PHYSICAL REVIEW E **89**, 053311 (2014)

Ab-initio reconstruction of complex Euclidean networks in two dimensions

S. R. Gujarathi, C. L. Farrow, C. Glosser, L. Granlund, and P. M. Duxbury[*]
Department of Physics and Astronomy, Michigan State University, East Lansing, Michigan 48824, USA

(Received 19 August 2013; published 20 May 2014)

Reconstruction of complex structures is an inverse problem arising in virtually all areas of science and technology, from protein structure determination to bulk heterostructure solar cells and the structure of nanoparticles. We cast this problem as a complex network problem where the edges in a network have weights equal to the Euclidean distance between their endpoints. We present a method for reconstruction of the locations of the nodes of the network given only the edge weights of the Euclidean network. The theoretical foundations of the method are based on rigidity theory, which enables derivation of a polynomial bound on its efficiency. An efficient implementation of the method is discussed and timing results indicate that the run time of the algorithm is polynomial in the number of nodes in the network. We have reconstructed Euclidean networks of about 1000 nodes in approximately 24 h on a desktop computer using this implementation. We also reconstruct Euclidean networks corresponding to polymer chains in two dimensions and planar graphene nanoparticles. We have also modified our base algorithm so that it can successfully solve random point sets when the input data are less precise.

DOI: 10.1103/PhysRevE.89.053311 PACS number(s): 05.10.−a, 05.20.−y, 61.43.−j

I. INTRODUCTION

Reconstruction of heterogeneous and complex systems using pair correlation functions or pair distance information is a problem that arises in many branches of materials physics [1,2], in biology [3–6], and in a variety of engineering applications [7,8]. We distinguish between two problems: first, where the objective is to find a statistical characterization of a heterogeneous system consistent with experimental information. In these cases the reconstruction is not unique, but instead generates an ensemble of structures that are on average consistent with the data. Reverse Monte Carlo methods [9] for the atomic structure of glasses and simulated annealing methods for a range of heterogeneous materials are in this class. Large samples are often used and the system is highly underconstrained as there are many more degrees of freedom in the model than there is information in the data. Second, the related but significantly different problem where we seek to reconstruct a specific, unique network or structure. The amount of information in the data must constrain the degrees of freedom in the structure. This problem can be hard for structures with only ten to hundreds of atoms or components. Uniqueness is lost when the model has too many degrees of freedom as compared to the available data. This unique structure problem is the focus of our study. Surprisingly, we find that it is possible to efficiently reconstruct large complex structures in two dimensions, given only Euclidean distance information.

The practice of crystallography represents the gold standard for structure determination and it provides methods to overcome the phase problem. If there are no homometric variants [10], it provides a unique crystal structure. When crystals are not available, but a unique structure is still the objective, new methods are required. One successful approach is the determination of protein structure in solution that may be found by using pair distance information extracted from NOESY NMR data [4,5,11–13]. Two other approaches are emerging. The first is determination of the structure of individual nanoparticles using lensless imaging algorithms [14–17]. The second approach is extracting a list of interatomic distances from scattering data and solving a new inverse problem to find the atom locations. Here we present a highly efficient method to solve the latter inverse problem for the case of complex networks or random point sets in two dimensions.

As discussed recently in [18–20] by Torquato and collaborators, reconstruction of heterogeneous systems in general requires multipoint correlation functions. However, pair correlations are by far the most readily available structural data for heterogeneous materials as they are found by a Fourier transform of elastic electron, x-ray, or neutron scattering data collected, for example, at national facilities. This provides a strong motivation to find methods to determine the extent to which we can reconstruct heterogeneous systems only using pair information. The most fundamental pair information is the list of distances between points or atoms in a structure, reducing the problem to an inverse problem, namely, given a set of interatomic distances find the location of the atoms, up to global rotations, translations, and reflections of the structure. This pair distance inverse problem (PD-IP) may be interpreted as a complex network reconstruction problem where the edge weights are equal to the Euclidean distances between nodes in the network.

The PD-IP is central to determining protein structure from NMR data, however there are vital differences between the problem we study and the NMR PD-IP problem. The most important difference is that the list of residues or sequence of a protein is known, enabling mutation and other experiments to be carried out to specify the points between which each distance lies. This leads to the *assigned* pair distance inverse problem (APD). In contrast, pair distances are not assigned in problems concerning, for example, many materials and most heterogeneous media. This is the unassigned pair distance inverse problem (UPD), and it is a significantly harder inverse

[*] duxbury@pa.msu.edu

problem. In fact, APD algorithms for reconstruction of atom locations from precise distances is known to be easy, being of order the number of atoms in the structure (N). However, NMR is plagued by uncertainties in the experimentally determined interatomic distances, typically of order 25% or higher [21]. The problem of finding protein structure from NMR data is then best treated using loose restraints rather than hard distance constraints. The energy landscape of the APD with loose constraints has many of the features of spin glass problems, and structure determination with imprecise or missing distances is widely believed to be computationally challenging (NP hard) [1,22–25].

In almost all other Euclidean network reconstruction problems the distances are not assigned, as the data do not indicate which nodes lie at the end of each distance. For example, the pair distribution function method is used for the analysis of the local structure of nanoparticles and complex materials. In many complex materials, such as high performance thermoelectric materials [26], high temperature superconductors [27], and manganites [28], crystalline order and heterogeneous local distortions co-exist so that crystallographic and PDF methods are complementary. Crystallography finds the average structure and the PDF of the local structure [29,30]. The pair distribution function gives a direct measure of the list of interatomic distances arising in the local structure, however the endpoints of the distances are not known so we face a computationally challenging UPD problem known as the nanostructure problem [31].

Recently, in collaboration with the Professor Billinge's group at Columbia University, we developed efficient algorithms for the UPD problem for cases where there is significant symmetry in the structure, including C_{60} and a range of crystal structures. In those cases we found that two types of algorithm worked well, genetic algorithms and the novel Liga algorithm [32–34] that uses a combination of ideas from dynamic programming with backtracking, and tournaments. Though these methods work well for structures with relatively high symmetry, solving structures with hundreds of points, they fail miserably for low symmetry problems such as random point sets due to the large number of unique pair distances in random structures. They thus fail for the general problem of complex Euclidean networks.

A formal statement of the UPD problem is as follows. We are given a list of distances $\{d_l\}$, $l = 1 \ldots M$, between N points in a D-dimensional Euclidean space, where $M = N(N-1)/2$. Our task is to find coordinates of the points $\{\vec{r}_i\}$, $i = 1, \ldots, N$ such that the distance between every pair of points $|\vec{r}_i - \vec{r}_j| = r_{ij}$ is a member of the distance list $\{d_l\}$. Moreover, we require that every distance in the list $\{d_l\}$ occurs for some pair of points (i,j) in the structure.

The only inputs to the Euclidean network reconstruction algorithm described below are the number of points in the network N and the interpoint Euclidean distances. Physically, it is useful to think of the Euclidean distances as natural lengths of Hookian springs, l_{ij}^0 so that we may define an energy function,

$$E\left(\{l_{ij}^0\}\right) = \sum_{ij} k_{ij}\left(l_{ij} - l_{ij}^0\right)^2. \tag{1}$$

In the ideal UPD problem the distance list is known precisely, but we don't know the mapping (i.e., assignment) $d_l \rightarrow l_{ij}^0$. This is the precise UPD. In the precise UPD the key computational difficulty is to find this mapping or assignment of d_l to l_{ij}. If the correct assignment is found the energy is zero, while wrong assignments lead to stretched or compressed springs and nonzero energy. Here, we present an algorithm that solves the unassigned problem (UPD) in the precise case in two dimensions and hope that it will offer insights that lead to techniques for solving the imprecise case.

Two examples of this problem are shown in Fig. 1. The top panel of Fig. 1 presents an example of a degenerate distance list (i.e., distances are repeated), typical of structures which have high symmetry, while the bottom panel of Fig. 1 is an example of a random point set where all distances are, with high probability, unique. Since the number of Euclidean distances is $M = N(N-1)/2$, a search over all permutations of the distances to find the correct assignment of d_l to l_{ij} requires computational time proportional to the factorial of M. This is worse than exponential time complexity and thus a very poor way to proceed.

The rest of this paper is organized as follows. Section II summarizes the theoretical concepts upon which the UPD reconstruction algorithm is based. The key concepts, based on constraint counting and generic graph rigidity, have a long history in the physics and mathematics literature. Section III discusses implementation of the procedure, which broadly consists of two steps: core finding and buildup. A naive implementation is quite inefficient, however a simple optimization where cores are found using a selected subset of the distance list provides a much more efficient implementation. We also develop a loose polynomial upper bound on the computational efficiency of the algorithm and compare it with the actual data. Large random point sets may yield distance lists that are close to degenerate, leading to problems with reconstruction. Section IV discusses potential applications and extensions of the algorithm. Finally in Section V, we make our concluding remarks.

II. RIGIDITY THEORY OF UNASSIGNED PD-IP

Graph rigidity theory addresses how many independent constraints are required to ensure that a graph is rigid [35–38]. This subject was initiated by James Clerk Maxwell, leading to the development of mathematical theories of graph rigidity and physical approximations to the rigidity of glasses. In D dimensions a point has D translational degrees of freedom, so a structure with N nodes has DN degrees of freedom. The number of internal degrees of freedom is $DN - D(D+1)/2$ as there are $D(D+1)/2 = D + D(D-1)/2$ degrees of freedom due to global translations (D) and rotations [$D(D-1)/2$]. An object is rigid when its internal degrees of freedom are constrained, leaving only its global rotations and translations.

A constraint such as an interpoint distance contributes to the rigidity of a structure only if it is linearly independent with respect to the other constraints in the structure, so that identification of such constraints is key to accurate constraint counting. Several mechanisms for the linear dependence of constraints in small structures are illustrated in [22].

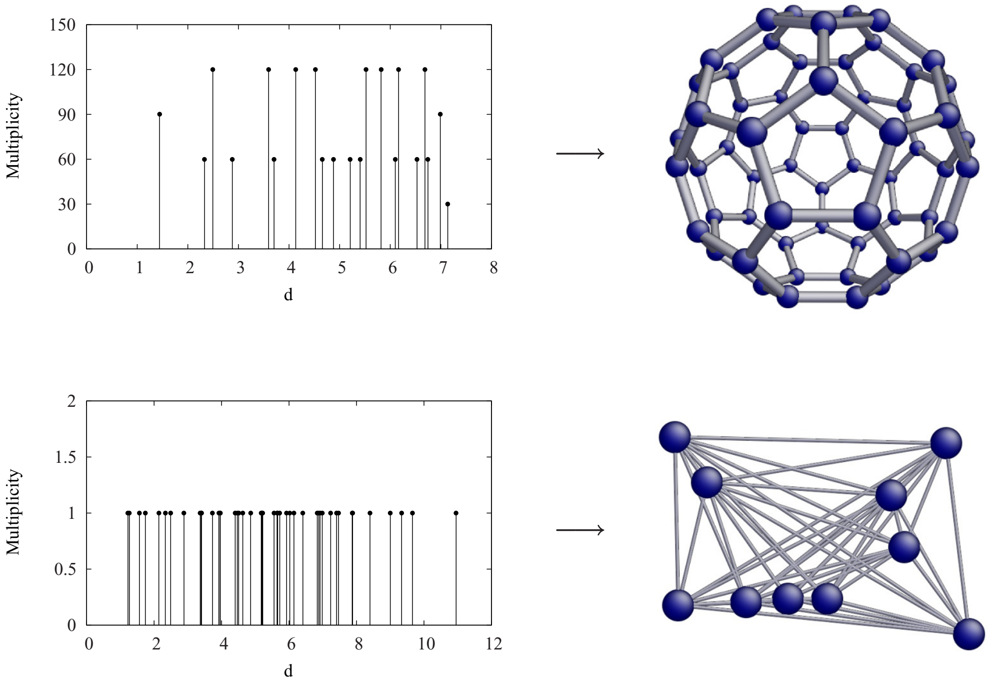

FIG. 1. (Color online) Simple examples of structures found from Euclidean distance lists. The figures on the left are plots of the distance lists for top: a C_{60} fullerene that has a degenerate distance list; and bottom: a random set of ten points in the plane that has a nondegenerate distance list. The fullerene has a total of 1770 interatomic distances, but only 21 unique distances. The random point set has 45 distances, which are with high probability unique. The multiplicity is on the vertical axis while the distance is on the horizontal axis (in arbitrary units). The figures on the right hand side are solutions to the inverse problem found using the Liga algorithm (fullerene) and Tribond (random point set) to find the structure from the given distance lists, without the use of any other information. For the random point set all interatomic distances are drawn in the figure. For clarity only the nearest neighbor bonds are drawn in the fullerene case. In this study, the distance lists are taken from the known structure and then we try to solve the inverse problem using only the distance list. In practice, the structure is unknown and the distance lists are derived from experiments, particularly x-ray and neutron scattering data.

In a classic paper, Laman [39] presented a combinatorial characterization of the rigidity of graphs in the plane and Henrickson [22] provided the basis for efficient algorithms that have been widely applied in physics, applied mathematics, and in biology. Note that if a graph is rigid it can support an applied stress. Addition of further bonds or edges (redundant bonds) to a rigid graph does not increase its rigidity, though of course the elastic moduli continue to increase as further bonds are added. Redundant bonds lead to overconstraint except in special cases. This is an important feature in physical situations, where energy is almost certainly nonzero, because they are a source of internal stress.

Now we are ready to address how much information is necessary to solve the UPD problem. The critical number of independent constraints, B_c, required to make the network rigid is

$$B_c = DN - D(D+1)/2. \tag{2}$$

In an ideal NMR or PDF experiment all interparticle distances would be extracted so that the number of interparticle constraints would be $N(N-1)/2$, which appears to be more than enough to constrain the structure. However, it is not clear that an interparticle distance corresponds to an independent constraint. In fact, the number of independent constraints is given by the number of independent rows in the rigidity matrix, the elements of which are derived from the vector differences between nodes in the relevant Euclidean vector space. Although an edge defined in this vector space may correspond to one of the independent rows, Euclidean distances alone are insufficient for constraint counting. This can be seen by considering the C_{60} molecule as illustrated in Fig. 1, where there are only 21 different interatomic distances. Since for a buckyball, $B_c = 3 \times 60 - 6 = 174 \gg 21$, it appears that there are far fewer distance constraints than required to find the correct structure using the distance list alone. However, distances with the same length need not be linearly dependent as they may have different directions in the structure. Mathematical analysis of this important issue is currently absent. In contrast, for the generic random point sets that are of interest here, all the distances and directions are, with high probability, unique. A random Euclidean network with $N = 60$ will therefore provide 1770 independent constraints, far more than required to specify the network in three dimensions.

The above discussion indicates that there are more than enough constraints in complex Euclidean networks to specify the network structure. As described in the next section, these rigidity concepts may be used to develop an efficient reconstruction algorithm. However, it is important to keep in mind that Laman's theorem only applies to planar structures.

The theoretical foundation of efficient algorithms for the UPD problem rests on rigidity theory discussed above that states that an isostatic (i.e., minimally rigid) structure in two dimensions [from Eq. (2)] has $B_c = 2N - 3$ independent distance constraints. However, the key test of whether the assignment of distances to natural lengths is correct is to place at least one additional, overconstrained Euclidean distance into the structure. A distance incompatible with the isostatic structure leads to a finite strain energy cost in Eq. (1), due to stretched or compressed springs, while a distance compatible with the isostatic structure has zero energy cost. Note that many isostatic structures that are *inconsistent* with the final structure can be made, but with high probability no overconstrained zero cost structures can be made that are inconsistent with the final reconstruction.

III. TRIBOND ALGORITHM

In two dimensions the smallest structure with at least one overconstrained bond is $N = 4$, where the total number of bonds is $\binom{4}{2} = 4 \times 3/2 = 6$, while the number required for isostaticity is [from Eq. (2)] $2N - 3 = 5$. The key observation is that if six unique Euclidean distances taken from a target structure form a four-point structure, and the cost function for this structure and distances is zero, then it is almost certainly a unique and correct substructure of the target. We call a zero cost correct substructure with six distances and four sites a *core*. If the distance list is nondegenerate, then with high probability, this core is a correct substructure of the target structure. We may then build up from the core iteratively to find the complete structure. At each step we have an existing, correct substructure. We then add one site and search for three edges that are compatible with the new node and with three nodes that are in the existing structure. The addition of one site and two edges is an isostatic addition, while the addition of one site and three edges is overconstrained. If we find three edges compatible with one additional site then, with high probability, this site is part of the target structure.

In practice, to construct a core (Fig. 2) we choose the smallest bond as the "base bond." We then test all the bond combinations using the triangle inequality to generate feasible triangle pairs. This is performed in two steps: first we fix a triangle as the "base triangle" and then search through all other candidate ("top") triangles that share the same base bond. After all the top triangles have been exhausted a new base triangle is selected and the process continues. For every triangle pair we calculate the length of the bond that connects the two apex points, which we call the bridge bond. The length of the bridge in the candidate core is tested against the lengths in the distance list. If the candidate bridge length matches an unused distance in the distance list, we have found a core.

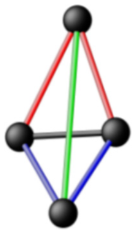

FIG. 2. (Color online) An example of a core. In two dimensions, it consists of four points. The horizontal bond is the base (in black), the bonds below it (in blue) make up the base triangle while those above it (in red) make up the top triangle. The vertical bond is the bridge (in green).

In the buildup procedure, we try to add more sites to the core. The addition of a site consists of generating candidate top triangles using the base bond and two distances from the distance list. After we place this site, we carry out bridge testing to determine whether the structure has zero strain energy. While core finding requires a search over all possible base and top triangles, buildup requires only a search through top triangles as the base triangle is a known part of the structure. Consequently, buildup requires significantly fewer computations than core finding.

Our Tribond implementation of the above procedure for the unassigned PD-IP algorithm may be summarized as follows:

We are given the sorted distance list $\{d_l\}$ with the number of nodes in the network N. (The target network is generated by randomly placing N points in a square box with sides of length N.) We start with an empty set, then

(A) Core finding procedure

(1) Choose the shortest bond as the base bond and a window (subset) of $W = 6$ smallest entries in the distance list for the core finding search.

(2) Iterate over all triangles constructed with the triangle inequality that have the same base bond using distances in the window W.

(3) Search over all pairs of feasible triangles generated above and calculate the bridge bond. Using a binary search, test if there is an unused distance that matches the bridge bond. If such a value is found, we have a core. Remove the edges used from the distance list and exit to the buildup procedure.

(4) Increment W by 6 and return to (1), making sure not to retest bond combinations.

(B) Buildup procedure

(1) Search over all sets of two edges from the distance list to find a set compatible with the base triangle in the existing structure. Search over the distance list to test the bridge bond.

(2) If successful, remove from the distance list the edges that are used in connecting the newly added node. If size of reconstructed structure is $<N$, return to previous step and resume the search.

In these procedures the choice of tolerance is important to ensure efficient reconstruction with few restarts, particularly for large structures. The results presented below are for optimal tolerance settings, so that even for large structures restarts are required only 10% of the time.

A coarse upper bound on the computational time for this procedure consists of two parts: (i) the time to find the core; (ii) the time to carry out the buildup procedure. The number of unique cores in the point set is $\binom{N}{4}$, the number of ways of choosing four sites from N total sites. The number of ways of choosing six distances from the set of $M = N(N-1)/2$ distances is $\binom{M}{6}$. If we had done a brute force search then we would find a core in computational time $\tau_{\text{core}} \sim \binom{M}{6}/\binom{N}{4} \sim N^8$. Similarly, using brute force for the buildup would take a computational time that scales as $\tau_{\text{buildup}} \sim \binom{M}{3} \sim N^6$. This clearly shows that the brute force approach is polynomial, although a high order one.

The simple methods we have developed reduce the computational time significantly from the coarse upper bounds of

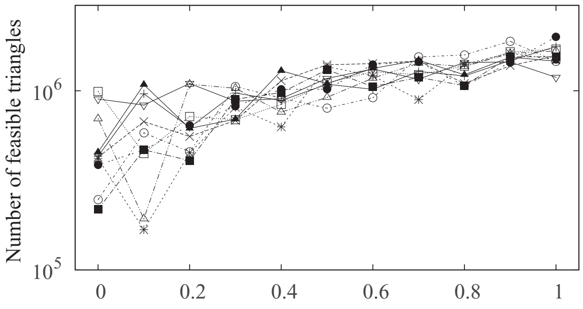

FIG. 3. Number of feasible triangles using the bonds from a given distance list go up when we choose a larger bond as base for the triangle. Statistically, using the shortest bond in the distance list as the base leads us to the core in the shortest time. This plot shows data from runs using 10 different structures with $N = 128$.

the last paragraph. The key observation is that many of the distances in the distance list violate the triangle inequality $d_1 + d_2 \geqslant d_3$. A large fraction of the computational time in a brute force search is spent exploring these trivially inconsistent distance combinations. If we fix the base bond, the bridge bond is found using binary search, using simple combinatorial arguments, $\tau_{\text{core}} \sim \binom{M}{4}\ln(N)/\binom{N}{2} \sim N^6\ln(N)$. For a triangle with base bond a and second side b, the range of values for third side c is $(b - a, b + a)$. So a larger base bond requires a much larger range of feasible values for the third side and, hence, the number of feasible triangles increases. But the actual number of triangles in the target structure is the same for any choice of base bond. This is seen in Fig. 3, where the number of feasible triangles increases with fractional position of the base bond in the distance list (for a list of distances l_i that are ordered smallest to largest, with a total number of distances M, the fractional position is i/M). Hence, statistically, we find a core in the least time if we choose the shortest bond as our base.

Distances are also more likely to satisfy the triangle inequality if they are drawn from a list of comparable, rather than disparate, lengths. Since the base bond is short, a core is more likely to be found quickly by searching over other short distances first (the small-core hypothesis, Fig. 4), and including longer distances only as necessary. This is implemented as a window of the W shortest distances in the distance list, which increases periodically as core finding proceeds. Of the six bonds in the core, the base is fixed, four are drawn from the window, and the bridge bond may appear anywhere in the distance list. We observe that a window of size $W \sim N$ is usually sufficient to find a core. Therefore, typical computation time is $\tau_{\text{core}} \sim \binom{N}{4}\ln(N) \sim N^4\ln(N)$.

From these arguments, supported by Figs. 3 and 4, we expect that using the smallest bond as the base will lead to the core finding and buildup in a much shorter time. Figure 5 shows that the improvement is about three orders of magnitude.

Attempting to find the core for large point sets ($N > 200$) frequently leads to bad cores. Bad cores are overconstrained substructures whose distances are part of the given distance list within a given numerical tolerance, but the substructure is not present in the target structure. This occurs due to finite tolerance when checking for the bridge bond and also finite precision while placing the points using triangulation. Triangles with both small and large distances are likely to have small angles, resulting in a greater loss of numerical precision. A base bond of intermediate length would limit this loss, but is not sufficient to forsake the performance benefits of a small base bond previously outlined. Instead, we try to use all six bonds (in the core) as the base bond and check if the corresponding bridge bond is valid or not. We only take cores for which the bridge bond is valid in all of the six cases. This check is very good at identifying bad cores.

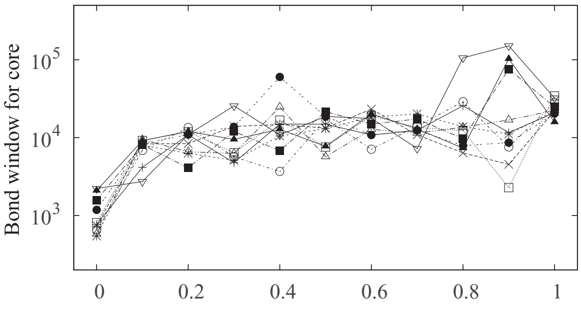

FIG. 4. Empirical example of the small-core hypothesis. The hypothesis states that there exists a core where at least five of the six total bonds are drawn from a relatively small window of the shortest bonds in the structure. Varying the base bond's fractional position in the distance list for ten different $N = 1024$ structures, core finding shows that using the smallest distance as the base bond reduces the typical size of the window required to find a core by an order of magnitude.

A structure comparison routine provides another test for bad cores by overlaying the points in the reconstructed structure ($\vec{r}$) with the points in the target structure ($\vec{R}$) and calculates an overlay error,

$$\epsilon_{\text{overlay}} = \sum_i |\vec{r}_i - \vec{R}_i|^2. \tag{3}$$

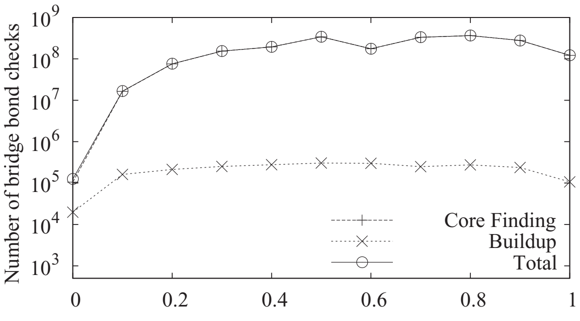

FIG. 5. Figure illustrating the effect of base bond size on the computational cost (bridge bond checks) of reconstruction for $N = 32$. The plots for the total and core finding steps are nearly indistinguishable because the core finding is orders of magnitude more expensive than buildup. If the smallest bond is chosen as the base, the total computational cost of reconstruction is nearly three orders of magnitude lower than when a longer base bond is utilized.

This error tells us if a given (sub)structure is part of the target structure. This proves useful for testing purposes only, as in principle the latter remains unknown. It is also used to verify the correctness of the final structure.

If the buildup step fails to add any points after looping over a certain number of bonds from the distance list, likely due to a bad core, then we discard the substructure. We resume the core finding step and attempt another buildup from a new core. This heuristic helps identify probable bad cores efficiently.

It is important to choose an appropriate tolerance when checking if the bridge bond is part of the distance list. Using a very loose tolerance leads to a large number of bad cores. On the other hand, using a very tight tolerance excludes good cores, due to finite precision when carrying out the triangulation to place the points in our substructure. All calculations were performed using 80-bit x86 extended precision floating point format (GNU GCC v.4.4.3 on an Intel Core 2 Duo processor), which provides about 18 digits of precision. We found that a relative tolerance of 10^{-12} is optimal to retain good cores and filter out bad ones. When trying to place points which are nearly collinear to the base bond a loss of precision is observed due to small angles, as discussed earlier. In such situations we relax the tolerance when checking for the bridge bond.

To check the validity of a new point while doing buildup, in addition to the bridge bond check, we check the ten largest distances that it creates with the points already in the substructure. Only if these are part of the distance list does the new point get added to the structure. The three bond lengths (two from the new triangle created and the third is the bridge) that were used are removed from further reconstruction. This reduces the list of available distances by 3. After placing the nth point, updating all $(n-1)$ distances created between the new point and the points already in the substructure reduces the number of available distances substantially. However, due to the computational cost of this update procedure, we see only a small speedup in the buildup routine.

If, after buildup, the structure has fewer than the desired number of points (N), relax the tolerance for the bridge bond checks by a few orders of magnitude. If the structure remains incomplete, choose a different bond as the base bond and buildup from the start. After reconstruction, we calculate the distance error, which is based on the agreement between the given distance list and the distances derived from the final structure.

The Tribond algorithm ran for $N = 8,16,\ldots,512$ and the computational cost was measured in a system-independent manner by counting the number of bridge bond checks while placing a point in both the core finding and buildup steps (Fig. 6). The time required for buildup is about an order of magnitude less than that for core finding. The scaling is $\tau_{\text{total}} \sim N^{3.32}$. This is better than the estimate obtained earlier using simple combinatorial arguments, which did not account for the speedup gained by exploiting the triangle inequality.

IV. APPLICATIONS

In the previous section we showed that Tribond is able to reconstruct random point sets, but it is not limited to such cases. Tribond is expected to solve any structure with all distinct distances. For example, an adsorbed polymer with varying nearest-neighbor distance modeled as a two-dimensional (2D) self-avoiding walk in the continuum will, with high probability, have a distance list with unique entries (Fig. 7).

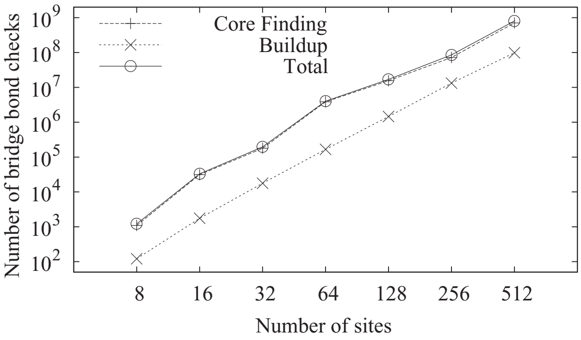

FIG. 6. Experimental results for a series of reconstructions from distance lists generated from random point sets in two dimensions. The computational cost (bridge bond checks) for finding the core, performing buildup, and their total is presented as a function of the number of points. The plots for the total and core finding steps are nearly indistinguishable because core finding takes orders of magnitude more time than buildup. Each point on the plots is an average over 25 different instances of random point sets. We find that the total time scales as $\tau_{\text{total}} \sim N^{3.32}$.

Structures occurring in nature often have symmetric or otherwise ordered features, with small deviations from ideal behavior due to, for example, finite size effects. Tribond is able to solve sufficiently perturbed lattice structures, as shown in Fig. 8. These perturbations create a distance list which has unique entries with respect to the algorithm's numerical precision.

A. Tribond for structures with high symmetry

The idea behind Tribond was refined to handle some structures with high symmetry (as indicated by a degenerate distance list). To handle degeneracy we restrict each step of core finding and buildup to a subset of the distance list consisting only of unique distances. Distance multiplicities are updated, however, in the buildup step. This cuts down on the number of bad cores and bad points (low cost, but wrong). Using this modified approach we attempted to solve square grids with up to $N = 1024$ (32×32) points. Nearly all the

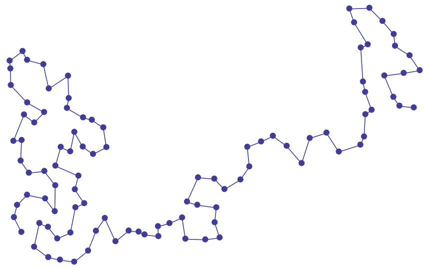

FIG. 7. (Color online) A self-avoiding walk is a sequence of moves that does not visit the same point more than once and is used to model polymers. Tribond was able to successfully reconstruct the above structure ($N = 100$) in a few minutes.

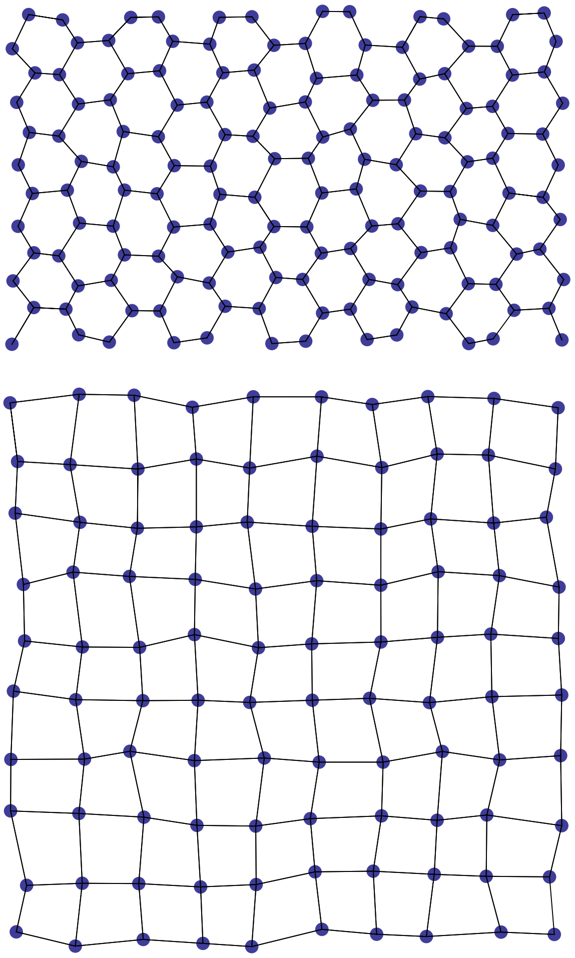

FIG. 8. (Color online) Lattice structures perturbed to resemble natural imperfections. Tribond reconstructs both in a few minutes. On the top is a hexagonal grid (graphene nanoparticle [40]) with 144 atoms and the bottom is a square grid with 100 points.

reconstructions were successful, with runs taking no more than 10 min on a desktop computer. In high symmetry cases it is possible for buildup to fail by finding a wrong substructure consistent with the distance list, preventing further addition of points. This problem could not be overcome for $N = 400$, 676, and 900, where reconstruction failed.

B. Structure buildup from known core with an imprecise distance list

Thus far distances have been known to a precision of about 18 digits, such that in our trials substructures are indistinguishable (to within a very small tolerance) to those consistent with a theoretical distance list of infinite precision. An imprecise distance list may be compatible with many substructures not part of the target structure, or may lead to high cost for any reasonable structure.

The inverse problem under these conditions is significantly more challenging, both theoretically and practically. We have attempted to address structure buildup from a known substructure with an imprecise distance list in the case of random point sets. The modification of the original buildup algorithm described in Sec. III is as follows.

Assume a known initial substructure (not necessarily a core) that serves as the starting point for reconstruction. The modified buildup step (adding a point) now has multiple stages; it uses a pool of candidate points which have low error with respect to the current substructure, and adds the two candidates which jointly lead to the lowest cost substructure. Because the pool examines many possible ways to grow the substructure, the likelihood of adding bad points is reduced. Adding two points at once is justified empirically, as this appeared to make the most acceptable tradeoff between success and run time. The detailed steps follow.

(1) Define an empty pool that saves the coordinates of $k_1 \leq 20$ candidate points to add to the current substructure. Associated with each candidate is the cost of the new substructure if that point were added. Populate the pool with candidate points. First, randomly choose a bond in the current substructure. Generate all triangles using two distances from the distance list which share the chosen bond. Calculate the cost for each candidate point (the new vertices). If this cost is below a user-defined threshold add it to the pool, and if the pool exceeds its maximum size remove the worst candidate. The threshold significantly improves run time without affecting the final structure.

(2) Randomly choose another bond in the current substructure and generate a new pool of size $k_2 \leq 20$ as described above.

(3) Select the best candidates from either pool to make a combined pool with $k \leq 20$ points.

(4) Calculate the pair cost for adding two candidates to the current substructure for each of the $\binom{k}{2}$ possible pairs.

(5) Add the two candidates with least pair cost to the substructure. If its size is less than target size then go to step 1.

Our results can be seen in Fig. 9, which shows the minimum initial substructure size needed to reconstruct structures of size $N = 26, 50, 76, 100$ for different values of the precision (P) of the input distance list. The units for the precision of the distances is the number of digits. Our criterion for

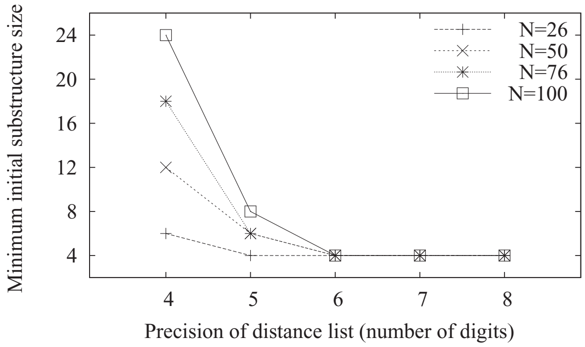

FIG. 9. Plot of minimum initial substructure size vs precision of the input distance list for $N = 26$, 50, 76, and 100. A larger initial substructure is needed for less precise distance lists. The typical run time for $N = 26, 50, 76, 100$ was about 1 min, 10 min, 4 h, and 20 h respectively, on a computer with a 2.2-GHz processor and 2 GB of memory.

success was that the algorithm successfully reconstructs at least five of ten different random point sets. We can see that as the distances become less precise, an initial substructure of larger size is needed for successful reconstruction. When the input data have a higher precision ($P \geq 6$), we found that an initial substructure size of 4 (i.e., a core) was sufficient for reconstruction.

A notable case with imprecise distances is the PDF of nanostructured materials, which can provide distance lists with uncertainties of order 0.01 Å. For a typical nanoparticle of size ~15 Å, this means the input distances from experimental data will have three to four digits of precision and our algorithm is a promising approach.

Experimental methods providing pair distance information of comparable precision and sufficient quantity are rare. Distances derived from the PDF may have uncertainties as low as order 0.01 Å, and in some nanostructured materials a sizable fraction of all pair distances can be resolved to this level [41]. Under these circumstances the input distances from the experimental data will have three to four digits of precision, and our algorithm is a promising approach. We are working on an algorithm that can handle incorrect or missing distances, in addition to imprecise distances. Chemical information like the presence of functional groups (aromatic rings, etc.) can serve as a core and help find the larger initial substructures necessary for buildup in the case of less precise distances. Some approaches to these issues are discussed in the context of reconstructing high symmetry nanostructures from experimental PDF data using the Liga algorithm [32–34]. A hybrid approach using Tribond (low symmetry) and Liga (high symmetry) could potentially solve structures of intermediate symmetry.

V. CONCLUSIONS

The problem of reconstructing complex Euclidean networks given only their unassigned Euclidean distances has unique theoretical and algorithmic challenges, reflecting the combinatorial explosion of possible assignments. We have concentrated on finding structure from precise distances. The Tribond algorithm consists of two steps: core finding and buildup. The core is the smallest substructure with at least one overconstrained bond. Choosing the smallest bond as the base bond for reconstruction had a dramatic improvement in performance. Computational cost of core finding was orders of magnitude more than buildup. Tribond was able to reconstruct random point sets in two dimensions of size ~1000.

A modified approach was presented for the buildup step with less precise data and given a known substructure. As precision decreases the minimum size of the starting substructure must increase, underscoring the importance of techniques, including core finding, which may help find good seeds for reconstruction. We successfully reconstructed random point sets of size 100, with the distances having four digits of precision, given a known substructure of size 24. Solving the unassigned distance problem arising from experimental data requires further effort. These must include methods for handling missing or incorrect distances and finding good substructures to seed the buildup step. A hybrid approach using Liga and Tribond could help overcome these issues. We are currently working on extending the Tribond algorithm to three dimensions, which will help us attack additional unassigned pair distance scenarios, including the general nanostructure problem.

ACKNOWLEDGMENTS

We gratefully acknowledge our collaborators at Columbia University, Professor Simon J. Billinge and Dr. Pavol Juhas, for useful discussions. This work was supported by the Center of Research Excellence in Complex Materials (CORE-CM) at Michigan State University. We also wish to acknowledge the support of the Michigan State University High Performance Computing Center and the Institute for Cyber-Enabled Research.

[1] G. M. Crippen and T. F. Havel, *Distance Geometry and Molecular Conformation* (Wiley and Sons, New York, 1988).

[2] G. Crippen, J. Math. Chem. **6**, 307 (1991).

[3] S. B. Lindström, D. A. Vader, A. Kulachenko, and D. A. Weitz, Phys. Rev. E **82**, 051905 (2010).

[4] K. Wuthrich, Acc. Chem. Res. **22**, 36 (1989).

[5] K. Wuthrich, Science **243**, 45 (1989).

[6] L. Liberti, C. Lavor, A. Mucherino, and N. Maculan, Int. Trans. Operation. Res. **18**, 33 (2011).

[7] M. Li, Y. Otachi, and T. Tokuyama, in *7th International Symposium on Algorithms for Sensor Systems, Wireless AD HOC Networks and Autonomous Mobile Entities, ALGOSENSORS 2011*, Lecture Notes in Computer Science 7111 (Springer-Verlag, Berlin, 2012), pp. 101–114.

[8] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino, SIAM Rev. **56**, 3 (2014).

[9] R. L. McGreevy and L. Pusztai, Mol. Simul. **1**, 359 (1988).

[10] A. Patterson, Nature (London) **143**, 939 (1939).

[11] J. Yoon, Y. Gad, and Z. Wu, technical report, 2000.

[12] J. C. Kendrew, R. E. Dickerson, B. E. Strandberg, R. G. Hart, D. R. Davies, D. C. Phillips, and V. C. Shore, Nature (London) **185**, 422 (1960).

[13] M. F. Perutz, M. Rossmann, A. Cullis, H. Muirhead, G. Will, and A. C. T. North, Nature (London) **185**, 416 (1960).

[14] J. Miao, H. N. Chapman, J. Kirz, D. Sayre, and K. O. Hodgson, Annu. Rev. Biophys. Biomol. Struct. **33**, 157 (2004).

[15] J. Miao, J. Kirz, and D. Sayre, Acta Crystallogr. Sect. D **56**, 1312 (2000).

[16] J. Wu, K. Leinenweber, J. C. H. Spence, and M. O'Keeffe, Nat. Mater. **5**, 647 (2006).

[17] V. L. Shneerson, A. Ourmazd, and D. K. Saldin, Acta Crystallog. Sect. A **64**, 303 (2008).

[18] Y. Jiao, F. H. Stillinger, and S. Torquato, Phys. Rev. E **81**, 011105 (2010).

[19] Y. Jiao, F. Stillinger, and S. Torquato, Phys. Rev. E **82**, 011106 (2010).

[20] D. Cule and S. Torquato, J. Appl. Phys. **86**, 3428 (1999).

[21] M. Nilges and S. I. O'Donoghue, Prog. Nucl. Magn. Reson. Spectrosc. **32**, 107 (1998).

[22] B. Hendrickson, SIAM J. Optim. **5**, 835 (1995).

[23] B. Berger, J. Kleinberg, and T. Leighton,' JACM **46**, 449 (1996).

[24] J. Moré and Z. Wu, technical report, 1995.

[25] J. Saxe, in *Proceedings of the 17th Allerton Conference on Communications, Control and Computing*, edited by J. B. Cruz and F. P. Preparata (University of Illinois Urbana-Champagne, Urbana, IL, 1979), pp. 480–489.

[26] H. Lin, E. S. Božin, S. J. L. Billinge, E. Quarez, and M. G. Kanatzidis, Phys. Rev. B **72**, 174113 (2005).

[27] L. Malavasi, G. A. Artioli, H. Kim, B. Maroni, B. Joseph, Y. Ren, T. Proffen, and S. J. L. Billinge, J. Phys.: Condens. Matter **23**, 272201 (2011).

[28] T. Proffen and S. Billinge, Appl. Phys. A **74**, s1770 (2002).

[29] S. J. Billinge, J. Solid State Chem. **181**, 1695 (2008).

[30] S. J. L. Billinge and M. G. Kanatzidis, Chem. Commun. (Cambridge, UK) **7**, 749 (2004).

[31] S. J. L. Billinge and I. Levin, Science **316**, 561 (2007).

[32] P. Juhás, D. M. Cherba, P. M. Duxbury, W. F. Punch, and S. J. L. Billinge, Nature (London) **440**, 655 (2006).

[33] P. Juhás, L. Granlund, P. M. Duxbury, W. F. Punch, and S. J. L. Billinge, Acta Crystallogr. Sect. A **64**, 631 (2008).

[34] P. Juhas, L. Granlund, S. R. Gujarathi, P. M. Duxbury, and S. J. L. Billinge, J. Appl. Crystallogr. **43**, 623 (2010).

[35] B. Roth, Am. Math. Mon. **88**, 6 (1981).

[36] H. Crapo, Struct. Topol. **1**, 26 (1979).

[37] L. Asimow and B. Roth, J. Math. Anal. Appl. **1**, 2572 (2010).

[38] L. Asimow and B. Roth, Trans. Am. Math. Soc. **245**, 279 (1978).

[39] G. Laman, J. Eng. Math. **4**, 331 (1970).

[40] L.-s. Li and X. Yan, J. Phys. Chem. Lett. **1**, 1595 (2010).

[41] T. Egami and S. J. L. Billinge, *Underneath the Bragg Peaks: Structural Analysis of Complex Materials* (Elsevier, Oxford, 2003).