{(k-1)} - R_i,\ R_i = r_i + \rho_q + 2\varepsilon.$$

We note that if $i \notin \Xi_2^k$, then we do not need to check the distance (or containment) constraint for the polyhedron $K_i(u_{a_i}^{(k-1)})$ and the object Ω_ε^k.

Step 6. Generate the *k*-th subproblem on solution subset $W_k = W \bigcap \Lambda_k^\varepsilon$ with deleted redundant phi-function inequalities and reduced dimension (O(*n*)):

$$\min_{u_{w_k} \in W_k \subset R^{\sigma - \sigma_k}} F(u_{w_k}), \tag{14}$$

$$W_k = \{u_{w_k} = (\varsigma, \tau_{w_k}) \in R^{\sigma-\sigma_k} : \widehat{\Phi}_{ij}^{'}(u_{a_i}, u_{a_j}) \ge 0, (i,j) \in \Xi_1^k, \widehat{\Phi}_i(u_{a_i}) \ge 0, i \in \Xi_2^k,$$
$$\Phi^{S_i \Omega_i^{k*}}(u_{a_i}) \ge 0, i = 1,\dots,n, l \ge l^{(k-1)} - \varepsilon, w \ge w^{(k-1)} - \varepsilon, h \ge h^{(k-1)} - \varepsilon\},$$

where Ξ_1^k and Ξ_2^k are defined on Step 5, $\sigma_k = 3(m - card(\Xi_1^k))$ is the number of all deleted auxiliary variables meeting in the appropriate redundant phi-function inequalities, $\sigma - \sigma_k = 3 + 6N + card(\Xi_1^k)$, $card(\Xi_1^k)$ is (O(*n*)).

Step 7. Generate a feasible starting point $u^{(k-1)} = (\varsigma^{(k-1)}, \tau_{w_k}^{(k-1)})$ for problem (14). Since a vector $\varsigma^{(k-1)}$ has already defined, we need to find values of the vector of auxiliary variables $\tau_{w_k}^{(k-1)} = (u_P^{(k-1)1}, \dots, u_P^{(k-1)s}, \dots, u_P^{(k-1)m})$ for such $s \in \{1,\dots,m\}$ that $(i,j) \in \Xi_1^k$.

To derive a vector $u_P^{(k-1)s}$ we employ the FAPA algorithm.

The key idea of the FAPA algorithm lies in the following (see Appendix C): we derive a vector $u_P^{(k-1)s}$ as a vector of feasible parameters of a separating plane for two spheres $S_i(u_{a_i}^{(k-1)})$ and $S_j(u_{a_j}^{(k-1)})$ if $\widehat{\Phi}^{S_i S_j} \ge 0$, using simple geometrical calculations, otherwise we find a vector $u_P^{(k-1)s}$, solving the following auxilary subproblem

$$\max \alpha \ \text{ s.t.} (u_P^s, \alpha) \in W_\alpha^{'}, \tag{15}$$

where

$$W_\alpha^{'} = \{(u_P^s, \alpha) \in R^4 : \widehat{\Phi}_{ij}^{'}(u_{a_i}^{(k-1)}, u_{a_j}^{(k-1)}, u_P^s) - \alpha \ge 0\},$$

$\alpha \in R^1$, $u_P^s = (\theta_P^{1s}, \theta_P^{2s}, \alpha_P^s)$, under fixed parameters $(u_{a_i}^{(k-1)}, u_{a_j}^{(k-1)})$ involving in the appropriate adjusted quasi phi-function $\widehat{\Phi}'_{ij}(u_{a_i}^{(k-1)}, u_{a_j}^{(k-1)}, u_P^s)$, $\forall (i, j) \in \Xi$. It should be noted that any nonnegative value of α in (15) provides feasible values of u_P^s.

Thus, all adjusted quasi phi-functions and phi-functions in (14) at the point $u^{(k-1)}$ take nonnegative values.

Step 8. Solve subproblem (14), starting from the feasible point $u^{(k-1)}$

$$\min_{u_{w_k} \in W_k \subset R^{\sigma - \sigma_k}} F(u_{w_k}), \tag{16}$$

and get a local minimum point $u_{w_k}^* = (\varsigma^{*k}, \tau_{w_k}^{*k})$.

If the point $u_{w_k}^*$ of local minimum of subproblem (16) belongs to the frontier of an auxiliary subset Λ_k^ε, i.e. $u_{w_k}^* \in fr\Lambda_k^\varepsilon$, then we take ς^{*k} as a starting vector ς^k for the next iteration of the procedure (set $k = k+1$ and go to Step 2), otherwise we stop the optimisation procedure.

We claim that the point $u^* = u^{*k} = (\varsigma^{*k}, \tau^{*k}) \in R^\sigma$ is a point of local minimum of problem (8)-(9), where τ^{*k} involves $\tau_{w_k}^{*k}$ and auxiliary variables that are deleted at the k-th iteration. Note that the σ_k previously deleted auxiliary variables can be redefined by FAPA algorithm. However we do not need to redefine the deleted auxiliary variables at the last step of the algorithm, since the values of auxiliary variables have no effect on the value of the objective function, i.e. $F(u_{w_k}^*) = F(u^{*k})$.

Figure 10 shows the diagram of the COMPOLY procedure to solve problem (8)-(9). We illustrate the procedure of solving a sequence of subproblems, given by (16), for k=2,3,4. Note, that feasible starting point $u^{(0)}$ is found by algorithm FPPA. Each auxiliary (artificial) set Λ_k^ε , described at Step 4 of the COMPOLY procedure, is shown as a square with the centre point $u^{(k-1)}$, k=1,2,3,4.

We take the feasible point $u^{(0)}$, form set Λ_1^ε with the center point $u^{(0)}$, solve subproblem (16) on subregion $W_1 = \Lambda_1^\varepsilon \bigcap W$ and get a local minimum point $u_{w_1}^*$. The point $u_{w_1}^*$ belongs to the frontier of set Λ_1^ε, therefore we form the next set Λ_2^ε with the center point $u^{(1)} = u_{w_1}^*$ and search for a local minimum point $u_{w_2}^*$ of subproblem (16) on subregion $W_2 = \Lambda_2^\varepsilon \bigcap W$. The point $u_{w_2}^*$ belongs to the frontier of set Λ_2^ε, therefore we form the next set Λ_3^ε with the center point $u^{(2)} = u_{w_2}^*$ and search for a local minimum point $u_{w_3}^*$ of subproblem (16) on subregion $W_3 = \Lambda_3^\varepsilon \bigcap W$. The point $u_{w_3}^*$ belongs

to the interior of set Λ_3^{ε}, i.e. $u_{w_3}^* \in \operatorname{int} \Lambda_3^{\varepsilon}$, therefore we stop our procedure. The point $u_{w_3}^* = u^*$ is a point of local minimum of problem (8)-(9).

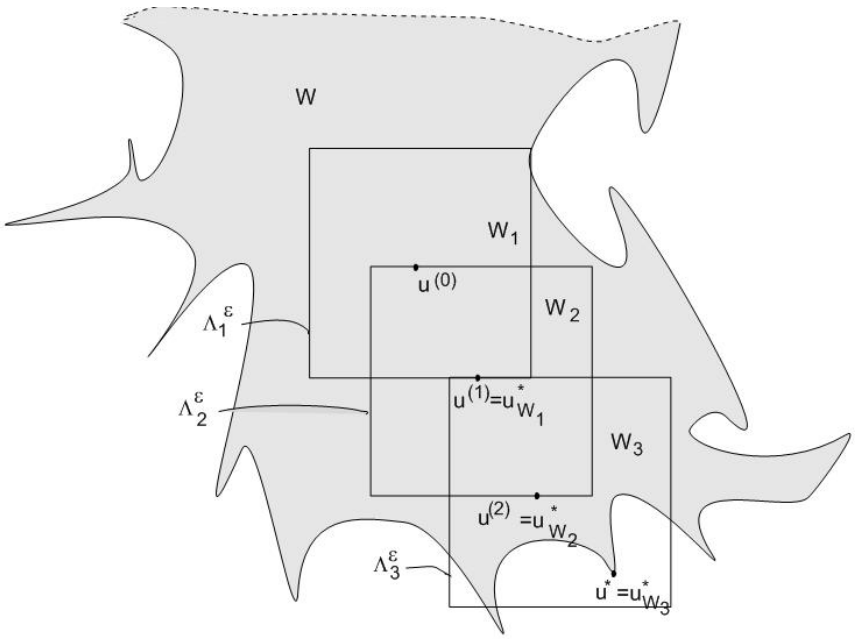


Fig. 10 – Diagram of the COMPOLY procedure.

Figure 11 illustrates the iterative procedure of packing concave polyhedra that is related to the Diagram shown in Figure 10.

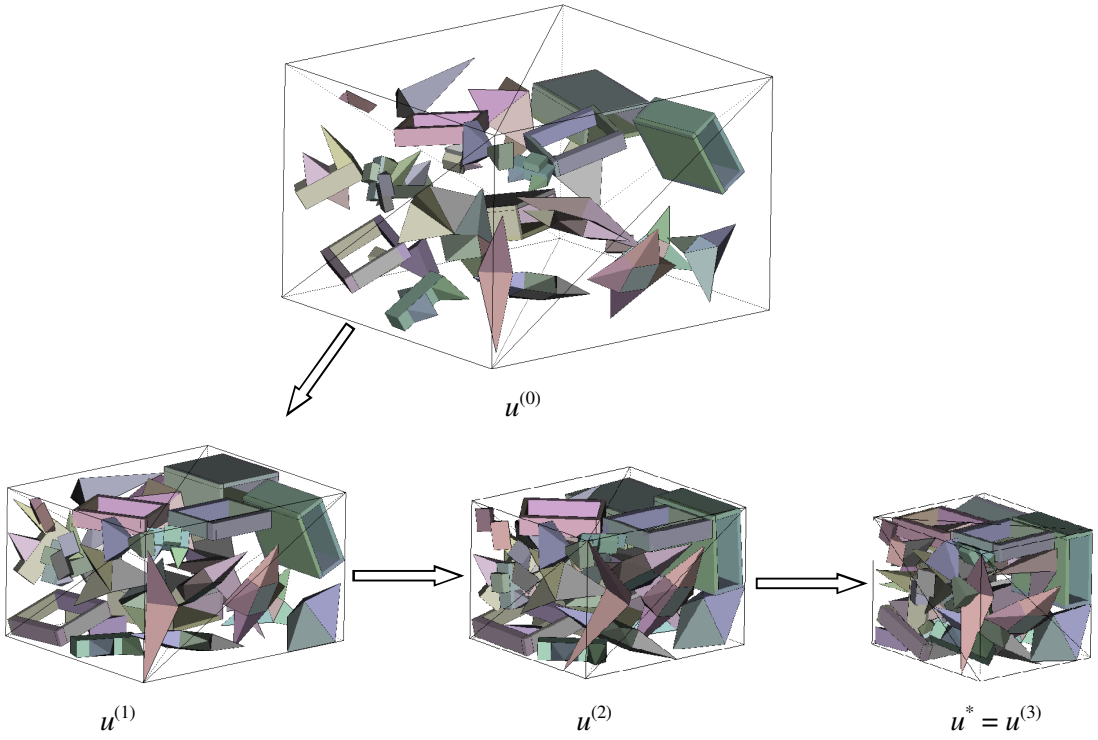


Fig. 11 – Arrangements of concave polyhedra, corresponding to the sequence of feasible points $u^{(0)}$, $u^{(1)}$, $u^{(2)}$, $u^* = u^{(3)}$ with respect to the Diagram.

We note that dist($u^*_{w_k}, u^*_{w_{k+1}}$) $\geq \varepsilon$, if $u^*_{w_{k+1}} \in fr\Lambda^{\varepsilon}_k$, and we take the value of ε that is considerably greater than the accuracy of IPOPT (10^{-8}). Thus, we can conclude that the stopping condition of the COMPOLY procedure is always reached in a finite number of iterations.

If the IPOPT program fails to find a local minimum of subproblem (14), we halve the value of ε and start up the COMPOLY procedure. If a local minimum is found under the half value of ε then we recover the initial value of epsilon and continue the COMPOLY procedure for a new feasible starting point, otherwise we terminate the procedure.

Our algorithm, in most cases, takes consideration of significantly fewer pairs of polyhedra than m (here m is the number of all pairs of convex polyhedra considered in problem (8)-(9)), because for each polyhedron only its "ε-neighbors" have to be monitored. It should be noted that the algorithm is not efficient for special cases when all objects are "ε-neighbors".

The parameter ε provides a balance between the number of inequalities in each nonlinear programming subproblem (14) and the number of the subproblems (12), which we need to generate and solve in order to get a local optimal solution of problem (8)-(9).

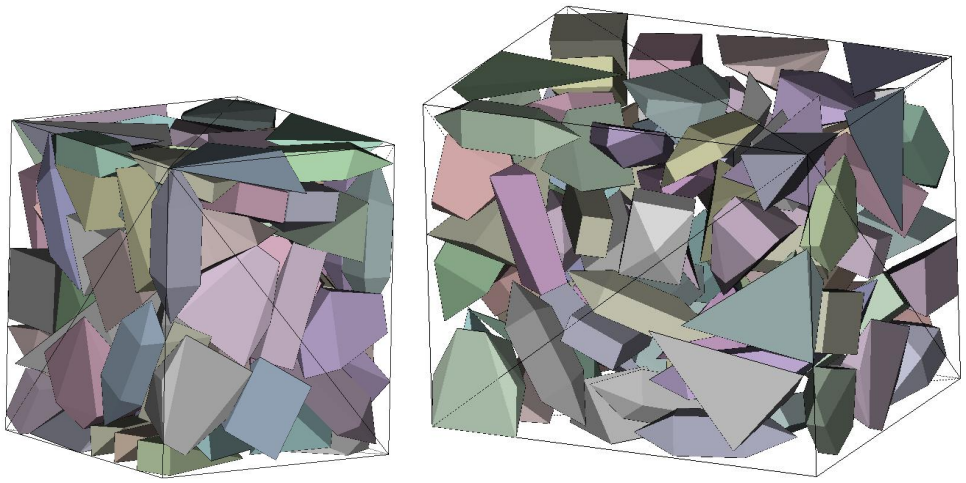
Thus the COMPOLY algorithm allows us to reduce the problem (8)-(9) with a large number of inequalities and dimension O(n^2) of the feasible set W, described by (9), to a sequence of subproblems (14) with a smaller number of nonlinear inequalities and dimension O(n) of solution subset W_k.

6. Computational experiments

We present a number of examples to demonstrate the efficiency of our methodology. We have run all experiments on an AMD Athlon 64 X2 5200+ computer, Programming Language C++, Windows 7. For the local optimisation we use the IPOPT code (https://projects.coin-or.org/Ipopt) by means of program interface using the default options.

The following examples set $\varepsilon = 5$ for the COMPOLY procedure.

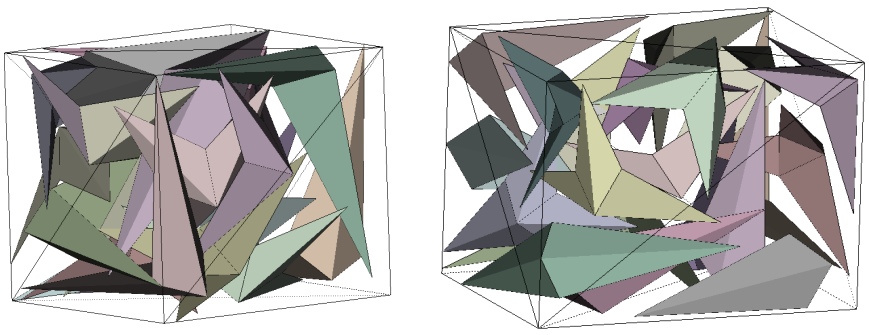
Example 1. We generate a collection of $n = 98$ convex polyhedra, consisting of the 7 types of polyhedra from example 1 given in [18] and in Appendix A. We include 14 of each type of polyhedra. Figure 12 shows the local optimal placement of the collection of convex polyhedra. The container has dimensions and volume: a) (l^*, w^*, h^*)=(30.9324, 28.1897, 26.5064) and $F(u^*) = 23113.06$ with $\rho = 0$ (Fig. 12a). One starting point is used. Computational time is 147967.3 sec.; b) (l^*, w^*, h^*)=(41.3510, 33.0721, 31.7988) and $F(u^*) = 43487.0040$ with $\rho = 1.5$ (Fig. 12b). One starting point is used. Computational time is 48152.79 sec.



(a) (b)

Fig. 12 – Local optimal placement of polyhedra in Example 1: a) $\rho = 0$; b) $\rho = 1.5$.

Example 2. We generate a collection of *N*=20 concave polyhedra, consisting of the 2 types of polyhedra given in [17] and in Appendix A. We include 10 of each type of polyhedra. Figure 13 shows the local optimal placement of the collection of concave polyhedra. The container has dimensions and volume: a) (l^*, w^*, h^*)=(26.3522, 23.7514, 24.4055) and $F(u^*) = 15275.4815$ with $\rho = 0$ (Fig. 13a). Two starting points are used. Computational time is 8729.45 sec.; b) (l^*, w^*, h^*)=(26.5890, 26.5239, 36.1706) and $F(u^*) = 25509.2576$ with $\rho = 1.5$. Ten starting points are used. Computational time is 24696.46 sec. (Fig. 13b).

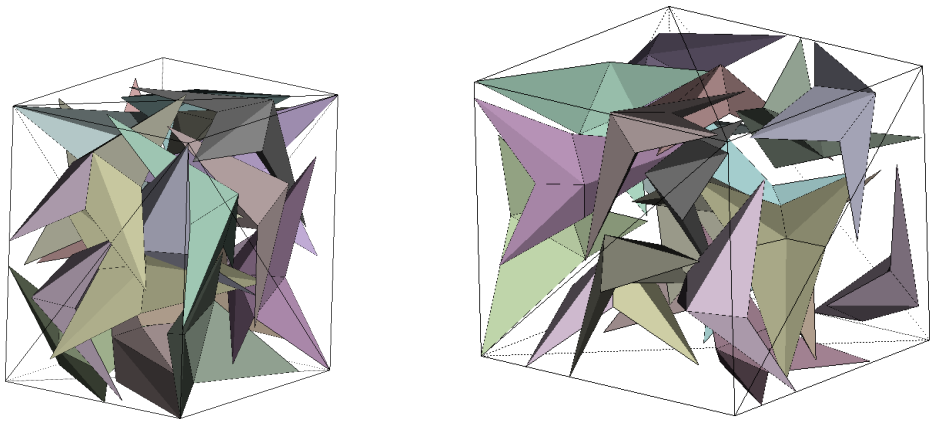


(a) (b)

Fig. 13 – Local optimal placement of polyhedra in Example 2: a) $\rho = 0$; b) $\rho = 1.5$.

Example 3. We consider a collection of *N*=20 equal concave polyhedra given in [17] and in Appendix A. Figure 14 shows the local optimal placement of *the collection* of concave polyhedra. The container has dimensions and volume: a) (l^*, w^*, h^*)=(23.7706, 26.6212, 20.2363) and $F(u^*) = 12805.6718$ with $\rho = 0$. Ten starting points are used. Computational time is 59497.9 sec. (Fig. 14a); b) (l^*, w^*, h^*) =(27.9795, 26.5408, 30.6725) and $F(u^*) = 22777.4233$ with $\rho = 1.5$. Ten starting points are used.

Computational time is 28700.16 sec. (Fig. 14b).



(a) (b)

Fig.14 – Local optimal placement of polyhedra in Example 3: a) $\rho = 0$; b) $\rho = 1.5$.

Example 4. We pack 45 concave polyhedra of 10 types given in Appendix A. We include 5 polyhedra of each type of the upper polyhedra row and 4 polyhedra of each type of the lower polyhedra row (Fig. A3). Figure 15 shows the local optimal placement of *the collection* of concave polyhedra. The container has dimensions $(l^*, w^*, h^*) =$ (39.7324, 34.8629, 44.6587) and volume $F(u^*) = 61860.807$. Three starting points are used. Computational time is 159884.0 sec.

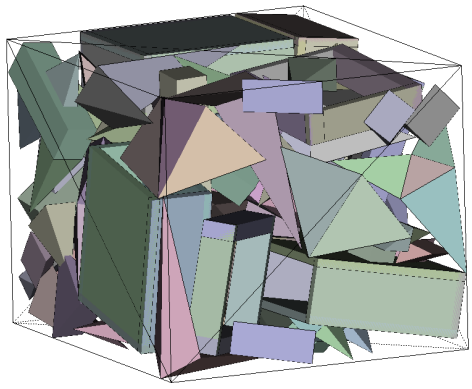


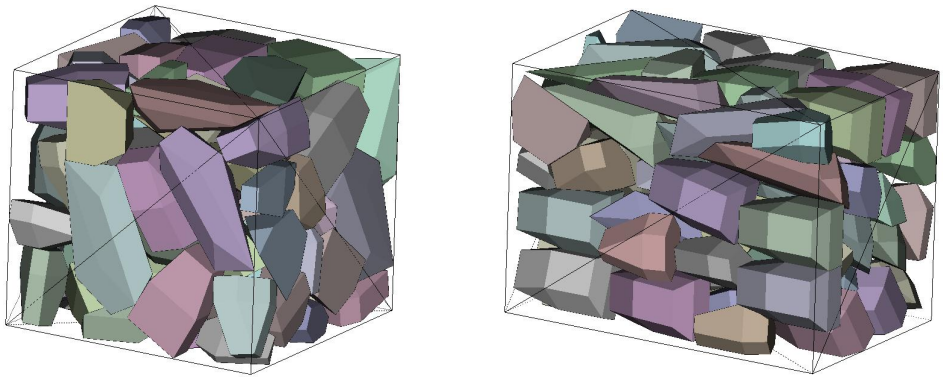
Fig.15 – Local optimal placement of concave polyhedra in Example 4.

Futher we compare our results to those given in [17] and [18]. We search for locally optimal solutions employing the compaction algorithm: a) starting from a feasible point generated by FPPA algorithm described in Section 5.1 and b) starting from a feasible point found by the algorithm developed in [17] and [18].

Example 5. We consider a collection of *n*=80 convex polyhedra, of example 1 given in [18] and in Appendix A. Figure 16 shows the local optimal placement of the collection of convex polyhedra. The container has dimensions and volume: a) $(l^*, w^*, h^*) =$(43.4338, 41.8435, 45.0059) and $F(u^*) =$ 81795.2169, starting from the feasible point found by FPPA. Computational time is 46035.78 sec.; b) $(l^*, w^*, h^*) =$(36.3569, 40.8764, 56.2557) and $F(u^*) = 83604.0544$, starting from the feasible point

found by the algorithm given in [18]. Computational time is 42950.4 sec.. Improvement of the value of objective function in comparison to the result given in [18]: a) 27.88%; b) 26.29%

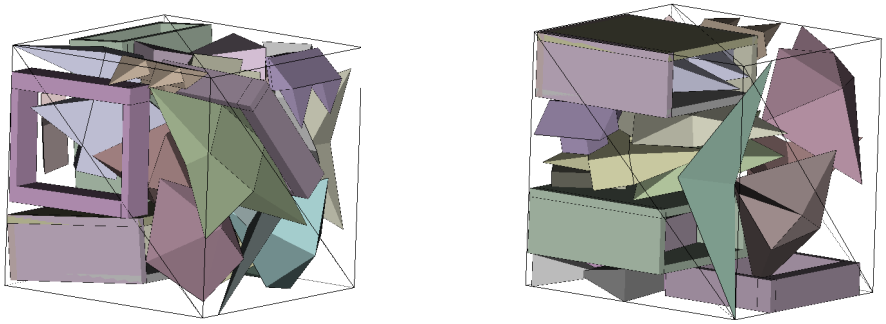
.



(a) (b)

Fig.16 – Local optimal placement of polyhedra in Example 5: a) starting from the feasible point found by FPPA; b) starting from the feasible point found by the algorithm given in [18].

Example 6. We consider a collection of *N*=20 concave polyhedra, of example 2 given in [17] and in Appendix A. Figure 17 shows the local optimal placement of the collection of concave polyhedra. The container has dimensions and volume: a) (l^*, w^*, h^*)=(29.7159, 30.6070, 30.1616) and $F(u^*)=$ 27432.6412, starting from the feasible point found by FPPA; b) (l^*, w^*, h^*)=(31.4820, 27.8994, 32.0000) and $F(u^*)=28106.6387$, starting from the feasible point found by the algorithm given in [17]. We generate 11 starting points, time limit is 10 hours. Improvement of the value of objective function in comparison to the result given in [17]: a) 18.36%; b) 16.35%

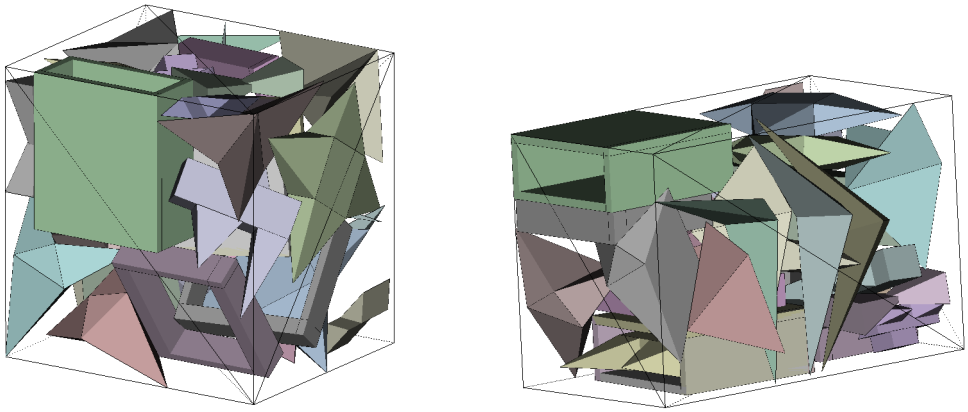


(a) (b)

Fig.17 – Local optimal placement of polyhedra in Example 6: a) starting from the feasible point found by FPPA; b) starting from the feasible point found by the algorithm given in [17].

Example 7. We consider a collection of *N*=30 concave polyhedra, of example 3 given in [17] and in Appendix A. Figure 18 shows the local optimal placement of the collection of concave polyhedra. The container has dimensions and volume: a) (l^*, w^*, h^*)=(36.9929, 36.3796, 30.9454) and $F(u^*)=$

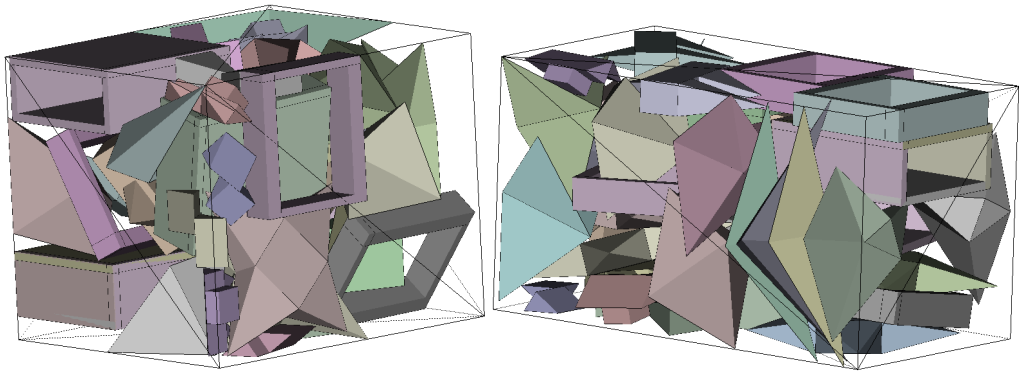
41646.1709, starting from the feasible point found by FPPA; b) (l^*, w^*, h^*)=(31.4376, 26.1920, 48.9148) and $F(u^*) = 40277.1892$, starting from the feasible point found by the algorithm given in [17]. We generate 11 starting points, time limit is 10 hours. Improvement of the value of objective function in comparison to the result given in [17]: a) 19.06 %; b) 21.72%



(a) (b)

Fig.18 – Local optimal placement of polyhedra in Example 7: a) starting from the feasible point found by FPPA; b) starting from the feasible point found by the algorithm given in [17].

Example 8. We consider a collection of *N*=40 concave polyhedra, of example 4 given in [17] and in Appendix A. Figure 19 shows the local optimal placement of the collection of concave polyhedra. The container has dimensions and volume: a) (l^*, w^*, h^*)=(34.9974, 36.9655, 43.2777) and $F(u^*) =$ 55988.4619, starting from the feasible point found by FPPA; b) (l^*, w^*, h^*)=(31.1419, 30.8086, 55.4061) and $F(u^*) = 53158.8838$, starting from the feasible point found by the algorithm given in [17]. We use 3 starting points, time limit is 10 hours. Improvement of the value of objective function in comparison to the result given in [17]: a) 15.64%; b) 19.91%

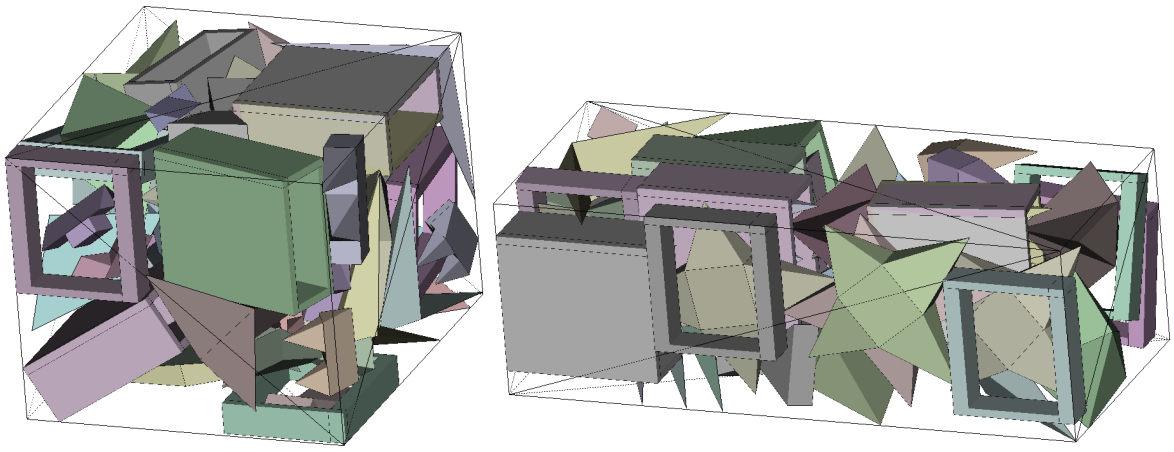


(a) (b)

Fig.19 – Local optimal placement of polyhedra in Example 8: a) starting from the feasible point found by FPPA; b) starting from the feasible point found by the algorithm given in [17].

Example 9. We consider a collection of *n*=50 concave polyhedra, of example 9 given in [17] and in

Appendix A. Figure 20 shows the local optimal placement of the collection of concave polyhedra. The container has dimensions and volume: a) (l^*, w^*, h^*)=(46.9742, 34.8305, 41.6923) and $F(u^*)$= 68214.5610, starting from the feasible point found by FPPA; b) (l^*, w^*, h^*)=(32.0000, 25.5894, 75.2637) and $F(u^*) = 61630.6754$, starting from the feasible point found by the algorithm given in [17]. One starting point is used, time limit is 10 hours. Improvement of the value of objective function in comparison to the result given in [17]: a) 17.45%; b) 25.42%



(a) (b)

Fig.20 – Local optimal placement of polyhedra in Example 9: a) starting from the feasible point found by FPPA; b) starting from the feasible point found by the algorithm given in [17].

Table 1 lists some examples presented in [13]. For each example the minimal volume of the container found by our method is smaller than the best solution reported in [13].

Table 1. Comparison of our results to those in [13]

Problem	the best volume from [13]	the best time (sec.) from [13]	found by FPPA* + COMPOLY volume	found by FPPA* + COMPOLY time (sec.)	found by [17]** + COMPOLY volume	found by [17]** + COMPOLY time (sec.)
20 from [17]	32550	26202.1	27432.64	34313.34	28106.64	5360.67
30 from [17]	48300	53741.5	41646.17	35289.34	40277.19	33008.89
40 from [17]	61950	99952.0	53158.88	201501.5	55988.46	195051.51
50 from [17]	77280	125210.6	68214.56	215144.55	61630.68	270654.84
36 from [13]	12480	9637.5	10461.67	23023.12	–	–

Note. In table 1: * – a starting feasible point found by FPPA; ** – a starting feasible point found by algorithm found in [17].

Example 10. We consider the collection of polyhedra of example 1 given in [13] and in Appendix A. Figure 21 shows the local optimal placement of *n*=36 concave polyhedra, starting from the feasible

point found by FPPA. The container has dimensions (l^*, w^*, h^*)=(21.5851, 19.8685, 24.3938) and volume $F(u^*) = 10461.67$. We generate 5 starting points, time limit is 10 hours. Improvement of the value of objective function is 16.18%.

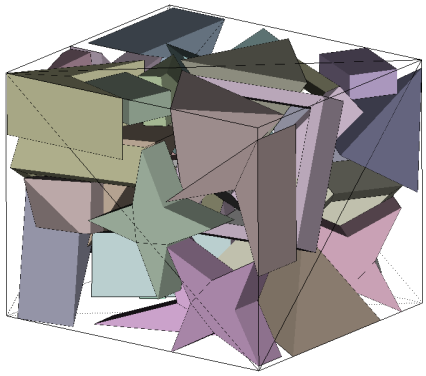


Fig.21 – Local optimal placement of polyhedra in Example 10.

To show the effectiveness of the COMPOLY procedure, some tests were performed. In the example for N = 10 concave polyhedra from Appendix A, the average computational time per one local extremum is: a) 1380 sec. without the use of the COMPOLY procedure; b) 283 sec. using the COMPOLY procedure. The number of variables and inequalities is: a) 1791 and 7934 without the use of the COMPOLY procedure; 626 and 3086 using the COMPOLY procedure at the last iteration.
In Example 6 for N=20 concave polyhedra, the average computational time per one local extremum is: a) 75026.31 sec. without the use of the COMPOLY procedure; b) 4980.74 sec. using the COMPOLY procedure. The number of variables and inequalities is: a) 7471 and 30916 without the use of the COMPOLY procedure; 1334 and 8028 using the COMPOLY procedure at the last iteration.

In Example 7 for N=30 concave polyhedra a local minimum has not been found within the time limit of 72 hours without using of the COMPOLY procedure. The average computational time per one local extremum is 35289.34 sec. using the COMPOLY procedure.

7. Conclusions and future work

We derive radical free adjusted quasi phi-functions to describe non-overlapping constraints for concave polyhedra and use adjusted phi-functions to describe containment constraints. These tools take into account continuous rotations of polyhedra and minimal allowable distances between objects. We introduce an exact mathematical model for the optimal polyhedron packing problem as a nonlinear programming problem with smooth functions. Our approach involves a fast starting point algorithm. We also propose the COMPOLY procedure to search for "good" local optimal solutions. It can be used as a compaction algorithm, starting from a feasible point found by any algorithm published before. The COMPOLY procedure allows us to reduce computational costs (time and memory) considerably. This reduction is of a paramount importance, since we deal with nonlinear optimisation problems. Our results on new instances and instances from the literature show our approach has superior performance. In the

near future, we intend to apply our methodology to pack arbitrary polyhedra into different shaped containers (a sphere, a cylinder, a polytope, a spheroid, an ellipsoid) with different objectives (e.g., maximum of the space usage) and additional constraints (e.g., behavior constraints).

References

1. Bennell, J., Oliveira, J. (2008). The geometry of packing problems:A tutorial.European Journal of Operational Research184:397–415.
2. Chazelle, B., Edelsbrunner, H., Guibas, L. J. (1989). The complexity of cutting complexes. Discr. & Comput. Geom., **4**(2), 139–181. DOI: 10.1007/BF02187720.
3. Chen, E. R., Klotsa, D., Engel, M., Damasceno, P. F., Glotzer, S. C. (2014). Complexity in surfaces of densest packings for families of polyhedra. Phys Rev X **4**(1), DOI:10.1103/PhysRevX.4.011024.
4. Chernov, N., Stoyan, Y., Romanova, T. (2010). Mathematical model and efficient algorithms for object packing problem. Comput. Geom.: Theory and Appl., **43**(9), 535–553. DOI:10.1016/j.comgeo.2009.12.003.
5. Egeblad, J., Nielsen, B. K., Brazil, M. (2009). Translational packing of arbitrary polyhedra. Comp. Geom., **42**(4), 269–288. DOI:10.1016/j.comgeo.2008.06.003.
6. Fasano, G. A. (2013). Global Optimisation point of view for non-standard packing problems. J. Glob. Optim., **55**(2), 279–299. DOI: 10.1007/s10898-012-9865-8.
7. Fischer, K., Gärtner, B. and Kutz, M. (2003). Fast Smallest-Enclosing-Ball Computation in High Dimensions. Algorithms - ESA 2003, **2832**, 630–641. DOI:10.1007/978-3-540-39658-1_57.
8. Galrão, R. A., Oliveira J. F., Gonçalves J. F., Lopes M. P. (2016) A container loading algorithm with static mechanical equilibrium stability constraints. Transportation Research, (Part B), **91**, 565-581. DOI: 10.1016/j.trb.2016.06.003.
9. Gomes, A. Miguel Irregular Packing Problems: Industrial Applications and New Directions Using Computational Geometry. (2014) Paper in special issue on “Cutting and Packing". Vol **11** | Part 1, 378-383. DOI: 10.3182/20130522-3-BR-4036.00113.
10. Kallrath, J. (2009). Cutting Circles and Polygons from Area-Minimizing Rectangles. Journal of Global Optimization, **43**(2), 299–328. DOI: 10.1007/s10898-007-9274-6.
11. Korte, A. C. J., Brouwers H. J. H. (2013). Random packing of digitized particles. Powder Techn., **233**, 319–324. DOI: 10.1016/j.powtec.2012.09.015.
12. Li, S. X., Zhao, J., Lu, P., Xie, Y. (2010). Maximum packing densities of basic 3D objects. Chin. Scien. Bull., **55**(2), 114–119. DOI: 10.1007/s11434-009-0650-0.
13. Liu, X., Liu, J., Cao, A., Yao, Z. (2015). HAPE3D – a new constructive algorithm for the 3D irregular packing problem. Frontiers of Information Technology & Electronic Engineering, **16**(5), 380–390. DOI: 10.1631/FITEE.1400421.
14. Pankratov, A., Romanova, T., Chugay, A. (2015). Optimal packing of convex polytopes using quasi phi-functions. Journal of Mechanical Engineering, **18** (2), 55-64.

15. Smeets, B., Odenthal, T., Vanmaercke, S., Ramon, H. (2015). Polygon-based contact description for modeling arbitrary polyhedra in the Discrete Element Method. Computer Methods in Applied Mechanics and Engineering, **290**, 277-289. DOI: 10.1016/j.cma.2015.03.004.
16. Stoyan, Y., Chugay, A. (2012). Mathematical modeling of the interaction of non-oriented convex polyhedra. Cyber. and Sys. Anal.,**48** (6), 837–845. DOI: 10.1007/s10559-012-9463-2.
17. Stoyan, Y. G., Gil, N. I., Pankratov, A. V., et al., (2004). Packing Non-convex Polyhedra into a Parallelepiped. Technische Universitat Dresden.
18. Stoyan, Y., Gil, N., Scheithauer, G., Pankratov, A., Magdalina, I. (2005). Packing of convex polyhedra into a parallelepiped. Optimisation, **54** (2), 215 – 235. DOI: 10.1080/02331930500050681.
19. Stoyan, Y., Pankratov, A., Romanova, T. (2016). Quasi phi-functions and optimal packing of ellipses. J. of Glob. Optim., **65** (2), 283–307. DOI: 10.1007/s10898-015-0331-2.
20. Stroeven, P. and He, H. (2013). Packing of non-spherical aggregate particles by DEM. Advances in Cement and Concrete Technology in Africa, Uzoegbo, H.C. and Schmidt, W. (Eds.) BAM Fed. Inst. Mat. Test., Berlin: 809-816.
21. Tasios, N., Gantapara, A. P., Dijkstra M. (2014). Glassy dynamics of convex polyhedra. The Journal of Chemical Physics. 141: 224502. PMID 25494755 DOI: 10.1063/1.49029922.
22. Torquato, S., Jiao, Y. (2009). Dense polyhedral packings: Platonic and Archimedean solids. Phys. Rev., **80**, 041104. DOI: 10.1103/PhysRevE.80.041104.
23. Wachter, A., Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program., **106** (1), 25–57. DOI: 10.1007/s10107-004-0559-y.
24. Wascher, G., Hauner, H., Schumann, H. (2007). An improved typology of cutting and packing problems. Eur. J. Oper. Res., **183**(3), 1109–1130. DOI: 10.1016/j.ejor.2005.12.047.

APPENDIX A: DATA FOR EXAMPLES IN SECTION 6

1. DATA FOR CONVEX POLYHEDRA

Data for Example 1

We consider 7 types of covex polyhedra K_1, K_2, K_3, K_4, K_5, K_6, K_7 (Fig. A1).

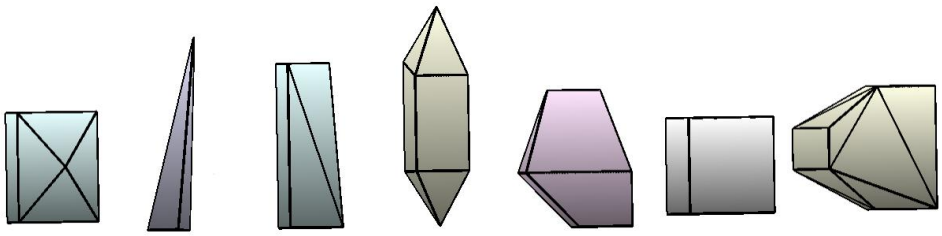


Fig. A1 – Types of convex polyhedra $\boldsymbol{K_i}$, i=1,…,7 in Example 1.

Vertex coordinates of polyhedron K_1:

$\{(x_j,y_j,z_j), j=1,2,\ldots,9\}=\{(3,6,0), (3,6,8), (3,0,8), (3,0,0), (0,6,0), (0,6,8), (0,0,8), (0,0,0), (5,3,4)\}$

Vertex coordinates of polyhedron K_2:

$\{(x_j,y_j,z_j), j=1,2,\ldots,4\}=\{(8,0,-4), (-3,4,-4), (6,2,10), (0,0,-4)\}$

Vertex coordinates of polyhedron K_3:

$\{(x_j,y_j,z_j), j=1,2,\ldots,7\}=\{(3,0,-4), (3,4,-4), (3,0,8), (0\ 4,-4), (0,4,8), (0,0,8), (0,0,-4)\}$

Vertex coordinates of polyhedron K_4:

$\{(x_j,y_j,z_j), j=1,2,\ldots,10\}=\{(2,0,0), (1,2,-4), (2,4,0), (-1,4,0), (-1,0,0), (2,0,7), (2,4,7), (1,2,12), (-1,4,8), (-1,0,8)\}$

Vertex coordinates of polyhedron K_5:

$\{(x_j,y_j,z_j), j=1,2,\ldots,11\}=\{(2,-4,0), (2,4,0), (1,2,6), (1,-2,6), (0,4,0), (-2,0,0), (-1,2,6), (0,-4,0), (2,0,-4), (0,0,-4), (2,4,-4)\}$

Vertex coordinates of polyhedron K_6:

$\{(x_j,y_j,z_j), j=1,2,\ldots,6\}=\{(4,7,0), (4,7,7), (6,0,7), (6,0,0), (0,0,0), (0,0,7)\}$

Vertex coordinates of polyhedron K_7:

$\{(x_j,y_j,z_j), j=1,2,\ldots,10\}=\{(4,-4,2), (4,-4,-1), (2,0,-4), (1,5,-4), (1,5,5), (3,0,5), (0,0,5), (0,0,-4), (-2,-5,-1), (-2,-5,2)\}$

Data for Example 5

We consider 5 types of covex polyhedra K_1, K_2, K_3, K_4, K_5 (Fig. A2).

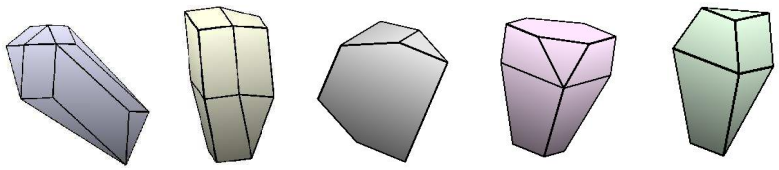


Fig. A2 – Types of convex polyhedra $\boldsymbol{K_i}$, i=1,2,…,5 in Example 5.

Vertex coordinates of polyhedron K_1:

$\{(x_j,y_j,z_j), j=1,2,\ldots,14\}=\{(4,2,0), (2,7,0), (0,3,3), (-11,8,-18), (1,5,-8), (3,0,-8), (-1,1,-5), (-14,10,-10), (-4,3,3), (-2,7,0), (-10,6,-10), (0,-1,3), (-10,10,-10), (4,-2,0)\}$

Vertex coordinates of polyhedron K_2:

$\{(x_j,y_j,z_j), j=1,2,\ldots,16\}=\{(3,-4,8), (3,-4,0), (3,6,0), (3,6,8), (-1,6,0), (-1,6,8), (-1,-4,8), (-5,-1,-1), (-5,-1,7), (-5,5,7),(-5,5,-1), (-1,-4,0), (2,-2,-8), (-2,-2,-8), (-2,4,-8), (2,4,-8)\}$

Vertex coordinates of polyhedron K_3:

$\{(x_j,y_j,z_j), j=1,2,\ldots,10\}=\{(8,0,10), (8,0,3), (6,5,0), (4,10,0), (4,10,13), (6,5,13), (2,5,13), (2,5,0), (0,0,3), (0,0,10)\}$

Vertex coordinates of polyhedron K_4:

$\{(x_j,y_j,z_j), j=1,2,\ldots,15\}=\{(7,0,8), (7,8,8), (7,4,12), (7,0,12), (0,8,8), (0,8,12), (4,8,12), (0,4,8), (0,4,12), (4,0,12), (6,2,0), (3,2,0), (3,6,0), (6,6,0), (4,0,8)\}$

Vertex coordinates of polyhedron K_5:

$\{(x_j,y_j,z_j), j=1,2,\ldots,11\}=\{(2,-4,0), (2,4,0), (1,2,4), (1,-2,4), (-1,-2,-8), (1,2,-8), (1,-2,-8), (0,-4,0), (-4,0,0), (-3,2,4), (-2,4,0)\}$

2. DATA FOR CONCAVE POLYHEDRA

Data for Examples 2- 4 and Examples 6 - 9

We consider 10 types of concave polyhedra (Fig.A3)

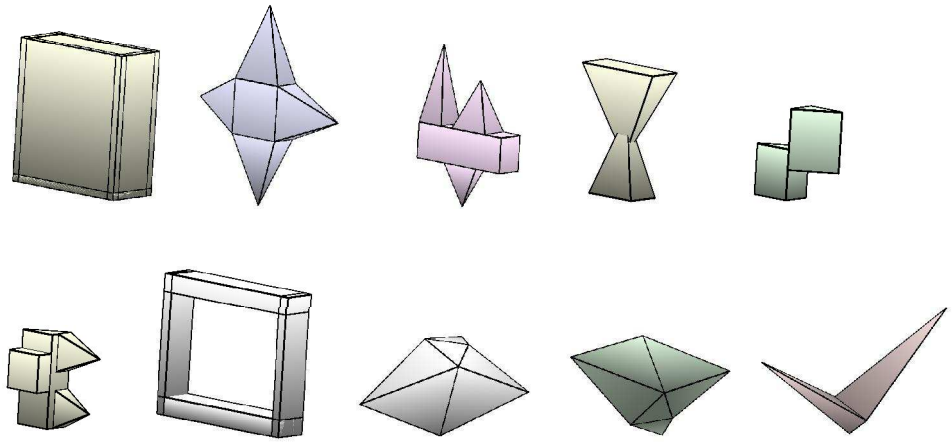


Fig. A3 – Types of concave polyhedra $\mathbb{Q}_q$, $q=1,2,\ldots,10$.

Each type of concave polyhedron is presented as a union of convex polyhedra given by the related collection of vertices in the local coordinate system of the appropriate concave polyhedron. Figure A4 shows decomposition of concave polyhedron $\mathbb{Q}_8$ with convex polyhedrons K_i, $i=1,2,3,4$.

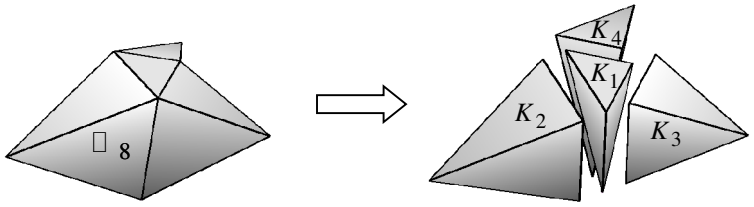


Fig. A4 – Concave polyhedron $\mathbb{Q}_8$ and convex polyhedrons K_i, $i=1,2,3,4$ that form the polyhedron.

We give here input data of vertices of convex polyhedrons that form concave polyhedra by two lists: list1 of vertex coordinates and list 2 of numbers of vertices (with respect to the list1) that define a collection of vertices of convex polyhedra that form appropriate concave polyhedron.

Remark. List 1 involves vertices of a concave polyhedron and, in general, additional vertices that appear as outcomes in construction of decomposition of the concave polyhedron with convex polyhedra.

List 1 of vertex coordinates (x_j,y_j,z_j), $j=1,2,\ldots,m_q$ for description of concave polyhedra:

$\mathbb{Q}_1$: $\{(x_j,y_j,z_j),\ j=1,2,\ldots,28\}$={(0,0,0), (8,0,0), (8,0,20), (0,0,20), (0,1,0), (0,1,20), (8,1,20), (8,1,0), (8,18,0), (8,18,20), (7,18,0), (7,18, 20), (7,0,20), (8,17,0), (0,17,0), (0,17,20), (8,17,20), (1,18,20), (0,18,20), (1,0,20), (1,0,0), (1,18,0), (0,18,0), (0,18,1), (0,0,1), (8,0,1), (8,18,1), (7,0,0)};

$\mathbb{Q}_2$:$\{(x_j,y_j,z_j),\ j=1,2,\ldots,12\}$={(0,0,0), (4,0,0), (4,8,0), (0,8,0), (4,0,8), (4,8,8), (0,8,8), (0,0,8), (2,-8,4), (2,4,-10), (2,18,4), (2,4,19);

$\mathbb{Q}_3$:$\{(x_j,y_j,z_j),\ j=1,2,\ldots,21\}$={(0,0,0), (4,0,0), (4,15,0), (0,15,0), (4,0,5), (4,15,5), (0,15,5), (0,0,5), (3,4,0), (1,4,0), (1,10,0), (3,10,0), (2,7,-6), (4,4,5), (0,4,5), (2,2,16), (4,6,5), (0,6,5), (0,12,5), (4,12,5), (2,9,12)};

$\mathbb{Q}_4$:$\{(x_j,y_j,z_j),\ j=1,2,\ldots,10\}$={(2,-3,0), (-2,-3,0), (-2,3,0), (2,3,0), (0,0,9), (0,0,4), (2,-5,14), (2,5,14), (-2,5,14), (-2,-5,14)};

$\mathbb{Q}_5$:$\{(x_j,y_j,z_j),\ j=1,2,\ldots,12\}$={(0,0,0), (3,0,0), (3,-4,0), (3,-4,5), (0,0,5), (3,0,5), (3,0,3), (0,4,3), (0,4,9), (3,0,9), (0,0,9), (0,0,3)};

$\mathbb{Q}_6$: $\{(x_j,y_j,z_j),\ j=1,2,\ldots,24\}$={(0,0,0), (4,0,0), (4,0,16), (0,0,16), (0,1,0), (4,1,0), (4,1,16), (0,1,16), (0,18,16), (4,18,16), (4,18,0), (0, 18,0), (0,17,16), (4,17,16), (4,17,0), (0,17,0), (4,0,2), (4,18,2), (0,18,2), (0,0,2), (4,0,14), (4,18,14), (0,18,14), (0,0,14)};

$\mathbb{Q}_7$: $\{(x_j,y_j,z_j),\ j=1,2,\ldots,22\}$={(3,0,0), (3,4,0), (0,4,0), (3,0,10), (3,4,10), (0,4,10), (5,0,8), (5,4,8), (5,4,4), (5,0,4), (3,0,8), (3,4,8), (3,4,4), (3,0,4), (2,4,6), (2,4,10), (1,9,8), (0,4,6), (2,4,4), (0,4,4), (2,4,0), (1,8,2)};

$\mathbb{Q}_8$: $\{(x_j,y_j,z_j),\ j=1,2,\ldots,7\}$={(0,0,0), (12,0,8), (-8,8,8), (-8,-8,8), (0,-4,12), (0,4,12), (-4,0,12)};

$\mathbb{Q}_9$: $\{(x_j,y_j,z_j),\ j=1,2,\ldots,7\}$={(0,0,0), (12,0,-8), (-8,8,-8), (-8,-8,-8), (0,-4,-12), (0,4,-12), (-4,0,-12)};

$\mathbb{Q}_{10}$: $\{(x_j,y_j,z_{j\,q}),\ j=1,2,\ldots,5\}$={(0,0,0), (0,-4,4), (0,4,4), (16,0,16), (-16,0,16)}.

List 2 of vertex numbers of the corresponding convex polyhedron K_i for each concave polytope $\mathbb{Q}_q$, q=1,2,…,10:

$\mathbb{Q}_1 = K_1 \cup K_2 \cup K_3 \cup K_4 \cup K_5$

K^1:{3,10,9,2,13,12,11,28}, K^2:{2,9,10,19,23,14,15,16,17}, K^3:{18,20,21,22,1,23,19,4}, K^4:{1,2,3,4,5,6,7,8}, K^5:{1,2,9,23, 24,25,26,27};

$\mathbb{Q}_2 = K_1 \cup K_2 \cup K_3 \cup K_4 \cup K_5$

$\mathbb{Q}_2$: K_1:{1,2,5,8,9}, K_2:{5,6,7,8,12}, K_3:{3,4,6,7,11}, K_4:{4,3,2,1,10}, K_5:{1,2,3,4,5,6,7,8};

$\mathbb{Q}_3= K_1 \cup K_2 \cup K_3 \cup K_4$

$\mathbb{Q}_3$: K_1:{1,2,3,4,5,6,7,8}, K_2:{5,8,15,14,16}, K_3:{17,18,19,20,21}, K_4:{9,10,11,12,13};

$\mathbb{Q}_4= K_1 \cup K_2$

$\mathbb{Q}_4$: K_1:{1,2,3,4,5}, K_2:{6,7,8,9,10};

$\mathbb{Q}_5= K_1 \cup K_2$

$\mathbb{Q}_5$: K_1:{1,2,3,4,5,6}, K_2:{7,8,9,10, 11,12};

$\mathbb{Q}_6= K_1 \cup K_2 \cup K_3 \cup K_4$

$\mathbb{Q}_6$: K_1:{1,2,3,4,5,6,7,8}, K_2:{9,10,11,12,13,14,15,16}, K_3:{17,18,19,20,1,2,11,12}, K_4:{21,22,23,24,3,4,9,10};

$\mathbb{Q}_7= K_1 \cup K_2 \cup K_3 \cup K_4$

$\mathbb{Q}_7$: K_1:{1,2,3,4,5,6}, K_2:{7,8,9,10,11,12,13,14}, K_3:{15,16,17,18,6}, K_4 :{19,20,21,22,3};

$\mathbb{Q}_8= K_1 \cup K_2 \cup K_3 \cup K_4$

$\mathbb{Q}_8$: K_1:{1,5,6,7}, K_2:{1,4,5,7}, K_3:{1,2,5,6}, K_4:{1,3, 6,7};

$\mathbb{Q}_9= K_1 \cup K_2 \cup K_3 \cup K_4$

$\mathbb{Q}_9$: K_1:{1,5,6,7}, K_2:{1,4,5,7}, K_3:{1,2,5,6}, K_4:{1,3,6,7};

$\mathbb{Q}_{10}= K_1 \cup K_2$

$\mathbb{Q}_{10}$: K_1:{1,2,3,4}, K_2:{1,2,3,5}.

APPENDIX B. FORMING INDEX SET Ξ_1^k IN COMPOLY ALGORITHM

Let three polyhedra are placed inside the container Ω^k at *k-th* iteration of COMPOLY algorithm (Fig. B1)

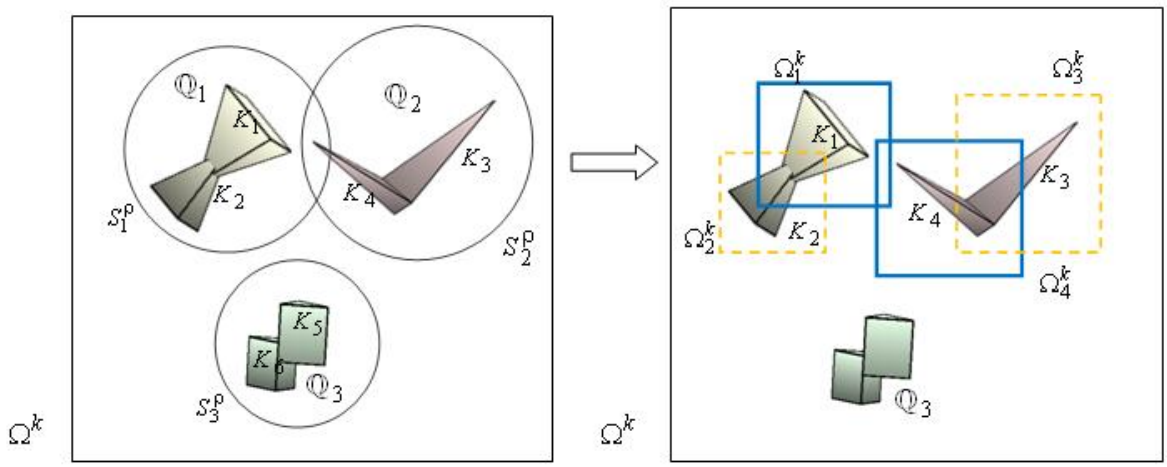


Fig. B1 – Illustration to construction of the index set Ξ_1^k at *k-th* iteration of COMPOLY algorithm.

For the example the index set Ξ defined by (7) has the form:

$\Xi = \{(1,3),(1,4),(1,5),(1,6),(2,3),(2,4),(2,5),(2,6),(3,5),(3,6),(4,5),(4,6)\}$.

Firstly we define the index set Ξ_1^{kS} (Fig B1a):

$\Xi_1^{kS} = \{(i,j) \in \Xi : \widehat{\Phi}^{S_{a_i} S_{a_j}}(v_1^{(k-1)}, v_2^{(k-1)}) < 0\} = \{(1,3),(1,4),(2,3),(2,4)\}$.

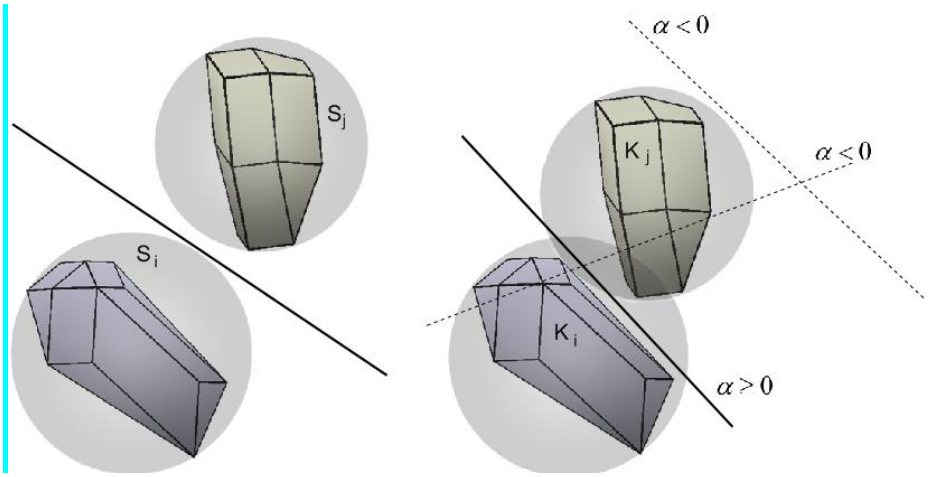
It means that only spheres S_1^ρ and S_2^ρ for concave polyhedra $\mathbb{Q}_1$ and $\mathbb{Q}_2$ have nonempty intersection, i.e. $\widehat{\Phi}^{S_1 S_2}(v_1^{(k-1)}, v_2^{(k-1)}) < 0$, and therefore it is sufficient to consider only possible intersection of convex polyhedra: K_1 and K_3, K_1 and K_4, K_2 and K_3, K_2 and K_4.

Then we form the index set Ξ_1^k (Fig B1b): $\Xi_1^k = \{(i,j) \in \Xi_1^{kS} : \varphi^{\Omega_i^k \Omega_j^k}(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}) < 0\} = \{(1,4)\}$.

It means that only individual containers Ω_1^k and Ω_4^k for convex polyhedra K_1 and K_4 have nonempty intersection, i.e. $\varphi^{\Omega_1^k \Omega_4^k}(v_1^{(k-1)}, v_2^{(k-1)}) < 0$ and therefore we need to include in our subproblem only quasi phi-function for polyhedra K_1 and K_4.

APPENDIX C. SEARCHING FOR FEASIBLE AUXALIRY VARIABLES IN THE FAPA ALGORITHM

On the seventh step of the COMPOLY algorithm we find values of the vector of auxiliary variables u_P, employing the FAPA algorithm. Figure C1 illustrates two cases to derive a vector of feasible parameters u_P of a separating plane: a) for two spheres S_i and S_j if $\operatorname{int} S_i \cap \operatorname{int} S_j = \varnothing$; for two convex polyhedra K_i and K_j if $\operatorname{int} S_i \cap \operatorname{int} S_j \neq \varnothing$.



(a) (b)

Fig. C1 – Illustration to 7th step at the *k-th* iteration of COMPOLY algorithm:

a) $S_i \bigcap S_j = \varnothing$; b) $\text{int}\, S_i \bigcap \text{int}\, S_j \neq \varnothing$.

For case a) we use trivial geometrical calculations to find u_P; for case b) we solve NLP subproblem (15) to find a nonnegative value of α that corresponds to the problem of searching for a nonnegative value of a quasi phi-function of two convex polyhedra K_i and K_j