A matheuristic approach with nonlinear subproblems for large-scale packing of ellipsoids

E. G. Birgin * R. D. Lobato †

September 27, 2017‡

Abstract

The problem of packing ellipsoids in the three-dimensional space is considered in the present work. The proposed approach combines heuristic techniques with the resolution of recently introduced nonlinear programming models in order to construct solutions with a large number of ellipsoids. The introduced approach is able to pack identical and non-identical ellipsoids within a variety of containers. Moreover, it allows the inclusion of additional positioning constraints. This fact makes the proposed approach suitable for constructing large-scale solutions with specific positioning constraints in which density may not be the main issue. Numerical experiments illustrate that the introduced approach delivers good quality solutions with a computational cost that scales linearly with the number of ellipsoids; and solutions with more than a million ellipsoids are exhibited.

Keywords: Packing, ellipsoids, nonlinear programming, algorithms, matheuristic.

1 Introduction

An usual way of studying material properties (from rocks to human tissues) is to assume that the material is composed by separate discrete particles, like, for example, grains. Ellipsoids are the simplest non-spherical shapes that can be considered in this case. As pointed out in [35], ellipsoids attract attention because of their shape-dependent anisotropic properties, since, in some applications such as photonic crystals, both positional and orientational order of the ellipsoidal packing are required. The problem of packing ellipsoids has a number of important applications, which include the design of high-density ceramic materials, the formation and growth of crystals [8, 33], the structure of liquids, crystals and glasses [3], the flow and compression of granular materials [14, 20, 21], the thermodynamics of liquid to crystal transition [1, 7, 32], the chromosome organization in human cell nuclei [36], and the modeling of vascular network formation [30]. See also [9, 10, 11, 12, 13, 27] and the references therein for more applications.

*Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão 1010, Cidade Universitária, 05508-090, São Paulo, SP, Brazil. E-mail: egbirgin@ime.usp.br

†Department of Applied Mathematics, Institute of Mathematics, Statistics, and Scientific Computing, State University of Campinas, Rua Sérgio Buarque de Holanda 651, 13083-859, Campinas, SP, Brazil. E-mail: lobato@ime.usp.br

‡Revision made on May 10, 2018.

The problem of packing ellipsoids has been studied in many different ways using a wide range of techniques. A few recent examples follow. In [19], the Brownian motion of isolated ellipsoidal particles in water is studied providing insights into processes that are potentially useful for understanding transport in membranes and the motions of anisotropic macromolecules. In [16], the authors study through simulations (using discrete element method) the force ratios, force network, and force probability distribution in the three-dimensional packing of fine ellipsoidal particles, as a function the size and the shape of the ellipsoids. The discrete element method is also used in [17], to study the effect of particle size and aspect ratio on packing structure of fine ellipsoids. In [16], an approach that combines computational fluid dynamics for gas phase and the discrete element method for particles is used to study flow and force structures of fine ellipsoids in gas fluidization. Effective properties of composite materials are evaluated in [38]. Periodic random packing of ellipsoids of different volume fractions and aspect ratios are build using a molecular dynamics-based method and packings with a volume fraction of up to 60% are built. In [18], dense structures of colloidal ellipsoids with a density of approx. 67% were self-assembled using direct current electric fields in conjunction with ultraviolet light. In that work, it is mentioned that studying dense packings is relevant since novel packing structures are predicted to occur at high volume fractions and crystal unit cells can contribute to a variety of applications such as, for example, structural color materials. In [37], the random packing structure of ellipsoids is studied using X-ray tomography.

In the last years, several works [4, 5, 15, 23, 24, 25, 26, 31, 34] addressed the problem of packing ellipsoids using nonlinear programming models and techniques. In [23, 24], global optimal solutions to small-sized instances (up to three ellipses or ellipsoids) were sought and local optimal solutions with up to one hundred ellipsoids were found by adding ellipsoids additively by means of a heuristic. Good quality solutions to medium- and large-sized instances were obtained in [4, 5], by seeking local minimizers of nonlinear programming models. The models proposed in [4] have a number of variables and constraints that is quadratic in the number of ellipsoids being packed; this being the main limitation for obtaining good quality solutions for instances with more than a hundred ellipsoids. Models with a number of variables and constraints that is expected to be linear with respect to the number of ellipsoids being packed were introduced in [5]. Using those models, solutions with up to five hundred ellipsoids can be obtained. Anyway, the nonlinear programming models that need to be solved are highly nonconvex and, in general, existent state-of-the-art methods are not capable of finding good quality local minimizers of instances with, say, a thousand ellipsoids. The aforementioned works considered the problem of packing a given collection of ellipsoids within a volume-minimizing container. If the number of ellipsoids m is very small, global minimizers (with a certificate of optimality) of the continuous and differentiable nonlinear programming models proposed in [4, 5], as well as the models considered in [23, 24, 25, 26, 31, 34], can be obtained considering state-of-the-art global optimization software. For medium- and large-sized instances (m up to, say, a thousand), state-of-the-art solvers for nonlinear programming may be able to find stationary points associated with “good quality” solutions to the packing problem. For instances with larger values of m, the nonconvexity of the models makes almost impracticable to find stationary points associated with reasonable solutions to the packing problem.

The problem of packing a large number of non-identical spheres with a wide range of “additional” constraints through nonlinear models and optimization techniques is in the heart of the software Packmol [29, 28]. Packmol is a software to pack molecules. The configuration of molecules that the method builds is then used as the initial configuration for performing molecular dynamics simulations. A molecule is composed by atoms. In simple terms, a molecule is represented by a set of small spheres with fixed distances among them. Thus, ultimately, Packmol simple packs spheres (with ad-hoc positioning constraints), which is a relatively simple and well-studied problem. However, the nonlinear programming problems that need to be solved are huge, highly nonlinear, and nonconvex; meaning that sate-of-the-art nonlinear programming methods are not even able to find a feasible point. Density is not an issue (in practical problems the "volume occupied by atoms in liquid water is approximately 30%"). Moreover, one of the problem constraints says that molecules must have a minimum distance among them. Otherwise, the equations that govern the dynamics simulation to be done later are not well defined. Another interesting application for the packing of ellipsoids in amorphous and not necessarily dense arrangements consists in building models of cytoplasmic solutions. Building initial configurations for molecular dynamics simulations of disordered systems is a known challenge, solved at the molecular scale by packing strategies, as that of Packmol [29]. For very large molecular systems, as a cellular environment, the packing of individual atoms becomes prohibitive. However, in a coarse-grained representation, globular proteins can be thought as spheres or ellipsoids of variable shapes and dimensions, and cellular environment consists in a crowded protein solution [22]. It can be constructed by packing ellipsoids of variable shapes, followed by fitting of the actual atomic representation of the protein structures inside the ellipsoids for subsequent simulation.

In the present work, we aim to investigate approaches that allow the possibility of representing molecules with spheres and ellipsoids within Packmol. Thus, we consider the problem of packing a large number of ellipsoids within a given container. We present a matheuristic approach based on the nonlinear programming models introduced in [4, 5]. The computational cost of the proposed method scales linearly with the number of ellipsoids and, therefore, huge instances can be considered. Moreover, the proposed approach can be applied to a variety of containers of different shapes as well as it can be used to pack identical and non-identical ellipsoids. In addition, ad-hoc positioning constraints that apply to a specific ellipsoid or a group of ellipsoids can be easily handled.

The rest of this paper is organized as follows. In Section 2, we state the problem considered in this work and introduce some notation. In Section 3, we present the model introduced in [5] to avoid the overlapping between ellipsoids. In Section 4.1, we present a simple and general algorithm to solve the problem of packing the largest possible number of ellipsoids inside a given container. In Section 4.2, we propose some strategies that can be used to compose the general algorithm. To deal with the case where the number of ellipsoids to be packed is large, we present what we call the isolation constraints in Section 4.3. These are additional constraints to the model to prevent large groups of ellipsoids from overlapping and thus reducing the total number of variables and constraints of the model. The complete nonlinear programming model and algorithm are presented in Section 5. Some implementation details are discussed in Section 6. Finally, we present numerical experiments in Section 7 and draw some conclusions in Section 8. The computer implementation of the method introduced in the present work and the solutions reported in Section 7 are freely available at `http://www.ime.usp.br/~lobato/`.

2 Problem definition and notation

We represent an ellipsoid in $\mathbb{R}^n$ by the set $\mathcal{E} = \{x \in \mathbb{R}^n \mid (x-c)^\top QP^{-1}Q^\top(x-c) \le 1\}$, where $c \in \mathbb{R}^n$ is the center of the ellipsoid, $Q \in \mathbb{R}^{n\times n}$ is an orthogonal matrix that determine the principal axes of the ellipsoid, and $P \in \mathbb{R}^{n\times n}$ is a positive definite diagonal matrix so that the eigenvalues of $P^{\frac{1}{2}}$ are the lengths of the semi-axes of the ellipsoid. We denote by $\text{int}(\mathcal{E})$ the interior of $\mathcal{E}$, i.e., $\text{int}(\mathcal{E}) = \{x \in \mathbb{R}^n \mid (x-c)^\top QP^{-1}Q^\top(x-c) < 1\}$. Also, we denote by $\partial\mathcal{E}$ the frontier of $\mathcal{E}$, i.e., $\partial\mathcal{E} = \{x \in \mathbb{R}^n \mid (x-c)^\top QP^{-1}Q^\top(x-c) = 1\}$. The k-th standard basis vector (i.e., the vector whose k-th components is equal to one and have all the other components equal to zero) is denoted by e_k. The largest eigenvalue of a matrix M is denoted by $\lambda_{\max}(M)$.

In this paper, we consider the problem of packing the maximum number of ellipsoids within a given container. The ellipsoids must not overlap each other and they must be entirely inside the container. Formally, given a set $\mathcal{C} \subseteq \mathbb{R}^n$, which we call the container, and a sequence of $(n\times n)$-dimensional positive definite diagonal matrices $\{P_i\}_{i=1}^{\infty}$, the objective is to find the maximum nonnegative number m^* and the ellipsoids $\mathcal{E}_i = \{x \in \mathbb{R}^n \mid (x-c_i)^\top Q_i P_i^{-1} Q_i^\top (x-c_i) \le 1\}$, for $i \in I = \{1,\dots,m^*\}$, in such a way that

1. $\text{int}(\mathcal{E}_i) \cap \text{int}(\mathcal{E}_j) = \emptyset$ for all $i, j \in I$ with $i \neq j$;
2. $\mathcal{E}_i \subseteq \mathcal{C}$ for all $i \in I$.

By finding an ellipsoid $\mathcal{E}_i$ we mean determining a vector $c_i \in \mathbb{R}^n$ and an orthogonal matrix $Q_i \in \mathbb{R}^{n\times n}$. If $P_i = P$ for all i then the problem reduces to the problem of packing as many identical ellipsoids (with semi-axis lengths given by the square roots of the diagonal entries of P) as possible within the container $\mathcal{C}$.

3 Non-overlapping and containment models

Nonlinear programming models for the non-overlapping of ellipsoids were introduced in [4, 5, 23]. Since the models presented in [4] and [5] are the foundation of the methodology proposed in this paper, we briefly summarize them in Section 3.1. Besides avoiding the overlap between the ellipsoids, it is required the ellipsoids to be inside a given container. In Section 3.2, it is presented a model to include an ellipsoid within a half-space, which was introduced in [4]. This model will be used to build a cuboidal container and also to construct the so called *isolation constraints* that will be presented in Section 4.3.

3.1 Non-overlapping model

Consider the ellipsoids $\mathcal{E}_i$ and $\mathcal{E}_j$ in $\mathbb{R}^n$ defined as

$$\mathcal{E}_i = \{x \in \mathbb{R}^n \mid (x-c_i)^\top Q_i P_i^{-1} Q_i^\top (x-c_i) \le 1\}$$

and

$$\mathcal{E}_j = \{x \in \mathbb{R}^n \mid (x-c_j)^\top Q_j P_j^{-1} Q_j^\top (x-c_j) \le 1\},$$

where c_i and c_j in $\mathbb{R}^n$ are their centers and Q_i and Q_j are orthogonal matrices in $\mathbb{R}^{n\times n}$ that determine their orientation. For example, for $n = 2$, we can represent Q_i as

$$Q_i = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix},$$

whereas, for $n = 3$, we can represent Q_i as

$$Q_i = \begin{pmatrix} \cos\theta_i\cos\psi_i & \sin\phi_i\sin\theta_i\cos\psi_i - \cos\phi_i\sin\psi_i & \sin\phi_i\sin\psi_i + \cos\phi_i\sin\theta_i\cos\psi_i \\ \cos\theta_i\sin\psi_i & \cos\phi_i\cos\psi_i + \sin\phi_i\sin\theta_i\sin\psi_i & \cos\phi_i\sin\theta_i\sin\psi_i - \sin\phi_i\cos\psi_i \\ -\sin\theta_i & \sin\phi_i\cos\theta_i & \cos\phi_i\cos\theta_i \end{pmatrix}.$$

The parameters θ_i (when $n = 2$) and θ_i, ϕ_i, and ψ_i (when $n = 3$) are called "rotation angles" of the ellipsoid. We denote by $\Omega_i \in \mathbb{R}^q$ the vector of rotation angles of the i-th ellipsoid ($q = 1$ when $n = 2$ and $q = 3$ when $n = 3$). The idea of the model presented in [4] is to transform one of the ellipsoids into a ball so that the problem of avoiding the overlap between two ellipsoids becomes the problem of preventing the overlap between a ball and an ellipsoid. By letting $T_{ij} : \mathbb{R}^n \to \mathbb{R}^n$ be the linear transformation defined by

$$T_{ij}(x) = P_i^{-\frac{1}{2}} Q_i^\top (x - c_j) \tag{1}$$

and applying this transformation to the ellipsoids $\mathcal{E}_i$ and $\mathcal{E}_j$, we obtain

$$\mathcal{E}_i^{ij} = \left\{ x \in \mathbb{R}^n \mid \left[x - P_i^{-\frac{1}{2}} Q_i^\top (c_i - c_j) \right]^\top \left[x - P_i^{-\frac{1}{2}} Q_i^\top (c_i - c_j) \right] \leq 1 \right\}$$

and

$$\mathcal{E}_j^{ij} = \{ x \in \mathbb{R}^n \mid x^\top S_{ij} x \leq 1 \},$$

respectively, where

$$S_{ij} = P_i^{\frac{1}{2}} Q_i^\top Q_j P_j^{-1} Q_j^\top Q_i P_i^{\frac{1}{2}}.$$

The result is that $\mathcal{E}_i^{ij}$ is a unit-radius ball and $\mathcal{E}_j^{ij}$ is an ellipsoid, so that avoiding the overlap between $\mathcal{E}_i$ and $\mathcal{E}_j$ is equivalent to requiring that the ball $\mathcal{E}_i^{ij}$ and the ellipsoid $\mathcal{E}_j^{ij}$ do not overlap (see Lemma 3.1 in [5]). Since $\mathcal{E}_i^{ij}$ is a unit-radius ball, it does not overlap the ellipsoid $\mathcal{E}_j^{ij}$ if and only if its center is outside of the ellipsoid and its distance to the frontier of the ellipsoid is at least one. Let c_i^{ij} denote the center of the ball $\mathcal{E}_i^{ij}$. By Proposition 4.1 in [4], if $c_i^{ij} \notin \operatorname{int}(\mathcal{E}_j^{ij})$, then we can write

$$c_i^{ij} = x_{ij} + \mu_{ij} S_{ij} x_{ij} \tag{2}$$

for a unique $x_{ij} \in \mathbb{R}^n$ in the frontier of $\mathcal{E}_j^{ij}$ and a unique scalar $\mu_{ij} \geq 0$. Moreover, $x_{ij} \in \mathbb{R}^n$ is the projection of c_i^{ij} onto $\mathcal{E}_j^{ij}$. By Proposition 4.2 in [4], if c_i^{ij} can be written as in (2) for some $x_{ij} \in \partial\mathcal{E}_j^{ij}$ and some $\mu_{ij} \geq 0$, then $c_i^{ij} \notin \operatorname{int}(\mathcal{E}_j^{ij})$ and x_{ij} is the projection of c_i^{ij} onto $\mathcal{E}_j^{ij}$. Hence, the distance between c_i^{ij} and $\mathcal{E}_j^{ij}$ is given by $\|c_i^{ij} - x_{ij}\| = \|\mu_{ij} S_{ij} x_{ij}\|$, which must be at least one for $\mathcal{E}_i^{ij}$ not to overlap with $\mathcal{E}_j^{ij}$. Hence, we obtain the following model for the non-overlapping of ellipsoids:

$$
\begin{aligned}
x_{ij}^\top S_{ij} x_{ij} &= 1, && \forall i, j \in I \text{ such that } i < j && (3)\\
\mu_{ij}^2 \left\| S_{ij} x_{ij} \right\|_2^2 &\geq 1, && \forall i, j \in I \text{ such that } i < j && (4)\\
P_i^{-\frac{1}{2}} Q_i^\top (c_i - c_j) &= x_{ij} + \mu_{ij} S_{ij} x_{ij}, && \forall i, j \in I \text{ such that } i < j && (5)\\
\mu_{ij} &\geq 0, && \forall i, j \in I \text{ such that } i < j && (6)
\end{aligned}
$$

where $I = \{1, \ldots, m\}$ is the set of indices of the ellipsoids being packed.

Since this model has a quadratic number of variables and constraints on the number of ellipsoids to be packed, it becomes rapidly hard to be solved as the number of ellipsoids grows. In order to alleviate this complexity, a model with a linear number of variables and constraints was introduced in [5].

To reduce the number of constraints, the constraints of model (3)–(6) are first replaced by their respective squared infeasibility measures

$$o(c_i, c_j, \Omega_i, \Omega_j, x_{ij}, \mu_{ij}; P_i, P_j) = 0, \quad \forall i, j \in I \text{ such that } i < j,$$

where

$$
\begin{aligned}
o(c_i, c_j, \Omega_i, \Omega_j, x_{ij}, \mu_{ij}; P_i, P_j) = \; & \left(x_{ij}^\top \big(P_i^{-\frac{1}{2}} Q_i^\top (c_i - c_j) - x_{ij} \big) - \mu_{ij} \right)^2 + \\
& \max\{0, \epsilon_{ij} - \mu_{ij}\}^2 + \left\| x_{ij} + \mu_{ij} S_{ij} x_{ij} - P_i^{-\frac{1}{2}} Q_i^\top (c_i - c_j) \right\|_2^2 + \\
& \max\left\{ 0, 1 - \left\| P_i^{-\frac{1}{2}} Q_i^\top (c_i - c_j) - x_{ij} \right\|_2^2 \right\}^2,
\end{aligned}
$$

and then combined into $m - 1$ constraints as follows:

$$\sum_{j=i+1}^{m} o(c_i, c_j, \Omega_i, \Omega_j, x_{ij}, \mu_{ij}; P_i, P_j) = 0, \quad \forall i \in I \setminus \{m\}.$$

To reduce the number of variables, for each $i, j \in I$ such that $i < j$, the variables x_{ij} and μ_{ij} are replaced with $\mathcal{X}_{ij} \equiv \mathcal{X}(c_i, c_j, \Omega_i, \Omega_j; P_i, P_j)$ and $\mathcal{U}_{ij} \equiv \mathcal{U}(c_i, c_j, \Omega_i, \Omega_j; P_i, P_j)$, respectively, where $\mathcal{X}_{ij}$ is a solution to the problem

$$
\begin{aligned}
\underset{x}{\text{minimize}} \quad & \left\| x - c_i^{ij} \right\|_2^2 \\
\text{subject to} \quad & x^\top S_{ij} x = 1,
\end{aligned}
$$

and $\mathcal{U}_{ij}$ is the corresponding Lagrange multiplier. Therefore, the non-overlapping constraints can be finally written as

$$\sum_{j=i+1}^{m} f(c_i, c_j, \Omega_i, \Omega_j; P_i, P_j) = 0, \quad \forall i \in I \setminus \{m\}, \tag{7}$$

where

$$f(c_i, c_j, \Omega_i, \Omega_j; P_i, P_j) = \max\left\{0, 1 - \left\| P_i^{-\frac{1}{2}} Q_i^\top (c_i - c_j) - \mathcal{X}(c_i, c_j, \Omega_i, \Omega_j; P_i, P_j) \right\|_2^2 \right\}^2 + \max\{0, \epsilon_{ij} - \mathcal{U}(c_i, c_j, \Omega_i, \Omega_j; P_i, P_j)\}^2 \tag{8}$$

and

$$\epsilon_{ij} = \epsilon(P_i, P_j) = \lambda_{\min}(P_i^{-1})\lambda_{\min}(P_i^{\frac{1}{2}})\lambda_{\min}(P_j)\lambda_{\min}(P_j^{-\frac{1}{2}}) > 0;$$

see [4, Prop. 4.3] for details. It is worth noticing that, if the ellipsoids $\mathcal{E}_i$ and $\mathcal{E}_j$ are far from each other then $f(c_i, c_j, \Omega_i, \Omega_j; P_i, P_j)$ is null and, thus, the quantities $\mathcal{X}(c_i, c_j, \Omega_i, \Omega_j; P_i, P_j)$ and $\mathcal{U}(c_i, c_j, \Omega_i, \Omega_j; P_i, P_j)$ do not need to be computed.

3.2 Containment model

The idea to include an ellipsoid within a half-space is similar to that of avoiding the overlap between ellipsoids. A transformation is applied to the ellipsoid so that it becomes a ball. The same transformation is then applied to the half-space, which transforms it into another half-space. The problem of including an ellipsoid within a half-space then becomes the equivalent problem of including a ball within a half-space.

Consider the ellipsoid $\mathcal{E}_i = \{x \in \mathbb{R}^n \mid (x - c_i)^\top Q_i P_i^{-1} Q_i^\top (x - c_i) \leq 1\}$, where $c_i \in \mathbb{R}^n$, $Q_i \in \mathbb{R}^{n \times n}$ is orthogonal, and $P_i \in \mathbb{R}^{n \times n}$ is positive definite and diagonal. Let $T_i : \mathbb{R}^n \to \mathbb{R}^n$ be the linear transformation defined by

$$T_i(x) = P_i^{-\frac{1}{2}} Q_i^\top x. \tag{9}$$

By applying transformation T_i to $\mathcal{E}_i$, we obtain the unit-radius ball

$$\mathcal{E}_{ii} = \{x \in \mathbb{R}^n \mid (x - P_i^{-\frac{1}{2}} Q_i^\top c_i)^\top (x - P_i^{-\frac{1}{2}} Q_i^\top c_i) \leq 1\}.$$

Now, consider the half-space $\mathcal{H} = \{x \in \mathbb{R}^n \mid w^\top x \leq s\}$, where $w \in \mathbb{R}^n$, $w \neq 0$, and $s \in \mathbb{R}$, and let $\mathcal{H}_i$ be the set obtained when transformation T_i is applied to the half-space $\mathcal{H}$, i.e.,

$$\mathcal{H}_i = \{x \in \mathbb{R}^n \mid w^\top Q_i P_i^{\frac{1}{2}} x \leq s\}.$$

Requiring $\mathcal{E}_i \subseteq \mathcal{H}$ is equivalent to requiring $\mathcal{E}_{ii} \subseteq \mathcal{H}_i$. For $\mathcal{E}_{ii}$ to be contained in $\mathcal{H}_i$, the center c_{ii} of $\mathcal{E}_{ii}$ must belong to $\mathcal{H}_i$, and the distance between c_{ii} and the frontier $\partial\mathcal{H}_i = \{x \in \mathbb{R}^n \mid w^\top Q_i P_i^{\frac{1}{2}} x = s\}$ of $\mathcal{H}_i$ must be at least one (the radius of the ball $\mathcal{E}_{ii}$). Since the distance $d(c_{ii}, \partial\mathcal{H}_i)$ from c_{ii} to the frontier of $\mathcal{H}_i$ is given by

$$d(c_{ii}, \partial\mathcal{H}_i) = \frac{|w^\top Q_i P_i^{\frac{1}{2}} c_{ii} - s|}{\left\| P_i^{\frac{1}{2}} Q_i^\top w \right\|_2},$$

these conditions are therefore

$$\frac{(w^\top Q_i P_i^{\frac{1}{2}} c_{ii} - s)^2}{\left\| P_i^{\frac{1}{2}} Q_i^\top w \right\|_2^2} \geq 1 \quad \text{and} \quad w^\top Q_i P_i^{\frac{1}{2}} c_{ii} \leq s. \tag{10}$$

Since $c_{ii} = P_i^{-\frac{1}{2}} Q_i^\top c_i$, conditions (10) can be equivalently written as

$$\frac{\left(w^\top c_i - s\right)^2}{\left\| P_i^{\frac{1}{2}} Q_i^\top w \right\|_2^2} \geq 1 \quad \text{and} \quad w^\top c_i \leq s.$$

4 Incremental packing of ellipsoids

In this section, the main ingredients of the matheuristic approach for packing ellipsoids are described in detail. The model algorithm is given in Section 4.1; the nonlinear strategy used for packing a set of new ellipsoids is described in Section 4.2; and the so-called isolation constraints, that aims to reduce the complexity of the nonlinear subproblems, are described in Section 4.3. A similar heuristic was considered in [23, 24], where a constructive approach with nonlinear programming subproblems and isolation constraints was also considered. However, it is worth noticing that the nonlinear models, the constructive way of adding new ellipsoids, and the isolation constraints presented here are different from the ones considered in [23, 24].

4.1 Model algorithm

Briefly, the algorithm to pack ellipsoids inside a given container is as follows. At each iteration, a certain number of ellipsoids (that were packed in previous iterations) are already arranged within the container. Once these ellipsoids are packed, they are fixed in their positions (their centers and rotations are fixed). Then, a new group of ellipsoids is packed, so that they do not overlap each other and do not overlap with the ellipsoids already fixed.

At the k-th iteration of the algorithm, let $\mathcal{F}_k = \{1, \dots, m_{k-1}\}$ be the set formed by the indices of the ellipsoids already packed and fixed in their positions and let $\mathcal{N}_k = \{m_{k-1} + 1, \dots, m_k\}$ be the set of indices of the new ellipsoids. In order to pack the new ellipsoids, we must ensure that (i) they are arranged inside the container, (ii) do not overlap each other, and (iii) do not overlap with the ellipsoids already fixed.

So, considering a container $\mathcal{C} \subseteq \mathbb{R}^n$ and the models presented in Sections 3.1 and 3.2, at the k-th iteration of the algorithm, we must find a solution to the feasibility problem given by

$$\mathcal{E}_i \subseteq \mathcal{C}, \quad \forall i \in \mathcal{N}_k, \tag{11}$$

$$\sum_{\substack{j \in \mathcal{N}_k \\ j > i}} f(c_i, c_j, \Omega_i, \Omega_j; P_i, P_j) = 0, \quad \forall i \in \mathcal{N}_k \cup \mathcal{F}_k, \tag{12}$$

where f is as defined in (8). The variables of this model are $c_i \in \mathbb{R}^n$ and $Q_i \in \mathbb{R}^{n \times n}$ for each $i \in \mathcal{N}_k$. Notice that c_i and Q_i for each $i \in \mathcal{F}_k$ are constants, since the ellipsoids in $\mathcal{F}_k$ have already been fixed.

4.2 Packing strategy

The algorithm described in the last section requires the new ellipsoids to be inside the container, not to overlap each other, and not to overlap with the ellipsoids already packed. However, those constraints describe a feasibility problem and they do not specify how the new ellipsoids should be packed. Since the goal is to pack as many ellipsoids as possible, the ellipsoids should stay tightly grouped within the container. An attempt to achieve this result is to minimize, in some sense, the heights of the ellipsoids to be packed. The idea is that the new ellipsoids become in contact with other ellipsoids already packed, so that the ellipsoids are well packed inside the container. Given an ellipsoid $\mathcal{E}$, we define two heights associated with it: the lower and the upper height. The lower height is defined as $\min\{x_n \mid x \in \mathcal{E}\}$ and the upper height is defined as $\max\{x_n \mid x \in \mathcal{E}\}$, where x_n is the n-th component of x. Since the goal is to minimize these heights, we need a simple way to model them. One way of doing this is to model the upper and lower heights of an ellipsoid by supporting hyperplanes. The idea is to consider hyperplanes that support the ellipsoid precisely at the points that realize the lower and upper heights.

Consider the half-space $\mathcal{S} = \{x \in \mathbb{R}^n \mid w^\top x \leq s\}$, where $w \in \mathbb{R}^n$ and $s \in \mathbb{R}$, and the ellipsoid $\mathcal{E}_i = \{x \in \mathbb{R}^n \mid (x - c_i)^\top Q_i P_i^{-1} Q_i^\top (x - c_i) \leq 1\}$, where $c_i \in \mathbb{R}^n$, $Q_i \in \mathbb{R}^{n\times n}$ is orthogonal, and $P_i \in \mathbb{R}^{n\times n}$ is diagonal and positive definite. We saw in Section 3.2 that, in order to ensure that the ellipsoid be contained in the half-space $\mathcal{S}$, we can simply require the center of the ellipsoid to belong to that half-space and the distance between the center of the ball $\mathcal{E}_{ii}$ and the frontier of the half-space $\mathcal{S}_i$, obtained by transformation T_i defined in (9), be at least one. To ensure that $\partial\mathcal{S}$ supports the ellipsoid $\mathcal{E}_i$, we can just change the minimum distance condition and require it to be exactly one. Therefore, the conditions

$$\frac{\left(w^\top c_i - s\right)^2}{\left\|P_i^{\frac{1}{2}} Q_i^\top w\right\|_2^2} = 1 \quad \text{and} \quad w^\top c_i \leq s \tag{13}$$

guarantee that the hyperplane $\partial\mathcal{S}$ supports the ellipsoid $\mathcal{E}_i$. Moreover, if we take $w = e_n$, the n-th standard basis vector, then $\partial\mathcal{S}$ will support the ellipsoid $\mathcal{E}_i$ at the point $\arg\max\{x_n \mid x \in \mathcal{E}_i\}$, and we will necessarily have $s = \max\{x_n \mid x \in \mathcal{E}_i\}$. If we take $w = -e_n$, then $\partial\mathcal{S}$ will support the ellipsoid $\mathcal{E}_i$ at the point $\arg\min\{x_n \mid x \in \mathcal{E}_i\}$, and we will have $s = -\min\{x_n \mid x \in \mathcal{E}_i\}$.

In order to minimize the upper height of the ellipsoid, we can then consider the problem of minimizing s subject to (11,12,13) with $w = e_n$ in (13). In an analogous way, in order to minimize the lower height of the ellipsoid, it is enough to consider the problem of minimizing $-s$ subject to (11,12,13) with $w = -e_n$ in (13).

As we will see in Section 7, experiments in the three-dimensional space show that the packed ellipsoid tends to have its semi-major axis parallel to the upper plane when its upper height is minimized (the ellipsoid is “standing”). On the other hand, when the lower height is minimized, the tendency is that the semi-minor axis remains parallel to the upper plane (the ellipsoid is “lying”). To avoid this kind of behavior, which can result in poor quality solutions, we can consider the minimization of a convex combination of the lower and upper heights.

Let s^i_{inf} and s^i_{sup} denote the lower and upper heights of ellipsoid $\mathcal{E}_i$, respectively. For a given $\xi \in [0,1]$, we define an intermediate height as $\xi s^i_{\text{inf}} + (1-\xi)s^i_{\text{sup}}$. Since $[c_i]_n$, the n-th component of the center of the ellipsoid, is equal to $\frac{1}{2}(s^i_{\text{inf}} + s^i_{\text{sup}})$, we can write $s^i_{\text{inf}} = 2[c_i]_n - s^i_{\text{sup}}$. Then,

$\xi s^i_{\text{inf}} + (1-\xi)s^i_{\text{sup}} = 2\xi[c_i]_n + (1-2\xi)s^i_{\text{sup}}$. Hence, to minimize the intermediate height, we can add the variable s^i_{sup} and the constraints

$$\frac{\left(e_n^\top c_i - s^i_{\text{sup}}\right)^2}{\left\| P_i^{\frac{1}{2}} Q_i^\top e_n \right\|_2^2} = 1 \quad \text{and} \quad e_n^\top c_i \leq s^i_{\text{sup}} \tag{14}$$

to the model. For $\xi = 1$, we have the minimization of the upper height of the ellipsoid being packed. For $\xi = 0$, we have the minimization of the lower height of the ellipsoid. For $\xi = \frac{1}{2}$, we have the minimization of $[c_i]_n$, the n-th component of the center of the ellipsoid (which we call the middle height). Notice that when $\xi = \frac{1}{2}$, the variable s^i_{sup} and the constraints (14) are not necessary.

When $|\mathcal{N}_k| > 1$, i.e., when there are more than one ellipsoid being packed at iteration k, we can minimize the sum of the heights of the ellipsoids:

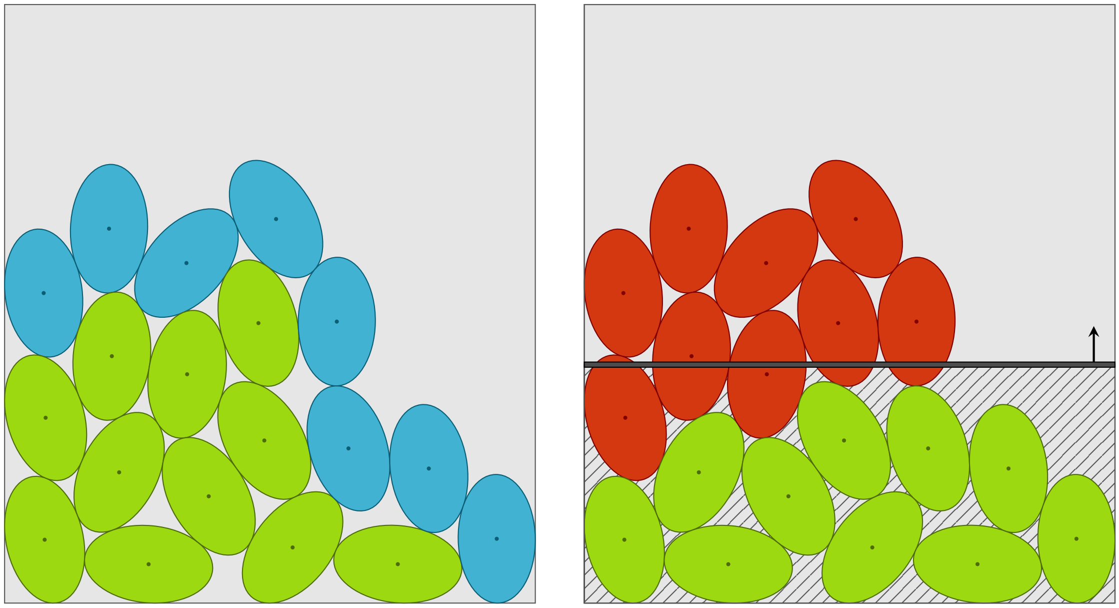
$$\sum_{i \in \mathcal{N}_k} 2\xi[c_i]_n + (1-2\xi)s^i_{\text{sup}}.$$

4.3 The isolation constraints

In addition to ensuring that the new ellipsoids (to be packed) do not overlap each other, we have to make sure that these ellipsoids do not overlap with the ellipsoids previously packed. Thus, the number of pairs of ellipsoids whose overlapping should be avoided grows as the number of previously packed ellipsoids increases. This makes the complexity of the evaluation of the constraints of each subproblem to increase, making each subproblem more and more difficult to be solved.

On the other hand, assuming that a sufficiently large number of ellipsoids has been packed, it is expected that there is no possibility for the new ellipsoids to be in contact with most of the fixed ellipsoids, since the latter should be surrounded by several other ellipsoids. Let $\mathcal{N}$ be the set of the new ellipsoids and $\tilde{\mathcal{F}}$ be the set formed by the ellipsoids already packed and that cannot touch the new ellipsoids in a feasible solution. By adding constraints to ensure that the ellipsoids in $\mathcal{N}$ are sufficiently distant from the ellipsoids in $\tilde{\mathcal{F}}$, we can remove the non-overlapping constraints between these two groups of ellipsoids. For this change in the model to have the desired effect (making the subproblems simpler), it is clear that the new constraints should be "easier" than the original non-overlapping constraints. By easy constraints we mean constraints that are smaller in number, defined by simpler functions, and/or involve a small number of variables. We will call these new constraints the isolation constraints. We say that an ellipsoid is isolated if it is possible to easily infer that the isolation constraints ensure that the new ellipsoids do not overlap with the ellipsoid in question.

We present Figure 4.1 to illustrate the isolation of ellipsoids. Consider the packing of ellipses inside a rectangle. In Figure 4.1(a), it is shown some ellipses already packed inside the rectangle. Now consider the problem of packing a new ellipse. Due to the non-overlapping constraints, this new ellipse could touch only the blue ellipses. The set $\tilde{\mathcal{F}}$ is formed by the green ellipses in Figure 4.1(a). Now, consider the isolation constraint that requires the new ellipse to lie above



(a) (b)

Figure 4.1: Illustration of the isolation constraints. (a) Ellipses already packed and fixed in their positions. A new ellipse to be packed should not overlap any of them. The blue ellipses are the ones that could possibly be touched by a new ellipse to be packed. (b) The isolation constraint requires a new ellipse to be packed to lie above the highlighted line. This implies that only the red ellipses need to be considered in the non-overlapping model; thus reducing the model complexity.

the line illustrated in Figure 4.1(b). Thus, the green ellipses are isolated and the original non-overlapping constraints associated with these ellipses can be removed.

Because of the simplicity of the isolation constraints, these constraints may isolate ellipsoids that could touch the new ellipsoids in a feasible solution (as it is the case for some green ellipses in Figure 4.1(b)). Anyway, it is important to point out that the isolation constraints ensure that the new ellipsoids do not overlap with the isolated ellipsoids. Even if the isolation constraints are not able to isolate all ellipsoids of $\tilde{\mathcal{F}}$, the expectation is that most of these ellipsoids are isolated and the subproblems have very low numbers of constraints and variables.

5 Complete model and algorithm

Consider the case where the container $\mathcal{C}$ is the following hypercube with side length l:

$$\mathcal{C} = \{x \in \mathbb{R}^n \mid -l \leq 2x_i \leq l, \forall i \in \{1, \ldots, n\}\}.$$

This hypercube can be modeled by $2n$ half-spaces, each one corresponding to a different side of the hypercube. Each side of the hypercube can then be modeled according to the model presented in Section 3.2. Hence, the inclusion of ellipsoid $\mathcal{E}_i = \{x \in \mathbb{R}^n \mid (x - c_i)^\top Q_i P_i^{-1} Q_i^\top (x - c_i) \leq 1\}$

within $\mathcal{C}$ can be modeled by the following constraints:

$$\frac{\left(\xi e_\ell^\top c_i - l/2\right)^2}{\left\| P_i^{\frac{1}{2}} Q_i^\top e_\ell \right\|_2^2} \geq 1, \quad \forall \ell \in \{1, \ldots, n\}, \forall \xi \in \{-1, 1\},$$
$$\xi e_\ell^\top c_i \leq l/2, \; \forall \ell \in \{1, \ldots, n\}, \forall \xi \in \{-1, 1\}.$$

In our experiments, we considered two types of isolation constraints. The first one constrains the new ellipsoids to remain within a certain hyperrectangle $\mathcal{R}$ centered at $u \in \mathbb{R}^n$ and whose sides have length $s > 0$, with the exception of the side along the n-th dimension, which has infinity length:

$$\mathcal{R} = \{x \in \mathbb{R}^n \mid -s/2 \leq x_i - u_i \leq s/2, \forall i \in \{1, \ldots, n-1\}\}. \tag{15}$$

Similarly to the hypercube model, the inclusion of ellipsoid $\mathcal{E}_i$ within $\mathcal{R}$ can be modeled as:

$$\frac{\left(\xi e_\ell^\top (c_i - u) - s/2\right)^2}{\left\| P_i^{\frac{1}{2}} Q_i^\top e_\ell \right\|_2^2} \geq 1, \quad \forall \ell \in \{1, \ldots, n-1\}, \forall \xi \in \{-1, 1\},$$
$$\xi e_\ell^\top (c_i - u) \leq s/2, \; \forall \ell \in \{1, \ldots, n-1\}, \forall \xi \in \{-1, 1\}.$$

The second type of isolation constraint requires the new ellipsoids to lie within the following half-space $\mathcal{H}$:

$$\mathcal{H} = \{x \in \mathbb{R}^n \mid x_n \geq h\}, \tag{16}$$

where $h \in \mathbb{R}$. Therefore, the inclusion of ellipsoid $\mathcal{E}_i$ within $\mathcal{H}$ can be modeled as:

$$\frac{\left(e_n^\top c_i - h\right)^2}{\left\| P_i^{\frac{1}{2}} Q_i^\top e_n \right\|_2^2} \geq 1 \;\text{ and }\; e_n^\top c_i \geq h.$$

Finally, the non-overlapping can be modeled as in (7) and the upper height of ellipsoid $\mathcal{E}_i$ as in (14).

Now, consider an iteration k of the algorithm. Let $\mathcal{F}_k$ be the set of indices of the ellipsoids packed in previous iterations, $\mathcal{N}_k$ be the set of indices of the ellipsoids that must be packed at this iteration, and $\bar{\mathcal{F}}_k \subseteq \mathcal{F}_k$ be the set of indices of fixed ellipsoids that should be considered in the non-overlapping constraints. After determining the isolation constraints (parameters $s > 0$, $u \in \mathbb{R}^n$, and $h \in \mathbb{R}$) and, consequently, the set $\bar{\mathcal{F}}_k$, the problem that must be solved at this iteration is the following:

$$\text{minimize} \quad \sum_{i \in \mathcal{N}_k} 2\xi [c_i]_n + (1 - 2\xi) s_{\text{sup}}^i \tag{17}$$

$$\text{subject to} \quad \sum_{\substack{j \in \mathcal{N}_k \\ j > i}} f(c_i, c_j, \Omega_i, \Omega_j; P_i, P_j) = 0, \quad \forall i \in \mathcal{N}_k \cup \bar{\mathcal{F}}_k, \tag{18}$$

$$\frac{\left(\xi e_\ell^\top c_i - l/2\right)^2}{\left\|P_i^{\frac{1}{2}} Q_i^\top e_\ell\right\|_2^2} \geq 1, \quad \forall i \in \mathcal{N}_k, \forall \ell \in \{1,\dots,n\}, \forall \xi \in \{-1,1\}, \tag{19}$$

$$\xi e_\ell^\top c_i \leq l/2, \quad \forall i \in \mathcal{N}_k, \forall \ell \in \{1,\dots,n\}, \forall \xi \in \{-1,1\}, \tag{20}$$

$$\frac{\left(\xi e_\ell^\top (c_i - u) - s/2\right)^2}{\left\|P_i^{\frac{1}{2}} Q_i^\top e_\ell\right\|_2^2} \geq 1, \quad \forall i \in \mathcal{N}_k, \forall \ell \in \{1,\dots,n-1\}, \forall \xi \in \{-1,1\}, \tag{21}$$

$$\xi e_\ell^\top (c_i - u) \leq s/2, \quad \forall i \in \mathcal{N}_k, \forall \ell \in \{1,\dots,n-1\}, \forall \xi \in \{-1,1\}, \tag{22}$$

$$\frac{\left(e_n^\top c_i - h\right)^2}{\left\|P_i^{\frac{1}{2}} Q_i^\top e_n\right\|_2^2} \geq 1, \quad \forall i \in \mathcal{N}_k, \tag{23}$$

$$e_n^\top c_i \geq h, \quad \forall i \in \mathcal{N}_k, \tag{24}$$

$$\frac{\left(e_n^\top c_i - s_{\text{sup}}^i\right)^2}{\left\|P_i^{\frac{1}{2}} Q_i^\top e_n\right\|_2^2} = 1, \quad \forall i \in \mathcal{N}_k, \tag{25}$$

$$e_n^\top c_i \leq s_{\text{sup}}^i, \forall i \in \mathcal{N}_k. \tag{26}$$

Considering that the problem (17)–(26) may be infeasible or that a local optimization solver may fail in finding a feasible point depending on the initial guess, we apply a multi-start strategy starting up to τ times from different initial guesses. The algorithm stops when, at a given iteration k, it is not possible to solve the problem (17)–(26) within τ trials. Therefore, we can summarize the algorithm as follows:

Algorithm 1.

Input: The container $\mathcal{C}$ and the lengths of the semi-axes of the ellipsoids given by the matrices $\{P_i\}_{i=1}^{\infty}$.

Output: m^* (the number of ellipsoids packed) and Q_i and c_i for $i \in \{1,\dots,m^*\}$.

Step 1. Let $k \leftarrow 0$.

Step 2. Let $k \leftarrow k+1$ and $t \leftarrow 0$.

Step 3. Let $t \leftarrow t+1$. If $t > \tau$, stop.

Step 3.1. Determine the set $\mathcal{N}_k$.

Step 3.2. Determine the isolation constraints.

Step 3.3. Determine the set $\bar{\mathcal{F}}_k$.

Step 3.4. Determine the initial solution.

Step 3.5. Try to solve the subproblem (17)–(26).

Step 3.6. Analyze the solution found.

Step 4. If the subproblem was solved, go to Step 2. Otherwise, go to Step 3.

6 Implementation details

6.1 Determining the isolation constraints and the set $\bar{\mathcal{F}}_k$

The hyperrectangle $\mathcal{R}$ is defined by $u \in \mathbb{R}^n$ and $s > 0$. The parameter $s > 0$ can be fixed since the beginning of the algorithm, but u must vary at each iteration of Step 3 of Algorithm 1 so that we can fill up the whole container with ellipsoids. We decided to choose each coordinate of u uniformly random on the interval $[-l/2, l/2]$ at Step 3.2. Once u is determined, we compute the set $\bar{\mathcal{F}}_k^0$, which will be used to determine the second type of isolation constraints (constraints (23) and (23)). This is the set of indices of ellipsoids that were packed in previous iterations of the algorithm and that could perhaps overlap with an ellipsoid that would be contained in $\mathcal{R}$. Ideally, $\bar{\mathcal{F}}_k^0$ should be the set

$$\{i \in \mathcal{F}_k \mid \mathcal{E}_i \cap \operatorname{int}(\mathcal{R}) \neq \emptyset\}. \tag{27}$$

But since it may be computationally costly to find the set (27), we check for sufficient conditions that guarantee that $\mathcal{E}_i \cap \operatorname{int}(\mathcal{R}) = \emptyset$. The set $\bar{\mathcal{F}}_k^0$ will then be formed by indices $i \in \mathcal{F}_k$ for which it was not possible show that $\mathcal{E}_i \cap \operatorname{int}(\mathcal{R}) = \emptyset$. Hence, $\bar{\mathcal{F}}_k^0$ will be a (potentially proper) superset of (27).

Let a_i denote the largest semi-axis length of ellipsoid $\mathcal{E}_i$, i.e., $a_i = \lambda_{\max}(P_i^{\frac{1}{2}})$. Let $\mathcal{B}_i$ be the minimal bounding sphere of $\mathcal{E}_i$, i.e.,

$$\mathcal{B}_i = \{x \in \mathbb{R}^n \mid (x - c_i)^\top (x - c_i) \leq a_i^2\}.$$

It is easy to verify whether $\mathcal{B}_i \cap \operatorname{int}(\mathcal{R}) = \emptyset$. And if $\mathcal{B}_i \cap \operatorname{int}(\mathcal{R}) = \emptyset$, then $\mathcal{E}_i \cap \operatorname{int}(\mathcal{R}) = \emptyset$. It may happen that $\mathcal{E}_i \cap \operatorname{int}(\mathcal{R}) = \emptyset$ but $\mathcal{B}_i \cap \operatorname{int}(\mathcal{R}) \neq \emptyset$. In this case, we verify whether there exist $\xi \in \{-1, 1\}$ and $\ell \in \{1, \dots, n-1\}$ such that

$$\frac{\left(\xi e_\ell^\top (c_i - u) - s/2\right)^2}{\left\| P_i^{\frac{1}{2}} Q_i^\top e_\ell \right\|_2^2} \geq 1 \quad \text{and} \quad \xi e_\ell^\top (c_i - u) \geq s/2. \tag{28}$$

If (28) is verified for some $\xi \in \{-1, 1\}$ and $\ell \in \{1, \dots, n-1\}$, then one of the sides of $\mathcal{R}$ separates $\mathcal{E}_i$ from $\mathcal{R}$ and, therefore, $\mathcal{E}_i \cap \operatorname{int}(\mathcal{R}) = \emptyset$. Notice that it may be the case that $\mathcal{E}_i \cap \operatorname{int}(\mathcal{R}) = \emptyset$ but none of those conditions could be verified (and then such an index i would unnecessarily belong to $\bar{\mathcal{F}}_k^0$). Figure 6.1 shows the projection onto the x-y plane of an ellipsoid in the three-dimensional space and the set $\mathcal{R}$. Although this ellipsoid does not intersect the interior of $\mathcal{R}$, none of the conditions above can be verified. The dashed circle represent the projection of the minimal bounding sphere of the ellipsoid.

Once the set $\bar{\mathcal{F}}_k^0$ is computed, we are ready to define the second type of isolation constraints. If $\bar{\mathcal{F}}_k^0 = \emptyset$, then the second type of isolation constraints is not necessary. Suppose that $\bar{\mathcal{F}}_k^0 \neq \emptyset$. As we see in (23) and (24), these isolation constraints are determined by the parameter $h \in \mathbb{R}$. Let h_0 be the highest middle height of an ellipsoid in $\bar{\mathcal{F}}_k^0$, i.e.,

$$h_0 = \max_{i \in \bar{\mathcal{F}}_k^0} [c_i]_n, \tag{29}$$

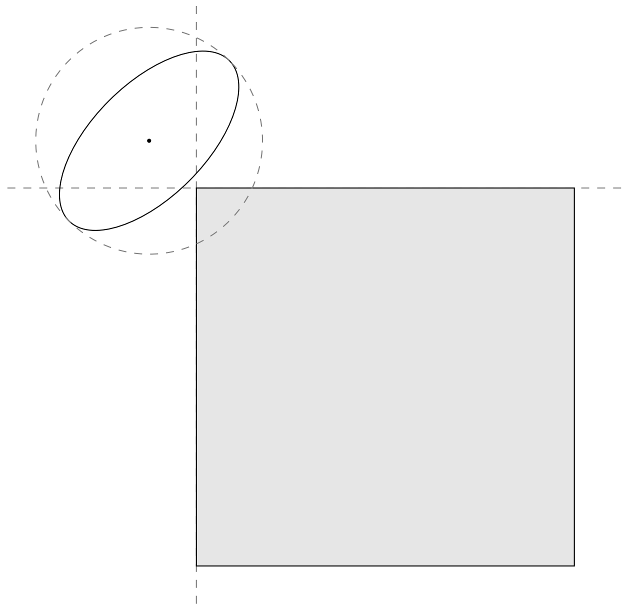


Figure 6.1: Projection of the three-dimensional set $\mathcal{R}$ and the three-dimensional ellipsoid onto the x-y plane.

and b denote the largest semi-axis length among the new ellipsoids, i.e.,

$$b = \max_{j \in \mathcal{N}_k} \lambda_{\max}(P_j^{\frac{1}{2}}).$$

For a given $\gamma \geq 0$, we define $h = h_0 - \gamma b$. Finally, we let

$$\bar{\mathcal{F}}_k = \{i \in \bar{\mathcal{F}}_k^0 \mid \mathcal{E}_i \cap \operatorname{int}(\mathcal{H}) = \emptyset\}.$$

An illustrative example of the construction of the set $\bar{\mathcal{F}}_k$ is given in Figure 6.2. The ellipsoids in $\mathcal{F}_k$, that were packed in previous iterations, are shown in Figure 6.2(a). The hyperrectangle with side length s is highlighted in Figure 6.2(b). A new ellipsoid that is placed inside this hyperrectangle can only possibly overlap with the blue ellipsoids, which therefore form the set $\bar{\mathcal{F}}_k^0$. Once $\bar{\mathcal{F}}_k^0$ is found, the second type of isolation constraints is defined. The hyperplane that determines the half-space $\mathcal{H}$ (see (16)) is placed at a distance γb from the center of the highest (in the sense of maximum middle height) ellipsoid in $\bar{\mathcal{F}}_k^0$; see Figure 6.2(c). Then, a new ellipsoid placed inside the hyperrectangle and above this hyperplane can only overlap with the red ellipsoids, which constitute the set $\bar{\mathcal{F}}_k$.

6.2 Removing unnecessary constraints

Let $\ell \in \{1, \ldots, n-1\}$ and $\xi \in \{-1, 1\}$. Consider the pairs of constraints (19,20) and (21,22) associated with ℓ and ξ. Notice that only one pair among these two are necessary in the model (17)–(26), as one will necessarily implies the other.

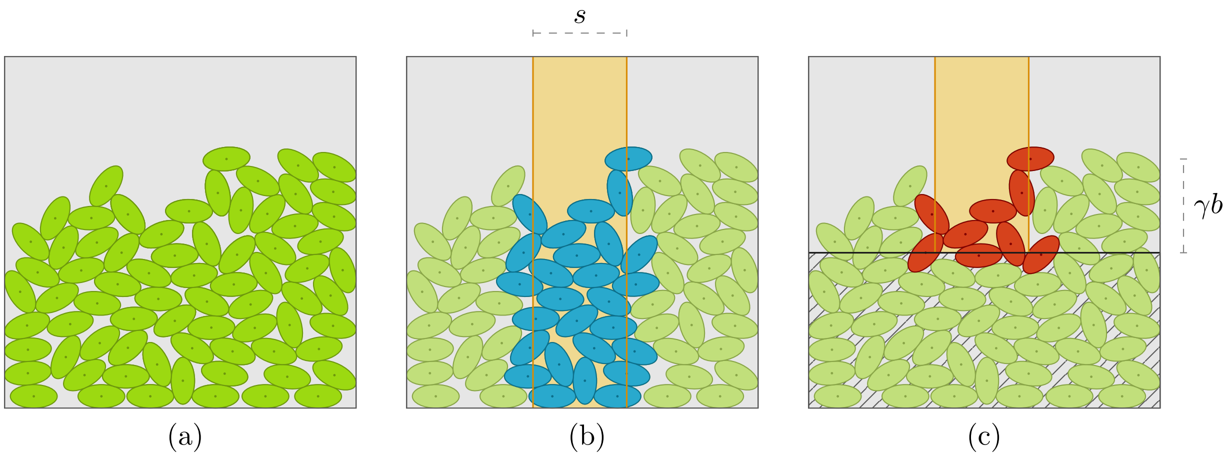


Figure 6.2: Selection of the ellipsoids to be considered in the non-overlapping constraints. (a) Fixed ellipsoids from the set $\mathcal{F}_k$. (b) First type of isolation constraints and determination of set $\bar{\mathcal{F}}_k^0$ formed by the blue ellipsoids. (c) Considering also the second type of isolation constraints, the set $\bar{\mathcal{F}}_k$ is then formed by the red ellipsoids.

Since the objective of the model is to minimize the height of the ellipsoids, they will be as low as possible from the "top lid" of the cube. In this case, the constraints (19,20) associated with $\ell = n$ and $\xi = 1$ would play no role in the model. We then remove these constraints and check whether they are satisfied when we obtain a solution to the problem. Some advantages of removing these constraints from the model are that we can easily construct an initial feasible solution when the container is almost full and the number of constraints are reduced.

6.3 Defining the initial solution

The initial solution is defined by the centers and rotation angles of the ellipsoids in $\mathcal{N}_k$. Each rotation angle of the ellipsoids is uniformly randomly chosen on the interval $[-\pi, \pi]$. The center of the ellipsoids are randomly chosen so that the ellipsoid are assuredly inside the container and satisfy the isolation constraints. For each $i \in \mathcal{N}_k$, we define the first $n-1$ components of c_i to be

$$[c_i]_\ell = \max\{-l/2 + a_i, \min\{l/2 - a_i, u_\ell + \beta(s/2 - a_i)\}\}, \text{ for each } \ell \in \{1, \dots, n-1\},$$

where β is a random variable that follows a uniform distribution on the interval $[-1, 1]$.

If $\bar{\mathcal{F}}_k = \emptyset$, let $\bar{h} = -l/2$. Otherwise, let $\bar{h}$ be defined as follows:

$$\bar{h} = \max_{j \in \bar{\mathcal{F}}_k}\{[c_j]_n + a_j\}.$$

Let $r = |\mathcal{F}_k| + 1$ and suppose that $\mathcal{N}_k = \{r, r+1, \dots, r + |\mathcal{N}_k| - 1\}$. For each $i \in \mathcal{N}_k$, we define the last component of the center of $\mathcal{E}_i$ to be

$$[c_i]_n = \bar{h} + a_i + 2\sum_{j=r}^{i-1} a_j.$$

This construction guarantees that the initial solution is feasible: every ellipsoid is inside the container, satisfy the isolation constraints, and do not overlap with any other ellipsoid.

6.4 Solving the subproblems and analyzing the solution found

We solve problem (17)–(26) with the nonlinear programming solver Algencan [2, 6] version 3.0.0. As we saw in Section 6.2, after a solution is returned by the solver, we must check whether it satisfies the constraints (19,20) associated with $\ell = n$ and $\xi = 1$, since we removed these constraints from the model. If they are not satisfied, then we declare that the solution is not feasible. Even if the solution is feasible, we must check whether this solution is acceptable. We say that a solution is acceptable if it is feasible (it satisfies all constraints of the model (17)–(26), including (19,20) associated with $\ell = n$ and $\xi = 1$), and each of the new packed ellipsoids is acceptably packed. An ellipsoid with index $i \in \mathcal{N}_k$ is acceptably packed if at least one of the following statements is true:

1. it touches the bottom side of the container (i.e., the constraint (19) associated with i, $\ell = n$ and $\xi = -1$ holds with equality);
2. it touches an ellipsoid packed in previous iterations;
3. it touches another acceptably packed ellipsoid.

If the solution found is acceptable, we declare that the subproblem was solved. Otherwise, we declare that the subproblem was not solved.

6.5 Reducing the size of $\mathcal{N}_k$

Consider the situation where, at an iteration k of Algorithm 1, we want to pack $|\mathcal{N}_k| > 1$ ellipsoids. Suppose that it is not possible to pack $|\mathcal{N}_k|$ ellipsoids after the τ trials of Step 3 of Algorithm 1. This situation naturally occurs when the container is almost full of ellipsoids. However, it could be the case that it is possible to pack less than $|\mathcal{N}_k|$ ellipsoids. For example, considering the container is almost full, it may not be possible to pack five more ellipsoids, but two new ellipsoids could fit in the container.

In order to consider this situation and improve Algorithm 1, we modify Step 3 in the following way. When $t > \tau$, we stop Algorithm 1 if and only if $|\mathcal{N}_k| = 1$. If $t > \tau$ but $|\mathcal{N}_k| > 1$, we reduce the size of $|\mathcal{N}_k|$ by one unit, let $t \leftarrow 0$, and continue again from Step 3.

6.6 Objective

Given $\xi \in [0, 1]$, the objective of problem (17)–(26) is to minimize the sum of the heights of the ellipsoids:

$$\sum_{i \in \mathcal{N}_k} 2\xi [c_i]_n + (1 - 2\xi) s^i_{\text{sup}}.$$

When $\xi = \frac{1}{2}$, the above expression becomes simply

$$\sum_{i \in \mathcal{N}_k} [c_i]_n.$$

In this case, the variables s^i_{sup}, for $i \in \mathcal{N}_k$, and the constraints (25)–(26) can be removed from the problem.

7 Numerical experiments and discussion

We implemented, in Fortran 90, the model (17)–(26) and the optimization procedure described in Section 5. To solve the nonlinear programming problems, we used Algencan [2, 6] version 3.0.0. The models, the optimization procedure, and Algencan were compiled with the GNU Fortran compiler (GCC) 5.4.0 with the `-O3` option enabled. The tests were run on a machine with Intel® Xeon® Processor X5650, 8GB of RAM memory, and Ubuntu 16.04 operating system. Our computer implementation of the method and the solutions reported in this section are freely available at `http://www.ime.usp.br/~lobato/`.

7.1 Evaluation of the isolation constraints and algorithmic parameters

In a first set of numerical experiments, we aim to evaluate the two types of isolation constraints described in Section 5. The first one constrains the new ellipsoids to remain within a hyperrectangle with infinite height. The second type of isolation constraint requires the new ellipsoids to lie above a certain plane parallel to the x-y plane. The isolation constraints depend on some parameters. The first type of isolation constraint depends on the choice of the lengths of the sides of the hyperrectangle (parameter $s \in \mathbb{R}$ in (15)). As for the second type, we need to decide at which point the plane must pass through (parameter $h \in \mathbb{R}$ in (16)). Ideally, the presence of isolation constraints should not affect the quality of the solution. Thus, we need to determine what would be good parameters for those constraints. Let b be the largest length of a semi-axis among the new ellipsoids to be packed. We shall let $s = \eta b$ and $h = h_0 - \gamma b$ (where h_0 is given by (29)) for the factors η and γ varying in the set $\{4, 5, \ldots, 10\}$. Notice that the values of these parameters will not change during the execution of Algorithm 1. Another parameter that must be chosen is the size of the set $\mathcal{N}_k$, i.e., the number of ellipsoids that must be packed at each iteration. We decided to let the size of this set be the same for all iterations (unless this size is reduced as explained in Section 6.5). We considered sets of sizes from 1 to 5.

To assess the influence of these parameters on the quality of the solution, we considered the packing of ellipsoids with semi-axis lengths 1, 0.75, and 0.5 within a cube with side length 30. The objective is to minimize the middle height of the ellipsoids (i.e., taking $\xi = 1/2$ in (17) and thus minimizing the sum of the n-th coordinate of the centers of the ellipsoids), according to Algorithm 1 presented in Section 5. At each iteration k of Algorithm 1, we use $\tau = 100$, i.e., we try to solve the subproblem at most 100 times. As explained in Section 6.5, if it was not possible to solve the subproblem after τ attempts, we reduce the size of $\mathcal{N}_k$ by one unit, and try to solve the subproblem again within τ new attempts. Once the number of ellipsoids to be packed is reduced, it is never increased again.

Tables 7.1 and 7.2 show the results we have obtained when packing the ellipsoids one by one, considering $|\mathcal{N}_k| = 1$ for each iteration k. Each entry in these tables has two numbers and is associated with a particular choice of η and γ. For Table 7.1, each entry shows the number of ellipsoids that were packed (left) and the CPU time in seconds (right).

As expected, the quality of the solution improves as the length of the side of the hyperrectangle increases. On the other hand, the behavior is not clear with respect to the γ parameter, which determine the height of the hyperplane. This suggests that even for $\gamma = 4$, the hyperplane is low enough not to affect the quality of the solution. We can also gauge the impact of η and γ

by checking whether the isolation constraints were active at the solution found at each iteration. Table 7.2 shows the number of iterations where the first isolation constraint was active (left) and the number of iterations where the second isolation constraint was active (right).

When $\eta = \gamma = 4$, the second type of isolation constraint is active only in two iterations out of 10272, which is a negligible amount. For any other combination of values for η and γ, the second type of isolation constraint is never active. This suggests that 4 can be a reasonable choice for the value of γ. Nevertheless, the first type of isolation constraint is active in a considerable number of iterations. For $\eta = 4$, this constraint is active around 48% of the iterations. For $\eta = 10$, this figure drops to 11%.

Tables 7.3 and 7.4 show the results when the ellipsoids are packed two by two; Tables 7.5 and 7.6 present the results when the ellipsoids are packed three at a time; Tables 7.7 and 7.8 show the results when the ellipsoids are packed four at a time; Tables 7.9 and 7.10 present the results when the ellipsoids are packed five by five.

		Hyperplane height factor γ													
		4		5		6		7		8		9		10	
Hyperrectangle side length factor η	4	10272	802	10319	911	10322	994	10313	1108	10313	1197	10322	1311	10322	1356
	5	10520	934	10523	1022	10489	1180	10497	1290	10494	1393	10515	1590	10465	1652
	6	10590	1097	10573	1248	10584	1408	10591	1649	10591	1817	10584	1903	10594	2086
	7	10648	1246	10653	1448	10640	1721	10633	1898	10645	2137	10644	2396	10640	2705
	8	10683	1425	10686	1764	10682	2010	10674	2383	10681	2683	10690	2883	10700	3498
	9	10712	1747	10713	2091	10711	2403	10706	2818	10716	3129	10693	3541	10706	4051
	10	10722	1949	10724	2412	10716	2876	10725	3415	**10732**	3736	10724	4399	10707	4851

Table 7.1: Number of ellipsoids packed (left) and CPU time in seconds (right) considering the strategy of packing one ellipsoid at a time.

		Hyperplane height factor γ													
		4		5		6		7		8		9		10	
Hyperrectangle side length factor η	4	4988	2	5026	0	5005	0	5057	0	5007	0	4913	0	4967	0
	5	3561	0	3480	0	3551	0	3561	0	3572	0	3561	0	3627	0
	6	2655	0	2721	0	2714	0	2663	0	2692	0	2592	0	2719	0
	7	2117	0	2085	0	2156	0	2154	0	2125	0	2167	0	2102	0
	8	1759	0	1775	0	1742	0	1687	0	1753	0	1670	0	1647	0
	9	1412	0	1358	0	1448	0	1475	0	1467	0	1457	0	1410	0
	10	1212	0	1180	0	1237	0	1248	0	1232	0	1248	0	1200	0

Table 7.2: Number of subproblems in which the first type of isolation constraint was active (left) and number of subproblems in which the second type of isolation constraint was active (right), considering the strategy of packing one ellipsoid at a time.

Let N be the number of ellipsoids that are packed at each iteration of the algorithm. We can observe that the CPU time increases when N increases. This is because the subproblems become harder to solve when there are more ellipsoids to pack at the same time. However, the quality of the solution is not considerably improved when N increases; it is almost the same for all $N \in \{1, 2, 3, 4, 5\}$.

	Hyperplane height factor γ													
Hyperrectangle side length factor η	4		5		6		7		8		9		10	
4	10136	3182	10199	3330	10213	3648	10192	3769	10196	3940	10177	4140	10207	4482
5	10453	2752	10460	3077	10458	3376	10472	3636	10475	4038	10463	4167	10471	4841
6	10572	3306	10591	3503	10577	4042	10575	4174	10588	4597	10592	5085	10572	5276
7	10637	2973	10631	3243	10638	3731	10632	4088	10633	4339	10652	4952	10647	5342
8	10683	3089	10663	3865	10668	4049	10681	4464	10690	4952	10672	5434	10687	6090
9	10701	3458	10690	3928	10713	4328	10697	4869	10702	5393	10699	5982	10686	6671
10	10704	3931	10708	4975	10712	5687	10711	6372	10711	6722	10724	7353	**10726**	8350

Table 7.3: Number of ellipsoids packed (left) and CPU time in seconds (right) considering the strategy of packing two ellipsoids at a time.

	Hyperplane height factor γ													
Hyperrectangle side length factor η	4		5		6		7		8		9		10	
4	4136	32	4207	8	4205	0	4181	0	4217	0	4197	0	4165	0
5	3304	7	3301	0	3317	0	3198	0	3329	0	3304	0	3337	0
6	2629	1	2486	0	2529	0	2629	0	2551	0	2582	0	2589	0
7	2094	0	2055	0	2096	0	2041	0	2039	0	2037	0	2072	0
8	1654	3	1639	0	1716	0	1714	0	1684	0	1686	0	1685	0
9	1326	2	1419	0	1435	0	1453	0	1472	0	1392	0	1459	0
10	1159	0	1191	0	1206	0	1195	0	1155	0	1240	0	1176	0

Table 7.4: Number of subproblems in which the first type of isolation constraint was active (left) and number of subproblems in which the second type of isolation constraint was active (right), considering the strategy of packing two ellipsoids at a time.

	Hyperplane height factor γ													
Hyperrectangle side length factor η	4		5		6		7		8		9		10	
4	10091	9828	10129	10081	10106	10880	10139	11003	10138	11968	10124	11856	10113	12921
5	10422	8666	10446	8503	10443	8722	10451	9424	10403	10166	10456	10499	10416	10823
6	10565	6917	10571	7864	10558	8362	10564	8934	10551	9116	10559	10190	10562	10267
7	10633	6328	10631	7528	10637	8221	10651	8718	10643	9697	10646	10070	10645	10411
8	10704	8212	10680	9091	10676	9924	10680	10304	10704	10954	10679	11630	10688	13076
9	10693	6870	10728	7323	10705	8480	10713	9583	10714	10004	10716	10708	10703	12014
10	10703	8015	10718	9070	10717	10178	10723	11057	10713	13129	10728	13889	**10729**	15097

Table 7.5: Number of ellipsoids packed (left) and CPU time in seconds (right) considering the strategy of packing three ellipsoids at a time.

Table 7.1 shows the results when we pack one ellipsoid at a time and minimize its middle height ($\xi = 1/2$). Now, we also consider the strategy of packing one ellipsoid at a time but minimizing a different height. We consider the minimization of the lower ($\xi = 1$), upper ($\xi = 0$), and a random height of the ellipsoid at each iteration. For the minimization of the random height, the value of ξ is determined right before Step 3.5 of Algorithm 1 and is chosen uniformly randomly on the interval $[0, 1]$. Table 7.11 shows the results for the minimization of the lower height. Table 7.12 shows the results for the minimization of the upper height. Table 7.13 shows

Hyperrectangle side length factor η		Hyperplane height factor γ													
		4		5		6		7		8		9		10	
	4	3200	60	3213	5	3214	10	3202	0	3190	0	3196	0	3206	0
	5	2905	28	2858	3	2870	0	2871	0	2878	0	2802	0	2869	0
	6	2388	10	2346	1	2407	0	2400	0	2387	0	2409	0	2370	0
	7	1953	6	1923	0	1914	0	1908	0	2000	0	1952	0	1933	0
	8	1591	11	1627	0	1642	0	1660	0	1559	0	1590	0	1580	0
	9	1368	11	1347	0	1371	0	1326	0	1367	0	1355	0	1341	0
	10	1154	2	1154	0	1152	0	1183	0	1122	0	1212	0	1228	0

Table 7.6: Number of subproblems in which the first type of isolation constraint was active (left) and number of subproblems in which the second type of isolation constraint was active (right), considering the strategy of packing three ellipsoids at a time.

Hyperrectangle side length factor η		Hyperplane height factor γ													
		4		5		6		7		8		9		10	
	4	9970	20733	9987	21274	10053	22357	10096	23086	10052	25256	10094	25904	10067	26082
	5	10374	16535	10381	17116	10432	18480	10412	19091	10419	20358	10420	21256	10419	24043
	6	10586	13691	10569	15675	10540	15807	10579	17302	10595	17735	10564	18643	10572	21131
	7	10640	11892	10615	13458	10642	15006	10663	15772	10648	16857	10655	18224	10641	18933
	8	10665	13736	10664	14151	10686	17169	10692	18058	10695	18977	10708	20193	10699	21173
	9	**10737**	12215	10719	12982	10699	13392	10724	15166	10731	16183	10705	17712	10700	20276
	10	10726	11885	10729	13707	10732	15965	10732	15920	10715	19010	10713	19049	10733	21747

Table 7.7: Number of ellipsoids packed (left) and CPU time in seconds (right) considering the strategy of packing four ellipsoids at a time.

Hyperrectangle side length factor η		Hyperplane height factor γ													
		4		5		6		7		8		9		10	
	4	2463	148	2473	76	2478	13	2498	0	2500	3	2494	0	2475	1
	5	2390	74	2401	33	2406	1	2388	0	2415	0	2395	0	2412	0
	6	2149	35	2100	4	2175	0	2100	0	2123	0	2159	0	2128	0
	7	1823	14	1852	1	1847	0	1843	0	1828	0	1793	0	1795	0
	8	1535	27	1495	1	1496	0	1528	0	1543	0	1487	0	1472	0
	9	1265	2	1317	2	1330	0	1279	0	1330	0	1347	0	1297	0
	10	1163	15	1137	0	1086	0	1116	0	1135	0	1110	0	1129	0

Table 7.8: Number of subproblems in which the first type of isolation constraint was active (left) and number of subproblems in which the second type of isolation constraint was active (right), considering the strategy of packing four ellipsoids at a time.

the results for the minimization of a random height. We can observe that the quality of the solutions is much lower than those found in previous experiments in which the middle height was minimized.

In Figure 7.1, we show the graphical representation of the best solution found for the min-

		Hyperplane height factor γ													
		4		5		6		7		8		9		10	
Hyperrectangle side length factor η	4	9775	37713	9964	39789	9995	42632	9957	44250	10030	43969	10068	50037	10059	50803
	5	10341	32992	10388	33020	10388	35112	10392	36862	10388	38651	10411	42044	10434	42090
	6	10534	25741	10571	24587	10558	26670	10558	26549	10552	29312	10597	30355	10571	32104
	7	10635	20672	10666	22122	10646	22820	10642	24923	10631	26333	10644	27509	10622	30696
	8	10694	22447	10715	24865	10686	25509	10700	28227	10702	30934	10709	32085	10712	33478
	9	10717	18196	10727	19469	10710	22592	10716	24535	10737	25709	10713	28514	10714	29317
	10	10736	17115	10740	19820	**10742**	21425	10731	22140	10729	24293	10727	26382	10731	28502

Table 7.9: Number of ellipsoids packed (left) and CPU time in seconds (right) considering the strategy of packing five ellipsoids at a time.

		Hyperplane height factor γ													
		4		5		6		7		8		9		10	
Hyperrectangle side length factor η	4	1963	214	1986	107	2006	43	1996	35	2016	9	2017	3	2012	2
	5	2006	94	2022	20	2013	6	2020	0	2027	1	2027	0	2019	0
	6	1888	52	1859	13	1874	1	1876	0	1851	0	1857	0	1885	0
	7	1660	20	1662	2	1675	0	1656	0	1678	0	1666	0	1677	0
	8	1416	12	1408	3	1416	0	1442	0	1419	0	1444	0	1419	0
	9	1262	14	1238	0	1266	0	1278	0	1246	0	1294	0	1283	0
	10	1074	7	1097	0	1090	0	1114	0	1091	0	1115	0	1118	0

Table 7.10: Number of subproblems in which the first type of isolation constraint was active (left) and number of subproblems in which the second type of isolation constraint was active (right), considering the strategy of packing five ellipsoids at a time.

		Hyperplane height factor γ													
		4		5		6		7		8		9		10	
Hyperrectangle side length factor η	4	9954	767	9955	859	9969	966	9974	1022	9967	1086	9952	1150	9941	1268
	5	10060	928	10110	1043	10125	1210	10127	1327	10102	1388	10099	1503	10095	1638
	6	10156	1104	10149	1319	10152	1443	10163	1611	10166	1762	10165	1992	10145	2173
	7	10178	1311	10182	1550	10190	1773	10190	2005	10186	2285	10202	2474	10205	2691
	8	10199	1498	10221	1880	10217	2196	10231	2546	10202	2838	10221	3157	10233	3361
	9	10260	1854	10258	2329	10272	2526	10235	3048	10251	3415	10255	3821	10244	4174
	10	10249	2098	**10273**	2582	10254	3022	10261	3491	10259	3989	10257	4540	10265	5199

Table 7.11: Number of ellipsoids packed (left) and CPU time in seconds (right) considering the strategy of packing one ellipsoid at a time and minimizing the lower height of the ellipsoid.

imization of each kind of height of the ellipsoid. Figure 7.1(a) represents the solution obtained by minimizing the middle height of the ellipsoid to be packed. In this case, 10732 ellipsoids were packed. Figure 7.1(b) represents the solution found by minimizing the lower height of the ellipsoid to be packed. In this case, 10273 ellipsoids were packed. We can notice that the semi-minor axis of the ellipsoids tends to be almost perpendicular to the base of the cube (the ellipsoids are almost "lying"). In Figure 7.1(c), we have the solution with 10281 ellipsoids obtained by minimizing the upper height of the ellipsoid. We observe in this case another trend: the ellip-

		Hyperplane height factor γ													
		4		5		6		7		8		9		10	
Hyperrectangle side length factor η	4	9748	908	9777	1019	9745	1123	9788	1248	9775	1329	9774	1585	9763	1529
	5	9982	1048	9997	1196	9995	1466	9985	1478	9977	1576	10003	1745	10010	1846
	6	10101	1196	10135	1364	10104	1605	10111	1761	10122	1916	10123	2076	10119	2251
	7	10168	1300	10175	1586	10166	1775	10185	2047	10157	2238	10160	2455	10176	2720
	8	10226	1664	10230	1948	10230	2269	10220	2654	10227	3019	10237	3521	10245	3998
	9	10249	2289	10261	2744	10243	3028	10264	3339	10250	3758	10250	4049	10241	3707
	10	10269	2204	10256	2670	10266	3105	**10281**	3591	10272	3983	10273	4490	10256	4899

Table 7.12: Number of ellipsoids packed (left) and CPU time in seconds (right) considering the strategy of packing one ellipsoid at a time and minimizing the upper height of the ellipsoid.

		Hyperplane height factor γ													
		4		5		6		7		8		9		10	
Hyperrectangle side length factor η	4	10052	1029	10027	1197	10049	1292	10040	1422	10039	1574	10038	1655	10041	1741
	5	10244	1540	10233	1679	10231	1820	10223	1924	10236	2136	10202	2310	10230	2514
	6	10315	2027	10306	2359	10311	2058	10298	2703	10308	3089	10320	3275	10294	3408
	7	10374	1775	10361	2371	10337	2769	10352	3318	10351	3152	10353	3759	10358	3796
	8	10388	2450	10383	2986	10368	3346	10389	3811	10380	4127	10371	4474	10372	4968
	9	10414	2379	10409	2809	10419	3339	10414	3870	10403	4204	10420	5496	10416	5231
	10	**10438**	2699	10419	3227	10418	4084	10410	4507	10429	5278	10425	5853	10430	6391

Table 7.13: Number of ellipsoids packed (left) and CPU time in seconds (right) considering the strategy of packing one ellipsoid at a time and minimizing a random height of the ellipsoid.

soids have their semi-major axes nearly perpendicular to base of the cube (the ellipsoids are almost "standing"). Figure 7.1(d) shows the solution with 10438 ellipsoids found by minimizing a random height of the ellipsoid. Contrary to what occurred in the minimization of the lower and upper heights, we cannot notice any positioning trend of the ellipsoids when we minimize the middle or a random height. They are positioned in a more varied way (they are "messier"), which should have contributed in getting a higher quality solution.

7.2 Comparison with other approaches

The works in [4] and [5] have dealt with the problem of packing a given collection of ellipsoids within a container whose volume must be minimized. The method proposed in [4] is suitable for small-sized instances having up to 100 ellipsoids. The method introduced in [5] is not appropriate for small-sized instances but for medium-sized instances with no more than 1,000 ellipsoids. Although the problem considered in this work is different from the ones considered in [4] and [5], we can try to compare how the method introduced in the present paper performs on those small- and medium-sized instances. The largest instance considered in [4] has 100 ellipsoids with semi-axis lengths $(1, 0.75, 0.5)$. For the problem that aims to minimize the volume of a cuboidal container, a cuboid with approximate side lengths $(5.17787, 5.73271, 8.26307)$ was found in [4] for that instance. To compute this solution required more than 2 days and 17 hours (in a computational environment similar to the one being considered in the present work). In [5], the largest instance considered for packing three-dimensional ellipsoids within a cuboid has 500 ellipsoids with semi-axis lengths $(1, 0.75, 0.5)$. For this instance, a solution with a cuboid with side lengths $(9.24587, 10.29180, 12.22661)$ was found after more than one day of computational time. Bringing each of these instances to the problem considered in the present work, we fix the container as the cuboid having one of those side lengths and try to pack as many ellipsoids with semi-axis lengths $(1, 0.75, 0.5)$ as possible. We have fixed the parameters $\eta = 10$, $\gamma = 10$, and $\tau = 100$, and packed the ellipsoids one by one. For the first instance, our method was able to pack 86 ellipsoids in less than 21 seconds. For the second instance, a solution with 431 ellipsoids was found in less than 155 seconds. For these instances, we see that the packings presented in [4] and [5] have higher densities than the ones produced by our current method. The explanation for this result relies, on the one hand, on the fact that the approaches introduced in [4] and [5] pack all the ellipsoids at the same time while minimizing the volume of the container; allowing the ellipsoids to be arranged in a more compact way and directly pursuing a higher density packing. On the other hand, it is important to notice that the approaches presented in [4] and [5] cannot be applied to large-sized instances and, even for small-sized instances, the method proposed here is much faster. Moreover, the method introduced in the present work was developed for large-sized instances, sacrificing quality of the solution in favor of a time complexity that grows linearly with the number of ellipsoids; the isolation constraints described in Section 4.3 being an example of that policy.

In [9, 27], studies on the density of ellipsoids' packings were reported. The authors were able to show how the density of packings of three-dimensional identical ellipsoids varies with the ellipsoids' shape. Experiments in [9] were conducted considering the packing of 1,000 ellipsoids within a cube with periodic boundary conditions. For ellipsoids with semi-axis lengths $(1, 1, 1.9^{-1})$ and $(1.3, 1, 1.3^{-1})$, for example, the packings found in [9] have densities close to 0.70 and 0.735, respectively; the latter being the packing with the highest density found in that work. We used our method to pack these ellipsoids within a cube with side length 20. Again, we fixed the parameters $\eta = 10$, $\gamma = 10$, and $\tau = 100$, and packed the ellipsoids one by one. For the first instance, our method found a packing containing 2,118 ellipsoids (which has a density around 0.583) within 10 minutes. For the second instance, a packing with density 0.585 and 1,119 ellipsoids was found within 7 minutes. In this case, the higher density obtained in [9] can be partially explained by the fact that (i) all ellipsoids are packed at the same time and (ii) the container has periodic boundary conditions, which increases the "useful space" of the container. Moreover, our method found both solutions in a few minutes; while computational times are not reported for the solutions presented in [9].

7.3 Applicability of the proposed approach

It is important to highlight the flexibility of the matheuristic approach being introduced in the present work, that was not developed seeking dense packings. The proposed approach allows the packing of non-identical ellipsoids, the addition of specific positioning constraints for each ellipsoid, and the addition of minimum distance constraints, among others. It is important to notice that, for some practical applications, the density is not the main focus. The approach presented here generalizes the one introduced in [29] for constructing initial configurations for molecular dynamics simulations. In this scenario, sphere packing problems with additional constraints and a density of approximately 30% are seek, as illustrated by the figure in [29, p.2162]. We finish the numerical experiments presenting three examples that illustrate the flexibility of the proposed approach.

In the first illustrative example, we consider the problem of packing non-identical ellipsoids within a cube. In this experiment, we chose the length of each semi-axis of each ellipsoid to be uniformly random on the interval $[0.1, 1]$. The ellipsoids were packed one at a time with their middle heights being minimized. Considering a cube with side length 30 and using the parameters $\eta = 10$, $\gamma = 6$, and $\tau = 100$, we were able to pack 23860 ellipsoids in 2h45m. Figure 7.2 illustrates this solution.

In a second illustrative example, we aim to show that the computational cost of the introduced strategy scales linearly with the number of ellipsoids being packed. We consider the packing of ellipsoids with semi-axis lengths $(1, 0.75, 0.5)$ within a cube with side length 140. We have chosen to pack one ellipsoid at a time and to minimize the middle height of the ellipsoid. We have also chosen $\eta = 10$, $\gamma = 4$, and $\tau = 10000$. Figure 7.3 shows the packing of 1,126,474 ellipsoids. This solution was found in 4d14h32m.

In the last illustrative example, we aim to show what can be achieved with additional positioning constraints. We consider the problem of simultaneously packing various kinds of ellipsoids within different regions that are determined by a cube $\mathcal{C}$, an ellipsoid $\mathcal{E}$, and a ball $\mathcal{B}$. The cube has side lengths 128; the ellipsoid has semi-axis lengths $(48, 24, 24)$; and the ball has radius 12. They are all centered at the origin, so that the ellipsoid lies within the cube and the ball lies within the ellipsoid. There are four groups of ellipsoids that must be packed. The first group has (red) ellipsoids with semi-axis lengths $(1, 0.75, 0.5)$. These ellipsoids must be packed so that they are inside one of the halves of the cube $\mathcal{C}$ but their centers are outside the ellipsoid $\mathcal{E}$. The second group has (green) ellipsoids with semi-axis lengths that are uniformly random on the interval $[0.25, 1]$. These ellipsoids must be inside the other half of the cube $\mathcal{C}$ but their centers must be outside the ellipsoid $\mathcal{E}$. The third group has (yellow) ellipsoids with semi-axis lengths $(0.75, 0.6, 0.5)$ whose centers must be inside the ellipsoid $\mathcal{E}$ but outside the ball $\mathcal{B}$. Finally, the fourth group has (blue) ellipsoids with semi-axis lengths $(0.5, 0.4, 0.3)$ and their centers must be inside the ball $\mathcal{B}$. We used the parameters $\eta = 10$, $\gamma = 10$, and $\tau = 1000$. Figure 7.4 illustrates the solution found containing 1,101,052 ellipsoids. Figure 7.5(a) shows only the ellipsoids whose last component of the center is nonpositive; while Figure 7.5(b) shows only the ellipsoids whose second component of the center is nonpositive.

8 Concluding remarks

The problem of packing ellipsoids in the three-dimensional space has been tackled through the application of global and local nonlinear optimization techniques in recent years. In all cases, only small- and medium-sized problems could be solved due to the nonconvexity of the highly complex considered models. In the present work, we introduced a matheuristic that uses nonlinear programming models and methods for solving small subproblems. In a constructive way, we were able to find solutions to packing problems with a huge number of ellipsoids. Problems with identical and non-identical ellipsoids can be tackled with the proposed approach. Moreover, the introduced method is flexible enough so it can easily handle positioning and other type of additional constraints. Assessing the quality of the obtained solutions, in the sense measuring in some way how far they are from a global solution is an open question that may be addressed in future research. On the other hand, the presented strategy is the first one based on nonlinear programming able to deliver solutions to that kind of huge ellipsoids' packing problems.

Acknowledgements. The authors are indebted to the anonymous referees whose comments helped to improve this paper. This work was supported by FAPESP (grants 2012/23916-8, 2013/03447-6, 2013/05475-7, 2013/07375-0, 2015/18053-9, 2016/01860-1, and 2017/05198-4) and CNPq (grant 309517/2014-1).

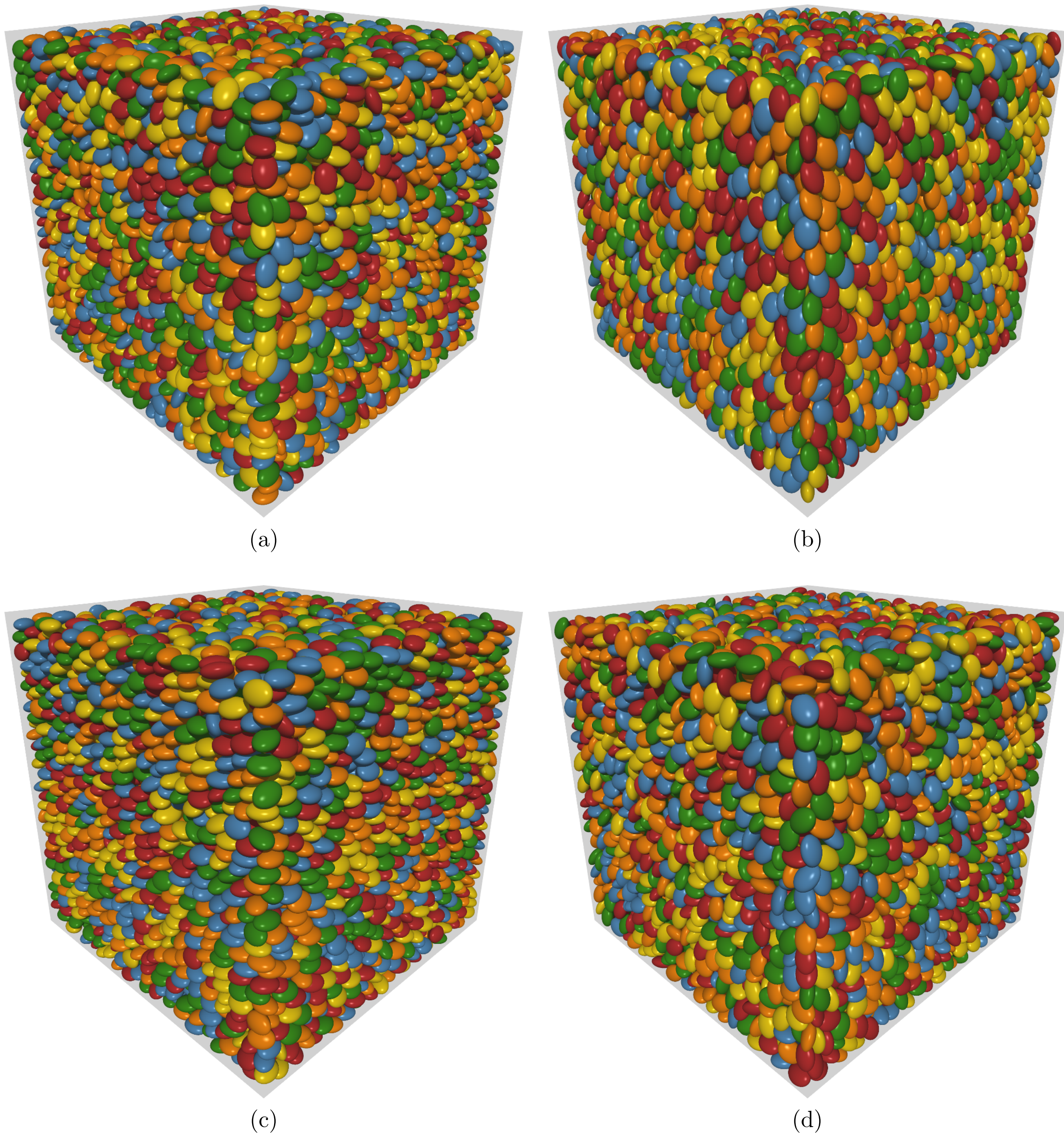


Figure 7.1: Packing of ellipsoids with semi-axis lengths $(1, 0.75, 0.5)$ within a cube with side length 30. (a) 10732 ellipsoids obtained by minimizing the middle height of the ellipsoid. (b) 10273 ellipsoids obtained by minimizing the lower height. (c) 10281 ellipsoids obtained by minimizing the upper height. (c) 10438 ellipsoids obtained by minimizing a random height.

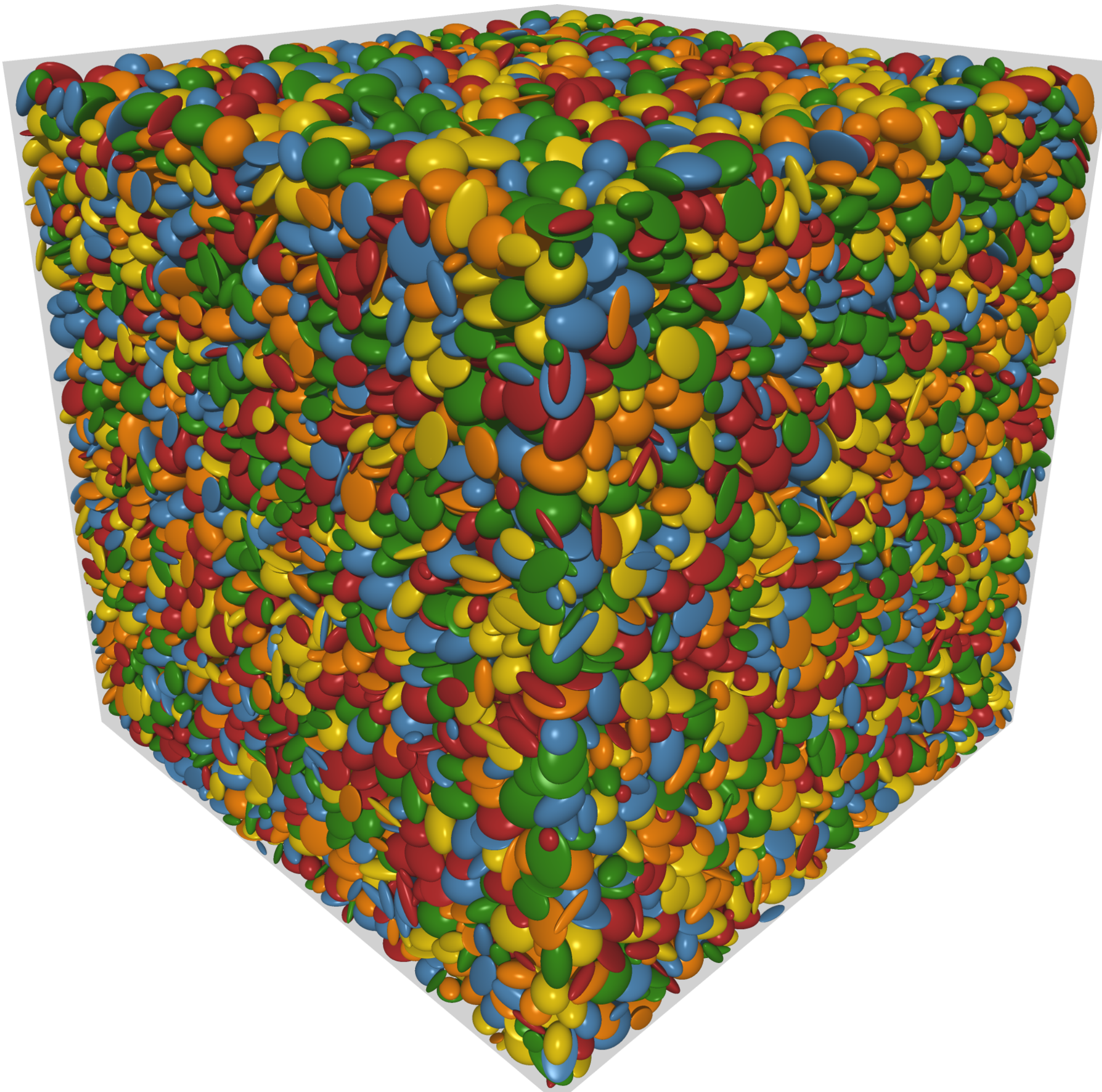


Figure 7.2: Packing of 23860 ellipsoids with uniformly random semi-axis lengths in the interval $[0.1, 1]$ within a cube with side length 30. This solution was found by packing ellipsoids one by one, minimizing the middle height, and using $\eta = 10$, $\gamma = 6$, and $\tau = 100$.

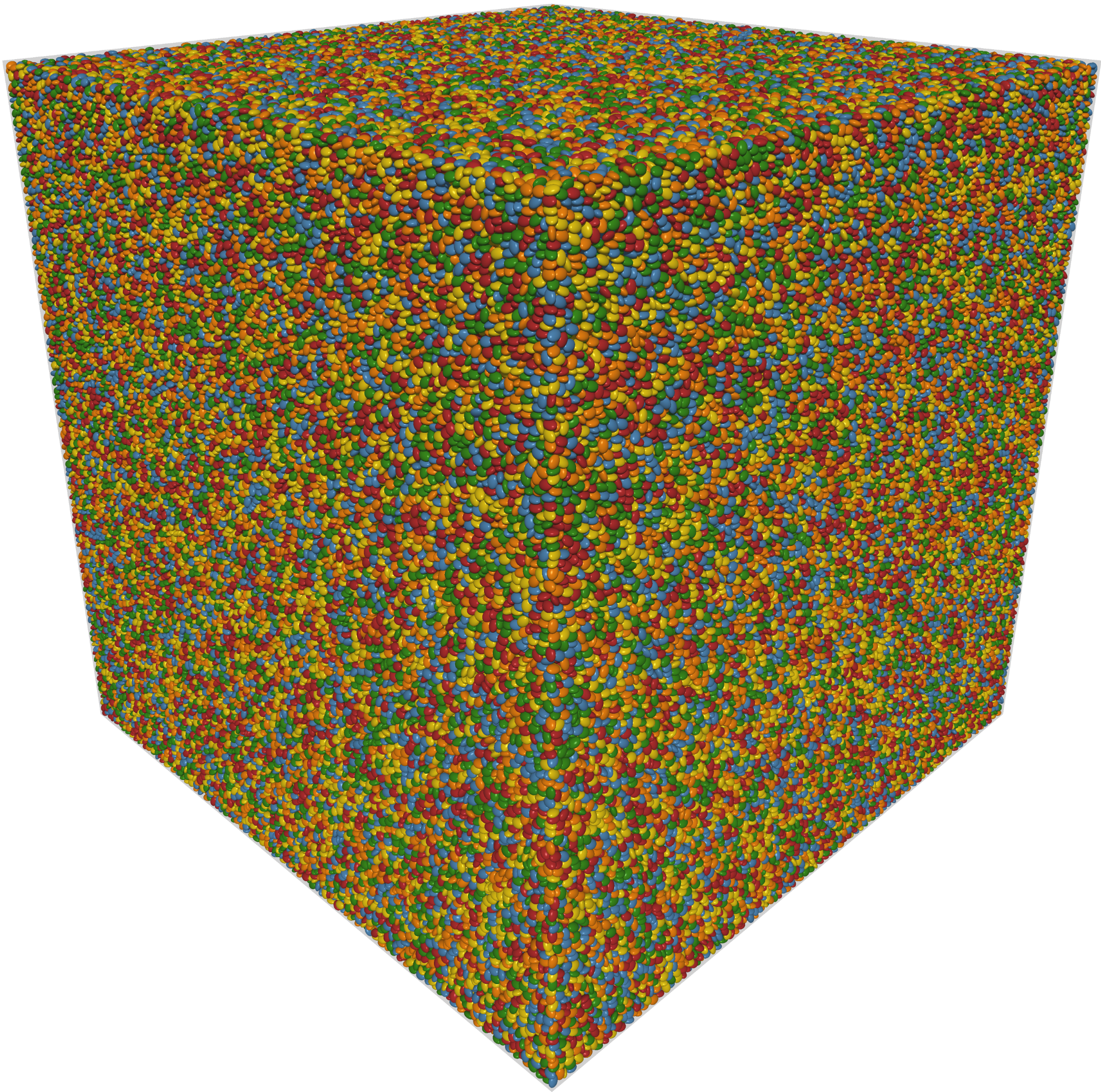


Figure 7.3: Packing of 1,126,474 ellipsoids with semi-axis lengths $(1, 0.75, 0.5)$ within a cube with side length 140. This solution was found by packing ellipsoids one by one, minimizing the middle height, and using $\eta = 10$, $\gamma = 4$, and $\tau = 10000$.

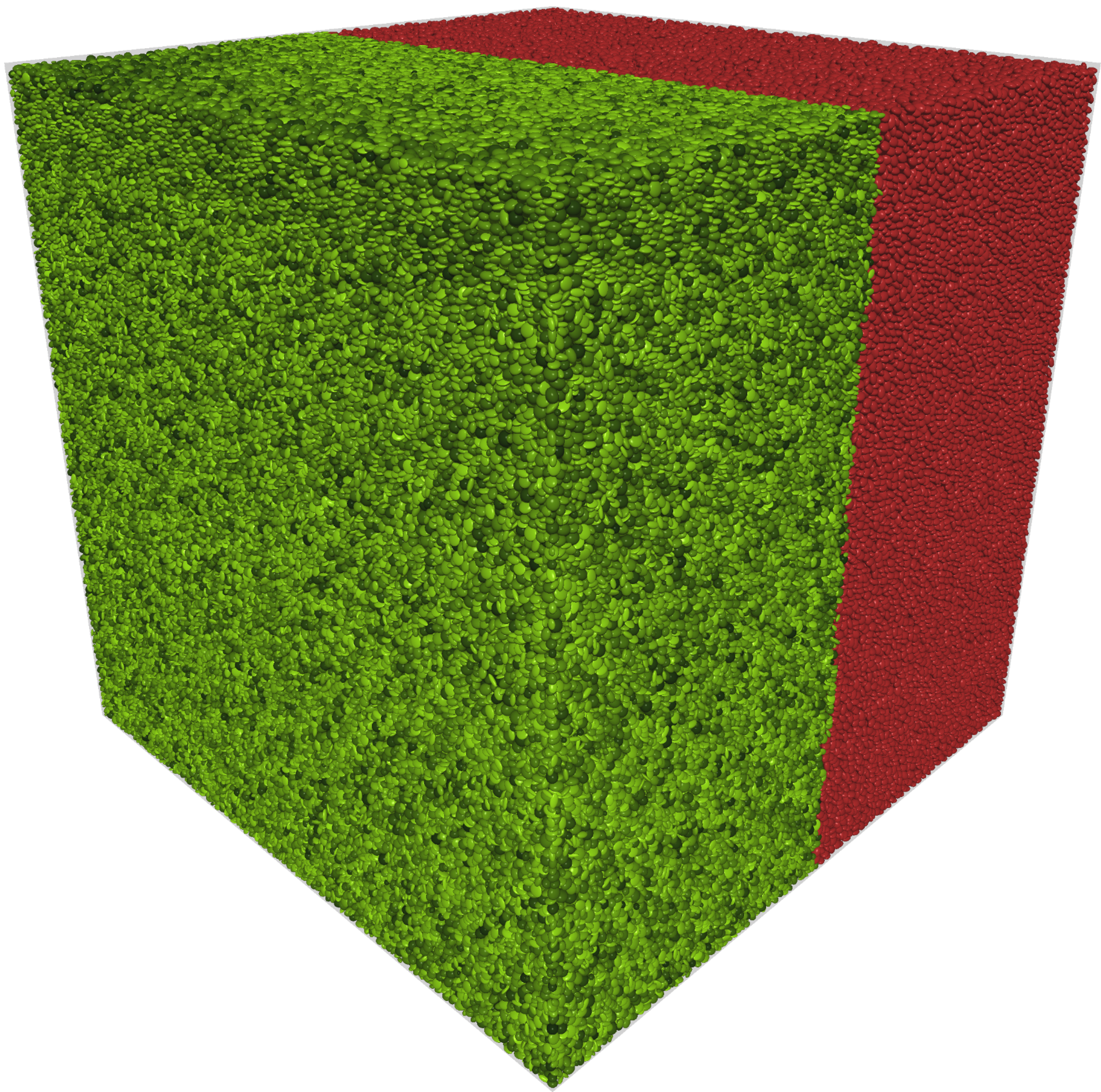
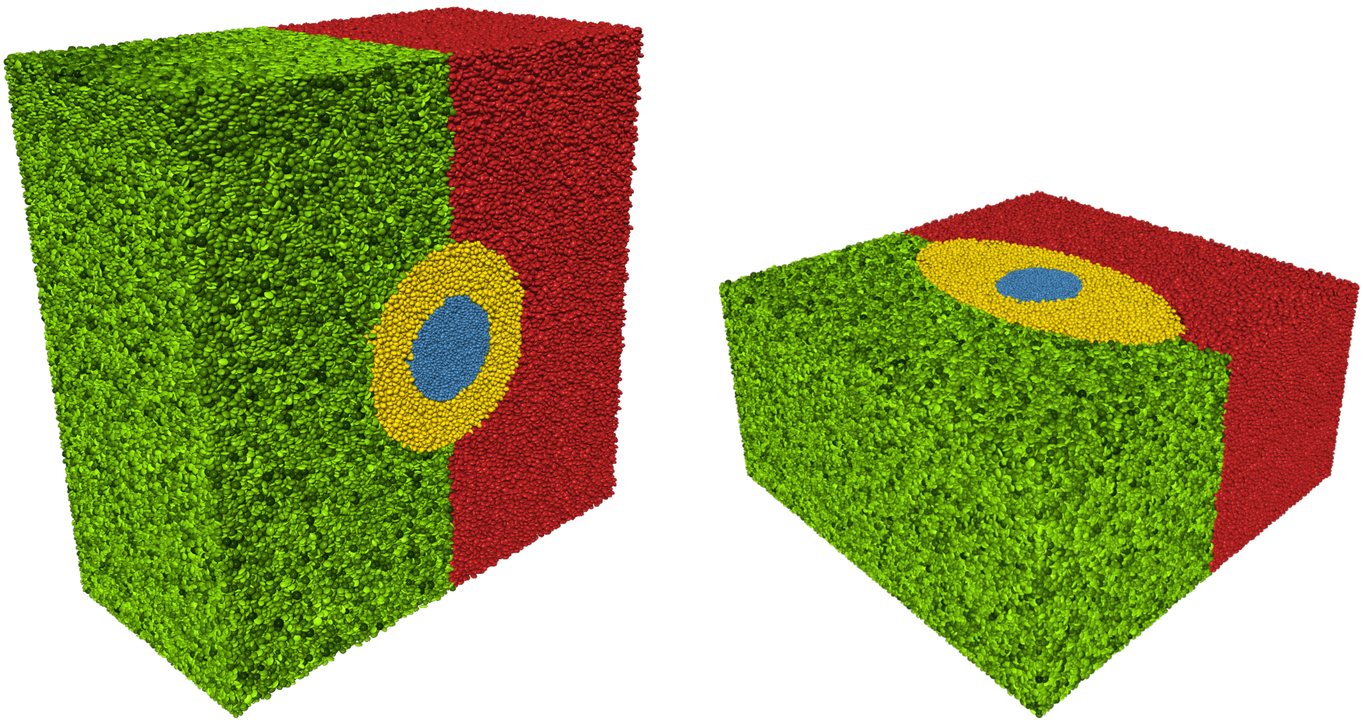


Figure 7.4: Solution found for the problem of packing various kinds of ellipsoids within different regions.



(a) (b)

Figure 7.5: Partial views of the interior of the solution found for the problem of packing various kinds of ellipsoids within different regions. (a) Only ellipsoids whose first component of the center is nonpositive are shown. (b) Only ellipsoids whose last component of the center is nonpositive are shown.

References

[1] B. J. Alder and T. E. Wainwright. Phase transition for a hard sphere system. *Journal of Chemical Physics*, 27(5):1208–1209, 1957.

[2] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt. On Augmented Lagrangian methods with general lower-level constraints. *SIAM Journal on Optimization*, 18(4):1286–1309, 2008.

[3] J. D. Bernal. Geometrical Approach to the Structure of Liquids. *Nature*, 183(4655):141–147, 1959.

[4] E. G. Birgin, R. D. Lobato, and J. M. Martínez. Packing ellipsoids by nonlinear optimization. *Journal of Global Optimization*, 65(4):709–743, 2016.

[5] E. G. Birgin, R. D. Lobato, and J. M. Martínez. A nonlinear programming model with implicit variables for packing ellipsoids. *Journal of Global Optimization*, 68(3):467–499, 2017.

[6] E. G. Birgin and J. M. Martínez. *Practical Augmented Lagrangian Methods for Constrained Optimization*, volume 10 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014.

[7] P. Chaikin. Thermodynamics and hydrodynamics of hard spheres: the role of gravity. In M. E. Cates and M. R. Evans, editor, *Soft and Fragile Matter: Nonequilibrium Dynamics, Metastability and Flow*, volume 53, pages 315–348. Institute of Physics Publishing, 2000.

[8] Z. D. Cheng, W. B. Russell, and P. M. Chaikin. Controlled growth of hard-sphere colloidal crystals. *Nature*, 401(6756):893–895, 1999.

[9] A. Donev, I. Cisse, D. Sachs, E. Variano, F. H. Stillinger, R. Connelly, S. Torquato, and P. M. Chaikin. Improving the density of jammed disordered packings using ellipsoids. *Science*, 303(5660):990–993, 2004.

[10] A. Donev, R. Connelly, F. H. Stillinger, and S. Torquato. Underconstrained jammed packings of nonspherical hard particles: Ellipses and ellipsoids. *Physical Review E*, 75(5):051304, May 2007.

[11] A. Donev, F. H. Stillinger, P. M. Chaikin, and S. Torquato. Unusually dense crystal packings of ellipsoids. *Physical Review Letters*, 92(255506), 2004.

[12] A. Donev, S. Torquato, and F. H. Stillinger. Neighbor list collision-driven molecular dynamics simulation for nonspherical hard particles. I. Algorithmic details. *Journal of Computational Physics*, 202(2):737–764, 2005.

[13] A. Donev, S. Torquato, and F. H. Stillinger. Neighbor list collision-driven molecular dynamics simulation for nonspherical hard particles.: II. Applications to ellipses and ellipsoids. *Journal of Computational Physics*, 202(2):765–793, 2005.

[14] S. F. Edwards. The Role of Entropy in the Specification of a Powder. In A. Mehta, editor, *Granular Matter: An Interdisciplinary Approach*, pages 121–140, New York, 1994. Springer–Verlag.

[15] Sh. I. Galiev and M. S. Lisafina. Numerical optimization methods for packing equal orthogonally oriented ellipses in a rectangular domain. *Computational Mathematics and Mathematical Physics*, 53(11):1748–1762, 2013.

[16] J. Q. Gan, Z. Y. Zhou, and A. B. Yu. Interparticle force analysis on the packing of fine ellipsoids. *Powder Technology*, 320:610 – 624, 2017.

[17] J.Q. Gan, A.B. Yu, and Z.Y. Zhou. {DEM} simulation on the packing of fine ellipsoids. *Chemical Engineering Science*, 156:64 – 76, 2016.

[18] M. Ganesan and M. J. Solomon. High-density equilibrium phases of colloidal ellipsoids by application of optically enhanced, direct current electric fields. *Soft Matter*, 13:3768–3776, 2017.

[19] Y. Han, A. M. Alsayed, M. Nobili, J. Zhang, T. C. Lubensky, and A. G. Yodh. Brownian Motion of an Ellipsoid. *Science*, 314(5799):626–630, 2006.

[20] H. M. Jaeger and S. R. Nagel. Physics of the granular state. *Science*, 255(5051):1523–1531, 1992.

[21] H. M. Jaeger, S. R. Nagel, and R. P. Behringer. Granular solids, liquids, and gases. *Reviews of Modern Physics*, 68(4):1259–1273, 1996.

[22] G. T. Johnson, L. Autin, M. Al-Alusi, D. S. Goodsell, M. F. Sanner, and A. J. Olson. cellPACK: a virtual mesoscope to model and visualize structural systems biology. *Nature Methods*, 12:85–91, 2015.

[23] J. Kallrath. Packing ellipsoids into volume-minimizing rectangular boxes. *Journal of Global Optimization*, 67(1-2):151–185, 2017.

[24] J. Kallrath and S. Rebennack. Cutting ellipses from area-minimizing rectangles. *Journal of Global Optimization*, 59(2):405–437, 2014.

[25] F. J. Kampas, I. Castillo, and J. D. Pintér. General ellipse packings in optimized regular polygons. Technical report, 2016. Available at `http://www.optimization-online.org/DB_HTML/2016/03/5348.html`.

[26] F. J. Kampas, J. D. Pintér, and I. Castillo. General ellipse packings in an optimized circle using embedded lagrange multipliers. Technical report, 2016. Available at `http://www.optimization-online.org/DB_HTML/2016/01/5293.html`.

[27] W. Man, A. Donev, F. H. Stillinger, M. T. Sullivan, W. B. Russel, D. Heeger, S. Inati, S. Torquato, and P. M. Chaikin. Experiments on random packings of ellipsoids. *Physical Review Letters*, 104(185501), 2010.

[28] J. M. Martínez and L. Martínez. Packing optimization for automated generation of complex system's initial configurations for molecular dynamics and docking. *Journal of Computational Chemistry*, 24(7):819–825, 2003.

[29] L. Martínez, R. Andrade, E. G. Birgin, and J. M. Martínez. Packmol: A package for building initial configurations for molecular dynamics simulations. *Journal of Computational Chemistry*, 30(13):2157–2164, 2009.

[30] D. Palachanis, A. Szabó, and R. M. H. Merks. Particle-based simulation of ellipse-shaped particle aggregation as a model for vascular network formation. *Computational Particle Mechanics*, 2(4):371–379, 2015.

[31] A. Pankratov, T. Romanova, and O. Khlud. Quasi-phi-functions in packing problem of ellipsoids. *Radioelectronics & Informatics*, 68(1):37–41, 2015.

[32] P. N. Pusey. Colloidal Suspensions. In J. P. Hansen and D. Levesque and J. Zinnjustin, editor, *Liquids, Freezing and Glass Transition, PT 2*, volume 51 of *Les Houches Summer School Session*, pages 763–942, Amsterdam, 1991. Elsevier Science Publishers B. V.

[33] M. D. Rintoul and S. Torquato. Metastability and crystallization in hard-sphere systems. *Physical Review Letters*, 77(20):4198–4201, 1996.

[34] Y. G. Stoyan, A. Pankratov, and T. Romanova. Quasi-phi-functions and optimal packing of ellipses. *Journal of Global Optimization*, 65(2):283–307, 2016.

[35] De-Wen Sun and M. Müller. Fabrication of ellipsoidal mesostructures in block copolymers via a step-shear deformation. *Macromolecules*, 51(1):275–281, 2018.

[36] C. Uhler and S. J. Wright. Packing Ellipsoids with Overlap. *SIAM Review*, 55(4):671–706, 2013.

[37] C. Xia, K. Zhu, Y. Cao, H. Sun, B. Kou, and Y. Wang. X-ray tomography study of the random packing structure of ellipsoids. *Soft Matter*, 10:990–996, 2014.

[38] X. Zhuang, Q. Wang, and H. Zhu. Effective properties of composites with periodic random packing of ellipsoids. *Materials*, 10(2), 2017.