The unassigned distance geometry problem

P.M. Duxbury a,∗, L. Granlund a, S.R. Gujarathi a, P. Juhas b, S.J.L. Billinge c

a Department of Physics and Astronomy, Michigan State University, East Lansing, MI, United States
b Condensed Matter Physics and Materials Science Department, Brookhaven National Laboratory, NY, United States
c Department of Applied Physics and Applied Mathematics, Columbia University, NY, United States

A R T I C L E   I N F O

Article history:
Received 22 March 2014
Received in revised form 1 September 2015
Accepted 17 October 2015
Available online 19 November 2015

Keywords:
Nanostructure
Rigid cluster
Percolation
Unassigned distances
LIGA
TRIBOND

A B S T R A C T

Studies of distance geometry problems (DGP) have focused on cases where the vertices at the ends of all or most of the given distances are known or assigned, which we call assigned distance geometry problems (aDGPs). In this contribution we consider the unassigned distance geometry problem (uDGP) where the vertices associated with a given distance are unknown, so the graph structure has to be discovered. uDGPs arise when attempting to find the atomic structure of molecules and nanoparticles using X-ray or neutron diffraction data from non-crystalline materials. Rigidity theory provides a useful foundation for both aDGPs and uDGPs, though it is restricted to generic realizations of graphs, and key results are summarized. Conditions for unique realization are discussed for aDGP and uDGP cases, build-up algorithms for both cases are described and experimental results for uDGP are presented.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

The foundational distance geometry problem (DGP) is an inverse problem where a list of pair distances associated with edges in a graph is inverted to find a structure or embedding in a Euclidean space ℜK, where K is the embedding dimension. This problem has a long and venerable history that has been recently reviewed [31]. That survey should be consulted for a broad background to the topic of this paper, and here we follow the notation used there, where applicable.

We make a strong distinction between two subproblems of DGPs. In the assigned distance geometry problem (aDGP) the vertices at the end of each distance or edge are known or assigned, while in the unassigned distance geometry problem (uDGP) no information about the assignment of vertices to distances is provided so the graph structure also has to be determined. Here we focus on uDGP problems that have precise distance lists, though some discussion and computational results for the imprecise case are also presented.

The fundamental distance geometry problem defined by Liberti et al. [31] is the assigned distance geometry problem:

Definition 1 (Assigned Distance Geometry Problem (aDGP)). Given an integer K > 0 and a simple undirected graph G = (V, E) whose edges are weighted by a non-negative function d : E → ℜ+, determine whether there is a function x : V → ℜK such that

∀{u, v} ∈ E, ∥x(u) − x(v)∥ = d({u, v}) = d uv.

(1)

∗ Corresponding author.
E-mail address: duxbury@pa.msu.edu (P.M. Duxbury).
1 Since 1855.
2 Since 1754.
In this paper the norm $\| \cdot \|$ is taken to be the Euclidean norm as is relevant to many applications, and in particular to the problem of nanostructure determination emphasized here. An optimization formulation of aDGP is defined as follows: Given a convex penalty function $f(y)$ with minimum at the origin and a set of vertex positions $X$, where $X$ is a $N \times K$ matrix where each row, $x$, represents a vertex position in dimension $K$, solve the minimization problem,

$$\min_x \sum_{[u,v] \in E} f(\|x(u) - x(v)\| - d_{uv}).$$

(2)

A mean square deviation is often used in applications to atomic structure determination in materials physics [13,1], so that $f(y) = ay^2$ where $a$ is a positive constant, while in applications to the determination of protein structure in solution [10,48,18,31], a distance interval is most often utilized [6,38,4]. The distance between protons in proteins is often extracted from Nuclear Magnetic Resonance (NMR) Nuclear Overhauser Effect Spectroscopy (NOESY) data, though these distances have significant errors so they are treated as restraints rather than constraints [10,48,18,31]. Moreover, considerable experimental work is carried out to determine the pair of protons assigned to each distance extracted from the data, which is essential to the aDGP approach.

The unassigned distance geometry problem (uDGP) [16,26–28] considers the additional complications arising when the vertices at the ends of the distances are not known, so the underlying graph structure is unknown and must be found along with the embedding. This problem arises in the determination of the atomic structure of molecules, nanoparticles and nanostructured bulk materials [26,2]. Suppose that the distance list $D$ contains $m$ distances. Let $L$ be the set of indices $\{1, 2, 3 \ldots m\} \subset \mathbb{N}$, where $\mathbb{N}$ is the set of natural numbers. Then $D = \{d_i \in \mathbb{N}_+ : i \in L\}$. The definition of the uDGP is then:

**Definition 2** (Unassigned Distance Geometry Problem (uDGP)). Given an integer $K > 0$ and a set $D$ of strictly positive distances, determine whether there is an assignment $\alpha : L \rightarrow E$, and an embedding $x : V \rightarrow \mathbb{R}^K$ such that,

$$\forall l \in L, u, v \in V : (u, v) = \alpha(l) \text{ and } \|x_u - x_v\| = d_l.$$  

(3)

This problem may be stated as an optimization problem as follows: Given a convex penalty function $f(y)$ with minimum at the origin and a set of vertex positions $X$, where $X$ is a $N \times K$ matrix whose rows represent the $N$ vertex positions in dimension $K$, solve the minimization problem,

$$\min_{\alpha : L \rightarrow E} \left[ \min_x \sum_{[u,v] \in E} f(\|x(u) - x(v)\| - d_{uv}) \right].$$

(4)

where the minimization is over all possible assignments of $d_i$ to $d_{uv}$ and over all placements of the positions of the vertices, $x(u)$.

The mapping $\alpha : L \rightarrow E$ is bijective when the cardinality of the index list $L$ is equal to the number of edges of an $N$-clique, that is $N(N - 1)/2$, so that for every distance in $D$ there is one and only one edge. However in many experiments the distance list is incomplete and in that case the mapping $\alpha$ is injective and non-surjective, as occurs for the fullerene case discussed in Section 6.

Examples of uDGP are schematically illustrated in Fig. 1 for two limiting cases. The top example is for a highly symmetric structure where the input distance list has high degeneracy, while the bottom example is for a case with low symmetry and in this example all of the distances are unique. Study of the uDGP is motivated by the fact that the functional properties of nanostructured materials and complex molecules are controlled by their atomic structure, making it essential to determine atomic structure to high precision to enable understanding and design of materials for most applications. Crystallography is the gold standard for determination of atomic structure at high precision, however many emerging complex materials have no periodic long-range order so their structures cannot be solved using crystallographic methods [2].

An alternative approach is to experimentally determine the distances between atoms in a material for example using pair distribution function (PDF) data, and to utilize uDGP algorithms to find the atomic structure as has been recently demonstrated [26,28]. In that work a stochastic heuristic algorithm called the “LIGA” algorithm was developed. This uDGP algorithm utilizes a combination of concepts from random sampling, dynamic programming, and tournaments. The LIGA algorithm works well for systems where the number of different distances in a nanostructure is relatively small. More recently a different approach, the “TRIBOND” algorithm, was developed for problems in two dimensions with a large number of unique distances [16].

In this paper a precise mathematical formulation of uDGP is presented (above), its relevance to the problem of nanostructure determination is elucidated (Section 2), and we provide a more rigorous basis for buildup algorithms such as TRIBOND (Section 3). The relevance of rigidity theory to aDGP and uDGP graph realization problems is surveyed in Sections 3 and 4, with Section 4 focusing on counting methods and their applications. Section 5 describes the TRIBOND algorithm in more detail, including a proof that a successful deterministic execution of the procedure is carried out in polynomial time. A brief overview of the LIGA algorithm is also presented. Section 6 collects together experimental results for the TRIBOND and LIGA methods and Section 7 contains a brief summary and conclusion.
2. The pair distribution function and the uDGP

The pair distribution function (PDF) is found by taking a Fourier transform of the structure factor measured using total X-ray, neutron or electron scattering from non-crystalline samples [13,1]. PDF analysis considers diffraction in the kinematical limit where each partial wave originates from one scattering event so the signal at a detector in the far field is given by (see e.g. [13,44]),

$$S(\vec{Q}) \propto \sum_{j,l} f_j^* f_l e^{i\vec{Q} \cdot (\vec{r}_j - \vec{r}_l)}$$

where the position of the $l$th atom is $\vec{r}_l$ and $\vec{Q} = \vec{k} - \vec{k}'$ is the scattering wavevector which is the difference of the incoming and outgoing wavevectors of the particle that is scattered. For elastic scattering, the magnitudes of the scattering wavevectors are the same, $k = k'$. $f_j$ is the scattering power of the atom at position $\vec{r}_j$, and $f_j^*$ is its complex conjugate. The term $j = l$ is called the self-scattering term, and is usually treated separately and normalized in the following way,

$$S(\vec{Q}) = 1 + \frac{1}{N\langle f \rangle^2} \sum_{j \neq l} f_j^* f_l \exp\left(i\vec{Q} \cdot \vec{r}_{jl}\right).$$

where $N$ is the number of atoms in the system and $\vec{r}_{jl} = \vec{r}_l - \vec{r}_j$. $\langle f \rangle^2$ is the mean square averaged scattering power, where the average is taken over all scatterers in the system. PDF analysis is usually used when the material samples are amorphous or powders so that the scattering is independent of sample orientation and an average over the scattering angles yields,

$$S(Q) = 1 + \frac{1}{N\langle f \rangle^2} \sum_{j \neq l} f_j^* f_l \frac{\sin(Q r_{jl})}{Q r_{jl}},$$

where $Q$ is related to the scattering angle $\langle \theta \rangle$ by $Q = 4\pi \sin(\langle \theta \rangle)/\lambda$, and $\lambda$ is the wavelength of the incident radiation. The pair distribution function is defined by,

$$g(r) = \frac{1}{r} \frac{1}{N\langle f \rangle^2} \sum_{j \neq l} f_j^* f_l \delta(r - r_{jl}).$$
From this expression it is clear that \( g(r) \) is proportional to the probability of finding an atom at position \( r \) given an atom at the origin, so that a list of the interatomic or interpoint distances can be extracted from \( g(r) \). For convenience in comparing different materials, the PDF above is often normalized so that it approaches one at large distances. From the relations (7) and (8) it is easy to see that the structure factor (the experimentally measured quantity) is related to the pair distribution function through,

\[
g(r) = \frac{2}{\pi} \int_0^\infty Q[S(Q) - 1] \sin(Qr)dQ. \tag{9}
\]

From the point of view of distance geometry, the key feature of the pair distribution function as defined in Eq. (8) is that it contains a list of interatomic distances without any specific reference to the particular atoms at the endpoints of each distance, so finding atomic structure from the PDF requires solution to a uDGP.

### 3. Conditions for unique graph realization

A graph realization is a framework defined by a vector co-ordinate, \( \vec{r}_i, i = 1, \ldots, N \) associated with each vertex, \( i \), in the graph \( G(V, E) \) where the cardinality of the vertex set is \( N \). In the aDGP a set of distance constraints or restraints are imposed and frameworks consistent with these conditions are the target of the optimization problem. In the uDGP the optimization problem also includes a search over all ways of assigning a distance list to the edges of the graph \( G(V, E) \).

Though graph realization is unsolved in general, important progress has been made using two approaches relevant to this work: generic rigidity of graphs and frameworks; build-up procedures that provide sufficient conditions for generic and non-generic frameworks. Graph rigidity applies to generic configurations \([7,17,8,19,15,9]\), which are defined as follows.

**Definition 3.** A graph realization or framework is generic when its rigidity is combinatorially equivalent to that of its underlying graph.

The rigidity of generic frameworks only depends on the connectivity and topology of the framework, and not on the particular geometric arrangements of the components of the framework. Moreover, given the same underlying graph, non-generic realizations may be either more rigid or less rigid than generic cases and there are no general results. Nevertheless most realizations are generic so that studies of generic rigidity provide useful insights in most cases.

Three frameworks that play a central role in rigidity theory are bar–joint frameworks, body–bar frameworks and body–hinge frameworks. Joints connect bars, or they connect a body and a bar, and in general body–body connections are also possible. A hinge is an extended linear connection between two bodies that allows rotation about the line that the hinge defines. A bar is a rigid rod with a joint at each of its ends, while a body is an extended object that may have several joints on its surface. Bars resist axial forces but they do not bend and they rotate freely about the joints that connect them. In the special case of body–bar networks, the joints on the surface of a body only connect to a bar, and there are no direct bar–bar or body–body connections. This restriction enables significant mathematical progress to be made. Whiteley introduced the “Molecular Conjecture” \([21,42,46,20,29]\), that enables progress in the characterization of the rigidity of molecules such as proteins and polymers where bond stretching and bond angle forces are strong. The rigidity of these molecules can be mapped to body–hinge networks, enabling considerable mathematical and algorithmic progress. It is important to distinguish between the following types of framework rigidity:

**Definition 4.** Rigidity of frameworks.

1. A framework is locally rigid in dimension \( K \) if it resists all local deformations in dimension \( K \).
2. A framework is globally rigid in dimension \( K \) if it has a unique realization in dimension \( K \), up to global congruences (rigid body rotations and translations).
3. A framework is universally rigid if it is globally rigid in all embedding dimensions.

Two broad classes of mathematical and algorithmic approaches to characterizing local and global rigidity of generic graphs and frameworks are useful. The first is a combinatorial characterization based on Laman’s theorem and its extensions \([30,17,46,9]\), that has found many applications in statistical physics and biology. In the next section, we discuss Laman characterization methods and their applications in more detail. An alternative characterization is based on matrix theory. Global rigidity of a framework is characterized by checking whether the associated stress matrix has rank \( N - K - 1 \) \([7,8,19,15]\). We do not discuss stress matrix methods further as they are beyond the scope of this work and they have not been utilized in structure determination applications yet. However, this is an interesting direction for further study.

The TRIBOND build-up algorithm for uDGP is based on aDGP methods for building realizations one vertex at a time \([11,49,50]\). The aDGP build-up procedures preserve uniqueness at each step in the build-up process, and provide sufficient but not necessary conditions for uniqueness. For precise aDGP Theorem 1 gives a constructive procedure that, if executed to completion, is sufficient to generate a unique framework in computational time that is \( O(N) \), where \( N = |V| \).
Theorem 1. Dong–Wu (DW) method for precise uDGP [11,49,50]: For dimensions $K = 2$ or $K = 3$, a deterministic procedure for finding a unique realization of a graph $G(V, E)$ given a precise Euclidean distance matrix is as follows.

Find a subgraph $S$ of $G$ and its realization $F$ in dimension $K$, such that $F$ is unique. Then add vertices one by one using the following process.

(i) Find a set of $K + 1$ vertices in $F$, such that the positions of these vertices do not lie in a manifold of dimension $K − 1$.
(ii) Find the unique position implied by the distances from a new vertex to the $K + 1$ vertices identified in (i).
(iii) Add the vertex and its position found in (ii) to the realization, $F$, being constructed.
(iv) If possible, continue this process until the positions of all vertices have been identified.

Proof. A set of $K$ linear equations that can be solved in order to find the position of the new vertex are constructed as follows. For $K = 3$, we define $r$ to be the position of the vertex to be added, with $r^T = (x, y, z)$, and the positions in the existing structure are $r_1, r_2, r_3, r_4$, with $r_i^T = (x_i, y_i, z_i)$. Define the distance between position $r_i$ and $r$ to be $d_i$, so that,

$$d_i^2 = \|r - r_i\|^2 = \|r\|^2 - 2r^Tr_i + \|r_i\|^2.$$  \hspace{1cm} (10)

Subtracting the first of these equations from each of the other three gives three linear equations in the co-ordinates of $r$,

$$d_i^2 - d_j^2 = 2r^T(r_j - r_i) + \|r_i\|^2 - \|r_j\|^2, \hspace{0.5cm} j = 2, 3, 4 \hspace{1cm} (11)$$

which in matrix form is $Mr = b$, where, the $i$th row of $M$ is,

$$M_i = (x_1 - x_{i+1}, y_1 - y_{i+1}, z_1 - z_{i+1}), \hspace{0.5cm} i = 1, 2, 3 \hspace{1cm} (12)$$

and

$$b_i^T = \|r_i\|^2 - \|r_j\|^2 - (d_i^2 - d_{i+1}^2) \hspace{0.5cm} i = 1, 2, 3. \hspace{1cm} (13)$$

A unique solution to these equations exists provided $M$ is non-singular. When a solution to the linear equations is found, its correctness for the non-linear starting problem is checked by substitution into Eq. (10). The requirement for a non-singular matrix $M$ is that $det(M) \neq 0$. Moreover, the determinant of $M$ gives the volume of the tetrahedron defined by the positions $r_1, r_2, r_3, r_4$. When this determinant is zero the volume of the tetrahedron is zero, hence the four points must lie in the same plane. The condition for a unique solution for $r$ is then equivalent to the condition for non-coplanarity of $r_1, r_2, r_3, r_4$. The calculations for $K = 2$ are analogous and the determinant of $M$ gives the area of the triangle defined by the three positions $r_1, r_2, r_3$. When this determinant is zero, the three points are on a common line. \hfill \Box

For generic graphs a necessary condition for success of the DW method is that the underlying graph be globally rigid. Nevertheless even in cases, like proteins in solution or in the unfolded state, where there may be rigid sub-structures connected by flexible components, a rigid cluster decomposition is useful as a preconditioning step. Once globally rigid sub-structures are identified, their realizations may be found using the DW method. A discussion of approaches to rigid cluster decomposition and rigidity percolation are discussed in the next section. Here we focus on extending the DW method to uDGPs.

The uDGPs requires further developments to treat the fact that the assignment of distances to edges in an underlying graph has to be discovered. First we define the true assignment (TA) of distances to edges as follows.

Definition 5. Amongst all of the possible assignments of a distance list to the edges in a graph, $\alpha : L \rightarrow E$, there is one assignment that corresponds to the structure from which the distance list was calculated. We call that assignment the true assignment (TA).

It is useful to consider a special case of the optimization formulation of uDG (Eq. (2)), where the penalty function is quadratic. The following definition provides a precise statement.

Definition 6. Given an assignment, the Hookian spring energy (HSE) is the minimum of expression (2) w.r.t. $X$, with $f(y) = ay^2$ and $a$ strictly positive.

From these definitions it is clear that the HSE of a TA is zero for precise distance lists. Since a uDG having as input a complete distance list leads to $|E|$ different ways that the distances can be assigned to the edges of the graph $G(V, E)$, a key question is whether there are assignments, other than the TA that have zero HSE. This question has been posed in the early days of X-ray scattering by Patterson [39] who introduced the term homometric for systems that have more than one structure for a given elastic scattering pattern. His formulation of the problem considers structures that have the same set of interpoint vectors. The less constrained cases where more than one structure may have the same set of interpoint distances are called weakly homometric structures [45,3,43]. A hexagon is an example with three weakly homometric structures. Though many such cases can be constructed, in situations where the underlying graph has many more edges than the rigidity threshold (see Section 4), homometric structures are unusual both in practical situations such as crystal structure determination or protein conformation, etc, and in random cases such as random point sets [3]. Of course if the number of distances in the distance list is smaller than the rigidity threshold and the underlying graph is not locally rigid, the possibility of weakly homometric or homometric sets occurring is markedly increased. Here we restrict attention to distance lists that yield a unique realization.
Definition 7. Precise distance lists that are not weakly homometric (NWH) have a unique framework with zero HSE.

The following remarks are useful.

Remark 1. The HSE of a TA is zero or in practice of the order of the machine precision.

This is trivial as by definition \( d_{uv} = \|x(v) - x(u)\| \) so \( y = 0 \) for all edges for a TA.

Remark 2. For NWH distance lists all assignments of distances to edges except the TA lead to realizations with strictly positive HSE.

This also follows from Eq. (2) as for any assignment except the TA, and provided the distance list is NWH, there exists at least one edge where \( d_{uv} \neq \|x(v) - x(u)\| \) which leads to a strictly positive HSE.

Extension of the DW buildup procedure to uDG is as follows.

PROCEDURE: uDG-buildup
For dimensions \( K = 2 \) or \( K = 3 \), a procedure for generating a realization of a graph \( G \) given a set of precise interpoint NWH distances is as follows. Find a subgraph \( S \) of \( G \) and its realization \( F \), such that \( F \) is unique. Then add vertices one by one using the following process.

(i) Find a set of \( K + 1 \) “connecting” vertices in \( F \), such that these vertices do not lie in a manifold of dimension \( K - 1 \).

(ii) Search the interpoint distance list to find a set of \( K + 1 \) distances and a position for the new vertex such that each new edge has zero HSE.

(iii) If step (ii) fails restart the process with a new starting condition. A new starting condition is either a new subgraph \( S \), or the use of a different set of edges to build from the same starting subgraph.

(iv) Continue this process until the positions of all vertices have been identified.

Theorem 2. For NWH distance lists, if procedure uDG-buildup runs to completion, then the final framework is the unique TA.

Proof. Remark 1 ensures that procedure uDG-buildup generates a unique realization, while Remark 2 ensures that if a final realization with zero HSE is found, then it is the TA. □

During uDG-buildup “decoy” vertex positions may occur when the set of connecting distances identified in (ii) have zero HSE, but the decoy position does not occur in the correct final structure. Decoy positions lead to failure of buildup at a latter stage. It is straightforward to construct distance lists where decoy positions occur, even for NWH distance lists, however for randomized point sets they are unlikely (see Section 5). When buildup does fail a simple restart of the procedure with a different starting subgraph \( S \), or a different set of \( K + 1 \) connecting vertices to the same \( S \) is usually successful for the cases we have tested.

4. Overview of constraint counting, rigid cluster decomposition and rigidity percolation

James Clerk Maxwell introduced a simple counting argument to assess the stability (local rigidity) of truss structures, which are structures consisting of rigid bars joined at joints [32,40,47], so they are bar–joint frameworks. Since bars only support axial forces Maxwell deduced that each bar provides at most one constraint on the degrees of freedom of the graph. Edges that do not provide additional constraints are called redundant edges, and vertices that have unconstrained degrees of freedom are called floppy vertices. Maxwell’s heuristic constraint counting method for embeddings in \( K \) dimensions assigns each vertex of a graph \( K \) degrees of freedom corresponding to \( K \) translations. Realizing that a rigid structure (body) in \( K \) dimensions has \( K(K + 1)/2 \) global rotations and translations, Maxwell estimated that the minimum number of bars required to ensure a truss structure is rigid is given by \( B_{min} = KN - K(K + 1)/2 \). In two and three dimensions the Maxwell estimate is then \( B_{min}^{(2)} = 2N - 3 \), \( B_{min}^{(3)} = 3N - 6 \). Maxwell counting is not exact as it ignores redundant (dependent) edges. If there are no redundant edges in the network, then the Maxwell estimate is correct and structure is isostatic. Isostatic structures have the property that removal of an edge leads to increased flexibility of the structure. In contrast removal of a redundant edge does not lead to increased flexibility. A landmark in the field of graph rigidity is Laman’s theorem [30] which builds on Maxwell counting to provide a complete characterization of the rigidity of graphs in two dimensions.

Laman’S Theorem. Let a graph \( G \) have exactly \( E = 2N - 3 \) edges, where \( N \) is the number of vertices in \( G \). Then \( G \) is “generically” rigid in \( \mathbb{R}^2 \) iff \( E' \leq 2N' - 3 \) for every subgraph of \( G \) having \( N' \) vertices and \( E' \) edges.

This theorem states that if there are no redundant edges in the graph, then the graph is rigid if the number of bars in the graph is equal to the Maxwell minimum \( B_{min} \). A major step forward in practical applications of Laman’s theorem occurred after Hendrickson noted that a bipartite matching procedure can be used to test whether an edge added to an isostatic graph is redundant [17]. Hendrickson suggested that this algorithm be called the bead game (private correspondence) and this name is used in some of the literature in the area [25]. A schematic of the idea of the matching algorithm is illustrated in Fig. 2.
Fig. 2. Illustration of Hendrickson’s matching algorithm for connectivity percolation (left) and rigidity percolation on a triangular lattice (right). Each bond is tested for redundancy using a bipartite matching algorithm, which fails for redundant bonds. The additional bonds utilized in the test account for the global degrees of freedom of a rigid (connected) cluster. An arrow pointing to a vertex indicates that the bond is matched to a degree of freedom at that vertex. The top two figures illustrate successful matchings indicating that the bond is not redundant, while the bottom two figures have failed matchings indicating that the tested bond is redundant (from [36]).

for the case of testing the rigidity of a subgraph of a triangular lattice embedded in two dimensions (where the condition for an isostatic graph is \( E = 2N - 3 \) provided there are no redundant edges). This subgraph has three global degrees of freedom (one rotation and two translations) and this is accommodated in the right figures of Fig. 2 by three additional “ghost” edges (light black rounded arrows). The edge being tested is the dark arrow that joins the same two vertices as the ghost edges. The idea of the algorithm is that each vertex has two degrees of freedom and that each edge (arrow) constrains one degree of freedom. In the figure an arrow pointing to a vertex indicates matching of that bond to a degree of freedom of the vertex that the arrow points to. If a matching of all arrows to vertices is possible, then the graph is isostatic or underconstrained, while if the matching fails the tested edge is redundant. The failed matching can also be used to find the overconstrained cluster that is associated with the redundant edge. An overconstrained cluster is a set of edges that are mutually redundant so that any one of the edges in the cluster may be removed without changing the number of floppy modes in the framework.

The ideas of constraint counting can be extended to the case of graph connectivity, and Hendrickson’s construction can be applied to this case as well, as illustrated in the left figures in Fig. 2. In connectivity problems isostatic graphs are trees, overconstrained graphs contain cycles, and only one ghost edge is required in the matching algorithm [36].

In practice rigidity algorithms based on Hendrickson’s method add edges one at a time, testing each edge for redundancy using a bipartite matching algorithm. If a tested edge is redundant it is not added to the graph so the procedure can be used to test other edges. As more edges are added the size of the rigid (isostatic) clusters grows and the bipartite matching procedure becomes more computationally intensive. Jacobs [24,22] and Moukarzel [35,33] found two procedures to resolve this problem and the most efficient algorithms based on these approaches are experimentally observed to be close to \( O(N) \) for sparse graphs. Generalization of Laman’s theorem to embeddings in higher dimensions is in general unsolved, however several important cases are reducible to a similar characterization; including graphs conforming to the molecular conjecture [21,42,46,20,29], and body–bar networks in any dimension [46,9].

Hendrickson’s algorithm and its extensions have proven very useful in statistical physics and in assessing protein rigidity, with one interesting application to characterizing behavior near the rigidity percolation threshold [24,35,25,36]. The rigidity percolation threshold is the threshold at which a giant rigid cluster emerges and is intuitively analogous to the emergence of a giant connected cluster in connectivity percolation. In studies of percolation, edges are added between randomly chosen vertex pairs until enough edges are present in the graph to induce an infinite rigid cluster. The geometry of the giant rigid cluster (right figure) and the giant connected cluster (left figure) for a triangular lattice at the percolation threshold are presented in Fig. 3, where darker bonds are overconstrained while lighter bonds are isostatic. Clearly the giant rigid cluster at its percolation threshold is denser than the giant connected cluster at its percolation threshold. These results were found by adding edges randomly between vertices in a triangular lattice and after each bond addition, Hendrickson’s matching method is used to test whether the edge is overconstrained with respect the edges already in the graph. Though both connectivity and bar–joint rigidity percolation transitions on triangular lattices are continuous, their critical exponents are different and the threshold for rigidity percolation is much higher than that for connectivity percolation. The difference between connectivity and rigidity percolation processes is more extreme for random graphs and for graphs in higher
Fig. 3. The geometry of the infinite cluster (giant cluster) at the percolation threshold on a triangular lattice: The left figure is for connectivity percolation, while the right figure is for rigidity percolation in a bar–joint system. Both of these transitions are second order with a fractal stress (thicker bonds in the right figure) or current carrying backbone (thicker bonds in the left figure) at their percolation thresholds. The lighter colored thin bonds in both figures are connected (left figure) or rigid (right figure) but they do not transmit current (left figure) or stress (right figure) that is applied at the top and bottom edges of the networks. (from [36]).

Fig. 4. The infinite cluster (circles) and stressed backbone (triangles) probability for a bar–joint rigidity percolation problem on a body-centered cubic lattice. In this case the rigidity percolation phase transition is strongly first order as found for random graphs. From [5].

dimensions, as demonstrated using analytic solutions at the tree level [37,12], and in general rigidity percolation is prone
to a strongly discontinuous transition, while connectivity percolation is typically a continuous transition.

An extension of connectivity percolation called $k$-core percolation considers vertices that have local $k$-connectivity, and
$k$-core clusters are found by recursively culling all vertices that are connected to less than $k$ neighbors. At the tree level
there are some close similarities between the emergence of a giant cluster in $k$-core percolation [41] and the emergence
of a giant rigid cluster. However there are important differences that move the rigidity threshold to higher values than the
$k$-core threshold on random graphs [12]. In both $k$-core percolation with $k > 1$ and rigidity percolation the phase transition
is most often discontinuous.

Bar–joint rigidity for graphs embedded in three dimensions does not adhere to Laman's theorem, however it is
straightforward to solve for the stress distribution in a network where the graph edges are replaced by Hookian springs,
to find the regions that carry stress. Below the rigidity threshold, there is no restoring force to an applied stress at long
length scales, however above the rigidity threshold there is. Using direct methods the nature of the rigidity transition for
body centered cubic lattices has been elucidated, with the result presented in Fig. 4 demonstrating a strongly discontinuous
behavior [5]. These studies support the assertion that rigidity percolation is often strongly first order.

Extensions of Hendrickson's matching methods have been used to study the rigidity of proteins when crosslinks such
as disulfide bonds and hydrogen bonds are added to the protein bonding network. This approach yields a rigid cluster
decomposition that indicates the regions of the protein that are most structurally rigid and other regions or mechanisms that
are either flexible or that can be made flexible by removing a small number of crosslinks [23,42,14]. More recently rigidity
concepts have been applied to the study of behavior near the dense packing threshold of random granular packings [34,51].
Fig. 5. Examples of a CORE in $K = 2$ (Left) and in $K = 3$ (Right). A core is the smallest cluster that contains a redundant bond in a generic graph rigidity sense. The colors indicate the manner in which the core is constructed algorithmically. For the two dimensional case (left figure), the horizontal bond is the base (in black), the bonds below it (in blue) make up the base triangle while those above it (in red) make up the top triangle. The vertical bond is the bridge bond (in green). The extension to $K = 3$ requires a base triangle (black), feasible tetrahedra compatible with the base triangle (blue, red) and finally a bridging bond (green) that is consistent with the target distance list. The bridging bond is the final bond added and is overconstrained. For precise distance lists, the bridge bond must have zero HSE for the core to be correct. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5. uDGP graph realization algorithms

Two types of embedding algorithms have been developed for uDGPs, both based on build-up procedures. The first algorithm called TRIBOND is based on the procedure described in Section 3 [16], and in most cases precise distance lists have been assumed. TRIBOND algorithms for embedding in two and three dimensions and for the cases of imprecise distance lists have been developed, and some preliminary results are presented. The second algorithm is called LIGA [26]. It is a stochastic build-up algorithm that incorporates methods based on tournaments and backtracking. LIGA has been used to reconstruct fullerenes and a variety of crystal structures from experimental PDF data [26–28]. These distance lists are imprecise and there are some missing distances.

The TRIBOND algorithm is an implementation of procedure-uDG build-up, and a procedure for finding a subgraph and its realization to start this procedure (CORE finding). We consider graphs with $N$ vertices and all $N(N − 1)/2$ interatomic distances in the input distance list. These uDGPs are highly overconstrained for $N > 10$. We start with a version of TRIBOND that is based on starting with the smallest overconstrained cluster or “CORE” which, as we show below, for two dimensional embeddings ($K = 2$) has four vertices and for embeddings in three dimensions ($K = 3$) has five vertices as illustrated in Fig. 5. In the discussion below the number of vertices in a subgraph is $n$ while the number of vertices in the graph is $N$. We define the CORE as follows:

Definition 8. The CORE of a buildup algorithm using precise distance lists is a zero HSE realization of the smallest-size overconstrained cluster.

The size of a CORE in dimension $K$ is given by the following theorem which follows from well known results in the theory of graph rigidity:[17].

Theorem 3. For precise distance lists, the CORE of a graph realization from a distance list $D$ in Euclidean dimension $K$ is a zero cost embedding of a $(K + 2) −$ clique using $(K + 1)(K + 2)/2$ distances from $D$.

Proof. A CORE is the smallest generic overconstrained cluster. The size of this cluster can be found by constraint counting. The number of global translations and rotations of a rigid cluster in $K$ dimensions is $K(K + 1)/2$, so the minimum number of redundant edges $R$ in a cluster is given by:

$$R = n(n − 1)/2 + K(K + 1)/2 − nK$$

(14)

where $n(n − 1)/2$ is the number of interatomic distances or edges in the underlying clique, $nK$ is the number of translational degrees of freedom of $n$ points in $K$ dimensions, and $K(K + 1)/2$ is the number of rotations and translations of a rigid body
in $K$ dimensions. Structures with $R = 0$ are isostatic, while those with $R = 1$ are the first overconstrained clusters. Setting $R = 1$ and solving, we find the solution $n_c(K) = K + 2$ as required. Note that for $R = 1$ all of the edges in the cluster are redundant in the sense that any one, but only one, of them can be removed to yield an isostatic cluster.

**TRIBOND Algorithm** (precise NWH distance list)

**INPUT:** An ordered list of interatomic distances, $D$, of cardinality $m$.

**CORE FINDING**

Search over all feasible arrangements of $(K + 2)(K + 1)/2$ distances selected from the input list of interatomic distances until a $(K + 2) - \text{clique}$ is found, such that the HSE cost of the clique embedding is smaller than machine precision.

FOR $i = 1, N - K - 2$

ADD A VERTEX

Choose $K + 1$ vertices in the existing structure that do not lie in a manifold of dimension $K - 1$.

Search over all ways of choosing $K + 1$ distances from the list of interatomic distances until a set with zero HSE is found to connect a new vertex to the $K + 1$ vertices in the substructure.

IF connecting edges with zero HSE cannot be found, RESTART.

A new starting condition is either a new core, or a different set of edges to start the buildup from the same core.

END

**OUTPUT:** A list of atomic co-ordinates for $N$ atoms.

### 5.1. Core finding procedure

The core of an embedding is found by choosing a starting distance, which in our implementation is usually the shortest distance. This is called the base of the core. A “window” of distances is selected and triangles are constructed using distances chosen from this window. For embeddings in two dimensions pairs of triangles are then tested for compatibility with distances in the distance list Fig. 5. This “bridge bond check” is carried out using half interval search. For most triangle pairs a compatible bridge bond does not exist and another pair of triangles is selected. For embeddings in three dimensions a similar procedure is carried out, except that two tetrahedra are generated and the bridge bond check tests the compatibility of these two tetrahedra with respect to the distances available in the input distance list.

### 5.2. Adding a vertex

Once a CORE is found, the TRIBOND algorithm adds vertices one at a time. For embeddings in two dimensions a “base triangle” is chosen from the existing globally rigid substructure. In our implementation this base triangle is usually a triangle including the original base bond. A second triangle is generated using a new vertex and two unused distances in the distance list. A bridge-bond-check is then carried out to determine whether a distance in the input distance list is compatible with these two triangles. For three dimensions, a base tetrahedron is selected and a new tetrahedron is generated using a triangle from the base tetrahedron, a new vertex and three unused distances from the input distance list. A bridge bond check is then carried out to test the compatibility of these two tetrahedra Fig. 5.

**Theorem 4.** Successful deterministic buildup without restarts using TRIBOND is in P, and is bounded above by $\tau < O(N^{(K+2)(K+1)})$ for $K \geq 2$ for dense lists where all $N(N - 1)/2$ pair distances are available.

**Proof.** This follows straightforwardly from the combinatorics of the problem. To ensure that we find a CORE of size $K + 2$ in $K$ dimensions we must search over all ways of choosing $n_c(n_c - 1)/2 = (K + 2)(K + 1)/2$ distances from a distance list of length $N(N - 1)/2$, which yields a coarse upper bound on the time for core-finding $\tau_c < O(N^{(K+2)(K+1)})$ which is in P. Buildup requires adding $N - K - 2$ vertices one at a time, with each vertex addition requiring correctly choosing $K + 1$ distances from a list of length $N(N - 1)/2$. A coarse upper bound on buildup is then $\tau_b < O(N^{2K+3})$, which is also in P and for $K \geq 2$ has a smaller exponent than core-finding.

For random point sets with $N < 1000$, we find that restarts are seldom required (see Section 6). In that case, the upper bounds presented above are higher than the typical time taken for TRIBOND as there are $\binom{N}{K+2}$ distinct cores so that the typical time for a naive core finding algorithm is $O(N^{K(K+2)})$. This is further reduced by noting (i) that the choice of the first distance does not require a search over the distance list, (ii) that subsequent searches can be constrained using the triangle inequality and (iii) that the search for the last edge can be carried out using a half interval search. Implementation
of these features leads to a core-finding algorithm that is typically $O(N^{K(K+2)-2\ln(N)})$. One further empirical observation is that choosing the first distance to be small leads to an improved performance due to the fact that the triangle inequality constraint is more effective for this case.

5.3. The liGA algorithm

LIGA is a stochastic algorithm that works well for high symmetry structures such as fullerenes and crystal structures [26–28], where distances with high multiplicity are typical. LIGA uses a combination of ideas from dynamic programming with backtracking, and tournaments. It builds up a candidate structure by starting with a base bond and adding atoms one at a time but instead of using systematic search it uses a stochastic procedure. As in TRIBOND, the most difficult part of the process is finding a CORE. Moreover, to account for the possibility that the core might be wrong or that the build-up of larger structures might be wrong, the LIGA algorithm keeps a set of low cost candidate sub-structures at each size. The algorithm then iteratively updates the candidate sub-structures at each size using a promotion and relegation procedure based on European football leagues (hence the name “LIGA”). In most cases the promotion and relegation procedures involve adding an atom to a sub-structure or removing a high cost atom from a sub-structure, however in some cases a sequence of vertex additions is invoked to rapidly advance very promising sub-structures. In LIGA, the cost of a vertex (atom) is the sum of the harmonic spring energies of all of the edges incident on the vertex.

6. Experimental results

Experimental measurements of the mean computational time of the TRIBOND algorithm for a series of precise random point sets in the plane of increasing size, $N = 8, 16, \ldots, 512$, are presented in (Fig. 6). From this figure it is evident that the time required for build-up is about an order of magnitude less than that for core finding, however both components of the algorithm require computational time that scales as $\tau \sim N^{3.32}$. In contrast build-up for precise aDGP problems is $O(N)$ [11,49,50]. The experimental exponent is much lower than the coarse upper bound of Theorem 4 or the reduced estimate in the section after this theorem, so that large random point sets can be reconstructed quite efficiently. For example, using TRIBOND in two dimensions we have reconstructed random point sets with several thousand vertices in less than a day on a 2-core 2.2 GHz desktop computer. Tribond is also able to reconstruct structures such as model polymer chains as illustrated in Fig. 7. In these calculations the structure that is found is as precise as the input distance list, which in this case is machine precision.

Extensions to treat imprecise distance lists is challenging as many small substructures may be compatible with imprecise distance lists and yet not be part of larger valid structures. However if we know a validated substructure, then many overconstrained or “bridge bond checks” can be carried out to verify if an added vertex and its associated edges are correct, that is, we check more than $K + 1$ bonds per vertex addition. Taking this one step further we consider the addition of two vertices along with all of the bond checks that these two vertices can have with a validated substructure. In many cases such as proteins and organic molecules validated substructures may be aromatic rings or fragments of amino acid residues or other motifs that are known to be globally rigid and structurally well characterized. These subunits can be used as validated substructures to use as a base for reconstructing the large scale structure of complex molecules even when the distances are not precise. To test these ideas we have modified the TRIBOND algorithm to build structures using a validated substructure as a base, and the results are presented in Fig. 8. From this figure it is evident that as the precision of the distance list data grows, the size of the validated substructure required for build-up reduces, and that for precision above one part in a million only a four vertex core is required in two dimensions.
Fig. 7. (Color online) A model polymer chain. Using only the list of interatomic distances extracted from this model, TRIBOND was able to successfully reconstruct the above structure, which has \( N = 100 \), in a few minutes on a desktop computer (dual-core 2.2 GHz processor and 2 GB of memory). From [16].

Fig. 8. Plot of minimum initial substructure size required to find a correct embedding vs precision of the input distance list for random point sets of different sizes (\( N = 26, 50, 76 \) and 100). A larger initial sub-structure is needed for less precise distance lists. The typical run time for \( N = 26, 50, 76, 100 \) was about 1 min, 10 min, 4 h and 20 h respectively, on a computer with a dual-core 2.2 GHz processor and 2 GB of memory. From [16].

We have recently extended TRIBOND to solve uDGPs in three dimensions. Fig. 9 demonstrates that for the reconstruction of random point sets in three dimensions, using the smallest bond in the distance list as the base bond leads to improved performance, as occurs for embeddings in two dimensions. The mean computational time, as measured by the number of bridge bond checks, for reconstructing random point sets in three dimensions is presented in Fig. 10. The scaling is found to be polynomial with an exponent close to five. We have compared the performance of TRIBOND and LIGA for a wide range of molecules, such as the quinine molecule (see Fig. 11) and have found that TRIBOND has a higher success rate than LIGA, especially for large and more complex molecules. Finally a preliminary study of the minimum core size required for buildup as a function of the precision of the distance list is presented in Fig. 12. Clearly further work to improve the computational efficiency of the TRIBOND algorithm, particularly for embeddings in three dimensions using imprecise distance lists is highly desirable.

Extracting a list of interpoint distances from experimental PDF data is challenging and leads to imprecise results. Nevertheless reconstruction is possible, and some intermediate structures found by LIGA during reconstruction of a fullerene using these distance lists, is illustrated in Fig. 13.

Fullerenes have only one atom type, however many molecules, nanoparticles and complex materials have binary or ternary atomic stoichiometries. To handle these more complex problems, we developed a multistep process 14 (see Fig. 14). The first step is to extract a list of distances from the PDF. The degeneracy of these distances is “soft” allowing for variation in the number of times a given distance is used in the reconstruction. Using this approach a list of low cost candidate structures is generated using TRIBOND or LIGA. In a second step a search over configurations of atom types on the vertices of the low cost structures is carried out. We call this the vertex coloring step. In this step the full PDF is generated for each candidate “coloring” and the PDF of a candidate structure is generated and compared to the experimental PDF. Vertex coloring is combinatorially hard, however in practice information about the chemistry of the system can be used to reduce the number...
Fig. 9. The size of the bond window that is required to find a core for reconstruction of random point sets with \( N = 50 \) vertices in three dimensions \( (K = 3) \). The smaller the base bond the smaller the window size required, so that the best practice is to choose the smallest bond as the base bond for the reconstruction.

Fig. 10. The average number of bridge or overconstrained bond checks required to reconstruct random point sets with \( N \) vertices in three dimensions \( (K = 3) \). Each point is an average over 10 instances of random points sets. A power law fit to this data yields a power law exponent of close to 5.

Fig. 11. (Color online) TRIBOND has been used to reconstruct a large number of complex molecules embedded in three dimensions, including the Quinine molecule illustrated here. This molecule has \( N = 48 \) (48 atoms), and chemical formula \( \text{C}_{20}\text{H}_{24}\text{N}_{2}\text{O}_{2} \). For exact distance lists, the time taken to reconstruct this molecule was 84.4 s on a computer with a dual-core 2.2 GHz processor and 2 GB of memory.

of configurations that need to be searched. For some cases this reduces the complexity of the chemical coloring problem dramatically [28].
Fig. 12. Plot of the minimum core size required for reconstruction vs precision of the input distance list for random point sets in three dimensions. As seen for the two-dimensional case, a bigger core is required as the distance lists becomes less precise. The time taken for reconstruction also rises rapidly as a much larger number of overconstrained bonds (bridge bonds) must be checked to ensure the added vertex or vertices are correct. Despite these checks the buildup sometimes fails and it is necessary to restart the process with a different base bond.

Fig. 13. (Color online) Reconstruction of a bucky-ball from experimental neutron scattering data. Figures a–c are various forms of the experimental data, starting with the raw structure factor data. The imprecise distance list of Fig. d extracted from this data is the only information that is used to find the correct structure. The LIGA algorithm finds this solution using a cluster growth procedure with backtracking as schematically illustrated in the lower figure and discussed in more detail in Section 5. From [26,27].

7. Summary and conclusions

Though both aDGP and uDGP are NP-hard, in practice uDGP requires more computational effort than aDGP due to the need to discover the underlying graph in addition to finding the positions of each vertex in the structure. Some first steps
Fig. 14. (Color online) Schematic of a procedure for reconstruction of a cluster with two types of atoms from imperfect pair distribution function data. The unassigned list of interatomic distances is extracted from the data and the structure is found from it, without enforcing the multiplicity strictly. This is followed by identification of the atom types using the multiplicity or PDF peak height information. An extension of this method is able to find the atomic structure of crystalline materials [28].

toward a mathematical basis for uDGPs arising in nanostructure determination were elucidated, and associated algorithms, TRIBOND and LIGA, were presented.

Weakly homometric distance lists do not have unique realizations and remain challenging. We considered distance lists that are not weakly homometric (NWH) and for these cases a buildup algorithm was constructed (TRIBOND) that generates a unique framework for precise distances lists. Though the algorithm may require restarts due to generation of decoy positions at intermediate steps, in the experiments presented here restarts are required in less than 10% of cases. In the absence of restarts we showed that TRIBOND is in P.

Experimental results for distance lists extracted from random point sets in two and three dimensions indicate that the computational time grows with the number of vertices in the graph to an exponent near three for embeddings in two dimensions and with an exponent near five for embeddings in three dimensions. TRIBOND is designed to solve problems where the distance lists have many unique distances, while the LIGA algorithm is designed to treat cases where the underlying structures are more symmetric and hence their distance lists have a great deal of multiplicity. The LIGA algorithm is stochastic and keeps a population of candidate structures at each size. By using a backtracking procedure with tournaments, high quality structures are eventually generated. LIGA does not use the deterministic strategies used in TRIBOND and the next generation algorithms should combine elements of the strategies used in TRIBOND and LIGA. The LIGA algorithm has already been used to solve structures from imprecise distance lists extracted from experimental PDF data.

The combinatorial approaches to structure determination described here provide an alternative to continuous global optimization approaches that have been applied to this class of problem. As occurs in determining protein structure from NMR distance restraints, the most commonly used approach to solving structure from PDF data utilizes simulated annealing. In the PDF area this approach is called reverse Monte Carlo or RMC and has been particularly successful in generating ensembles of structures in underconstrained problems. For more challenging problems where a unique structure is sought, RMC or simulated annealing methods work quite well if sufficient prior information is available to confine the search space, however we have found that RMC is not able to invert uDGP distance lists to find the structure of random point sets of size greater than ten vertices, and this approach also fails for the experimental problems solved by LIGA.

The two leading algorithms for uDGP at the present time, LIGA and TRIBOND, can typically solve problems up to hundreds of atoms for high precision distance lists. Further progress is needed to extend the range of applications of uDGPs, to improve algorithms for imprecise distances and to consider problems where the distance lists are sparse.

Acknowledgments

Support for work at Michigan State University by the MSU foundation is gratefully acknowledged. PMD would like to thank Carlile Lavor and Leo Liberti for interesting and motivating discussions and the referees for very constructive