

Dense packing of general-shaped particles using a minimization technique

Irvin Pérez Morales¹ · Roberto Roselló Valera¹ ·
Carlos Recarey Morfa¹ · Márcio Muniz de Farias²

Received: 28 August 2015 / Revised: 20 January 2016 / Accepted: 23 January 2016 / Published online: 9 February 2016
© OWZ 2016

Abstract The discrete element method (DEM) is a numerical method that has achieved general acceptance as an alternative tool to model discontinuous media, with a wide range of practical applications. Given that spheres are not always a suitable shape for DEM simulations, other particle shapes need to be used. However, for shapes different from spheres, there are not many advancing front packing algorithms, which are, in many cases, the best algorithms that allow obtaining an appropriate initial set of particles for a DEM simulation. This lack of advancing front packing algorithms for shapes different from spheres is mostly due to the difficulty of solving the problem of placing a mobile particle in contact with other two (in 2D) or three (in 3D) particles. In this paper, a new method for solving the problem of the particle in contact is proposed, and it is compared with the well-established wrappers method. It is shown that the new proposed method is a promising alternative for spherocylinders. For other shapes the formulation of the new method is shown to be correct, but it was clearly outperformed by

the wrappers method and the efficiency of the proposed formulation needs to be improved by optimizing the solution procedure.

Keywords DEM · Packing · Particle in contact · Optimization · Spherocylinders

1 Introduction

The discrete element method (DEM) is a numerical method that has achieved great recognition as an alternative tool to model discontinuous media. Several professional [1,2] and free software [3–5] are available for this purpose. Practical applications to a wide range of problems may be found in the recent literature [6–26]; however, most of them assume spherical particles. Spheres are simple to code and easy to use, but in many cases they cannot capture the basic dynamic mechanisms and therefore do not provide the most adequate geometric model for the particles. For instance, an individual disk (or sphere) will always roll down over a rough slope; however, a generic particle, such as a cluster of disks, may stay in static equilibrium, slide or roll, depending on the slope angle, the tangential friction coefficient, and the particle shape [27]. Particle types other than disks or spheres used in DEM include: clusters of spheres [1–3,28,29], which enable to model a wide range of different shapes, polyhedra [30], ellipses and ellipsoids [27], superquadrics [31] and spherocylinders [32], among others.

When DEM began to be applied some years ago, one of its major problems was the cost of obtaining an initial set of particles with a high volume (area) fraction, which is defined as the ratio of solid volume (area) to the total volume (area). Most of the initial applications used some kind of dynamic algorithm, in which a loose packing of non-

Electronic supplementary material The online version of this article (doi:10.1007/s40571-016-0103-x) contains supplementary material, which is available to authorized users.

✉ Irvin Pérez Morales
hotei.mat@gmail.com

Márcio Muniz de Farias
muniz@unb.br

¹ Centro de Investigación de Métodos Computacionales y Numéricos en la Ingeniería (CIMCNI), Aula UCLV-CIMNE, Universidad Central Marta Abreu de Las Villas (UCLV), Carretera a Camajuaní Km 5 1/2, C.P. 54830, Santa Clara, Villa Clara, Cuba

² Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, UNIVERSIDADE DE BRASILIA - ENC/FT/UNB. Asa Norte, Brasília, DF 70910-900, Brazil

Pseudocode 1 General steps that are common to all advancing front packing algorithms

1. Initialize the packing (usually with two or three particles).
2. Generate or select the particle to be packed.
3. Select an active front and determine the position at which the particle just touches the particles in the front.
4. Check if the particle at this position overlaps with any existing particles.
5. If no overlap occurs, accept the new particle and return to step 2 for the next particle. Otherwise go to step 6.
6. Reject the position and repeat step 3 for another active front.

overlapping particles is generated at random positions, and later the particles are rearranged by imposing some loading and boundary conditions [33–37]. Dynamic algorithms are computationally costly because they require a previous DEM simulation. Hence it was necessary to develop constructive packing methods, which are characterized by the sequential placement of particles at their final positions [38–45]. The class of constructive methods includes “advancing front algorithms”.

An advancing front is a group of particles in the surroundings of the evolving system of particles under generation. A group of previously placed particles lie inside the advancing front, while new particles are placed in contact with the outer particles of the front. The packing usually starts with a set of two or three particles at any given position, or one or two particles in contact with the walls defining the domain (walls are also considered particles in this context). These particles comprise the initial advancing front. Then a new particle is generated or chosen from a repertory of particles to be added to the packing. Next, the new particle is placed at a position that just touches other particles in the advancing front. Then the advancing front is updated and the process continues. Pseudocode 1 summarizes the basic steps of a generic advancing front algorithm [40].

In order to carry out step 3 of Pseudocode 1, the problem of placing a particle in contact with others must be solved (see Sect. 3). In this sense, some authors state that a higher local density is achieved if each new particle added to the media is placed in contact with other two existing particles in the two-dimensional (2D) case [41]. In the analogous three-dimensional (3D) case, the contact involves other three existing particles. For spherical particles of equal size, Kepler’s conjecture [46] is the solution for a maximum global volume fraction. Apollonius circle problem [47] is also related to placing particles in contact, but it is not exactly the problem that is solved further in this paper using minimization.

The problem of placing a particle in contact with other two (in 2D) or other three (in 3D) fixed particles has been solved using a direct approach, for some types of particles, as part of advancing front packing algorithms. Such direct approach is briefly explained in Sect. 3.1, and the types of

particles mentioned above are circles [41, 48], polygons [49], ellipses [49], and spheres [50]. The solution of the problem can be not unique, as will be seen in Sect. 3.

Even for simple shapes such as ellipses, the previously mentioned direct approach can be very difficult to apply, given the complexity of the analytical expressions that have to be obtained. That is why an alternative procedure based on minimization is presented in Sect. 3.2, together with a comparison with the direct approach, for several types of particles used in DEM.

Optimization techniques have been used as an auxiliary tool in the process of packing particles for DEM. For example, the position and dimension of particles can be modified iteratively in order to decrease the empty space in the domain, and in order to eliminate the gap between the domain boundary and the particles [51]. Also, the remaining heterogeneities in the packing can be removed, even without modifying the shape or dimensions of particles [38]. However, to the best of the authors’ knowledge, optimization has never been used before by other researchers in order to place a particle in contact with other two (in 2D) or other three (in 3D) fixed particles.

2 Advancing front particle packing algorithm

Let G be a sub-domain of the physical space \mathbb{R}^n ($n = 2$ for 2D or $n = 3$ for 3D) that should be filled with a dense set of particles whose dimensions \tilde{r} follow a given distribution. The dimensions, for instance, correspond to $\tilde{r} = r$ where r is the radius in case the particles are disks or spheres, and $\tilde{r} = (a, b)$ when the particles are ellipses having semi-axis a and b , respectively.

Alternative versions of advancing front algorithms have been proposed [38–41, 45], differing on details in each step of Pseudocode 1. The specific advancing front algorithm in 2D used by the authors, based on their previous work [42–44] is described in Pseudocode 2. This pseudocode, when adapted to 3D, may become more complex in practice, but remains essentially the same.

Step 2 of Pseudocode 2 is illustrated in Fig. 1, for the particular case of disks. Sub-domain G is represented by the

Pseudocode 2 Detailed steps of advancing front packing algorithm in 2D**Input:** Bounding geometry G (of finite area)**Output:** set E of particles**Step 1)** (Initialization)Create two initial tangent particles p_1 , and p_2 $E := \{p_1, p_2\}$ $C_{ext} = \{p_1, p_2\}$ **Step 2)** $\tilde{r} :=$ set of parameters that define the next particle except for its center**While** $C_{ext} \neq \emptyset$ **Step 2.1)** (Definition of local front) $p_0 :=$ select particle of C_{ext} at random $V :=$ all neighbor particles of p_0 **Step 2.2)****For all** $p \in V$ **Step 2.2.1)****If** $W(p_0, p, G, V, \tilde{r}) \neq \emptyset$ $p_{new} :=$ any element of $W(p_0, p, G, V, \tilde{r})$ $C_{ext} := C_{ext} \cup p_{new}$ $E := E \cup p_{new}$ $\tilde{r} :=$ set of parameters that define the next particle except for its center

Go to step 2.1

Step 2.3) $C_{ext} := C_{ext} - \{p_0\}$ **Step 3)** End

surrounding square box in bold black lines. The complete set of particles, represented by black and gray disks at a given time step of the procedure, is memorized in a set E . The most external particles of the packing comprise the present global advancing front and are represented by the dark disks in Fig. 1a. These particles are memorized in a set C_{ext} which is initialized with two tangent particles $C_{ext} := \{p_1, p_2\}$, in Step 1. The initial particles are generally positioned at the center or at a corner of the domain defined by the bounding geometry, G . Then the new particle p_{new} is generated (or chosen from an initial pool) with defining parameters \tilde{r} as represented in the lower right corner of Fig. 1a. In order to

select the position of p_{new} , a particle $p_0 \in C_{ext}$ is selected at random as pointed by the arrow in top right Fig. 1b, or in Fig. 1c in the following iteration of the example. This particle p_0 is a candidate to be tangent to p_{new} . The next task within substep 2.1 of the pseudocode is to find a local advancing front, defined by the sub-set $V \supseteq E$ with existing particles in the neighborhood of p_0 . Neighbor particles are in contact with p_0 or at distance not greater than the gap necessary to accommodate the new particle. These particles are marked with a little star in Fig. 1b, c.

The next substep 2.2 comes once p_0 and one of its neighbors p have been determined in the previous substep 2.1. Step

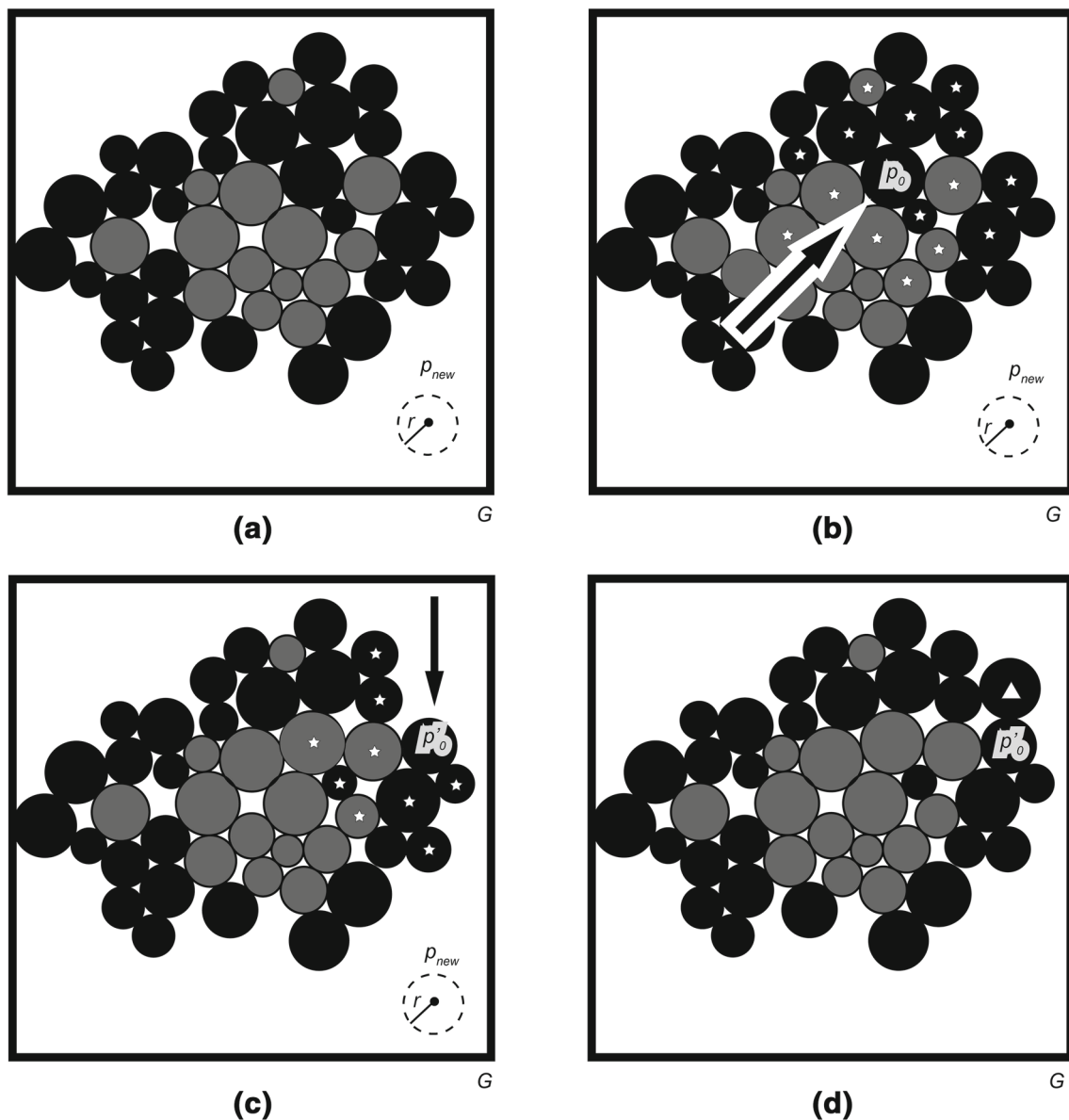


Fig. 1 Step 2 of Pseudocode 2, for the particular case of disks. Particles in contact with p_0 or at distance not greater than the gap necessary to accommodate the new particle, are marked with a *little star*. **a** configuration of a given step with C_{ext} in dark and new particle p_{new} represented

in the *lower right corner of the square*; **b** random selection of front member $p_0 \in C_{ext}$; **c** rejection of front member p_0 (now in gray color) and selection of new front member $p'_0 \in C_{ext}$; **d** final position of p_{new} tangent to p'_0

2.2 is a vital part of the algorithm and is based on a function that generates the set W of all possible new particles, in contact with p_0 and p (the entire Sect. 3 is dedicated to this issue). The rules set by this function are explained after the next paragraph, but it is conjectured that there exist at most two possible positions when particles are convex. If set W is empty then particle p_0 is removed from the advancing front C_{ext} and a new element $p'_0 \in C_{ext}$ is randomly selected as pointed by the arrow in Fig. 1c. If set W is not empty, one of its elements is chosen randomly and added to the packing. In the example of Fig. 1d such element is marked with a tri-

angle. Then the advancing front and total set of particles are updated and the process continues until the whole domain is filled.

It is important to notice that, in Pseudocode 2, the advancing front C_{ext} , which contains the external particles of the media, is completely determined by the initial assignment $C_{ext} := \{p_1, p_2\}$ of step 1; by the assignment $C_{ext} := C_{ext} \cup p_{new}$ of step 2.2.1, which takes place whenever a new particle is added to the media; and by the assignment, $C_{ext} := C_{ext} - \{p_0\}$ of step 2.3, which takes place whenever it is not possible to place a new particle in contact with

particle p_0 . This way of defining C_{ext} may eventually leave some “external” particles in the interior of the media, as can be seen in Fig. 1, but they will be removed as soon as they are selected in substep 2.1, because it will not be possible to place another particle in contact with them.

In order to define the set W of Pseudocode 2, let p_1 and p_2 be two particles already placed in the sub-domain G . Then $W(p_1, p_2, G, V, \tilde{r})$ is defined as a set of particles (or the empty set) such that: (a) they are determined by parameters \tilde{r} except for their position; (b) they are in outer contact with p_1 and p_2 simultaneously; and (c) they are completely contained in G and do not overlap with any element of V . It can be said that this set $W(p_1, p_2, G, V, \tilde{r})$ has at most two elements, as can be seen in Sect. 3. A new way to calculate such elements is presented in the next section.

Finally, some words about the complexity of Pseudocode 2 should be said. It has been proven that it has a complexity of order $O(n)$, where n is the total number of particles. The interested reader can verify the derivation of this result in [42]. The comparison of Pseudocode 2 with respect to dynamic methods, for the case of spheres, can also be seen in the latter reference.

3 Construction of a particle in contact with others

Let $p[\mathbf{c}]$ denote a particle in \mathbb{R}^n such that $\mathbf{c} \in \mathbb{R}^n$ is a point with the property that any rotation or translation applied to $p[\mathbf{c}]$ must also be applied to \mathbf{c} and vice versa. Now consider the following problem:

Placing a particle in contact with others

Let p_1, \dots, p_n be n fixed particles in \mathbb{R}^n ($n \in \{2, 3\}$), and let $p_{\text{mob}}[\mathbf{c}]$ be another particle that must be translated, without making rotations, in such a way that $p_{\text{mob}}[\mathbf{c}]$ be in outer contact with all the particles p_i simultaneously, $i = \overline{1, n}$, without overlapping with any of them. Find the points \mathbf{c} that satisfy this condition. From now on, particle $p_{\text{mob}}[\mathbf{c}]$ will be referred to as the “mobile particle”, in order to simplify the terminology, despite it is not actually moving. The phrase “without making rotations” can be better understood by looking at Fig. 4. The mobile particles there, $E_{\text{mob}}[\mathbf{c}]$ and $S_{\text{mob}}[\mathbf{c}]$, change their positions but preserve their inclination, in such a way that they are not rotated.

It has been verified in practice that in the general case, the problem of placing a particle in contact with others has at most two solutions when particles p_1, \dots, p_n and $p_{\text{mob}}[\mathbf{c}]$ are convex and are close enough to each other (Fig. 2a). This can degenerate to only one solution when the mobile particle fits exactly in the gap between the fixed particles (Fig. 2b). Obviously, there is no solution when p_1, \dots, p_n are apart from each other by a distance greater than the larger Feret dimension of the particle to be placed (Fig. 2c).

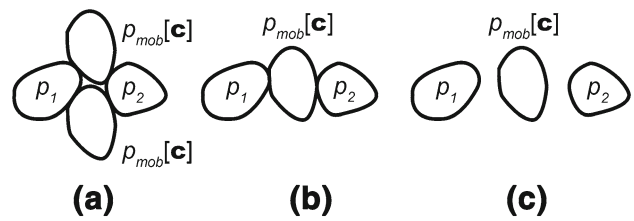


Fig. 2 Number of solutions for the problem of placing a particle in contact with others

In the case of spherical particles and clusters of spheres it is possible to develop an analytical solution for the problem proposed above based on the concept of wrapper’s intersection [38,52,53], explained in the following Sect. 3.1. However, the analytical procedures may become too cumbersome in the case of polyhedra and there is no analytical solution for particles with general shape. An alternative methodology that may be eventually generalized for these cases is explored in Sect. 3.2. The two solutions are compared when possible.

3.1 Wrappers intersection method for placing a particle in contact with others

Let p_{fix} be a fixed particle and $p_{\text{mob}}[\mathbf{c}]$ be a mobile particle. The locus defined by all points \mathbf{c} such that p_{fix} and $p_{\text{mob}}[\mathbf{c}]$ are in outer contact, will be called *wrapper*.

In two dimensions, if the fixed and mobile particles are circles with radii equal to r_{fix} and r_{mob} , respectively, then the corresponding wrapper is obviously a circle with radius $r_{\text{fix}} + r_{\text{mob}}$ (Fig. 3a). When the two particles are described by polygons, the wrapper is a polygon with twice the number of sides of the fixed one (Fig. 3b). Similar geometries are generated in the three-dimensional case.

The method of wrappers intersection in \mathbb{R}^n , for translating a mobile particle $p_{\text{mob}}[\mathbf{c}]$ in such a way that it is in outer contact with other fixed particles p_1, \dots, p_n , without overlapping, consists of finding the loci described by \mathbf{c} when sliding $p_{\text{mob}}[\mathbf{c}]$ around each of the fixed particles, then finding the intersections of these loci, and finally translating \mathbf{c} to make it coincide with these intersections. It is important to notice that the choice of \mathbf{c} is irrelevant as long as its position remains unchanged with respect to particle $p_{\text{mob}}[\mathbf{c}]$.

Figure 3a shows an example of a mobile circle $C_{\text{mob}}[\mathbf{c}]$ of radius r_{mob} placed in contact with the fixed circles C_1 and C_2 , of center (radii) equal to \mathbf{c}_1 (r_1) and \mathbf{c}_2 (r_2), respectively. In this case, the point \mathbf{c} is taken as the center of $C_{\text{mob}}[\mathbf{c}]$, and it can be seen that the wrappers obtained by sliding $C_{\text{mob}}[\mathbf{c}]$ around C_1 and C_2 are the circles C'_1 and C'_2 , concentric with C_1 and C_2 , respectively, and having radii equal to $r_1 + r_{\text{mob}}$ and $r_2 + r_{\text{mob}}$, respectively. The solution of the problem are the circles C_{31} and C_{32} . Similarly, Fig. 3b shows the case of the fixed squares P_1 and P_2 , which yield the wrappers P'_1 and

Fig. 3 Particle in contact with other two, with its position obtained by the wrappers intersection method. **a** Case of circles; **b** Case of polygons

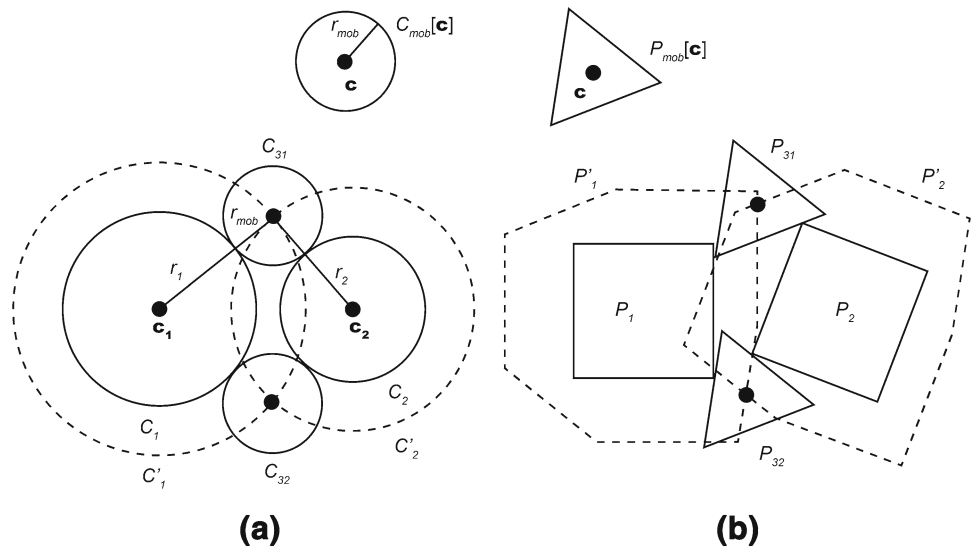
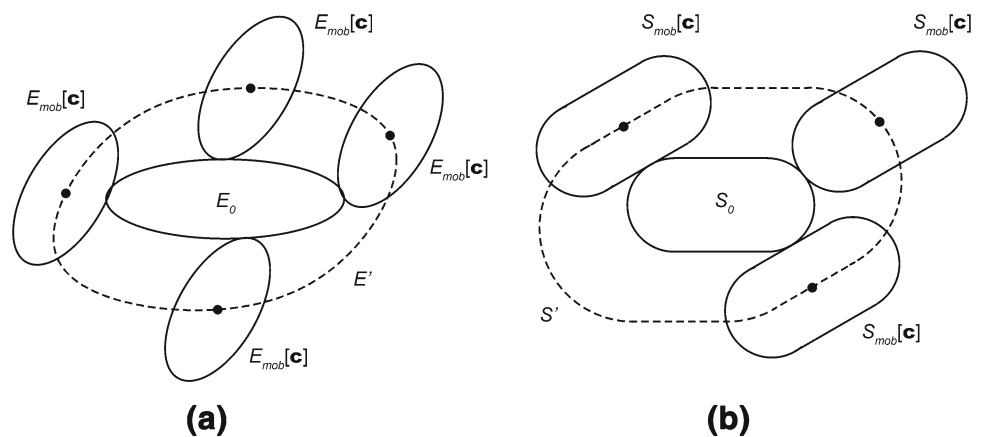


Fig. 4 Examples of wrappers. **a** Ellipses; **b** Spherocylinders



P'_2 after sliding the mobile triangle $P_{mob} [c]$ around them. In this case, the solution is the triangles P_{31} and P_{32} . Notice that $P_{mob} [c]$, P_{31} , and P_{32} are all identical except for their position.

Despite the simplicity of wrappers in Fig. 3, other cases can be quite complex. Consider, for example, the wrapper E' formed by the center c of an ellipse $E_{mob} [c]$, when sliding it around another ellipse E_0 in horizontal position, centered at the origin of coordinates. It can be seen in Fig. 4a that the wrapper E' is not even an ellipse in the general case. The points $(x_c, y_c) \in E'$ are given by expression

$$(x_c, y_c) = (a \cos \theta + u\lambda, b \sin \theta + v\lambda), \tag{1}$$

where a and b are positive constants, $\theta \in [0, 2\pi)$ and u, v , and λ depend on θ (see [40] for more details).

The wrapper S' corresponding to a fixed and a mobile spherocylinders S_0 and $S_{mob} [c]$, respectively, is not defined straightforwardly either. It consists of a set of line segments and circumference arcs interleaved, which is more complex than just a spherocylinder. The analogous wrappers in 3D for

ellipsoids and spherocylinders are even more complicated, let alone finding their intersection for placing the mobile particle in contact with the fixed ones. Greater difficulties are expected for other shapes such as clusters of circles or spheres, polyhedra, etc. In some of these cases, the theory explained in the next section can be an alternative.

3.2 Potential minimization method for placing a particle in contact with others

The method corresponding to this section uses an optimization approach to solve the problem of the particle in contact. In some cases, it can be easier to apply than wrappers intersection because it only requires the definition of a continuous function $\omega(p_1, p_2)$ for a pair of particles (p_1, p_2) such that:

$$\omega(p_1, p_2) \begin{cases} \leq 0 & \text{if } p_1 \cap p_2 \neq \emptyset \\ > 0 & \text{otherwise} \end{cases} \tag{2}$$

Function $\omega(p_1, p_2)$ is a measure of the gap between the surfaces of the two particles. Condition (2) implies that p_1

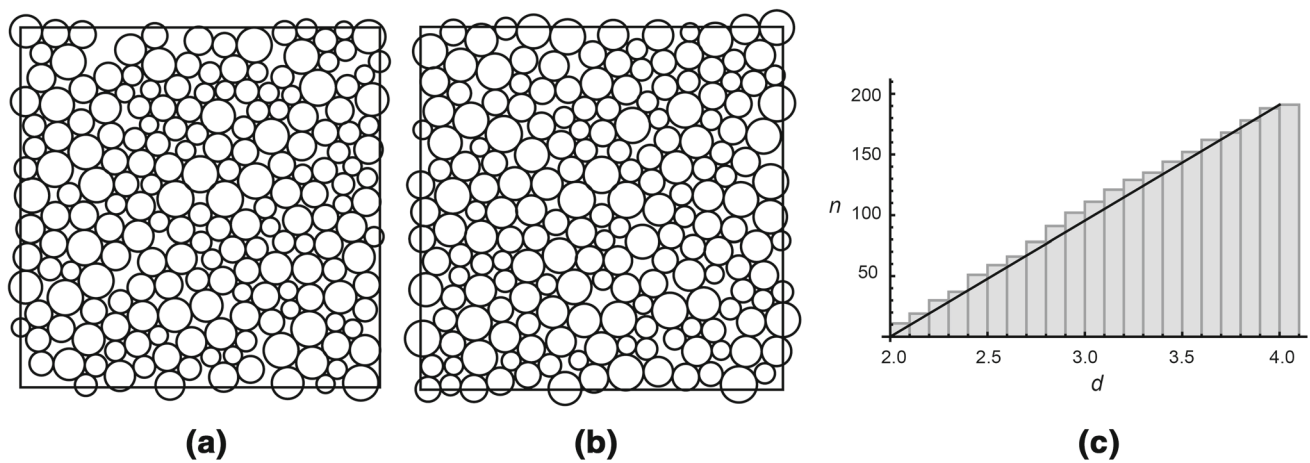


Fig. 5 Group of disks and particle size distribution. **a** packing generated with wrappers; **b** packing generated with minimization; **c** Cumulative histogram of disks’ diameters d versus number of particles n of the packing generated with minimization

and p_2 are in outer contact without overlapping if and only if $\omega(p_1, p_2) = 0$. Function ω is usually not unique. Explicit formulas for function $\omega(p_1, p_2)$ will be given in next sections for the cases of disks, ellipses, and superquadrics in 2D and for the cases of spheres and spherocylinders in 3D.

Once the gap function ω has been chosen, the solution to the problem of placing a particle in contact with others can be obtained by solving the following optimization problem:

$$\begin{aligned} &\text{minimize} && |\omega(p_1, p_{\text{mob}}[\mathbf{x}])| + \dots + |\omega(p_n, p_{\text{mob}}[\mathbf{x}])| \\ &\text{subject to} && \mathbf{x} \in \mathbb{R}^n \end{aligned} \tag{3}$$

Condition (3) means that a particle is in simultaneous outer contact with other two particles in 2D (or three in 3D) when the sum of the gaps is minimized (in this case the minimum should be zero). Since two solutions for problem (3) are being searched in most cases (see Fig. 2), such problem has to be solved twice each time in practice, with an additional restriction that indicates which solution is being searched. Such restriction is based on the fact that the centers of the two solution particles usually lie on different half-spaces defined by the centers of the fixed particles. In the 2D case, the half-spaces are the half-planes determined by the line joining the centers of the two fixed particles, while in the 3D case the half-spaces are determined by the plane containing the centers of the three fixed particles. In order to solve the minimization problem the authors used the Nelder–Mead method [54] already validated and included in a commercial software for the 2D cases, and the same method available in a free C++ library [55], for the 3D cases. This method was initially chosen because it requires relatively few evaluations to reach the global minimum, and does not require derivative information of the objective function. More details about the optimization process are given in Sect. 3.3.

3.2.1 Circles or spheres

For any two circles or spheres p_1 and p_2 , $\omega(p_1, p_2)$ can be defined by the equality

$$\omega(p_1, p_2) = \|\mathbf{c}_1 - \mathbf{c}_2\|^2 - (r_1 + r_2)^2, \tag{4}$$

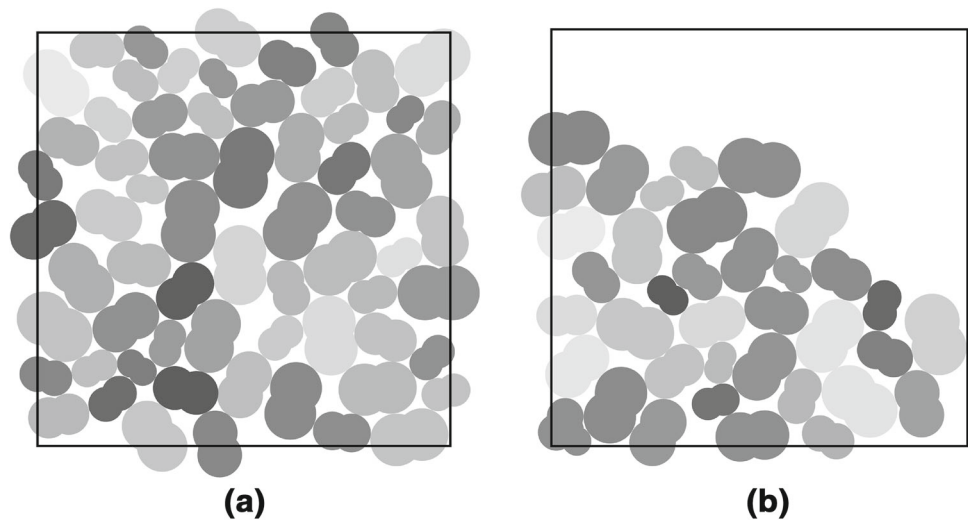
where \mathbf{c}_1 and \mathbf{c}_2 are the coordinates of centers of the particles and r_1 and r_2 their radii, respectively. It is possible to verify that expression (4) satisfies (2).

In order to check the proposed function and minimization procedure, the authors present the case of packing a square box with 40 unit side with disks whose radii follow a random continuous uniform distribution in the interval [1, 2]. The notation $U[a, b]$ will be used for the random continuous uniform distribution in the interval $[a, b]$.

Figure 5a shows the final packing of 195 disks using the wrappers technique described in [40]. The average placing time using this direct approach was 40.94 particles per second. The result using minimization is shown in Fig. 5b and this final configuration of 191 disks was achieved at a placing rate of 0.037 particles per second. So, it can be seen that in the case of disks, the direct approach is much faster than the minimization technique used. The area fraction in both cases is very high: 80.01 % for the packing of Fig. 5a and 82.00 % for the packing of Fig. 5b.

A histogram showing the cumulative diameter distribution d versus the number of particles n of the latter packing is shown in Fig. 5c. It can be seen that it corresponds to the distribution $U[2, 4]$. The histograms corresponding to all the other packings in the paper are omitted since they are almost identical to this one. By the way, one of the advantages of Pseudocode 2 is that the prescribed particle size distribution remains unchanged. This is because, for example, in the case of disks, if a random radius r has been generated from the

Fig. 6 Group of clusters.
a packing generated with wrappers; **b** packing generated with minimization



prescribed distribution, no other radius will be generated until a disk of radius r has been added to the packing.

3.2.2 Clusters of disks

The expression $\omega(p_1, p_2)$ for a pair of clusters of disks or spheres, p_1 and p_2 , is also relatively simple. It can be defined as the minimum of the evaluations of the expression (4) applied to all pairs of disks or spheres comprised in the clusters p_1 and p_2 :

$$\omega(p_1, p_2) = \min \left\{ \|\mathbf{c}_{1i} - \mathbf{c}_{2j}\|^2 - (r_{1i} + r_{2j})^2 : i = \overline{1, n_1} \text{ and } j = \overline{1, n_2} \right\} \quad (5)$$

where \mathbf{c}_{1i} is the center of the i -th disk or sphere comprised in the cluster p_1 , \mathbf{c}_{2j} is the center of the j -th disk or sphere comprised in p_2 , and n_1 and n_2 are the number of disks or spheres comprised in clusters p_1 and p_2 , respectively. The radii r_{1i} and r_{2j} have an analogous definition.

The procedure is tested for clusters formed by two disks each, with circumscribed radii distribution $U[1, 2]$. The aspect ratio of all clusters is $2/3$. Figure 6a shows a packing of 65 clusters obtained with the direct geometric approach, while Fig. 6b shows a packing of 38 clusters obtained with the minimization procedure using a commercial program. Both packings are inside a square of side 20 units. Again the direct approach was much faster, achieving a placement rate of 9.05 particles per second against only 0.023 particles per second using minimization.

Furthermore, in several instances the minimization of expression (3) was not achieved, and several particles inside the domain could not be surrounded by others, therefore a few particles were misplaced, thus producing an invalid packing. In theory, for any particle shape, expression (3) can always be used to solve the problem of the particle in contact stated

in the beginning of Sect. 3. However, despite the only difference between expressions (4) (for disks) and (5) (for clusters of disks), being the min operator, the Nelder–Mead method used by authors did not succeed in finding the global minima for all instances of (3) when the particles are clusters of disks. So, in this respect, more research is needed in order to find suitable optimization methods calibrated with appropriate parameters, and used with better initial points.

3.2.3 Ellipses in 2D

The expression of $\omega(p_1, p_2)$, for any two elliptical particles p_1 and p_2 can be given by

$$\omega(p_1, p_2) = \min_{t \in [0, 2\pi[} P_1(\tilde{p}_2(t)) \quad (6)$$

where $\tilde{p}_2(t) \in \mathbb{R}^2$, with $t \in [0, 2\pi[$, is a parametric representation of p_2 , and $P_1(\mathbf{x}) = 0$, for $\mathbf{x} = (x, y) \in \mathbb{R}^2$, is a cartesian representation of p_1 . $P_1(x, y)$ and $\tilde{p}_2(t)$ are given by the following expressions, respectively:

$$P_1(x, y) = A_1x^2 + B_1y^2 + C_1xy + D_1x + E_1y + F_1, \quad (7)$$

$$\tilde{p}_2(t) = M_{\theta_2} \begin{pmatrix} a_2 \cos t \\ b_2 \sin t \end{pmatrix} + \mathbf{c}_2, \quad (8)$$

where A_1, B_1, a_2 , and b_2 are positive real numbers, $C_1, D_1, E_1, F_1 \in \mathbb{R}$, $\mathbf{c}_2 \in \mathbb{R}^2$, and M_{θ_2} is a matrix representing a rotation of angle θ_2 with respect to the origin of coordinates in 2D.

Expression (6) was derived from the known fact that $P_1(\mathbf{x}) < 0$ ($P_1(\mathbf{x}) > 0$) if and only if \mathbf{x} lies inside (outside) P_1 , and that $P_1(\mathbf{x}) = 0$ if and only if \mathbf{x} lies over P_1 . There are other alternatives for evaluating (6) using Lagrange multipliers [27].

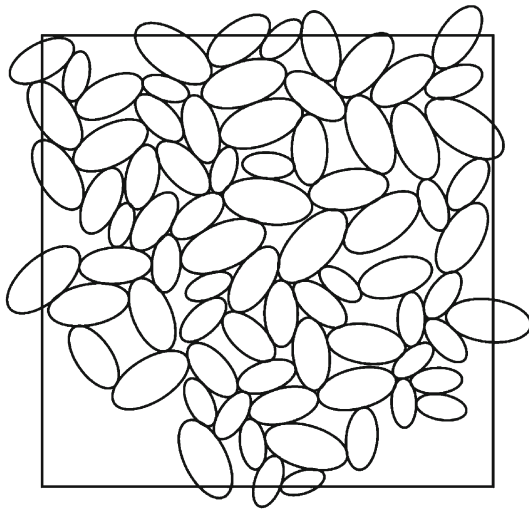


Fig. 7 Packing of ellipses generated with wrappers

One initial packing of 77 ellipses inside a square of side 20 units, was obtained with the direct approach (Fig. 7). Such ellipses have an aspect ratio equal to 0.5, and their major semi-axis follow the distribution $U [1, 2]$. Due to numerical complications in the computational implementation of expression (6), some solutions were lost, which resulted in large gaps between the particles and the lower part of the domain boundary. The packing took 1647 s to be generated. On the other hand, the minimization approach was so slow and inaccurate for ellipses, that a simple instance of two ellipses in contact with other two (Fig. 13d) took 3484.8 s to be generated.

3.2.4 Superquadrics

Another case of preliminary application of the potential minimization method will be shown with superquadrics. In mathematics, the superquadrics or super-quadrics (also superquadratics) are a family of geometric shapes defined by formulas that resemble those of ellipsoids and other quadrics, except that the squaring operations are replaced by arbitrary powers. The canonical equation for a superquadric [56] in two dimensions is given by the following expression:

$$f(x, y) = \left| \frac{x}{a} \right|^\alpha + \left| \frac{y}{b} \right|^\beta - 1 = 0, \tag{9}$$

with $\alpha, \beta > 0$. Depending on the values of α and β , different shapes can be obtained. Figure 8 shows three examples.

One also has that $f(x, y) < 0$ for points inside the curve, $f(x, y) = 0$ for points lying on the curve, and $f(x, y) > 0$ for points outside the curve. Taking advantage on this property of f , the function ω applied to two superquadrics can be defined analogously to the case of ellipses, by taking the minimum evaluation of all the points of one of them into

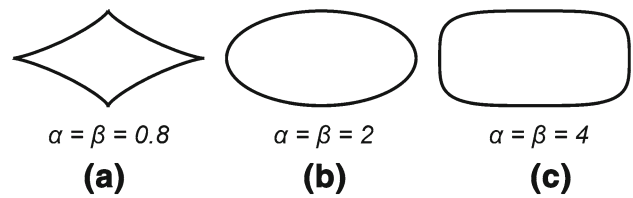


Fig. 8 Several shapes can be obtained depending on values of α and β in (9). **a** sharp superquadric; **b** ellipse; **c** rectangle-shaped superquadric

the cartesian equation of the other. This idea has been previously used in other works in order to, for example, check the intersection of two ellipses [27]. There also exist distance calculation methods [57,58] that can be used in order to evaluate function ω applied to two superquadrics.

The implementation of this particle construction method does not work properly yet, and the high computational cost makes practical applications almost impossible for now. Figure 13 shows two examples. For sharp superquadrics (Fig. 13e), the minimization algorithm failed in the upper solution, while for squared superquadrics (Fig. 13f), the minimization algorithm failed in the lower solution. The generation time of the two sharp and rectangle-shaped superquadrics were 462.78 and 1401.1 s, respectively.

3.2.5 Spheres

This and the following section are about bodies in 3D, and all the packings shown in them were generated with implementations in C++, unlike packings shown in previous sections, which were all obtained from implementations in a commercial command interpreter.

Equation (3) used for disks in 2D also applies for spheres in 3D. Two preliminary sphere packings were generated (Fig. 9), with radii following the $U [1, 2]$ distribution, inside cubes of side equal to 40 units. As expected, following the results obtained with disks, the direct approach also outperforms minimization in the case of spheres. The generation speed with the direct approach was more than 20 times faster than with minimization. The volume fraction in the packing obtained with the direct approach in Fig. 9a is 58.9%, while the volume fraction in the packing obtained in Fig. 9b with minimization is 60.1%. Both values of volume fraction can be considered high for the radii distribution adopted [42].

3.2.6 Spherocylinders

A spherocylinder is a capsule-like body determined by a line segment and a positive real number called radius, and is defined as the set of all points that lie at a distance from the segment equal to or smaller than the radius. For this type of particle, the potential minimization method is perhaps the

Fig. 9 Group of spheres. **a** packing of 1973 spheres generated with wrappers; **b** packing of 1945 spheres generated with minimization

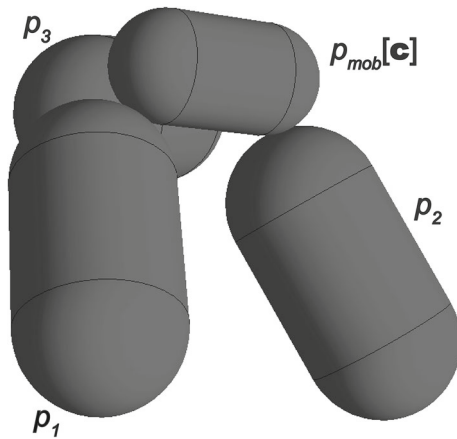
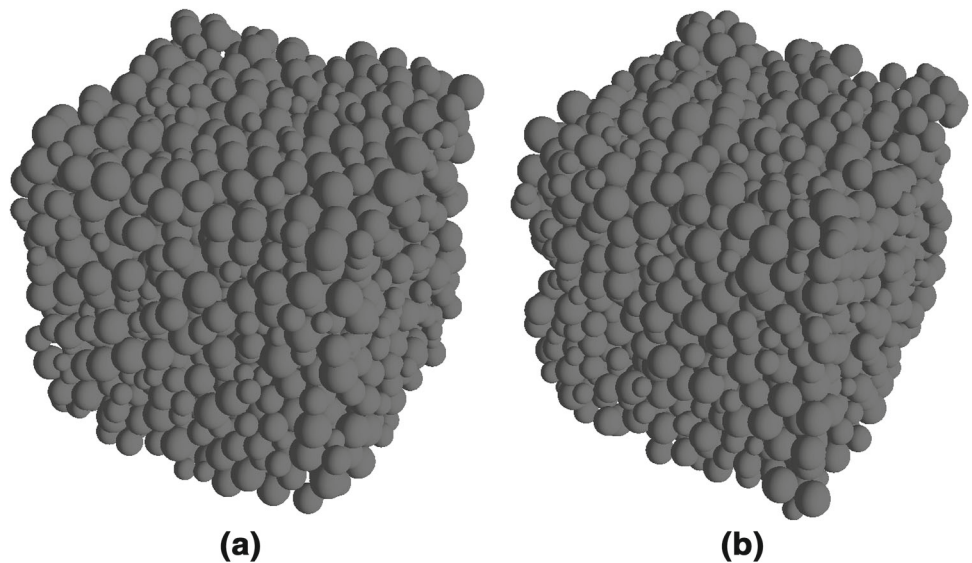


Fig. 10 Spherocylinder built in contact with other three using the potential minimization method

most suitable in order to build the particle in contact. If p_1 and p_2 are two spherocylinders defined by segments s_1 and s_2 , and radii r_1 and r_2 , respectively, then the function $\omega(p_1, p_2)$ can be defined by the following equality:

$$\omega(p_1, p_2) = d_1(s_1, s_2)^2 - (r_1 + r_2)^2, \quad (10)$$

where $d_1(s_1, s_2) = \inf \{d(\lambda_1, \lambda_2) : \lambda_1 \in s_1, \lambda_2 \in s_2\}$ is the distance between segments s_1 and s_2 (a procedure for calculating the distance between two line segments can be seen in [59]), being d the usual distance in \mathbb{R}^n . An example of contacting spherocylinder $p_{\text{mobl}}[c]$ obtained by solving (3) after substituting ω by (10) can be seen in Fig. 10.

Given that for spherocylinders the wrappers were very complicated to describe, especially in 3D, a preliminary comparison in 2D between wrappers and minimization (Fig. 11) was carried out by approximating spherocylinders with clusters of 4 disks each in the case of wrappers. In [45] the reader

can find approximations of some simple shapes with clusters. A packing of spherocylinders in 3D was also obtained (Fig. 12).

The two packings can be seen in Fig. 11. In both packings, contained in squares of side equal to 20 units, the particles have an aspect ratio equal to 0.5, and circumscribed radii following the $U[1, 2]$ distribution. The packing of 73 clusters (Fig. 11a), obtained by wrappers intersection, was generated at a speed of 1.05 particles per second, while the packing of 77 spherocylinders (Fig. 11b) obtained by minimization, was generated at a speed of 0.0094 particles per second. This suggests that if the generation of spherocylinders using wrappers was possible, it would be by far faster than generation using minimization. However, as was already mentioned, the formulation of wrappers intersection for spherocylinders, especially in 3D, is not a trivial task. The area fractions of the packings were equal to 77.21 and 82.20% for the cases of Fig. 11a and b, respectively.

The packing of spherocylinders generated in 3D can be seen in Fig. 12b. It comprises 5901 particles generated at a speed of 3.90 particles per second, and is contained within a cube of side equal to 40 units. This speed is so much higher than the analogous speed in 2D, because in this case an efficient implementation in C++ was used. Each particle has an aspect ratio of 0.5, and the circumscribed radii of the particles follow the $U[1, 2]$ distribution. The volume fraction of the packing, measured with respect to the circumscribed box, is equal to 45.77%.

For the sake of comparison, another packing of 4778 spherocylinders approximated with clusters was generated (Fig. 12a). Given that in this case the generation speed with wrappers was very slow, an approximate wrappers method was implemented, producing a packing with a much less volume fraction equal to 36.68%, but generated at the convenient speed of 172.30 particles per second. This packing is

Fig. 11 Comparison between wrappers and minimization in 2D spherocylinders. **a** packing of spherocylinders approximated with clusters of disks and generated with wrappers intersection; **b** packing of spherocylinders generated with minimization

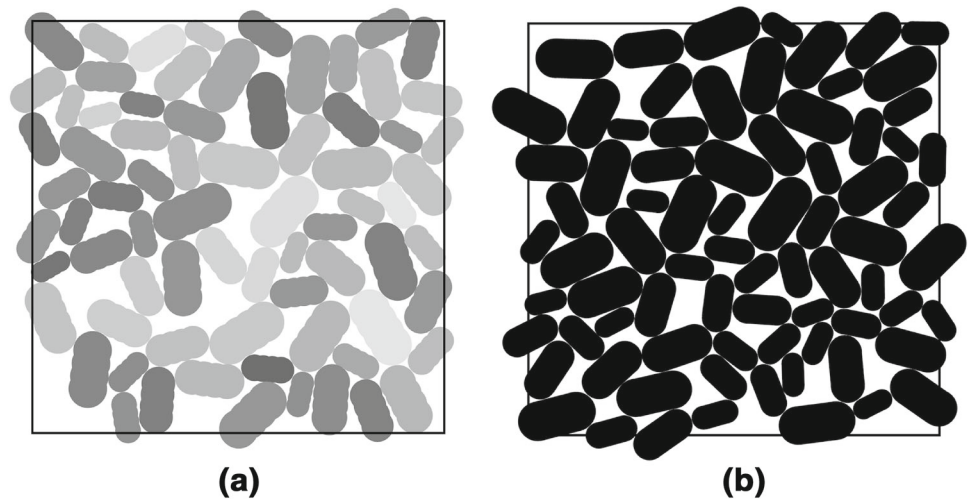
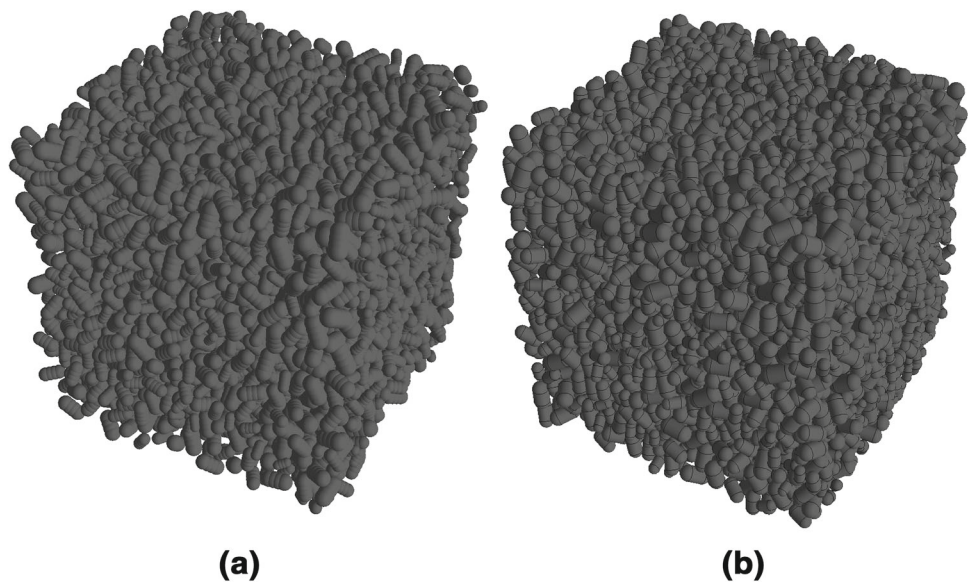


Fig. 12 Comparison between wrappers and minimization in 3D spherocylinders. **a** Packing of 4778 spherocylinders approximated with clusters; **b** Packing of 5901 spherocylinders generated with minimization



also contained in a cube of side equal to 40 units. It is interesting that not only in this case, but also in all packings presented in this paper, the volume fraction of packings obtained with minimization is higher than the volume fraction of analogous packings obtained using wrappers.

3.3 Performance of the optimization methods

Three global optimization methods were tested for several types of particles, in finding the two global minima of problem (3). The methods were Nelder–Mead [54], Differential Evolution [60], and Simulated Annealing [61]. The results obtained with Nelder–Mead in a preliminary test in 2D with circles, spherocylinders, clusters of three circles, ellipses, sharp superquadrics, and squared superquadrics can be seen in Fig. 13, where the points obtained during the iteration process are joined in dashed lines. In this figure, all particles except for the circles, have aspect ratio equal to 0.5,

and circumscribed radius equal to 2 units. The initial points of the optimization algorithm were set in the line bisecting the segment joining the centers of the fixed particles. It can be seen that one of the two solutions are incorrect for the cases of ellipses and superquadrics (Fig. 13d, e, f). The solutions for circles, spherocylinders, and clusters of three circles (Fig. 13a, b, c) are not only correct, but were also obtained in shorter times, as can be seen in Figs. 14 and 15. In these two figures, the lines represent, for each iteration point, the time consumed in seconds (horizontal axis) versus the distance to the converged solution (vertical axis). The results obtained with Differential Evolution and Simulated Annealing were less accurate than those shown in Fig. 13 for Nelder–Mead, and the solution times required by the latter are shorter in all cases. Figure 16 shows the time comparison for the case of spherocylindrical particles, for which the Nelder–Mead has the shortest times. The same happens for the other particles but the corresponding

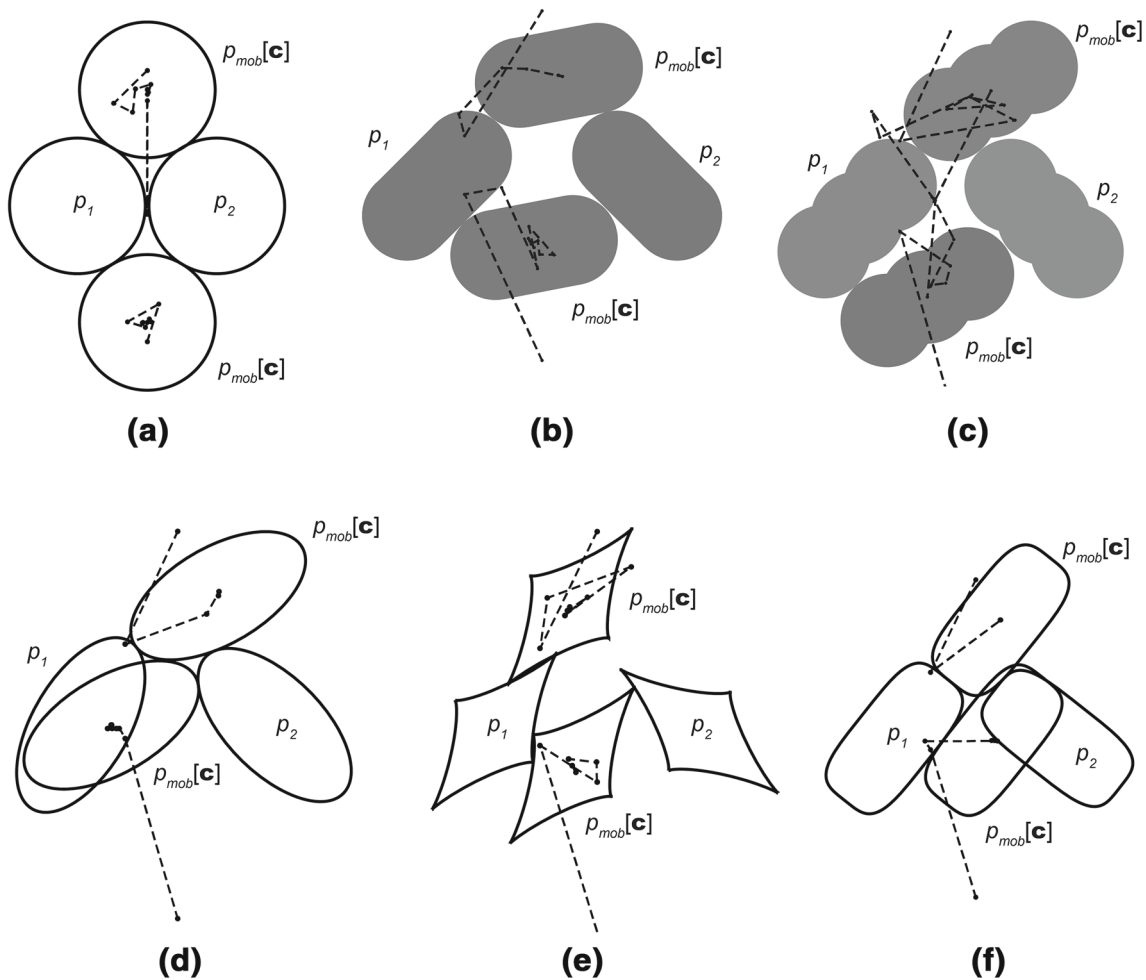


Fig. 13 Solution of problem (3) with Nelder–Mead for circles (a), spherocylinders (b), clusters of three circles (c), ellipses (d), sharp superquadrics (e), and squared superquadrics (f)

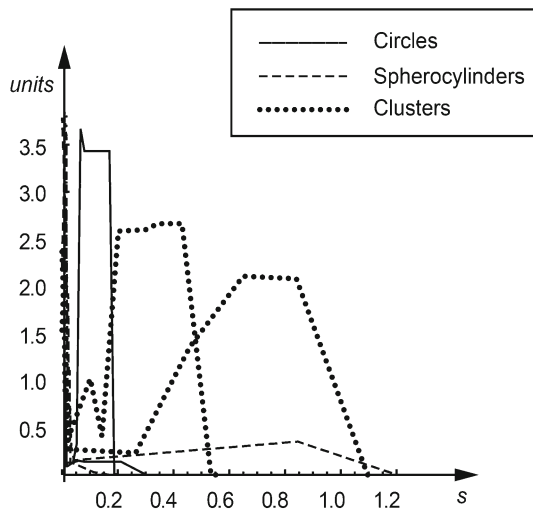


Fig. 14 Curves representing CPU time in seconds versus distance to the solution, for the iteration points of the Nelder–Mead optimization, for the cases of circles, spherocylinders, and clusters of three circles shown in Fig. 13a, b, c

figures are omitted for the sake of brevity. Two curves correspond to each legend entry in Figs. 14, 15, 16, because there exist two solutions $p_{mob}[c]$ in each of the six subplots in Fig. 13.

4 Conclusion

The problem of placing a mobile particle in contact with other two (in 2D) or three (in 3D), as part of advancing front particle packing algorithms in the context of DEM simulations, has been little studied in the available literature. The geometric solution of such problem only exists for a few particle shapes, and is only based on the direct approach.

In this paper, a new solution method for the problem, based on minimization, has been proposed, and some preliminary comparisons with respect to the existing direct approach have been carried out, for several particle shapes used in DEM.

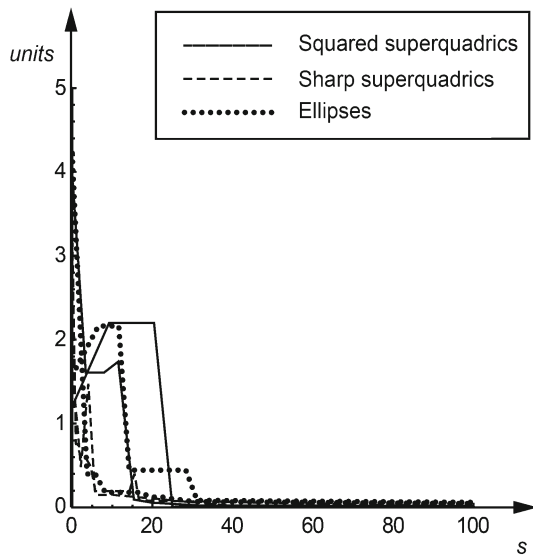


Fig. 15 Curves representing CPU time in seconds versus distance to the solution, for the iteration points of the Nelder–Mead optimization, for the cases of ellipses, sharp superquadrics, and squared superquadrics shown in Fig. 13d, e, f

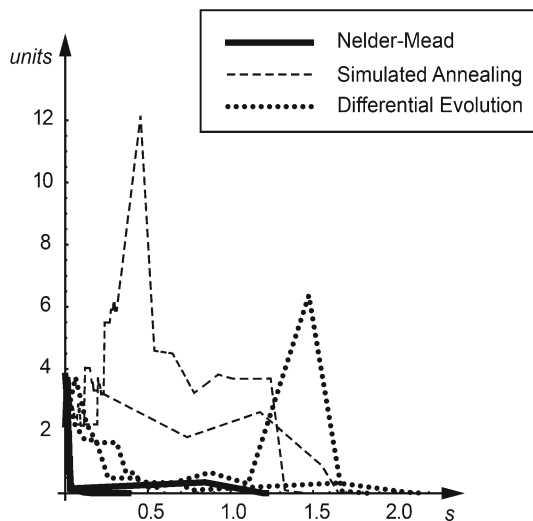


Fig. 16 Curves representing CPU time in seconds versus distance to the solution, for the iteration points of the spherocylindrical particles, for the methods Nelder–Mead, Simulated Annealing, and Differential Evolution shown in Fig. 13b

The minimization approach to build a mobile particle in contact with other two (in 2D) or other three (in 3D) particles has been applied to eight different shapes, and compared to the direct approach whenever possible. For all of the shapes, the correctness of the minimization formulation could be observed. Moreover, for all packings obtained with minimization, a higher volume or area fraction was obtained too.

For almost all cases analyzed herein the direct wrappers approach was by far the most efficient. However, for the case of spherocylinders, which have complex wrappers, the min-

imization approach can be a valid alternative. The efficiency of the proposed approach must be increased by improving the optimization process, in order to obtain results of practical value.

Acknowledgements The authors are deeply grateful to the CAPES project 208/13.

References

1. DEM Solutions. Engineering with confidence (2014). <http://www.dem-solutions.com>. Accessed 14 July 2014
2. ITASCA Consulting Group, Inc. (2014) PFC general purpose distinct-element modeling framework. <http://www.itascacg.com/software/pfc>. Accessed 14 July 2014
3. Smilauer V, Catalano E, Chareyre B, Dorofenko S, Duriez J, Gladky A, Kozicki J, Modenese C, Scholtès L, Sibille L, Stránský J, Thoeni K (2010) Yade Documentation, 1st edn. The Yade Project
4. LIGGGHTS Open Source Discrete Element Method Particle Simulation Code. <http://www.cfdem.com/liggghts-open-source-discrete-element-method-particle-simulation-code>. Accessed Nov 2015
5. ESyS-Particle. <https://launchpad.net/esys-particle>. Accessed Nov 2015
6. Abraham CL, Maas SA, Weiss JA, Ellis BJ, Peters CL, Anderson AE (2013) A new discrete element analysis method for predicting hip joint contact stresses. *J Biomech* 46(6):1121–1127
7. Dang HK, Meguid MA (2013) An efficient finite-discrete element method for quasi-static nonlinear soil-structure interaction problems. *Int J Numer Anal Methods Geomech* 37(2):130–149
8. Hahn M, Bouriga M, Kröplin B-H, Wallmersperger T (2013) Life time prediction of metallic materials with the discrete-element-method. *Comput Mater Sci* 71:146–156
9. Hassanpour A, Pasha M, Susana L, Rahmanian N, Santomaso AC, Ghadiri M (2013) Analysis of seeded granulation in high shear granulators by discrete element method. *Powder Technol* 238:50–55
10. Li X, Yu H-S, Li X-S (2013) A virtual experiment technique on the elementary behaviour of granular materials with discrete element method. *Int J Numer Anal Methods Geomech* 37(1):75–96
11. Lim K-W, Andrade JE (2014) Granular element method for three-dimensional discrete element calculations. *Int J Numer Anal Methods Geomech* 38:167–188
12. Nakamura H, Fujii H, Watano S (2013) Scale-up of high shear mixer-granulator based on discrete element analysis. *Powder Technol* 236:149–156
13. Pennec F, Alzina A, Tessier-Doyen N, Nait-ali B, Mati-Baouche N, Baynast HD, Smith DS (2013) A combined finite-discrete element method for calculating the effective thermal conductivity of bio-aggregates based materials. *Int J Heat Mass Transf* 60:274–283
14. Bourrier F, Kneib F, Chareyre B, Fourcaud T (2013) Discrete modeling of granular soils reinforcement by plant roots. *Ecol Eng* 61(C):646–657
15. Catalano E, Chareyre B, Barthélémy E (2014) Pore-scale modeling of fluid-particles interaction and emerging poromechanical effects. *Int J Numer Anal Methods Geomech* 38:51–71
16. Duriez J, Darve F, Donzé FV (2013) Incrementally non-linear plasticity applied to rock joint modelling. *Int J Numer Anal Methods Geomech* 37:453–477
17. Epifancev K, Nikulin A, Kovshov S, Mozer S, Brigadnov I (2013) Modeling of peat mass process formation based on 3D analysis of the screw machine by the code yade. *Am J Mech Eng* 1:73–75

18. Grujicic M, Snipes JS, Ramaswami S, Yavari R (2013) Discrete element modeling and analysis of structural collapse/survivability of a building subjected to improvised explosive device (IED) attack. *Adv Mater Sci Appl* 2:9–24
19. Hadda N, Nicot F, Bourrier F, Sibille L, Radjai F, Darve F (2013) Micromechanical analysis of second order work in granular media. *Granul Matter* 15:221–235
20. Hilton JE, Tordesillas A (2013) Drag force on a spherical intruder in a granular bed at low froude number. *Phys Rev E* 88:062203
21. Lominé F, Scholtès L, Sibille L, Poullain P (2013) Modelling of fluid-solid interaction in granular media with coupled LB/DE methods: application to piping erosion. *Int J Numer Anal Methods Geomech* 37:577–596
22. Nicot F, Hadda N, Darve F (2013) Second-order work analysis for granular materials using a multiscale approach. *Int J Numer Anal Methods Geomech* 37(17):2987–3007
23. Nicot F, Hadda N, Guessasma M, Fortin J, Millet O (2013) On the definition of the stress tensor in granular media. *Int J Solids Struct* 50(14–15):2508–2517
24. Scholtès L, Donzé FV (2013) A DEM model for soft and hard rocks: role of grain interlocking on strength. *J Mech Phys Solids* 61:352–369
25. Thoeni K, Lambert C, Giacomini A, Sloan SW (2013) Discrete modelling of hexagonal wire meshes with a stochastically distorted contact model. *Comput Geotech* 49:158–169
26. Tran VDH, Meguid MA, Chouinard LE (2013) A finite-discrete element framework for the 3D modeling of geogrid-soil interaction under pullout loading conditions. *Geotext Geomembr* 37:1–9
27. Mohammadi S (2003) *Discontinuum mechanics using finite and discrete elements*. WIT Press, Southampton
28. Smilauer V, Catalano E, Chareyre B, Dorofenko S, Duriez J, Gladky A, Kozicki J, Modenese C, Scholtès L, Sibille L, Stránský J, Thoeni K (2010) *Yade Reference Documentation*. Yade Documentation, 1st edn. The Yade Project
29. Smilauer V, Gladky A, Kozicki J, Modenese C, Stránský J (2010) *Yade using and programming*. Yade Documentation, 1st edn. The Yade Project
30. Van Liedekerke P, Tijskens E, Dintwa E, Anthonis J, Ramon H (2006) A discrete element model for simulation of a spinning disc fertilizer spreader I. Single particle simulations. *Powder Technol* 170(2):71–85. doi:10.1016/j.powtec.2006.07.024
31. Han K, Feng YT, Owen DRJ (2006) Polygon-based contact resolution for superquadrics. *Int J Numer Methods Eng* 66:485–501. doi:10.1002/nme.1569
32. Pourmin L (2005) On the behavior of spherical and non-spherical grain assemblies, its modeling and numerical simulation. Ph.D. thesis, École Polytechnique Fédérale de Lausanne, Lausanne
33. Cheng YF, Guo SJ, Lai HY (2000) Dynamic simulation of random packing of spherical particles. *Powder Technol* 107(1–2):123–130
34. Jia X, Williams RA (2001) A packing algorithm for particles of arbitrary shapes. *Powder Technol* 120(3):175–186
35. Han K, Feng YT, Owen DRJ (2005) Sphere packing with a geometric based compression algorithm. *Powder Technol* 155(1):33–41
36. Mueller GE (2005) Numerically packing spheres in cylinders. *Powder Technol* 159(2):105–110
37. Fraige FY, Langston PA, Chen GZ (2008) Distinct element modelling of cubic particle packing and flow. *Powder Technol* 186(3):224–240
38. Benabbou A, Borouchaki H, Laug P, Lu J (2010) Numerical modeling of nanostructured materials. *Finite Elem Anal Des* 46(1–2):165–180
39. Bagi K (2005) An algorithm to generate random dense arrangements for discrete element simulations of granular assemblies. *Granul Matter* 7:31–43
40. Feng YT, Han K, Owen DRJ (2002) An advancing front packing of polygons, ellipses and spheres. Paper presented at the discrete element methods, numerical modeling of discontinua, Santa Fe
41. Feng YT, Han K, Owen DRJ (2003) Filling domains with disks: an advancing front approach. *Int J Numer Methods Eng* 56:699–713. doi:10.1002/nme.583
42. Valera R, Morales I, Vanmaercke S, Morfa C, Cortés L, Casañas H-G (2015) Modified algorithm for generating high volume fraction sphere packings. *Comput Part Mech* 2(2):1–12. doi:10.1007/s40571-015-0045-8
43. Pérez Morales I, Roselló Valera R, Recarey Morfa C, Cerrolaza M (2011) Particle-packaging for computational modeling of bone. *CMES: Comput Model Eng Sci* 79(3):183–200
44. Pérez Morales I, Pérez Brito Y, Cerrolaza M, Recarey Morfa C (2010) Un Nuevo Método de Empaquetamiento de Partículas: Aplicaciones en el Modelado del Tejido óseo. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería* 26(4):305–315
45. Löhner R, Oñate E (2004) A general advancing front technique for filling space with arbitrary objects. *Int J Numer Methods Eng* 61:1977–1991. doi:10.1002/nme.1068
46. Weisstein EW (2015) Kepler Conjecture. *Wolfram MathWorld*. <http://mathworld.wolfram.com/KeplerConjecture.html>. Accessed Nov 2015
47. Weisstein EW (2015) Apollonius Circle. *Wolfram MathWorld*. <http://mathworld.wolfram.com/KeplerConjecture.html>. Accessed Nov 2015
48. Bagi K (2005) An algorithm to generate random dense arrangements for discrete element simulations of granular assemblies. *Granul Matter* 7:31–43. doi:10.1007/s10035-004-0187-5
49. Feng YT, Han K, Owen DRJ (2002) An advancing front packing of polygons, ellipses and spheres. In: Cook BK, Jensen RP (eds) *Discrete element methods. Numerical modeling of discontinua*, Santa Fe, Sept 23–25 2002. *Geotechnical Special Publication*. American Society of Civil Engineers, pp 93–98
50. Benabbou A, Borouchaki H, Laug P, Lu J (2009) Geometrical modeling of granular structures in two and three dimensions. Application to nanostructures. *Int J Numer Methods Eng* 80(4):425–454. doi:10.1002/nme.2644
51. Labra C, Oñate E (2008) High-density sphere packing for discrete element method simulations. *Communications in Numerical Methods in Engineering*. doi:10.1002/cnm.1193
52. Hernández Ortega JA (2003) Simulación numérica de procesos de llenado mediante elementos discretos. Polytechnic University of Catalonia
53. Pérez Morales I (2012) Método de Elementos Discretos: desarrollo de técnicas novedosas para la modelación con métodos de partículas. Universidad Central Marta Abreu de Las Villas, Santa Clara
54. Nelder JA, Mead R (1965) A simplex method for function minimization. *Comput J* 7:308–313
55. NLOpt Algorithms (2015). http://ab-initio.mit.edu/wiki/index.php/NLOpt_Algorithms#Nelder-Me...1. Accessed 21 July 2015
56. Hogue C (1998) Shape representation and contact detection for discrete element simulations of arbitrary geometries. *Eng Comput* 15(3):374–390
57. Sohn K-A, Juttler B, Kim M-S, Wang W (2002) Computing distances between surfaces using line geometry. Paper presented at the proceedings 10th Pacific conference on computer graphics and applications, 9–11 Oct 2002

58. Portal R, Sousa L, Dias J (2010) Contact detection between convex superquadric surfaces. *Arch Mech Eng* LVII 2:165–186. doi:[10.2478/v10180-010-0009-8](https://doi.org/10.2478/v10180-010-0009-8)
59. Eberly D (2015) Geometric Tools. <http://www.geometrictools.com/Documentation/Documentation.html>. Accessed Nov 2015
60. Price K, Storn R (1997) Differential evolution: a simple evolution strategy for fast optimization. *Dr Dobb's J* 264:18–24
61. Ingber L (1993) Simulated annealing: practice versus theory. *Math Comput Model* 18(11):29–57. doi:[10.1016/0895-7177\(93\)90204-C](https://doi.org/10.1016/0895-7177(93)90204-C)