Discrete Applied Mathematics, 354:83-93, 2024

An impossible combinatorial counting method in distance geometry

Germano Abud[a], Jorge Alencar[b], Carlile Lavor[c], Leo Liberti[d], Antonio Mucherino[e]

[a]*FAMAT, Federal University of Uberlândia, Minas Gerais, Brazil*
[b]*IFTM, Federal Institute of Triângulo Mineiro, Brazil*
[c]*IMECC, University of Campinas, Brazil*
[d]*LIX CNRS, École Polytechnique, Institut Polytechnique de Paris, F-91128 Palaiseau, France*
[e]*IRISA and University of Rennes I, Rennes, France*

Abstract

The Distance Geometry Problem asks for a geometric representation of a given weighted graph in $\mathbb{R}^K$ so that vertices are points and edges are segments with lengths equal to the corresponding weights. Two problem variants are defined by a vertex order given as part of the input, which allows the use of a branching algorithm based on K-lateration: find two possible positions for the next vertex j using the positions of K predecessors and their distances to j, then explore each position recursively, pruning positions at need. Whereas the first variant only requires the K predecessors to exist, the second variant also requires them to be contiguous and immediately preceding j. For this variant, fixed-parameter tractability of the algorithm can be established by means of a solution counting method that only depends on the graph edges (rather than their weights). Only in the first variant, however, it is possible to efficiently construct a suitable vertex order directly from the graph. Since both fixed-parameter tractability and efficient vertex order construction are desirable properties, one would need an analogous solution counting method for the first variant. In this paper we prove that such a counting method cannot be devised for the first variant.

Keywords: DGP, DMDGP, DDGP, Branch-and-Prune, solution symmetry.

1. Introduction

The Distance Geometry Problem (DGP) [33, 38] is as follows. Given an integer $K > 0$ and a simple undirected edge-weighted graph $G = (V, E, d)$ with $|V| = n$ and $d : E \to \mathbb{R}_+$, find a set of points $\{x_1, \ldots, x_n\} \subset \mathbb{R}^K$ that *realizes* the

Email addresses: germano.abud@ufu.br (Germano Abud), jorgealencar@iftm.edu.br (Jorge Alencar), clavor@unicamp.br (Carlile Lavor), liberti@lix.polytechnique.fr (Leo Liberti), antonio.mucherino@irisa.fr (Antonio Mucherino)

graph so that, for each $\{i, j\} \in E$, we have $\|x_i - x_j\|_2 = d_{ij}$. This problem is used to model the retrieval of positions (the x_i's) from some given distances (the d_{ij}'s) in various application settings, e.g. synchronization in network protocols [40], restriction site mapping in molecules [41], sensor network localization [3, 6, 22], protein conformation from distance data [17, 18, 33], geometry of nanostructures [5], localization of underwater vehicles [4], natural language processing [21, 28], machine learning and data science [29].

In this paper we consider two DGP variants where a certain order on the vertices is given as part of the input. This order is used to construct realizations in a build-up fashion [13]. Both variants have desirable algorithmic properties, so a reconciliation of these properties into a single DGP variant would represent an important progress. In this paper we prove that the best strategy we currently have to achieve this reconciliation cannot work. The rest of this section explains the goal of this paper in more detail.

1.1. The two variants

In the first variant, the order specifies that, for any graph vertex j beyond the K-th one, there are at least K vertices $i < j$ (i.e., preceding j in the order) such that $\{i, j\} \in E$. In other words, distances to j are known from a set U_j of at least K vertices $\{i_1, \ldots, i_K\}$ preceding j. This variant is known as Discretizable DGP (DDGP) [37]. We note that, in general, many vertex orders may satisfy this requirement.

In the second variant, the order specifies that the vertices in U_j precede j *immediately* and *contiguously*, i.e. $i_1 = j - K$, $i_2 = j - K + 1$, $\ldots$, $i_K = j - 1$. It is very easy to show that, under this condition, every U_j induces a clique in G (see the beginning of Sect. 2.1 about induced subgraphs). Again, many orders may satisfy this requirement. This variant is a subclass of DDGP instances known as Discretizable Molecular DGP (DMDGP) [27]. The reference to molecules in the name arises from the application of the DGP to protein conformation from Nuclear Magnetic Resonance (NMR) experiments [43], where, in a first approximation, the order can follow the amino acid sequence defined by the protein backbone [26].

1.2. Branching and pruning

Both variants can be solved by an algorithm known as Branch-and-Prune (BP) [32]. This algorithm exploits the vertex order to find (almost certainly) at most two possible realizations for vertex j by means of the at least K predecessors of j, and then it branches on each of the two positions. The realization of vertex j can be obtained from the realizations x_i of its predecessors $i \in U_j$ by an algorithm known as K-*lateration* [38]. It consists in solving a linear system of $K - 1$ equations in K unknowns plus a single quadratic equation, which is why it yields at most two points x_j^+, x_j^- almost certainly.

If there are more predecessors of j aside from those in U_j, the BP checks the feasibility of x_j^+, x_j^- w.r.t. the distances from these other predecessors, pruning either or both of the points at need. The BP algorithm has worst-case exponential time complexity.

The BP algorithm can be stopped as soon as the first realization is found, or be allowed to continue until it exhausts all non-pruned branchings. In the second case, it finds all incongruent realizations (up to an initial reflection) of the given instance.

1.3. Fixed-parameter tractability of the DMDGP

A Fixed-Parameter Tractable (FPT) algorithm has worst-case complexity $O(2^{\kappa} p(\iota))$ where $p(\iota)$ is a polynomial in the input size ι, and κ is a part of the input that is usually small or can be fixed in practice. An FPT problem is one for which an FPT algorithm exists. See [14] for more details.

On DMDGP instances, the BP algorithm turns out to be FPT (up to the usual approximations Turing Machines are subjected to when dealing with algorithms involving real numbers [20]) w.r.t. K and another parameter v_0 indicating the vertex at which the BP stops branching "too frequently" [34], which can be considered fixed for most proteins (but not all). So the DMDGP is FPT and justifiably very fast on most protein instances, whereas an equivalent theory is not yet known for the DDGP.

The proof showing that the DMDGP is FPT involves the theory of partial reflection symmetries of the DMDGP [35], which allows one to count the number of BP branchings in function of K and v_0 (and hence the number of partial solutions up to a certain vertex j) in a way that only depends on the edges E of G, but not on their weights d. Note that the number of branchings up to order rank v is the same as the number of partial realizations up to v, which is the same as the number of realizations of the DMDGP instance defined on the subgraph of G induced by $\{1, \ldots, v\} \subset V$.

We label E as a "combinatorial" data type, and d a "numeric" data type, and define a method to count realizations *combinatorially* as long as it only uses combinatorial data types. We note that the combinatorial counting method in [35] counts DMDGP realizations correctly only almost certainly.

This use of the term "combinatorial" comes from the graph rigidity literature (a prominent part of distance geometry), the fundamental problem of which [16, Ch. 5] is to find a *combinatorial* method to determine the rigidity of graphs in $K > 2$ dimensions [24], i.e. a method that only depends on the graph topology, not on the edge weights or the geometric realization of the graph.

We remark that the FPT-ness of the DMDGP is clearly a desirable feature.

1.4. Finding the vertex orders from the graphs

Suitable vertex orders for the DDGP and the DMDGP can be found algorithmically from the input of the DGP. So, while the two variants are not special cases of the DGP because their definition requires the vertex order as an input, it makes sense to speak of the "recognition problems" of the two variants: when is a given DGP instance a DDGP or a DMDGP one?

In the protein application, constructing a suitable vertex order from the graph is important because NMR experiments do not always provide enough inter-atomic distances to warrant using the natural order of the atoms in proteins.

It turns out that finding a suitable DDGP order is FPT w.r.t. K [25, §3.2], which is fixed for most DGP applications [29, §3.3]. Hence, in practice, DDGP orders can be found in polytime. Finding a suitable DMDGP order, however, is **NP**-complete for any fixed K [9].

We remark that the FPT-ness of the DDGP recognition problem is clearly a desirable feature.

1.5. Our contribution

From the foregoing discussion, it is important to have both of the desirable FPT-related features of the two DGP variants at the same time.

The **NP**-hardness of the DMDGP recognition problem for fixed K makes it impossible to attempt to find a corresponding FPT recognition algorithm w.r.t. K, unless $\mathbf{P} = \mathbf{NP}$. We can therefore only hope to be able to construct an FPT solution algorithm for the DDGP. The first and most natural attempt towards this goal is to extend the FPT property of the DMDGP solution algorithm to the DDGP. For this purpose, we need a combinatorial solution counting method for the DDGP, because it is precisely the feature that makes the FPT analysis of the BP possible for DMDGP instances.

We finally come to state the objective of this paper more precisely. We show that the number of solutions of a DDGP instance may depend on the edge weights with some positive probability. This negates the existence of a combinatorial method for counting DDGP solutions that is similar to the one we have for the DMDGP. In turn, this implies that extending the FPT analysis we currently have for the DMDGP to the DDGP is doomed to failure. See Sect. 3.1.1 for more details.

As for the rest of this paper, preliminary notions are given in Sect. 2. In Sect. 3 we prove that we cannot easily extend our FPT analysis from the DMDGP to the DDGP; we also present a simple subclass of the DDGP where combinatorial counting *is* possible, to show that our impossibility result does not cover the whole problem.

2. Notation and preliminary notions

In the following, we use the formal language of first order logic in the framework of the ZFC axiom system [23], including the usual symbols for logical connectives (e.g. $\wedge$, $\vee$, $\neg$) and quantifiers (e.g. $\forall$, $\exists$). In particular, we use "$<$" to denote orders on sets of vertices, and also on sets of integers. Thus, we may have quantifications such as "$\forall k \le K, i \le K, i < j$", where k, K are integers and i, j are vertices of a graph. The apparent ambiguity is dispelled by the fact that there is always a natural bijection between countable ordered vertex sets and sets of integers having the same cardinality. In particular, the quantification above states "for each integer k less than or equal to K, each i-th vertex in the vertex set with ordinal label i less than or equal to K, and each vertex pair having ordinal labels i, j such that i is strictly less than j". We use the standard notation with angular brackets for the inner product of two vectors.

2.1 Definition
Given an integer K and a simple undirected graph $G = (V, E)$, an embedding of G in $\mathbb{R}^K$ is a function $x : V \to \mathbb{R}^K$. Given an edge weight function $d : E \to \mathbb{R}_+$, a realization of G in $\mathbb{R}^K$ is an embedding that also satisfies

$$\forall \{i, j\} \in E \quad \|x_i - x_j\|_2^2 = d_{ij}^2, \tag{1}$$

where $x_i = x(i)$ for all $i \in V$, and $d_{ij} = d(\{i, j\})$ for all $\{i, j\} \in E$. ■

As mentioned above, the DDGP arises in applications such as the determination of protein structure [38], as well as in the study of rigid graphs constructed by "Henneberg type 1 moves" [19, 42]. Its formal definition is as follows:

> Given an integer $K > 0$, a simple undirected graph $G = (V, E)$ with an edge weight function $d : E \to \mathbb{R}_+$, and a vertex order $<$ on $V = \{1, \ldots, n\}$ such that:
>
> (i) $G[U_0]$ (the subgraph of G induced by U_0) is a clique of size K, where $U_0 = \{1, \ldots, K\}$
>
> (ii) $\forall j \in \{K + 1, \ldots, n\}\ \exists U_j \subseteq \{1, \ldots, j - 1\}$ with
> $|U_j| = K \wedge \forall i \in U_j\ \{i, j\} \in E$,
>
> determine if there is a realization $x : V \to \mathbb{R}^K$ satisfying Eq. (1).

Remarks

1. The size of the clique U_0 is always equal to the dimension: this is the reason why we use the same symbol K for both.
2. As already mentioned, the DMDGP differs from the DDGP because U_j consists of immediate and contiguous predecessors of j.
3. For any $j \in V$, we let $\ell(j) = \max_< U_j$ and $\overline{U}_j = U_j \cup \{j\}$. In other words, $\ell(j)$ is the last vertex in the set U_j according to the vertex order.
4. We partition the edge set E into the *discretization edges* $E_D = \{\{i, j\} \in E \mid i \in U_j\}$ and *pruning edges* $E_P = E \smallsetminus E_D$.

2.1. Euclidean distance and Gram matrices

A K-dimensional realization x of a graph on n vertices can be represented as an $n \times K$ matrix: the i-th row is the position vector $x_i \in \mathbb{R}^K$ of the i-th vertex. We shall make use of this representation in some of the proofs below.

The $n \times n$ symmetric zero-diagonal matrix D having $\|x_i - x_j\|_2^2$ as its (i, j)-th entry is a *squared Euclidean Distance Matrix* (EDM). By Schoenberg's theorem [39], the matrix $\Gamma = -\frac{1}{2} JDJ$ (where $J = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ is the *centering matrix* and $\mathbf{1}$ is the all-one vector) is the *Gram matrix* of the realization x, i.e. $xx^\top = \Gamma$ [11]. Using spectral decomposition and the positive semidefiniteness of Gram matrices, one can derive x from Γ in polynomial time [10]. Moreover, $D =$

$\mathsf{diag}(\Gamma)\mathbf{1}^\top - 2\Gamma + \mathbf{1}\mathsf{diag}(\Gamma)^\top$, which implies that $\mathsf{rank}(D) \le \mathsf{rank}(\Gamma) + 2$ [12]; since $\mathsf{rank}(\Gamma) = \mathsf{rank}(x) \le K$, we also have $\mathsf{rank}(D) \le K + 2$.

Notation-wise, for a realization x of $G = (V, E, d)$ and a subset $U \subseteq V$ we define the *restriction* of x to U by $x[U] = (x_i \mid i \in U)$, which turns out to be a realization of the *induced subgraph* $G[U] = (U, \{i \in U, j \in U \mid \{i, j\} \in E)$.

Given a realization x of G, we can compute the corresponding EDM by evaluating all Euclidean distances between all pairs x_i, x_j. Conversely, given an EDM D, we can compute a realization (in some dimension $K \in \{1, \ldots, n\}$) by obtaining the Gram matrix Γ in function of D as explained above, and then factoring G using the spectral decomposition $\Gamma = P^\top \Lambda P$, where P is a matrix of eigenvectors and Λ a diagonal matrix of corresponding eigenvalues. Then $x = P^\top \sqrt{\Lambda}$ is a realization of D.

2.2. More on the BP algorithm

The BP algorithm [32] works by finding a position for a vertex $j > K$ in V by using K-lateration on a set U_j of K adjacent predecessors of j. This assumes that the set U_0 of the first K vertices in the order already has a realization. Otherwise, since the definition of DDGP and DMDGP ensures that $G[U_0]$ is a clique, realizing the initial clique can simply be carried out in polynomial time in K (constant if K is fixed) [2], by using $\hat{K}$-lateration repeatedly for $\hat{K} < K$.

Repeated K-lateration applied to DDGP and DMDGP instances yields an exact algorithm in the real RAM computational model [7], which endows a theoretical Turing machine with the ability of computing with real numbers exactly — an actual computer would have to approximate real numbers with floating point numbers, and might possibly yield inexact solutions. A similar consideration is also valid during the pruning operation of the BP: checking if the two positions for vertex j found by K-lateration are compatible with the positions of its other adjacent predecessors $h \notin U_j$ (if any) involves a check of the type $\|x_h - x_j\|_2 = d_{hj}$, which in practice is transformed into $|\, \|x_h - x_j\|_2^2 - d_{hj}^2 \,| \le \varepsilon$ for some given $\varepsilon > 0$.

By repeated branching and pruning operations, the BP yields a tree search over the set S of possible positions for vertices $\{K + 1, \ldots, n\}$. This tree (call it T) has width at most 2^{n-K} and depth at most n. If the BP stops when T has depth $< n$, then no positions could be found for vertex n, which means that the instance is NO. Otherwise, the instance is YES; and any sequence $(x_1, \ldots, x_K, \ldots, x_j^{s_j}, \ldots, x_n^{s_n})$ of positions found by the BP for all vertices in V, where $(s_j \mid K < j \le n)$ is a sequence of $+, -$, is a realization of G which certifies a YES (up to ε).

We recall that this certificate is only valid in the real RAM model, which describes a computer able to represent real numbers exactly. In practice, we take $d : E \to \mathbb{Q}_+$, perform operations in floating point, and attempt at minimizing numerical errors using a variety of techniques [8, 15, 36, 37].

2.3. Almost always and almost never

Given any probability space, an event happens *almost always* when the corresponding set has measure 1 in the space, and *almost never* when the corresponding set has measure 0. In the rest of the paper we shall discuss events happening almost always without necessarily making the probability space explicit. In some cases our probability space will be a bounded subset of the realization space $\mathbb{R}^{nK}$, while in others it will be a bounded subset of the edge weight space $\mathbb{R}^m$. The correct setting should be clear from the context.

It is always possible to construct infinite families of DDGP (respectively, DMDGP) instances where the edge weight function d is carefully chosen so that there may be more than two possible positions for vertex j (see Example 3.3 below, and [35] for more examples). But these families all have measure zero ("almost never") in the set of all DDGP (respectively, DMDGP) instances.

Most of the properties discussed in this paper hold almost always: this occurs because K-lateration may fail to work as expected almost never, notably when the points realizing vertices in U_j are not in *general position* [16, p. 20]. If x is a realization of G in general position and $W \subseteq V$ then, for each $W \subseteq V$ with $|W| = h + 1$, $x[W]$ spans an affine subspace of dimension h.

2.4. How K-lateration actually works

Given K points $\{x_1, \ldots, x_K\} \subset \mathbb{R}^K$ and their distances d_i to an unknown point $y \in \mathbb{R}^K$, y can be determined by solving the quadratic system of K equations in K unknowns $y = (y_1, \ldots, y_K)$

$$\forall i \le K \quad \|x_i - y\|_2^2 = d_i^2. \tag{2}$$

The K-lateration operation is as follows:

1. Rewrite Eq. (2) as $\forall i \le K \ \|x_i\|_2^2 + \|y\|_2^2 - 2\langle x_i, y\rangle = d_i^2$.
2. Arbitrarily choose one of these K equations, e.g. the K-th one, and form the system of $K - 1$ equations in K unknowns given by the difference of the i-th equation with the K-th one; this removes the term $\|y\|_2^2$ from all equations, leaving the following (after some rearrangements):
$$\forall i < K \ 2\langle x_i - x_K, y\rangle = (\|x_i\|_2^2 - \|x_K\|_2^2) - (d_i^2 - d_K^2), \tag{3}$$
which is a linear underdetermined system in y.
3. We assume that Eq. (3) has full rank $K - 1$ almost always, so we can express $K - 1$ of the unknowns in function of the remaining one, which we assume without loss of generality (wlog) to be y_K:
$$\forall i < K \ y_i = b_i - B_i y_K, \tag{4}$$
for some $B, b \in \mathbb{R}^K$ [30, §3.3].
4. We replace $y_1, \ldots, y_{K-1}$ in $\|x_K - y\|_2^2 = d_K^2$ and obtain a quadratic equation in the single unknown y_K. We solve this equation and obtain two solutions y_K^+, y_K^- almost always, yielding two positions y^+, y^- for y by using Eq. (4).

5. Finally, we check that y^+, y^- satisfy the original equations Eq. (2). If they do, the system has two solutions almost always. Otherwise, it is infeasible.

We denote by S_j the result of the K-lateration operation in function of y, U_j, namely the position of vertex $j \leq n$ in function of the positions $y[U_j] = (y_{i_1}, \ldots, y_{i_K})$. We remark that either $S_j = \varnothing$ or $|S_j| = 2$ almost always.

The following example shows how trilateration determines: (a) zero or two positions with probability one, and (b) one or infinitely many positions with probability zero.

2.2 Example
Consider a triangle graph over $V = \{1, 2, 3\}$ *with* $d_{12} = 2$ *and* $d_{13} = d_{23} \in \alpha = [1, 2]$, *embedded in* $\mathbb{R}^2$. *If* $x_1 = (0, 0)$ *and* $x_2 = (2, 0)$, *then* x_3 *moves continuously on the segment* $(1, t)$ *for* $t \in [-\sqrt{3}, \sqrt{3}]$ *as* d_{13}, d_{23} *move continuously in* α *(see first picture in Fig. 1).*

- *At* $t = 0$ *(corresponding to* $d_{13} = d_{23} = 1$*) the three points* x_1, x_2, x_3 *are aligned, and therefore their affine span has deficient rank equal to* 1*: this situation yields a single position for vertex* 3 *(point* x_3*). Since three points on the plane almost never determine a single line, the corresponding realization occurs almost never. All of the other values in the interval* α *define nontrivial isosceles triangles having full affine span rank* 2*, yielding two distinct positions for vertex* x_3*. This situation occurs almost always.*

- *A different choice of* α*, e.g.* $[\frac{3}{2}, 2]$*, might have yielded a situation where the affine span rank of the associated realizations is always full. For more complicated graphs it is possible to have situations where both endpoints of the interval yield realizations with deficient ranks.*

- *Suppose now that we add another vertex (labelled by* 4*) to the triangle graph above, and another spatial dimension (so* $K = 3$*). We let* 4 *be adjacent to* 1, 2, 3 *with edge weights* $d_{14} = d_{24} = 2$ *and* $d_{34} = \sqrt{3}$*. We consider realizations in* $\mathbb{R}^3$*. When we apply the* K*-lateration operation to the realization* $x_1 = (0, 0)$*,* $x_2 = (2, 0)$*,* $x_3 = (1, 0)$ *(which occurs almost never),* x_4 *can move in a circle of radius* $\sqrt{3}$ *and centered at* $(1, 0)$*. In other words, this* K*-lateration operation finds an uncountable number of positions for* x_4 *(see last picture in Fig. 1). The "almost always" case occurs when* x_1, x_2, x_3 *are not collinear — and, although it is not shown here (but see [33, Fig. 3.8]), it yields two distinct positions for vertex* 4*.*

We also note that the determination of the positions in $\mathbb{R}^K$ *for the last point given* K *known points is the intersection of* K *spheres. If the spheres intersect at all, this intersection almost always consists of two points, but it may also yield a single point or uncountably many points almost never.* ■

2.5. Search space symmetry

The tree T is a graph defined over $S \subset \mathbb{R}^K$, and is therefore itself naturally embedded in $\mathbb{R}^K$ as the union of the (partial) realizations of G explored during

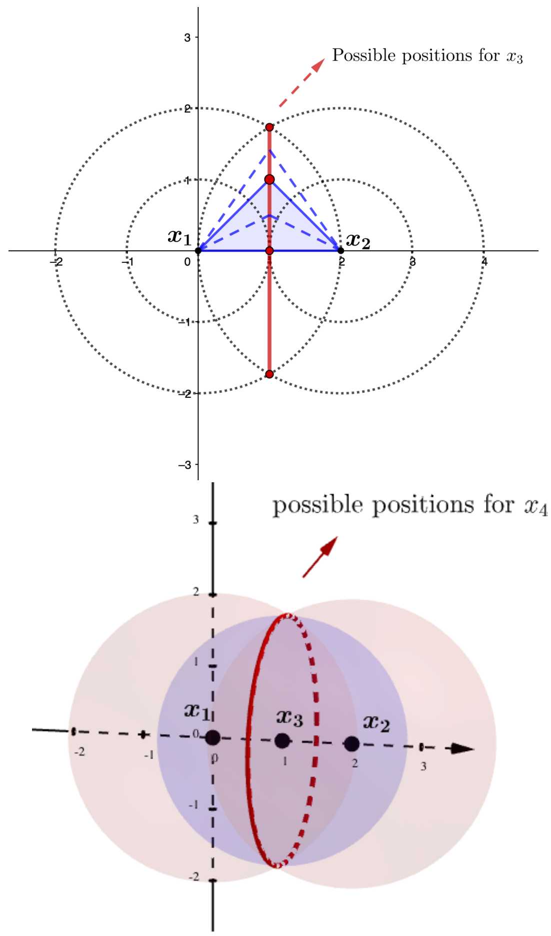


Figure 1: The two situations depicted in Eg. 2.2. Top: x_3 is almost always in any of two positions in $\mathbb{R}^2$, and almost never in just one. Bottom: x_4 is almost never in uncountably many positions in $\mathbb{R}^3$.

the BP. Limited to the DMDGP only, two invariant groups of the embeddings of T were described in [35]. Both groups are reflection groups acting on the realizations of G in $\mathbb{R}^K$. The group action is both geometric (in $\mathbb{R}^K$) and permutative (w.r.t. the tree T seen as a graph).

The *discretization group* is the invariant group of maximum width trees T with 2^{n-K} leaf nodes, where each vertex j is only adjacent to the K immediate predecessors in U_j (as well as perhaps some successors); unsurprisingly, this group has cardinality power of two (more precisely, 2^{n-K}) almost always.

The *pruning group*, which is a subgroup of the discretization group, is the invariant group of the more general case where vertices j may be adjacent to the immediate predecessors in U_j but also to other predecessors that allow the BP to prune. Surprisingly, it was proved (see [35] and [33, §3.3.8]) that the pruning group also has cardinality power of two, where the exponent depends on how many order ranks are not within the range of vertex pairs defining pruning edges, up to a K rank offset. The argument implies that the number of nodes at each tree level only depends on the graph adjacencies rather than the edge weights. This gives rise to a combinatorial method for counting the number of solutions of any DMDGP instance [31].

If $U \subset V$ is such that U is an initial segment of the vertex order of V, it is evident that $G[U]$ is also a DMDGP instance. The combinatorial counting method above can therefore also be used to count the number of partial realizations of G during a run of the BP algorithm. In particular, if there is an order rank v_0 starting with which the number of solutions increases at most logarithmically, the width of T remains polynomially bounded. This was used to prove that the BP algorithm is FPT in K and v_0 [34].

2.5.1. The elusive DDGP symmetries

The difference between DMDGP and DDGP is that the sets U_j of adjacent predecessors must also be immediate and contiguous in the DMDGP case. By an easy induction, this implies that each $G[U_j]$ must be a clique of size K in G, while this need not hold in the DDGP. The fact that each $G[U_j]$ is a clique is the key property used in the analysis of DMDGP symmetries. A similar study for the DDGP does not exist yet.

An attempt to lay some groundwork in extending the study of symmetries from the DMDGP to the DDGP was made in [1]. For a number of years we debated on how to progress without making any actual advance. We argue in this paper that such an extension is impossible because it would require a combinatorial method for counting the solutions of a DDGP instance. However, as shown in Sect. 3 below, DDGP instances may have different numbers of solutions depending on the edge weights only.

3. Can we count DDGP realizations combinatorially?

In this section we claim that we can extend the existing DMDGP combinatorial counting algorithm only to a special subclass of DDGP instances, namely when

$$\forall K < j \leq n \quad U_j \text{ induces a clique of size } K \text{ in } G. \tag{5}$$

We already recalled in Sect. 2 that Property (5) need not hold in all DDGP instances.

We shall first focus on the contra-positive of this claim: whenever some U_j does *not* induce a clique in G, combinatorial counting for the DDGP *must* fail. Since this observation is independent on the presence of pruning edges, our argument is based on the (easier) case of YES instances without pruning edges, i.e. when every edge in G is an edge necessary to carry out the K-lateration operation.

For each $j \in \{1, \dots, n\}$ let a_j be the number of positions, found by the BP algorithm for vertex j, which will eventually lead to a realization of G. We assume that the given DDGP instance is YES, and, wlog, that $a_1 = \cdots = a_K = 1$. Moreover, since the only possible choice for U_{K+1} is $\{1, \dots, K\}$, which are the immediate predecessors of $K+1$, the DMDGP and DDGP coincide on instances of size $K+1$, which implies [27] that $a_{K+1} = 2$.

We start with the trivial observation that, by a repeated application of K-lateration, there are two positions for vertex j for each position of the vertex $\ell(j)$ with largest label in U_j:

$$a_j \leq 2a_{\ell(j)}. \tag{6}$$

By [35], the condition $a_j = 2a_{\ell(j)}$ is necessary to ensure combinatorial counting in DMDGP instances without pruning edges. We therefore look at conditions that yield $a_j = 2a_{\ell(j)}$ almost always, to observe what could go wrong when $a_j < 2a_{\ell(j)}$. We assume in the following that all realizations of G are in general position.

Given a realization x of $G = (V, E)$ we let $D^x(V)$ be the EDM of x. If $W \subseteq V$ we also let $D^x(W)$ be the EDM of $x[W]$. We recall that $\overline{U}_j = U_j \cup \{j\}$.

3.1 Remark

If x is a realization of G and $K < j \leq n$, then we can write the EDM of $x[\overline{U}_j]$ in the form below:

$$\begin{aligned} D^x(\overline{U}_j) &= \begin{pmatrix} D^x(U_j) & d^2_{\cdot,j} \\ d^2_{j,\cdot} & 0 \end{pmatrix} \\ &= \left(\begin{array}{cccc|c} 0 & \|x_{i_1} - x_{i_2}\|_2^2 & \cdots & \|x_{i_1} - x_{i_K}\|_2^2 & d^2_{i_1,j} \\ \|x_{i_2} - x_{i_1}\|_2^2 & 0 & \cdots & \|x_{i_2} - x_{i_K}\|_2^2 & d^2_{i_2,j} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \|x_{i_K} - x_{i_1}\|_2^2 & \|x_{i_K} - x_{i_2}\|_2^2 & \cdots & 0 & d^2_{i_K,j} \\ \hline d^2_{j,i_1} & d^2_{j,i_2} & \cdots & d^2_{j,i_K} & 0 \end{array}\right), \end{aligned} \tag{7}$$

where $D^x(U_j)$ is expressed in function of x, whereas the last row and column is expressed in function of the known edge weights d. ■

3.2 Proposition

Consider a YES DDGP instance (K, G) without pruning edges, and let $j \in V$ such that $K < j \leq n$. If, for any realization z of $G[U_j]$ the matrix $D^z(\overline{U}_j)$ is a EDM, then $a_j = 2a_{\ell(j)} > 1$ almost always.

Proof. If there is no z realizing $G[U_j]$ then there cannot be any realization of G in $\mathbb{R}^K$, making the DDGP instance a NO instance, against the assumption. So we assume there is at least one realization z of $G[U_j]$, which yields $a_{\ell(j)} \geq 1$. Now we use the hypothesis that for any z realizing $G[U_j]$, $D^z(\overline{U}_j)$ is a EDM. By Schoenberg's theorem, we can find the Gram matrix of a realization z of $G[\overline{U}_j]$. From this we infer that K-lateration on U_j to determine the position of vertex j will find a solution set $S_j \neq \varnothing$. By Sect. 2.4, this means that $|S_j| = 2$

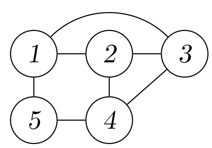
almost always. Since different realizations of $G[U_j]$ almost always yield different positions of vertex $\ell(j)$, we conclude that the number of realizations of $G[\overline{U}_j]$ is almost always twice the number of positions of vertex $\ell(j)$ over all realizations of $G[U_j]$. This concludes the proof. □

Proposition 3.2 suggests looking at cases where $a_j < 2a_{\ell(j)}$ in order to find conditions that prevent combinatorial counting.

3.1. An impossibility result

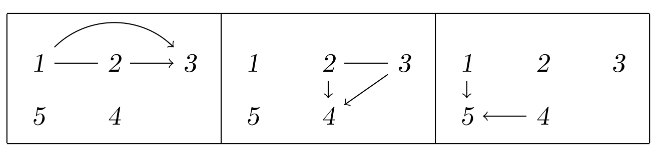
The counterexample below shows what can go wrong if the hypothesis of Prop. 3.2 does not hold.

3.3 Example
Consider the graph $G = (V, E)$ below:



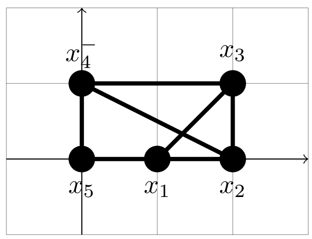
with edge weights $d_{12} = d_{15} = d_{23} = d_{45} = 1$, $d_{34} = 2$, $d_{13} = \sqrt{2}$, $d_{24} = \sqrt{5}$, realized in $\mathbb{R}^2$.

First, we remark that G is a DDGP graph without pruning edges. We show U_j for each $j \in \{3, 4, 5\}$ in the table below (arc tails), as well as the induced edges in $G[U_j]$ (undirected edges). It is evident that $U_5 = \{1, 4\}$ but $\{1, 4\} \notin E$: in other words, $G[U_5]$ is not a 2-clique in G (against Property (5)).



We assume that $x_1 = (1, 0)$, $x_2 = (2, 0)$, $x_3 = (2, 1)$. There are two possible positions for vertex 4, namely $x_4^+ = (4, 1)$, $x_4^- = (0, 1)$, as shown in Fig. 2. However, $\|x_1 - x_4^+\|_2 = \sqrt{10}$ cannot form a triangle with segments realizing $\{1, 5\}, \{4, 5\}$ both having unit length, since $d_{15} + d_{45} = 2 < \sqrt{10}$, which negates the triangular inequality on $1, 4, 5$. On the other hand, the position $x_5 = (0, 0)$ is compatible with x_4^-. In this case, K-lateration would return the singleton $\{x_4^-\}$ as the set S_4 of positions for vertex 4, rather than ensuring $|S_4| \in \{0, 2\}$ as expected. Note that the above instance is not "almost never", as all U_j's are realized in general position. We shall exploit this in Thm. 3.4. Generalizations of this counterexample can be obtained for all K. ■

3.4 Theorem
The solutions of the DDGP cannot be counted combinatorially.



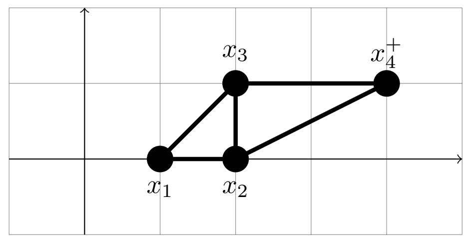


Figure 2: Left: the realization with x_4^- is feasible: we can find a position for x_5. Right: the realization with x_4^+ is infeasible: there is no position for x_5 compatible with the given distances.

Proof. A counting method is combinatorial if it does not use the edge weights (Sect. 1.3). We now construct an uncountable family of DDGP instances for which K-lateration finds 0,1 or 2 positions for a certain vertex, all with positive probability. This shows that the edge weights must necessarily be taken into account by any counting method, and hence that this counting method cannot be combinatorial. We consider the case of Example 3.3: our strategy is to define intervals for d_{24} and d_{34} such that: (i) at the lower extrema K-lateration on 5 finds two positions for x_5, (ii) at the upper extrema K-lateration on 5 only finds one position (and hence fails) for x_5, and (iii) there are neighbourhoods of these extrema for which the same behaviours hold. This will show that the K-lateration output cardinality depends on the edge weights only. Therefore, by inclusion of this DDGP subclass in the whole DDGP, there can be no general combinatorial counting method for the DDGP.

In the rest of the proof (which simply consists of a long but easy symbolic calculation) we sometimes indicate the distance between two vertices u, v by $\overline{uv}$ for brevity. We generalize the instance in Example 3.3 to the uncountable family of instances given by $d_{24} \in [\sqrt{1+\varepsilon^2}, \sqrt{5}]$ and $d_{34} \in [\varepsilon, 2]$ for some small enough $\varepsilon > 0$. If we take the lower extrema of both intervals $d_{24} = \sqrt{1+\varepsilon^2}$ and $d_{34} = \varepsilon$ we obtain $x_4^+ = (2+\varepsilon, 1)$ and $x_4^- = (2-\varepsilon, 1)$, whence

$$\begin{aligned}
\overline{14^+} = \|x_1 - x_4^+\|_2 &= \sqrt{(-1-\varepsilon)^2+1} = \sqrt{2+2\varepsilon+\varepsilon^2} \\
\overline{14^-} = \|x_1 - x_4^-\|_2 &= \sqrt{(-1+\varepsilon)^2+1} = \sqrt{2-2\varepsilon+\varepsilon^2}.
\end{aligned}$$

When ε is negligible, we have $\overline{14^+} \approx \sqrt{2} < 2 = 1+1 = \overline{15} + \overline{4^+5} = d_{15} + d_{45}$ and the same for $\overline{14^-}$, which implies that both positions for vertex 4 yield a distance $\overline{14}$ that satisfies the triangular inequality. As ε grows, $\overline{14^-}$ decreases, which means that it satisfies the triangular inequality for all values of d_{24}, d_{34} in the respective intervals (as verified in Eg. 3.3). We want to find the value of ε at which x_4^+ satisfies the triangular inequality at equality, namely $\overline{14^+} = \overline{15} + \overline{4^+5} = d_{15} + d_{45} = 2$. This happens at $\sqrt{2+2\varepsilon+\varepsilon^2} = 2$, namely $\varepsilon^2 + 2\varepsilon - 2 = 0$, i.e. when $\varepsilon = \frac{-2\pm\sqrt{4+8}}{2} = -1 \pm \sqrt{3}$. Since we assumed $\varepsilon > 0$, $\varepsilon = -1+\sqrt{3}$ is the only value for which $\overline{14^+} = d_{15} + d_{45} = 2$. Thus, the family of DDGP instances under scrutiny has the property that vertex 5 has two positions (almost always) for $d_{24} \in [1, \sqrt{5-2\sqrt{3}}]$, $d_{34} \in [0, -1+\sqrt{3}]$, only one position ($x_4^-$) for $d_{24} \in [\sqrt{5-2\sqrt{3}}, \sqrt{5}]$, $d_{34} \in [-1+\sqrt{3}, 2]$, and zero positions in the remaining cases where no position for vertex 4 exists.

In other words, assuming uniform probability distributions over the two distance intervals for d_{24}, d_{34}, we have shown that this DDGP instance family has (almost always) $2p$ solutions (for some $p \in \mathbb{N}$) with probability $pi_2 = \frac{\sqrt{5-2\sqrt{3}}+\sqrt{3}-2}{\sqrt{5}+1} \approx 0.3$, p solutions with probability $\pi_1 = \frac{\sqrt{5}-\sqrt{5-2\sqrt{3}}-\sqrt{3}+1}{\sqrt{5}+1} \approx 0.08$, and 0 solutions in the remaining events where d_{24} is towards the lower extremum while d_{34} is towards the upper one and *vice versa*, which have joint probability $\pi_0 = 1 - \frac{\pi_2+\pi_1}{\sqrt{5}+1} = \frac{\sqrt{5}+1-(\sqrt{5}-1)}{\sqrt{5}+1} = \frac{2}{\sqrt{5}+1} \approx 0.62$. Note that $\pi_0, \pi_1, \pi_2 > 0$, as claimed. □

Thm. 3.4 does not prevent the existence of counting techniques for subclasses of the DDGP, or based on a condition involving other parameters than G, d, K, or taking into account special structures in the pruning edges.

3.1.1. Just what is impossible here?

Our impossibility result states that there cannot be an extension to the DDGP of the FPT analysis of the BP on DMDGP instances. It does that by showing that the combinatorial counting method that is valid for the DMDGP cannot apply to the DDGP. In turn, this is implied by Thm. 3.4: there exist infinitely many DDGP instances, defined on the same graph, such that their number of solutions varies in function of the edge weights only.

Thm. 3.4 raises a new question: could an FPT analysis of the DDGP rely on an FPT parameter (which controls the exponential behaviour of the solution algorithm) defined in function of the edge weights, and not just of the graph itself? This would yield a worst-case complexity such as $2^\kappa p(\iota)$ where κ is a rational number, which is unusual and somewhat perplexing. One might try and reduce the rational part of the instance to integers, which could however result in an inordinately large integer κ. Most of all, with κ depending on numerical rather than combinatorial parameters, it would be extremely hard to argue that the FPT parameter remains small in all cases (think of infinite families of instances with a few exponentially long distances). These considerations suggest that an FPT analysis of the DDGP is unlikely to exist.

Our belief is that, unless all of the U_j's induce cliques in G, there is no *exact* combinatorial counting method for the DDGP. However, since FPT analyses argue towards worst-case complexity estimates, it is possible that there may exist *approximate* combinatorial counting methods for the DDGP that yield valid, albeit slack, FPT analyses for a DDGP solution algorithm (such as the BP).

In summary, our impossibility result destroys the most direct avenue of thought towards an FPT analysis of the BP for the DDGP, but some hope remains for more convoluted approaches.

3.2. A sufficient condition (without pruning edges)

We derive here a sufficient condition to count solutions combinatorially in DDGP instances without pruning edges and satisfying Property (5).

3.5 Proposition
Let $(K, G = (V, E, d))$ be a YES DDGP instance without pruning edges and satisfying property (5). Let $j \in V$ such that $K < j \leq n$. If $G[U_j]$ is a clique of size K in G, then $a_j = 2a_{\ell(j)}$ almost always.

Proof. Let x by any realization of $G = (V, E, d)$. Then $x[U_j]$ is a realization for $G[U_j]$. Therefore $D^x(U_j)$ must be such that its (i, h)-th component $\|x_i - x_h\|_2$ is equal to d_{ih} for every $\{i, h\}$ in the edges of $G[U_j]$. But since $G[U_j]$ is assumed to be a clique, $\|x_i - x_h\|_2 = d_{ih}$ for every $i < h \in U_j$, i.e. none of the entries of $D^x(U_j)$ depends on the given realization x: we can therefore rename $D^x(U_j)$ to $\hat{D}_j$, since it is a constant matrix. Now we can compute any realization z of $\hat{D}_j$ by using Schoenberg's theorem [39] and spectral decomposition [10], then applying congruence operators to z. For any one of these realizations we can apply K-lateration to find two positions for vertex j almost always (since the DDGP instance is YES), yielding corresponding realizations z' in $\mathbb{R}^K$ for $G[\overline{U}_j]$. This shows that $D^x(\overline{U}_j)$ is an EDM. The claim now holds by Prop. (3.2). □

We also remark that Prop. 3.5 cannot be improved in general terms, for example by asking that $G[U_j]$ is a clique without one or a few edges, since Eg. 3.3 portrays a failure when a single edge is missing from the clique on $G[U_5]$.

This shows that a combinatorial counting of the number of solutions of DDGP instances prior to actually solving the instance may only be possible in the special case where all of the U_j's induce cliques of size K in G. We call the class of such DDGP instances the *combinatorial DDGP*.

3.6 Corollary
For a combinatorial DDGP instance without pruning edges, the number of realizations of G (excluding those that are congruent by rotations and translations) is 2^{n-K} almost always.

Proof. This follows by $a_1 = \cdots = a_K = 1$, $a_{K+1} = 2$, and Prop. 3.5. □

We remark that Cor. 3.6 applies to DMDGP instances. This provides an alternative proof to the result that DMDGP instances without pruning edges almost always have 2^{n-K} incongruent solutions [33, §3.3.8.1].

Acknowledgements

CL is grateful to FAPESP and CNPq for support. LL is partly supported by the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement n. 764759 ETN "MINOA". AM and LL are grateful to ANR for partly supporting this research under PRCI grant "MultiBioStruct". All authors are grateful to several anonymous reviewers for helping making the paper much, much clearer with respect to our mystifying original submission.

References

[1] G. Abud, J. Alencar, C. Lavor, L. Liberti, and A. Mucherino. The k-discretization and k-incident graphs for discretizable distance geometry. *Optimization Letters*, 14:469–482, 2020.

[2] J. Alencar, C. Lavor, and L. Liberti. Realizing euclidean distance matrices by sphere intersection. *Discrete Applied Mathematics*, 256:5–10, 2019.

[3] J. Aspnes, D. Goldenberg, and R. Yang. On the computational complexity of sensor network localization. In S. Nikoletseas and J. Rolim, editors, *Algorithmic Aspects of Wireless Sensor Networks*, volume 3121 of *LNCS*, pages 32–44, Berlin, 2004. Springer.

[4] A. Bahr, J. Leonard, and M. Fallon. Cooperative localization for autonomous underwater vehicles. *International Journal of Robotics Research*, 28(6):714–728, 2009.

[5] S. Billinge, P. Duxbury, D. Gonçalves, C. Lavor, and A. Mucherino. Assigned and unassigned distance geometry: Applications to biological molecules and nanostructures. *4OR*, 14:337–376, 2016.

[6] P. Biswas, T. Lian, T. Wang, and Y. Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions in Sensor Networks*, 2:188–220, 2006.

[7] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions, and universal machines. *Bulletin of the AMS*, 21(1):1–46, 1989.

[8] A. Cassioli, B. Bordeaux, G. Bouvier, A. Mucherino, R. Alves, L. Liberti, M. Nilges, C. Lavor, and T. Malliavin. An algorithm to enumerate all possible protein conformations verifying a set of distance constraints. *BMC Bioinformatics*, 16:23–38, 2015.

[9] A. Cassioli, O. Günlük, C. Lavor, and L. Liberti. Discretization vertex orders for distance geometry. *Discrete Applied Mathematics*, 197:27–41, 2015.

[10] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman & Hall, Boca Raton, 2001.

[11] J. Dattorro. *Convex Optimization and Euclidean Distance Geometry*. $\mathcal{M}\epsilon\beta oo$, Palo Alto, 2015.

[12] I. Dokmanić, R. Parhizkar, J. Ranieri, and M. Vetterli. Euclidean distance matrices: Essential theory, algorithms and applications. *IEEE Signal Processing Magazine*, 1053-5888:12–30, Nov. 2015.

[13] Q. Dong and Z. Wu. A geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data. *Journal of Global Optimization*, 26:321–333, 2003.

[14] R. Downey and M. Fellows. Fixed-parameter tractability and completeness I: Basic results. *SIAM Journal on Computing*, 24(4):873–921, 1995.

[15] D. Gonçalves and A. Mucherino. Discretization orders and efficient computation of cartesian coordinates for distance geometry. *Optimization Letters*, 8:2111–2125, 2014.

[16] J. Graver, B. Servatius, and H. Servatius. *Combinatorial Rigidity*. AMS, Providence, RI, 1993.

[17] T. Havel and K. Wüthrich. An evaluation of the combined use of nuclear magnetic resonance and distance geometry for the determination of protein conformations in solution. *Journal of Molecular Biology*, 182(2):281–294, 1985.

[18] B. Hendrickson. The molecule problem: exploiting structure in global optimization. *SIAM Journal on Optimization*, 5:835–857, 1995.

[19] L. Henneberg. *Die Graphische Statik der starren Systeme*. Teubner, Leipzig, 1911.

[20] D. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and vlsi. *Journal of the ACM*, 32(1):130–136, 1985.

[21] S. Khalife, D. Gonçalves, and Leo Liberti. Distance geometry for word representations and applications. *Journal of Computational Mathematics and Data Science*, 6:100073, 2023.

[22] N. Krislock and H. Wolkowicz. Explicit sensor network localization using semidefinite representations and facial reductions. *SIAM Journal on Optimization*, 20:2679–2708, 2010.

[23] K. Kunen. *Set Theory. An Introduction to Independence Proofs*. North Holland, Amsterdam, 1980.

[24] G. Laman. On graphs and rigidity of plane skeletal structures. *Journal of Engineering Mathematics*, 4(4):331–340, 1970.

[25] C. Lavor, J. Lee, A. Lee-St. John, L. Liberti, A. Mucherino, and M. Sviridenko. Discretization orders for distance geometry problems. *Optimization Letters*, 6:783–796, 2012.

[26] C. Lavor, L. Liberti, B. Donald, B. Worley, B. Bardiaux, T. Malliavin, and M. Nilges. Minimal NMR distance information for rigidity of protein graphs. *Discrete Applied Mathematics*, 256:91–104, 2019.

[27] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino. The discretizable molecular distance geometry problem. *Computational Optimization and Applications*, 52:115–146, 2012.

[28] L. Liberti. A new distance geometry method for constructing word and sentence vectors. In *Companion Proceedings of the Web Conference (DL4G Workshop)*, volume 20 of *WWW*, New York, 2020. ACM.

[29] L. Liberti. Distance geometry and data science. *TOP*, 28:271–339, 220.

[30] L. Liberti and C. Lavor. *Euclidean Distance Geometry: An Introduction*. Springer, New York, 2017.

[31] L. Liberti, C. Lavor, J. Alencar, and G. Abud. Counting the number of solutions of ^{k}DMDGP instances. In F. Nielsen and F. Barbaresco, editors, *Geometric Science of Information*, volume 8085 of *LNCS*, pages 224–230, New York, 2013. Springer.

[32] L. Liberti, C. Lavor, and N. Maculan. A branch-and-prune algorithm for the molecular distance geometry problem. *International Transactions in Operational Research*, 15:1–17, 2008.

[33] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino. Euclidean distance geometry and applications. *SIAM Review*, 56(1):3–69, 2014.

[34] L. Liberti, C. Lavor, and A. Mucherino. The discretizable molecular distance geometry problem seems easier on proteins. In Mucherino et al. [38], pages 47–60.

[35] L. Liberti, B. Masson, C. Lavor, J. Lee, and A. Mucherino. On the number of realizations of certain Henneberg graphs arising in protein conformation. *Discrete Applied Mathematics*, 165:213–232, 2014.

[36] T. Malliavin, A. Mucherino, C. Lavor, and L. Liberti. Systematic exploration of protein conformational space using a distance geometry approach. *Journal of Chemical Information and Modeling*, 59:4486–4503, 2019.

[37] A. Mucherino, C. Lavor, and L. Liberti. The discretizable distance geometry problem. *Optimization Letters*, 6:1671–1686, 2012.

[38] A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, editors. *Distance Geometry: Theory, Methods, and Applications*. Springer, New York, 2013.

[39] I. Schoenberg. Remarks to Maurice Fréchet's article "Sur la définition axiomatique d'une classe d'espaces distanciés vectoriellement applicable sur l'espace de Hilbert". *Annals of Mathematics*, 36(3):724–732, 1935.

[40] A. Singer. Angular synchronization by eigenvectors and semidefinite programming. *Applied and Computational Harmonic Analysis*, 30:20–36, 2011.

[41] S. Skiena and G. Sundaram. A partial digest approach to restriction site mapping. *Bulletin of Mathematical Biology*, 56(2):275–294, 1994.

[42] T.-S. Tay and W. Whiteley. Generating isostatic frameworks. *Structural Topology*, 11:21–69, 1985.

[43] K. Wüthrich. NMR studies of structure and function of biological macromolecules (Nobel lecture). *Angewandte Chemie*, 42:3340–3363, 2003.