Efficient linear programming algorithm to generate the densest lattice sphere packings

Étienne Marcotte and Salvatore Torquato

Phys. Rev. E **87**, 063303 — Published 7 June 2013

DOI: 10.1103/PhysRevE.87.063303

An Efficient Linear Programming Algorithm to Generate the Densest Lattice Sphere Packings

Étienne Marcotte[1] and Salvatore Torquato[1,2,3,4*]

[1] *Department of Physics,* [2] *Department of Chemistry,*

[3] *Program in Applied and Computational Mathematics,*

[4] *Princeton Institute of the Science and Technology of Materials,*

Princeton University, Princeton, New Jersey 08544, USA

Abstract

Finding the densest sphere packing in d-dimensional Euclidean space $\mathbb{R}^d$ is an outstanding fundamental problem with relevance in many fields, including the ground states of molecular systems, colloidal crystal structures, coding theory, discrete geometry, number theory, and biological systems. Numerically generating the densest sphere packings becomes very challenging in high dimensions due to an exponentially increasing number of possible sphere contacts and sphere configurations, even for the restricted problem of finding the densest lattice sphere packings. In this paper, we apply the Torquato-Jiao packing algorithm, which is a method based on solving a sequence of linear programs, to robustly reproduce the densest known lattice sphere packings for dimensions 2 through 19. We show that the TJ algorithm is appreciably more efficient at solving these problems than previously published methods. Indeed, in some dimensions, the former procedure can be as much as three orders of magnitude faster at finding the optimal solutions than earlier ones. We also study the suboptimal local density-maxima solutions (inherent structures or "extreme" lattices) to gain insight about the nature of the topography of the "density" landscape.

*Electronic address: torquato@princeton.edu

I. INTRODUCTION

There has been great interest in understanding the packings of hard (*i.e.* nonoverlapping) particles because they serve as useful models for a variety of many-particle systems arising in the physical and biological systems, such as liquids [1, 2], glasses [3–5], crystals [6–8], granular media [9–12], and living cells [13]. One outstanding problem is to find the densest packing of identical spheres in d-dimensional Euclidean space $\mathbb{R}^d$. This seemingly simple problem has proved to be a challenge for all but the most simple systems; it was not until 2005 that a proof was successfully presented to confirm the centuries-old Kepler conjecture [14], which states that the densest packing of spheres in three dimensions is the face-centered cubic lattice. For $d \geq 4$, there are no proofs for the densest sphere packings, although for $d = 8$ and $d = 24$ they are almost surely the E_8 and Leech lattices, respectively [15]. Interestingly, these two lattices have also been used to construct 10- and 26-dimensional string theories, respectively [16, 17].

In recent years, high-dimensional dense sphere packings have attracted the attention of physicists because of the insights they offer about condensed-phase systems in lower dimensions [5, 12, 18–20]. It is noteworthy that the general problem of finding the densest sphere packings in $\mathbb{R}^d$ (and other spaces) is directly relevant to making data transmission over communication channels resistant to noise [21, 22] and of intense interest in discrete geometry and number theory [22, 23]. The densest sphere packing problem is also deeply linked to the covering, quantizer, number variance, and kissing number problems, with which it shares the best known solutions in a variety of dimensions [22, 24, 25]. Clever analytical methods have been used to discover dense packings in high dimensions (*i.e.*, $d \geq 4$) but this approach becomes less efficient as d increases, especially because lessons learned in lower dimensions cannot be used to construct packings in higher dimensions [22, 26].

Numerical methods have only recently emerged to discover the densest packings in high-dimensional spaces. One such method devised by Kallus, Elser, and Gravel [19], is based on the "divide and concur" framework in which a dense arrangement of overlapping spheres is gradually relaxed until none of the spheres overlap. Another method formulated by Andreanov and Scardicchio [20] takes advantage of the fact that all densest lattice packings are also *perfect lattices* (defined precisely in Sec. IV), which are finite in number [22]. The densest lattice packings can therefore be obtained by randomly exploring the space of perfect

lattices. The efficiency of both algorithms plummets as d grows larger, preventing them from being effectively used in very high dimensions [27].

In the past twenty years, the Lubachevsky-Stillinger (LS) algorithm [28] has served as a standard for generating dense packing of various shaped hard particles in two and three dimensions [29–31]. However, since the LS algorithm is based on a particle-growth molecular dynamics simulation, it is extremely computationally costly to use it to generate jammed dense packings with high numerical accuracy, especially as d grows beyond three dimensions. A recent improvement on the LS algorithm is the Torquato-Jiao (TJ) algorithm [32], which replaces the molecular dynamics with an optimization problem that is solved using sequential linear programming. In particular, the density ϕ of a sphere packing (fraction of space covered by the spheres) within an adaptive fundamental cell subject to periodic boundary conditions is maximized. The design variables are the sphere positions (subject to nonoverlap), and the shape and size of the fundamental cell. The linear programming solution of this optimization problem becomes exact as the packing approaches the *jamming point* [12]. The TJ algorithm has been found to be a very powerful packing protocol to generate both maximally-dense packings (global maxima) and disordered jammed packings (local maxima) with a large number of identical spheres (per fundamental cell) across space dimensions [12] as well as maximally dense binary sphere packings [33, 34].

In this paper, we specialize the TJ algorithm to the restricted problem of finding the densest *lattice* sphere packings in high dimensions. In a lattice packing, there is only one sphere per fundamental cell [35]. Even this limited problem for $d \geq 4$ brings considerable challenges; its solution has been proven only for $d \leq 8$ [49] and $d = 24$ [15], and it is closely related to the shortest-vector problem, which is of NP-hard complexity [36]. Additionally, most of the densest known sphere packings for $d \leq 48$ are lattice packings [22, 26]. Tackling the lattice problem is thus a necessary first step prior to attempting to solve the much more complicated general problem of finding the densest periodic packings. A *periodic* packing of congruent particles is obtained by placing a fixed configuration of N particles where $N > 1$ with in one fundamental cell of a lattice, which is then periodically replicated without overlaps.

The outline of the rest of the paper is as follows: Sec. II describes the implementation of the TJ algorithm for the special case of lattice sphere packings. In Sec. III we motivate the choices that we make for the initial conditions and relevant parameters in order the various

problems across dimensions. In Sec. IV, we apply the TJ algorithm for $2 \leq d \leq 19$, and show that it is able to rapidly and reliably discover the densest known lattice packings without *a priori* knowledge of their existence. The TJ algorithm is found to be appreciably faster than previously published algorithms [19, 20]. We also demonstrate that the suboptimal-lattice solutions (i.e., the local maxima "inherent structures") are particularly interesting because they reveal features of the "density" landscape. In Sec. V, we close with some concluding remarks and a discussion about possible improvements and other applications of the TJ algorithm.

II. APPLICATION OF THE TJ ALGORITHM TO FINDING THE DENSEST LATTICE SPHERE PACKINGS

The basic principle behind the TJ algorithm [32] resides in the fact that finding the densest sphere packing can be posed as an optimization problem with a large number of nonlinear constraints (such as nonoverlap conditions between pairs of particles) which can be solved by solving a series of linear approximations of the original problem. Its solution eventually converges toward a local or global optimum. While global optimality cannot be guaranteed, it has been shown that the TJ algorithm frequently reaches the globally densest packings [32]. The TJ algorithm was formulated for the general problem of finding dense periodic sphere packings. Here we describe its implementation for the special case of determining the densest lattice sphere packings, which reduces the problem to optimizing the shape and size of the fundamental cell, since no sphere translations are involved. It is interesting to note that the TJ algorithm can be viewed as a hard-core analog of a gradient descent in the space of lattices for energy minimizations for systems of particles interacting with soft potentials as described by Cohn, Kumar, and Schürmann [38].

Before explaining the numerical details of the TJ algorithm, we need to define some mathematical quantities. A d-dimensional lattice Λ is composed of all vectors that are integer linear combinations of a set of d basis vectors $\mathbf{m}_1$, ..., $\mathbf{m}_d$,

$$\mathbf{P} = n_1\mathbf{m}_1 + n_2\mathbf{m}_2 + \cdots + n_d\mathbf{m}_d, \tag{1}$$

where n_j are the integers ($j = 1, 2, \ldots, d$) and we denote by $\mathbf{n}$ the corresponding column vector with such components. Using the generator matrix $\mathbf{M}_\Lambda$, whose columns are the basis

vectors, allows us to explicitly write the lattice set:

$$\Lambda = \left\{ \mathbf{M}_\Lambda \mathbf{n} : \mathbf{n} \in \mathbb{Z}^d \right\}. \tag{2}$$

One useful property of $\mathbf{M}_\Lambda$ is that its determinant is equal (up to a sign) to the volume of the lattice fundamental cell. We can then write the lattice packing density ϕ as the ratio of the volume occupied by spheres of diameter D to the volume of the fundamental cell:

$$\phi(\Lambda) = \frac{v(D/2)}{|\det \mathbf{M}_\Lambda|}, \tag{3}$$

where

$$v(R) = \frac{\pi^{d/2} R^d}{\Gamma(1 + d/2)} \tag{4}$$

is the d-dimensional volume of a sphere of radius R and $\Gamma(n)$ is the Euler gamma function.

The problem of finding the densest lattice packing of spheres in d dimensions can be expressed as: *Find the $d \times d$ generator matrix $\mathbf{M}_\Lambda$ with minimal determinant $|\det \mathbf{M}_\Lambda|$, under the constraint that all non-zero lattice vectors $\mathbf{M}_\Lambda \mathbf{n}$, $\mathbf{n} \in \mathbb{Z}^d \setminus \{\mathbf{0}\}$, are at least as long as D.*

For this problem, the Torquato-Jiao algorithm consists of the following four steps:

1. Randomly create a generator matrix $\mathbf{M}_\Lambda$ according to some stochastic process.
2. For a given *influence sphere radius* $R_I > D$, find all of the non-zero lattice vectors it contains, *i.e.*, compute $\left\{ \mathbf{v} = \mathbf{M}_\Lambda \mathbf{n} : \mathbf{n} \in \mathbb{Z}^d \setminus \{\mathbf{0}\} \wedge |\mathbf{v}| \leq R_I \right\}$.
3. Solve a linearized version of a problem, for which the objective is to maximize ϕ (equivalent to minimizing $|\det \mathbf{M}_\Lambda|$) and the constraints are that none of the vectors calculated in step 2 become shorter than D.
4. Consider whether the algorithm has converged to a lattice that is a stable maximum in ϕ (either the densest lattice packing or a local maximum *inherent structure* [37]). If it is the former, repeat the procedure starting from step 2. If it is the latter, the solution has converged to a local or global optimum and the procedure is terminated.

In what follows, we provide a more detailed explanation of these four steps.

A. Initialization

There are many possible methods to initialize the generator matrix $\mathbf{M}_\Lambda$. Any candidate procedure must both satisfy the minimal length constraint and adequately sample the space

of all lattices. The former is trivially satisfied by rescaling the matrix if the minimal length constraint is violated. In order to satisfy the latter condition, we mainly use Gaussian initial lattices, in which each coefficient of their generator matrix $\mathbf{M}_\Lambda$ is an independent normal variable $N(0, \sigma^2)$ with a variance σ^2. These matrices have the property that each of their lattice vectors (columns of $\mathbf{M}_\Lambda$) have independent orientations with no given preference for any particular direction. To compare this against a different initialization method, we also consider initial lattices for which $\mathbf{M}_\Lambda$ is the sum of the generator matrix of a specific lattice packing (such as the d-dimensional checkerboard lattice D_d or the hypercubic lattice Z_d, see Appendix A for the definitions of these lattices) and one of a Gaussian initial lattice.

B. Finding short vectors

Finding all of the vectors for an arbitrary lattice that are within a small given radius R_I from the origin is a complex problem in high dimensions. Indeed, the problem of finding the *shortest lattice vector* for a given lattice Λ grows superexponentially with d and is in the class of NP-hard (nondeterministic polynomial-time hard) problems [36]. One efficient method to solve this problem can be found in Ref. 39. The influence sphere radius R_I can be any value larger than the sphere diameter D, and may vary from one iteration to the next. It is found that the algorithm is largely insensitive to the value chosen for R_I, which is to be contrasted to the results for periodic packings, where larger R_I values favor the densest packings over inherent structures [32]. Since the computational cost of this and the following steps quickly increases with R_I, we opt to use the nearly minimal value $R_I = 1.1D$.

C. Solving the linearized problem

The only linearized problem variables in the case of the implementation of the TJ algorithm in the case of a lattice packing are the coefficients of the $d \times d$ symmetric strain tensor $\boldsymbol{\varepsilon}$ [40]. The modified generator matrix is then

$$\mathbf{M}_\Lambda \to \mathbf{M}_\Lambda + \boldsymbol{\varepsilon}\mathbf{M}_\Lambda. \tag{5}$$

The constraint that a vector originally at position $\mathbf{v} = \mathbf{M}_\Lambda \mathbf{n}$ remains at least as large as D can then be written as

$$\begin{aligned} \mathbf{n}^\top \mathbf{M}_\Lambda^\top \mathbf{M}_\Lambda \mathbf{n} + 2\mathbf{n}^\top \mathbf{M}_\Lambda^\top \boldsymbol{\varepsilon} \mathbf{M}_\Lambda \mathbf{n} + \mathbf{n}^\top \mathbf{M}_\Lambda^\top \boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon} \mathbf{M}_\Lambda \mathbf{n} &\geq D^2, \\ \mathbf{v}^\top \mathbf{v} + 2\mathbf{v}^\top \boldsymbol{\varepsilon} \mathbf{v} + \mathbf{v}^\top \boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon} \mathbf{v} &\geq D^2. \end{aligned} \tag{6}$$

This constraint is linearized by dropping the term that is quadratic in $\boldsymbol{\varepsilon}$:

$$2\mathbf{v}^\top \boldsymbol{\varepsilon} \mathbf{v} \geq D^2 - \mathbf{v}^\top \mathbf{v}. \tag{7}$$

It should be noted that the term $(\mathbf{v}^\top \boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon} \mathbf{v})$ that has been dropped is non-negative, which means that every set of variables that satisfies inequality (7) also satisfies inequality (6). This is different from the equivalent constraints for periodic packings, for which the quadratic term may be negative due to the interaction between the lattice deformation and the particle displacements. This avoids the necessity of either adding a constant term to the constraint or rescaling the system if spheres are found to overlap, which is the case for the general periodic packing problem [32].

Additionally, extra constraints must be added to prevent vectors that could be outside the influence sphere from becoming shorter than D:

$$\begin{aligned} 2\mathbf{v}^\top \boldsymbol{\varepsilon} \mathbf{v} &\geq D^2 - R_I^2 \\ \frac{\mathbf{v}^\top \boldsymbol{\varepsilon} \mathbf{v}}{\mathbf{v}^\top \mathbf{v}} &\geq \frac{D^2/R_I^2 - 1}{2} \equiv -\lambda, \end{aligned} \tag{8}$$

where the length of the vector has been chosen as its smallest possible value (R_I). A simple yet robust method to ensure that inequality (8) is satisfied for all vectors outside of the influence sphere is to bound the lowest eigenvalue of $\boldsymbol{\varepsilon}$ from below by $-\lambda$. There are multiple ways to write linear constraints on $\boldsymbol{\varepsilon}$ such that its eigenvalues are all larger than $-\lambda$. One such way is given by

$$-\frac{\lambda}{2} \leq \text{Diagonal element of } \boldsymbol{\varepsilon} < \infty, \tag{9}$$

$$-\frac{\lambda}{2(d-1)} \leq \text{Off-diagonal element of } \boldsymbol{\varepsilon} \leq \frac{\lambda}{2(d-1)}. \tag{10}$$

Finally, the determinant of the modified generator matrix (assuming that $\det \mathbf{M}_\Lambda > 0$) is

$$\det \mathbf{M}_\Lambda \det (\mathbf{I} + \boldsymbol{\varepsilon}) = \det \mathbf{M}_\Lambda \left(1 + \operatorname{tr} \boldsymbol{\varepsilon} + O(\boldsymbol{\varepsilon}^2)\right), \tag{11}$$

where $\mathbf{I}$ is the d-dimensional identity matrix. The linearized density ϕ is thus

$$\phi \simeq \phi_0 \left[1 - \operatorname{tr} \boldsymbol{\varepsilon}\right], \tag{12}$$

where ϕ_0 is the density for the initial generator matrix $\mathbf{M}_\Lambda$ and we used the fact that the density is inversely proportional to the fundamental cell volume. We can see from the above relation that maximizing the lattice density is equivalent to minimizing the trace of the strain tensor $\boldsymbol{\varepsilon}$. Unlike the linearized constraints (7), (9) and (10), which are conservative in that as long as they are satisfied the nonlinearized constraints will always be satisfied, the objective function (12) may have the wrong sign due to the nonlinear term having an unknown sign. In the situation where the updated lattice has a larger determinant than the original matrix, we halve $\boldsymbol{\varepsilon}$ (multiple times if necessary) to ensure a lower updated determinant. This prevents the algorithm from oscillating between multiple lattices and forces it to eventually converge.

D. Convergence criterion

The algorithm is considered to have converged if the sum of the squared coefficients of $\boldsymbol{\varepsilon}$ is below a small threshold value (10^{-12} for this paper). This is numerically equivalent to saying that all lattices in the neighborhood of the current lattice are less dense. This resulting lattice is therefore a local density maximum ("inherent structure" or "extreme" lattice, as elaborated in Sec. IV B). Such a lattice is also *strictly jammed*, since any possible deformation requires an increase in the volume of its fundamental cell [11, 41, 42].

III. STUDY OF PARAMETERS AND INITIAL CONDITIONS

The ability of TJ algorithm to discover the densest lattice packings can potentially be affected by the influence sphere radius R_I, the lowest eigenvalue of the strain matrix λ, and by the choice of the initial lattice. This section is dedicated to the study of their impact on the algorithm and to explain our choices for them in the following sections.

The TJ algorithm is deterministic [43], and therefore the initial lattice fully controls the resulting final lattice for given parameters R_I and λ. For example, employing initial lattices that are very close to the known densest lattice, not surprisingly, results in a very

high success rate in obtaining that lattice. On the flip side, it would almost certainly never be able to discover a hypothetical denser lattice. It would therefore be misguided to use configurations that are near the known densest lattice as the initial conditions. However, allowing initial lattices that are very bad packers could result in a low success rate or a large convergence time for success. Thus, good choices for initial lattices involve a delicate balance between their diversity and an ability to relax quickly to dense lattices.

TABLE I: Frequency at which the densest known lattice packing in 13 dimensions, the K_{13} lattice [22, 26], is obtained for various parameters using the TJ algorithm. For all sets of influence sphere radii and initial conditions, 10000 lattice packings have been generated, excepted for $R_I = 2.0$ where only 3000 packings were generated. The calculations were performed on a single thread on a 2.40 GHz processor using the Gurobi linear programming library [44]. Since the run time strongly depends on the computer running the program and how well the code is optimized, it should only be used as a rough indication of the program efficiency.

Sphere of influence radius	Initial conditions	Success rate (%)	Average time per trial (sec)
$R_I = 1.1D$	Gaussian	8.61	5.0
$R_I = 1.1D$	D_{13} + noise	8.21	5.5
$R_I = 1.1D$	$\mathbb{Z}^{13}$ + noise	8.58	5.2
$R_I = 1.1D$	Invariant distribution	8.08	29.2
$R_I = 1.02D$	Gaussian	8.53	12.0
$R_I = 1.5D$	Gaussian	7.61	69.9
$R_I = 2.0D$	Gaussian	6.87	1938.5
variable R_I, ~ 200 constraints	Gaussian	7.97	6.3
variable R_I, ~ 2000 constraints	Gaussian	7.95	17.6
variable R_I, ~ 2000 constraints, reduced λ	Gaussian	8.58	108.7

Table I shows numerical results in 13 dimensions. The initial lattices are taken from four different distributions, using six different influence sphere radii. The TJ algorithm typically succeeds at generating the densest known lattice packing with a high probability. However, it has a relatively lower success rates for the cases $d = 13$ and $d \geq 17$. We thus purposely choose the 13-dimensional case to probe the best choices for the initial conditions and algorithmic parameters because of its abnormally low success rate in comparison to cases

$d \leq 16$. Its low success rate results in better sensitivity to algorithm parameters compared with dimensions that have naturally higher success rates. Similar parameter dependence has been observed for other dimensions.

The *Gaussian* initial condition, as previously explained in Sec. II A, selects each coefficient of $\mathbf{M}_\Lambda$ from independent normal distributions with variances $\sigma^2 = D^2$. The initial conditions referred to as D_d + noise and $\mathbb{Z}^d$ + noise starts with the generator matrices for the checkerboard D_d and hypercubic $\mathbb{Z}^d$ lattices (these lattices are defined in Appendix A), respectively, with nearest-neighbor distance equal to D plus some noise. Specifically, we add normal noise to each coefficient of $\mathbf{M}_\Lambda$ with a variance $\sigma^2 = D^2/100$. The final initial condition type that we attempt to employ, which we call an *invariant distribution*, generates the lattice from an approximation of the invariant lattice distribution, using the algorithm described in Ref. 45 with $p = 10007$. For all of these initial conditions, the nearest neighbor distance is calculated and the lattice is rescaled to avoid any sphere overlap.

As can be seen in Table I, the different initial conditions that we have used result in similar success rates. We therefore use the Gaussian initial condition to generate the initial lattices for all subsequent calculations, since it lacks both the potential bias that the D_d + noise and $\mathbb{Z}^d$ + noise initial conditions share, and it does converge much faster than the invariant distribution.

The main parameter influencing the efficiency of the TJ algorithm is the influence sphere radius R_I, which can either be fixed or vary from one iteration to the next. A radius that is too large leads to a large number of extra constraints for the linear program, greatly increasing its complexity. By contrast, if R_I is too close to D, then the constraints on the shear matrix $\boldsymbol{\varepsilon}$ will be too restrictive [see Eqs. (8), (9) and (10)]. This, in turn, only allows the lattice to deform very slowly, thereby requiring many iterations before convergence. A compromise between both is to use a variable R_I, such that the number of vectors inside the sphere of influence stays relatively constant, thus initially allowing a fast convergence when ϕ is small, without needing numerous constraints when ϕ gets close to its maximum. We use the following rough approximation to select R_I:

$$\text{Number of constraints} \sim \frac{1}{2}\frac{v(R_I)}{|\det \mathbf{M}_\Lambda|}, \tag{13}$$

where the factor of one-half comes from the observation that for every vector $\mathbf{v}$ in a lattice, there is another one of identical length $-\mathbf{v}$ which does not need to be explicitly constrained.

A final parameter that can be modified is how much the lattice is allowed to deform at every iteration. As a test case, we divide the value of λ by 10 to check whether an increased value of R_I provides benefits other than allowing larger strain matrices.

From Table I, we can see that increasing R_I does not increase the success rate (it actually negatively affects it), while it significantly increases the run time. Therefore, the following calculations will be done using a small influence sphere radius of $R_I = 1.1D$. We attempted to adjust R_I as a function of dimension d to improve success rates for large d, but this proved to be fruitless. The radius R_I only weakly impacts the success rate, but its value has a dramatic influence on the time per trial, which gets multiplied by 400 when R_I is increased from $1.1D$ to $2.0D$. Therefore, one should decide on a choice of R_I so as to prioritize a faster execution speed over an increased probability of reaching the densest lattice packing.

IV. RESULTS

Here we describe the results we obtain by applying the TJ algorithm to find the densest lattice packings in dimensions 2 through 19. We compare our results with those obtained in previous investigations [19, 20]. We also provide the frequency of time that the TJ algorithm finds local versus the densest known global maxima.

A. Finding the densest lattice packings

We have applied the TJ algorithm for dimensions $d = 2$ through $d = 19$, and found the densest currently known lattice packing for each of them. The algorithm is robust in that it converges rapidly to the optimal solutions in most dimensions. Not surprisingly, except for the trivial $d = 2$ and $d = 3$ cases, it does not reach the optimal solution for all initial conditions. Therefore, even though the probabilities of finding the densest packing on the first attempt was high (greater than 19% for $d \leq 12$ and $14 \leq d \leq 16$), we typically needed multiple trials (i.e., different random initial conditions) to guarantee that the densest lattice packings were among these. Consequently, the quality of such a global optimization algorithm is preferably measured using the time required per successful trial instead of simply the time per trial or the success rate. Table II describes the rate at which the TJ algorithm produced the densest known lattice packings for dimensions $d = 2$ through $d = 19$ and the

TABLE II: Frequency at which the densest known lattice packing is obtained using the TJ algorithm for $d = 2$ through $d = 19$ together with the lattices packing fraction ϕ and kissing number Z. The number of lattice packings generated is 10000 for $d \leq 18$ and 100000 for $d = 19$. The influence sphere radius $R_I = 1.1D$ and the initial lattices are generated using the Gaussian initial condition. See Appendix A for the definitions of the various lattices. The comments in Table I concerning computational times also apply here.

d	Densest lattice packing	ϕ	Z	Success rate (%)	Time per trial (sec)	Time per successful trial (sec)
2	A_2	0.9069	6	100	1.7×10^{-5}	1.7×10^{-5}
3	D_3	0.7405	12	100	8.0×10^{-5}	8.0×10^{-5}
4	D_4	0.6169	24	74.31	5.6×10^{-4}	7.5×10^{-4}
5	D_5	0.4653	40	97.41	8.0×10^{-3}	8.2×10^{-3}
6	E_6	0.3729	72	89.72	0.019	0.022
7	E_7	0.2953	126	91.91	0.046	0.050
8	E_8	0.2537	240	84.16	0.33	0.40
9	Λ_9	0.1458	272	43.82	0.21	0.49
10	Λ_{10}	0.09202	336	22.74	0.49	2.1
11	K_{11}	0.06043	432	19.39	1.1	5.7
12	K_{12}	0.04945	756	33.30	2.7	8.2
13	K_{13}	0.02921	918	8.61	5.0	58
14	Λ_{14}	0.02162	1422	20.69	10	51
15	Λ_{15}	0.01686	2340	23.78	16	65
16	Λ_{16}	0.01471	4320	22.50	51	227
17	Λ_{17}	0.008811	5346	1.65	55	3.4×10^3
18	Λ_{18}	0.005928	7398	0.10	79	7.9×10^4
19	Λ_{19}	0.004121	10668	0.009	162	1.8×10^6

average time required per successful trial. We determine whether we achieved the densest known packings primarily by comparing the packing density ϕ and the kissing number Z (the number of spheres that are in contact with any given sphere) with published data [22, 26]. Additionally, we calculate theta series (the generating functions for the number of vectors

with specific lengths in the lattices [22]) up through the first few coordination shells.

The time required by the TJ algorithm to generate the densest known lattice packings is appreciably smaller than the times reported in Ref. 19: approximately 4000 and 25000 seconds per successful packing for $d = 13$ and $d = 14$, respectively. The times required by the TJ algorithm of 58 and 51 seconds are orders of magnitude lower, indicating a genuine algorithmic improvement that cannot be attributed to the type of computer employed nor to implementation details.

The authors in Ref. 20 do not state precise run times for all dimensions, but report that, after generating more than 10^5 lattices, their algorithm is unable to discover the densest known lattices for $d = 14$ through $d = 19$. Since generating 10^5 lattices using their algorithm takes at least several hours, the TJ algorithm's ability to successfully generate the densest lattice packings in minutes for $d \leq 16$ is a tremendous speed-up improvement. Using more computing power, the authors in Ref. 20 are able to reliably obtain the densest known lattice for $d \leq 17$ using their algorithm [46]. For example, their calculations took four days ($\sim 3 \times 10^5$ seconds) for $d = 14$, which is three to four orders of magnitude longer than our own calculations (see Table II).

The fact that the TJ algorithm was unable to find any denser lattice packings than the densest known lattice packings reinforces the evidence that these are indeed the densest lattice packings for $d = 2$ through $d = 19$. Although this evidence is not as strong for $d = 18$ and $d = 19$, due to the rare occurrences of the densest lattice packings, the evidence is quite strong for $d \leq 17$.

One particular aspect of the success rates shown in Table II is that they do not decrease monotonically with increasing dimension. Dimensions that are notably difficult are $d = 4$ and $d = 13$, and neither case can be explained by lattice packings with unusual properties, since $d = 5$ and $d = 12$, respectively, share similar packings, but not the relatively low success rates. We will attempt to explain this phenomenon, along with the sharp decrease in success rates at $d = 17$, in the following section.

B. Inherent structures

The TJ algorithm is intrinsically a local density maximization algorithm. As such, it can, and often does, converge locally to the densest lattice packing associated with a given initial

configuration, i.e., an *inherent structure* [32], that are not necessarily the global maxima. These local maxima are analogous to the inherent structures of a continuous potential. The study of these inherent structures are of fundamental interest in their own right because they offer insight about the nature of topography of the "density" landscape and understanding the frequency of their occurrence could potentially lead to improvements on the algorithm.

One interesting property of the density landscape associated with the lattice packing problem is that all of its inherent structures are *extreme* lattices, i.e., they are both *perfect* and *eutactic* [47]. Only a finite number of distinct extreme lattices exists for any dimension, which explains how the TJ algorithm is able to always reach the ground state for $d = 2$ and $d = 3$, for each of which only a single extreme lattice exists. However, as d increases, the number of extreme lattices grows quickly, possibly exponentially fast. It is thus remarkable that the TJ algorithm can reliably yield the densest lattice packing from the large set of possible end states. This indicates that the "basin of attraction" of the ground state is much larger than the basins of attraction of the local-maxima inherent structures. The relatively lower success rates for some dimensions ($d = 4$, $d = 11$, $d = 13$, and $d \geq 17$) can then be understood as being due to smaller than usual basins for the corresponding ground states. The cause of this reduction and whether the symmetry of the inherent structure is lower than that of the ground state or some other effect is still unknown and warrants further investigation.

As seen in Table III, some inherent structures are degenerate in the sense that multiple lattices share the same packing density. A peculiar property that these degeneracies share is that their appearance rate is far from constant. For example, it goes from 9.24% for the $\Lambda_{12}^{\min}$ to a mere 0.03% for the $\Lambda_{12}^{\max}$. Since both of these are laminated lattices, why does one occurs more frequently than the other? One possible reason is that for all these degeneracies but one, the lattices with smaller kissing number are more likely to be generated. In the case of $\Lambda_{12}^{\min}$ and $\Lambda_{12}^{\max}$, their kissing numbers are respectively 624 and 648. This is consistent with previous work which has shown that for packings with many particles per fundamental cell, the TJ algorithm has a propensity to generate isostatic packings from random initial conditions, where the number of interparticle contacts is equal to the number of degrees of freedom of the problem [32].

Figure 1 shows that as the dimensionality increases, the inherent-structure densities tend to become concentrated around a specific value instead of being spread over a range of pos-

TABLE III: Second and third highest-density inherent structures (locally densest lattice packings), including their packing density ϕ, kissing number Z, and success rate from the TJ algorithm. See Table II to compare to the densest lattice packings. The number of lattice packings generated for each dimension is 10000 for $d \leq 18$ and 100000 for $d = 19$. Multiple lattices with equal density are grouped together and written in ascending kissing number order. See Ref. 26 for the definitions of the following lattices: A_5^{+3}, E_6^*, $P7.3$, $P7.5$, K_9^2, Dim11 (named dim11kis422 in the reference), K_{14}^1, K_{14}^2, Λ_{15}^2, K_{15}^1, Λ_{16}^2, and K_{16}^1. Lattices that were not identified in Ref. 26 and found here are denoted as U_d^n, where n is used to distinguish different lattices at some fixed dimension d.

	Second densest				Third densest			
d	Lattice	ϕ	Z	Rate (%)	Lattice	ϕ	Z	Rate (%)
2	—	—	—	—	—	—	—	—
3	—	—	—	—	—	—	—	—
4	A_4	0.5517	20	25.69	—	—	—	—
5	A_5^{+3}	0.4136	30	1.51	A_5	0.3799	30	1.08
6	E_6^*	0.3315	54	1.53	D_6	0.3230	60	7.70
7	$P7.3$	0.2143	72	0.88	$P7.5/D_7$	0.2088	72/84	1.92/0.11
8	U_8^1	0.1691	142	0.41	U_8^2	0.1530	116	3.75
9	U_9^1	0.1383	258	2.60	K_9^2	0.1190	198	14.09
10	U_{10}^1	0.08282	294	0.42	U_{10}^2	0.08231	308	0.05
11	$\text{Dim11}/\Lambda_{11}^{\text{min}}/\Lambda_{11}^{\text{max}}$	0.05888	422/432/438	5.32/7.80/0.30	U_{11}^1	0.05551	408	0.81
12	$\Lambda_{12}^{\text{min}}/\Lambda_{12}^{\text{mid}}/\Lambda_{12}^{\text{max}}$	0.04173	624/632/648	9.24/2.96/0.03	$U_{12}^1/U_{12}^2/U_{12}^3$	0.03732	550/560/566	1.38/0.18/0.05
13	$\Lambda_{13}^{\text{min}}/\Lambda_{13}^{\text{mid}}/\Lambda_{13}^{\text{max}}$	0.02846	888/890/906	12.17/1.50/0.29	U_{13}^1	0.02683	828	2.97
14	U_{14}^1	0.01934	1260	0.69	K_{14}^2/K_{14}^1	0.01922	1242/1248	2.26/0.38
15	Λ_{15}^2/U_{15}^1	0.01376	1872/1890	1.57/0.02	K_{15}^1	0.01298	1746	0.92
16	Λ_{16}^2	0.01040	2982	0.69	K_{16}^1/U_{16}^1	0.009805	2772/2820	0.67/0.03
17	U_{17}^1	0.007194	4266	0.63	U_{17}^2	0.006661	3942	0.09
18	U_{18}^1	0.005134	6336	0.03	U_{18}^2	0.004743	5820	0.02
19	U_{19}^1	0.003686	9480	0.012	U_{19}^2	0.003475	8910	0.002

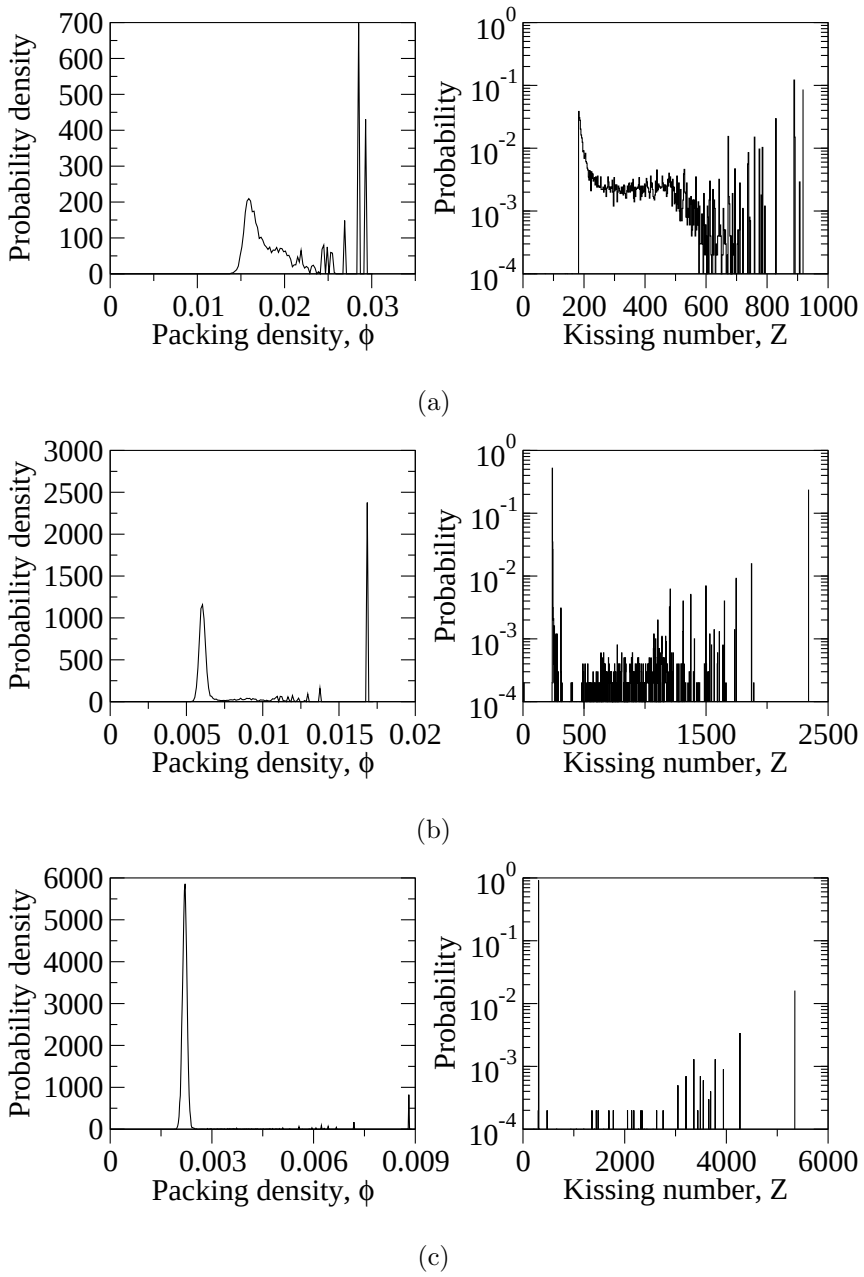

FIG. 1: Probability density functions for the packing density ϕ (left) and probabilities for the kissing number z (right) of the lattice resulting from the TJ algorithm for (a) $d = 13$, (b) $d = 15$, and (c) $d = 17$. The minimal value of the kissing number $Z_{min} = d(d + 1)$ is 182 for $d = 13$, 240 for $d = 15$, and 106 for $d = 17$.

sible densities. This concentration tendency is caused by the rapid increase in the number of such low-density inherent structures for large d, which eventually overwhelms the algorithmic bias toward high-density lattices. This explains the dramatic reduction in success rates in Table II for $d \geq 17$. The kissing number has a similar behavior to the packing density,

resulting in the fact that most of the generated lattices for $d \geq 17$ have an identical low kissing number. Since these are locally-optimal solutions, a local deformation of the lattice would either decrease its packing fraction or makes the central sphere and its neighboring spheres overlap. Therefore, we can define a lower bound on the kissing number by exploiting the fact that, for a linear program to have a unique feasible solution, it requires at least one more active inequality constraint than the number of degrees of freedom. Since the problem possesses $d(d+1)/2$ degrees of freedom (the number of independent components of $\boldsymbol{\varepsilon}$), $1 + d(d+1)/2$ active inequality constraints are required for the problem to be fully constrained. One of these constraints comes from the density being at a local maximum, while each pair of kissing spheres adds a single constraint. Consequently, the minimum kissing number of a lattice inherent-structure in d dimensions is $Z_{min} = d(d+1)$. Referring to Fig. 1, we observe that as d increases, the proportion of generated configurations with a kissing number equal to Z_{min} increases rapidly relative to all other kissing numbers. Since the best known lattice packings have high kissing numbers (nearly the same or equal to highest known kissing numbers), the tendency of the TJ algorithm tendency to favor lattices with minimal kissing numbers further explains its low success rates for $d \geq 17$.

V. CONCLUSIONS AND DISCUSSION

In this paper, we have shown that the Torquato-Jiao algorithm is able to quickly find the densest known lattice packings for $d \leq 19$. The TJ algorithm is found to be orders of magnitude faster than the previous state-of-the-art lattice packing methods [19, 20]. This makes the TJ algorithm the fastest current numerical method to generate the densest lattice packings in high dimensions.

While we limited our present study to $d \leq 19$, the TJ algorithm can be employed to generate dense lattice packings in higher dimensions at greater computational cost. We expect that dimensions $d = 20$ and $d = 21$ would be manageable with more computing resources, but improvements to the algorithm would be required to study $d \geq 22$. One possible approach to increase the likelihood of generating a dense lattice packing for $d \geq 22$ would be to include *ad hoc* methods in between the TJ-algorithm steps that favor denser packings, such as thermal equilibration of the system (e.g., via Monte Carlo methods to solve the "adaptive shrinking cell" optimization problem [51, 52]) or relaxation under pair potentials

known to favor high-density configurations. Another possibility would be to combine the strengths of the TJ algorithm with those of other lattice packing methods. The ability of the TJ algorithm to quickly generate extreme lattices (the inherent structures) could be used as a starting point for an algorithm that performs an exhaustive search in the space of perfect lattices [20]. Moreover, its efficiency in finding locally-densest lattice packings from arbitrary initial conditions could be used to rapidly obtain such packings starting from intermediate-density packings generated using other methods [19]. As d increases from one, the first dimension in which the densest known packing that is not a Bravais lattice (periodic packing with a multiple-particle basis) is $d = 10$, which has a basis of 40. Since the TJ algorithm was successfully used to obtain the densest known packings for $d \leq 6$ with a large multiple-particle basis (up to a basis of 729 for $d = 6$) [32], it would be interesting to explore whether the TJ algorithm could be used to discover currently unknown denser non-lattice packings in 10 dimensions or higher.

For $d \geq 17$ dimensions, the TJ algorithm mainly produces lattices that have both a low packing density and a minimal kissing number, while still being locally densest, revealing a richer and more complex density landscape than in most dimensions less than 17. This phenomenon could possibly be exploited to quickly generate low-density extreme lattices in very high dimensions. Since these lattices are strictly jammed and have the minimal kissing number to ensure mechanical stability, they can be considered to be the lattice analogs to the maximally random jammed packings (disordered local-maxima inherent structures) that have been generated using the TJ algorithm with many particles per fundamental cell [32]. Such configurations could be generated in much higher dimensions than those considered in this paper, since the requirement of reaching the ground state would be removed, and the TJ algorithm is less resource-intensive when generating suboptimal kissing configurations (through the reduced number of constraints).

Acknowledgments

We thanks Alexei Andreanov, Henry Cohn, Veit Elser, Yoav Kallus, and Antonello Scardicchio for very helpful discussions. This work was partially supported by the Materials Research Science and Engineering Center Program of the National Science Foundation under Grant No. DMR- 0820341 and by the Division of Mathematical Sciences at the National

Science Foundation under Award No. DMS-1211087. This work was partially supported by a grant from the Simons Foundation (Grant No. 231015 to Salvatore Torquato). S.T. also thanks the Department of Physics and Astronomy at the University of Pennsylvania for their hospitality during his stay there.

Appendix A: Lattice definitions

In this appendix, we define some common lattices, following the notation and nomenclature used in Refs. 22 and 26.

The *hypercubic* $\mathbb{Z}^d$ lattice is defined by

$$\mathbb{Z}^d = \{(x_1, \ldots, x_d) : x_i \in \mathbb{Z}\} \quad \text{for } d \geq 1 \tag{A1}$$

where $\mathbb{Z}$ is the set of integers $(\ldots -3, -2, -1, 0, 1, 2, 3 \ldots)$ and $x_1, \ldots, x_d$ denote the components of a lattice vector. The kissing number of $\mathbb{Z}^d$ is $2d$. A d-dimensional generalization of the face-centered-cubic lattice is the *checkerboard* D_d lattice defined by

$$D_d = \{(x_1, \ldots, x_d) \in \mathbb{Z}^d : x_1 + \cdots + x_d \ \text{ even}\} \quad \text{for } d \geq 2. \tag{A2}$$

Its kissing number is $2d(d-1)$. Note that D_2 is simply the square lattice $\mathbb{Z}^2$. Another generalization of the face-centered-cubic lattice is the *root* lattice A_d, which is a subset of points in $\mathbb{Z}^{d+1}$, i.e.,

$$A_d = \{(x_0, x_1, \ldots, x_d) \in \mathbb{Z}^{d+1} : x_0 + x_1 + \cdots + x_d = 0\} \quad \text{for } d \geq 1. \tag{A3}$$

The kissing number of A_d is $d(d+1)$. In three dimensions, D_3 and A_3 are identical, but D_d and A_d are inequivalent for $d \geq 4$. Another set of root lattices is denoted E_d, for $d = 6$, $d = 7$, and $d = 8$. The root lattice E_8 is equal to the union of D_8 and the translation of D_8 by $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. The root lattice E_7 is the section of E_8 where the sum of the lattice coefficients is set equal to zero, and the root lattice E_6 is the section of E_7 where the sum of the first and eight coefficients is also set equal to zero. Alternatively, vectors in E_8 perpendicular to any A_2-sublattice in E_8 also form E_6.

The *laminated* lattice Λ_d is constructed by stacking layers of a $(d-1)$-dimensional laminated lattice Λ_{d-1} as densely as possible such that the shortest vector in Λ_d is of equal or longer length than the shortest vector in Λ_{d-1}. This definition does not uniquely define Λ_d

for all dimensions. For $d = 11$, $d = 12$, $d = 13$, and $d \geq 25$, there exist multiple laminated lattices of equal densities, which we distinguish using superscripts. Many of the previously defined lattices are also laminated lattices. For example, $\Lambda_1 = \mathbb{Z}^1$, $\Lambda_2 = A_2$, $\Lambda_3 = D_3$, $\Lambda_4 = D_4$, $\Lambda_5 = D_5$, $\Lambda_6 = E_6$, $\Lambda_7 = E_7$, and $\Lambda_8 = E_8$. A particularly interesting laminated lattice is the 24-dimensional Leech lattice Λ_{24}. Finally, the *Coxeter-Todd* lattice K_{12} can be defined for 18 dimensions:

$$K_{12} = \{(x_{11}, \cdots, x_{16}, x_{21}, \cdots, x_{26}, x_{31}, \cdots, x_{36}) : x_{ij} \in \mathbb{Z}\}, \tag{A4}$$

where x_{ik} denotes the components of a lattice vector, subject to the following conditions

$$x_{i1} + x_{i2} + x_{i3} = 0 \quad i \in \{1, \cdots, 6\}, \tag{A5}$$

$$x_{i1} - x_{j1} \equiv x_{i2} - x_{j2} \equiv x_{i3} - x_{j3} \mod 3 \quad i, j \in \{1, \cdots, 6\}, \text{and} \tag{A6}$$

$$x_{1k} + x_{2k} + x_{3k} + x_{4k} + x_{5k} + x_{6k} \equiv 0 \mod 3 \quad k \in \{1, 2, 3\}. \tag{A7}$$

This lattice can be generalized to other dimensions in the range $6 \leq d \leq 18$ by requiring that K_d is the densest section of K_{d+1} which either contains or is contained in K_{12} and taking $K_{18} = \Lambda_{18}$.

---

[1] J. P. Hansen and I. R. McDonald, *Theory of Simple Liquids* (Academic, New York, 1986).

[2] P. M. Chaikin and T. C. Lubensky, *Principles of Condensed Matter Physics* (Cambridge University Press, New York, 2000).

[3] R. Zallen, *The Physics of Amorphous Solids* (Wiley, New York, 1983).

[4] S. Torquato, T. M. Truskett, and P. G. Debenedetti, Phys. Rev. Lett. 84, 2064 (2000).

[5] G. Parisi and F. Zamponi, Rev. Mod. Phys. 82, 789 (2010).

[6] Z. W. Salsburg and W. W. Wood, J. Chem. Phys. 37, 798 (1962).

[7] S. Torquato and F. H. Stillinger, J. Appl. Phys. 102, 093511 (2007); 103, 129902 (2008).

[8] T. Aste and D. Weaire, *The Pursuit of Perfect Packing* (Taylor & Francis, New York, 2008).

[9] A. Mehta, *Granular Matter: An Interdisciplinary Approach* (Springer-Verlag, New York, 1994).

[10] S. F. Edwards and D. V. Grinev, Phys. Rev. Lett. 82, 5397 (1999); Chem. Eng. Sci. 56, 5451 (2001).

[11] S. Torquato and F. H. Stillinger, J. Phys. Chem. B 105, 11849 (2001).

[12] S. Torquato and F. H. Stillinger, Rev. Mod. Phys. 82, 2633 (2010).

[13] S. Torquato, *Random Heterogeneous Materials: Microstructure and Macroscopic Properties* (Springer, New York, 2002).

[14] T. C. Hales, Ann. Math., **162**, 1065 (2005).

[15] H. Cohn and A. Kumar, Ann. Math., **170**, 1003 (2009).

[16] D. J. Gross, J. A. Harvey, E. Martinec, and R. Rohm, Phys. Rev. Lett. **54**, 502 (1985).

[17] G. Chapline, Phys. Lett. B **158**, 393 (1985).

[18] S. Torquato and F. Stillinger, Exp Math. **15**, 307 (2006); A. Scardicchio, F. H. Stillinger, and S. Torquato, J. Math. Phys. **49**, 043301 (2008).

[19] Y. Kallus, V. Elser, and S. Gravel, Phys. Rev. E **82**, 056707 (2010).

[20] A. Andreanov and A. Scardicchio, Phys. Rev. E, **86**, 041117 (2012).

[21] C. E. Shannon, Bell Syst. Tech. J. **27**, 379 (1948); **27**, 623 (1948).

[22] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices, and Groups* (Springer, New York, 1998).

[23] P. Sarnak and A. Strömbergsson, Invent. Math. **165**, 115 (2006).

[24] S. Torquato, Phys. Rev. E **82**, 056109 (2010).

[25] H. Cohn, Y. Jiao, A. Kumar, and S. Torquato, Geom. Topology **15**, 2235 (2011).

[26] G. Nebe and N. J. A. Sloane, *Catalogue of Lattices*, `http://www.math.rwth-aachen.de/~Gabriele.Nebe/LATTICES/`.

[27] It should be noted that the algorithm presented in Ref. 20 was designed to explore general suboptimal perfect lattice packings (not necessarily the densest lattice packings) and whether their statistics have implications for bounds on the maximal density.

[28] B. D. Lubachevsky and F. H. Stillinger, J. Stat. Phys. **60**, 561 (1990).

[29] A. Donev, S. Torquato, and F. H. Stillinger, J. Comput. Phys. **202**, 737 (2005); **202**, 765 (2005).

[30] Y. Jiao, F. H. Stillinger, and S. Torquato, Phys. Rev. Lett. **100**, 245504 (2008).

[31] Y. Jiao, F. H. Stillinger, and S. Torquato, Phys. Rev. E **79**, 041309 (2009).

[32] S. Torquato and Y. Jiao, Phys. Rev. E, **82**, 061302 (2010).

[33] A. B. Hopkins, Y. Jiao, F. H. Stillinger, and S. Torquato, Phys. Rev. Lett. **107**, 125501 (2011).

[34] A. B. Hopkins, F. H. Stillinger, and S. Torquato, Phys. Rev. E **85**, 021130 (2012).

[35] In the physical sciences and engineering, a lattice is usually referred as a Bravais lattice.

[36] M. Ajtai, *Generating Hard Instances of Lattice Problems*, Proc. 28th Annual ACM Symp. Theory of Computing, (1996).

[37] Finding the lattice which maximizes ϕ is equivalent to determining the ground states (global minima) in the "energy landscape" in which the "energy" is $-\phi$, where the degrees of freedom are the components of $\mathbf{M}_\Lambda$. Following Torquato and Jiao [32], we call the stable local/global density maxima (or energy minima) *inherent structures*.

[38] H. Cohn, A. Kumar, and A. Schürmann, Physical Review E **80**, 061116 (2009).

[39] U. Fincke and M. Pohst, Math. Comp., **44**, 463 (1985).

[40] An asymmetric shear tensor $\boldsymbol{\varepsilon}$ could have been used, but the set of inherent-structure solutions would have been unchanged. Under this more general asymmetric form, in all of the linearized Eqs. (7), (9), (10), and (12), $\boldsymbol{\varepsilon}$ would be replaced by the symmetrized tensor $(\boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}^\top)/2$. The only impact from the antisymmetric portion of such an $\boldsymbol{\varepsilon}$ would be to add trivial $d(d-1)/2$ rotational degrees of freedom to Λ, which are irrelevant as far as the packing density is concerned.

[41] S. Torquato, A. Donev, and F. H. Stillinger, Int. J. Solids Struct. **40**, 7143 (2003).

[42] The strict jamming of the resulting configuration can only be guaranteed as long as the packing is restricted to a lattice, i.e., one sphere per fundamental cell. For any such strictly jammed lattice sphere packing, it is possible that density-preserving or density-increasing deformations exist on a larger torus involving this structure that would lead to unjamming motions. This possibility increases with increasing dimension, since all lattice packings almost surely become "unsaturated" (holes exist that can accommodate extra spheres) in sufficiently high dimensions [22].

[43] It is important to realize that the linear programs that the TJ algorithm solves often do not have unique solutions. Therefore, the TJ algorithm is guaranteed to be deterministic only if the selection of an optimal solution is also guaranteed to be deterministic. An example of multiple equivalent solutions occurs if all of the lattice non-zero vector lengths are larger than the sphere of influence radius R_I, in which case the only linearized problem constraints will be those on the shear matrix $\boldsymbol{\varepsilon}$. Then, as long as they satisfy inequality (10), the non-diagonal elements of $\boldsymbol{\varepsilon}$ can take any value, without any impact on the objective function.

[44] *Gurobi Optimizer*, version 5.0.2, Gurobi Optimization, `www.gurobi.com`.

[45] D. Goldstein and A. Mayor, Forum Math. **15**, 165 (2003); H. Cohn and G. Minton, private communication, 2013.

[46] A. Andreanov and A. Scardicchio, private communication, 2013.

[47] *Perfect* lattices in d dimensions have the property that any $d \times d$ symmetric matrix S can be written as a linear combination of the lattice shortest vector projectors $\mathbf{v}_i^{min}$, *i.e.*

$$S = \sum_{i=1}^{Z} \alpha_i \mathbf{v}_i^{min} \mathbf{v}_i^{min\top},$$

where Z is the lattice kissing number and α_i are linear combination coefficients. All perfect lattices for $1 \leq d \leq 8$ have been identified. There are 1, 1, 1, 2, 3, 7, 33, and 10916 perfect lattices for $d = 1$ through $d = 8$, respectively [26]. *Eutactic* lattices in d dimensions have the property than each of their shortest vector $\mathbf{v}_i^{min}$ is associated with a positive *eutactic coefficient* $\beta_i > 0$ such that the norm of any vector $\mathbf{x}$ can be written as:

$$|\mathbf{x}|^2 = \sum_{i=1}^{Z} \beta_i \left(\mathbf{v}_i^{min\top} \mathbf{x} \right)^2.$$

A lattice is a local maximum in density (i.e., an inherent structure) if and only if it is an *extreme* lattice, which is both perfect and eutactic [48]. There are 1, 1, 1, 2, 3, 6, 30, and 2408 extreme lattices for $d = 1$ through $d = 8$, respectively [49, 50].

[48] G. Voronoi, J. reine angew. Math. **133**, 97 (1908).

[49] J. Martinet, *Perfect lattices in Euclidean Spaces* (Springer, New York, 2003).

[50] C. Riener, J. Th. Nombres Bordeaux **18**, 677 (2006).

[51] S. Torquato and Y. Jiao, Phys. Rev. E **80**, 041104 (2009).

[52] S. Atkinson, Y. Jiao, and S. Torquato, Phys. Rev. E **86**, 031302 (2012).