Optimization Letters manuscript No.
(will be inserted by the editor)

Discretization orders for distance geometry problems

Carlile Lavor · Jon Lee · Audrey Lee-St. John · Leo Liberti · Antonio Mucherino · Maxim Sviridenko

Received: ? / Accepted: ?

Abstract Given a weighted, undirected simple graph $G = (V, E, d)$ (where $d : E \to \mathbb{R}_+$), the Distance Geometry Problem (DGP) is to determine an embedding $x : V \to \mathbb{R}^K$ such that $\forall \{i, j\} \in E \; \|x_i - x_j\| = d_{ij}$. Although, in general, the DGP is solved using continuous methods, under certain conditions the search is reduced to a discrete set of points. We give one such condition as a particular order on V. We formalize the decision problem of determining whether such an order exists for a given graph and show that this problem is **NP**-complete in general and polynomial for fixed dimension K. We present results of computational experiments on a set of protein backbones whose natural atomic order does not satisfy the order requirements and compare our approach with some available continuous space searches.
Keywords: molecular distance geometry, proteins, sensor network localization, graph drawing.

1 Introduction

In this paper, we discuss a problem that is auxiliary to solving the following problem by means of a discrete search method.

> Distance Geometry Problem (DGP). Given a weighted, undirected, simple graph $G = (V, E, d)$, where $d : E \to \mathbb{R}_+$, and given a positive integer K,

C. Lavor
Dept. of Applied Math. (IMECC-UNICAMP), State University of Campinas, Campinas - SP, Brazil, E-mail: clavor@ime.unicamp.br

J. Lee and M. Sviridenko
IBM T.J. Watson Research Center, NY USA, E-mail: {jonlee,sviri}@us.ibm.com

A. Lee-St. John
Comp. Sci. Dept., Mount Holyoke College, MA USA, E-mail: astjohn@mtholyoke.edu

L. Liberti
LIX, École Polytechnique, Palaiseau, France, E-mail: liberti@lix.polytechnique.fr

A. Mucherino
CERFACS, Toulouse, France, E-mail: antonio.mucherino@cerfacs.fr

establish whether there exists an embedding $x : V \to \mathbb{R}^K$ such that:

$$\forall \{u, v\} \in E \quad \|x_u - x_v\| = d_{uv} \,. \tag{1}$$

We denote explicit dependence of the DGP on K by DGP_K. The DGP has three main applications.

- *Embedding molecules in space.* In the Molecular Distance Geometry Problem (MDGP), G is a molecule graph, where the E is the set of pairs of atoms with known interatomic distances, and $K = 3$. Because the behavior of a molecule depends strongly on its spatial configuration, finding an embedding of V in $\mathbb{R}^3$ is of practical interest (see [22,14,16]). A distinguishing property is that, because of the experimental techniques involved, most distances are bounded above by 6Å. Related problems in molecular conformation involve the minimization of an energy function (see [31, 8,10]).
- *Localizing wireless sensors.* The Sensor Network Localization Problem (SNLP) aims to embed a wireless sensor network in $\mathbb{R}^2$ (so $K = 2$). Pairs of sensors can estimate the distance between them by measuring the power used for a two-way communication. Because sensor networks often include a wired backbone (allowing the link between the sensor network and the external world), and the position of the wired backbone components is usually known, the distinguishing property of the SNLP is that a *partial embedding* $x' : U \to \mathbb{R}^2$ may be given, where $U \subseteq V$ is the set of wired backbone components, called *anchors* in the SNLP literature (see [6,33]).
- *Drawing graphs.* Graph Drawing is a discipline studying algorithms for drawing graphs. The embedding might be defined for any $K \geq 1$, but of course only projections in 2D and 3D are actually represented visually. See `www.graphdrawing.org` for more information.

Even when the DGP input is all integer, the solution might still be irrational (e.g. take the equilateral triangle graph $(\{u, v, w\}, \{\{u, v\}, \{v, w\}, \{u, w\}, d_{uv} = d_{vw} = d_{uw} = 1)$ with $K = 2$), so it is not clear that the is in **NP**. The DGP has a strong connection with the Euclidean Distance Matrix Completion Problem (EDMCP), which is essentially the same problem with the difference being that K is part of the output: the EDMCP asks to find a minimum K such that there exists an embedding $x : V \to \mathbb{R}^K$ satisfying (1) (see [11]).

Although the DGP implicitly requires a search in continuous Euclidean space (see [24,30]), if an appropriate order on the vertices is given, we can show that the feasible space becomes discrete. The main intuitive idea behind this discretization is that, in general, the intersection of K spheres in $\mathbb{R}^K$ determines at most *two* points (assuming the distances in d obey strict simplex inequalities). As long as an order on V is given that ensures that each vertex $i > K$ has at least K adjacent predecessors, it is easy to derive a binary tree search where there are at most two alternatives for placing the next vertex in the sequence (see [16,22]). Search methods based on this principle were proposed in [21,35,3]. The same vertex order for the problem restricted to $K = 3$ was discussed within the context of the Discretizable Molecular Distance Geometry Problem (DMDGP), see [15], and an application to real proteins discussed in [27,19].

The object of this paper is that of describing the auxiliary problem of deciding whether, given a weighted undirected simple graph $G = (V, E, d)$ (with d obeying strict simplex inequalities), there exists a vertex order with the desired properties. The significance of this auxiliary problem is paramount: the atomic order found in the Protein Data Bank (PDB) (see [1]) does not guarantee the ability to use the discrete

approach described above. Currently, methods based on continuous searches are slower than discrete search methods and may produce approximate solutions that are far from satisfying all the distance constraints. Our computational results show the efficiency and solution quality differences obtained on protein embeddings with and without the auxiliary problem discussed in this paper.

The rest of this paper is organized as follows. In Sect. 2 we define the Discretizable Distance Geometry Problem (DDGP), which rests on a certain vertex order guaranteeing a discrete search; in Sect. 3 we discuss the Discretizable Vertex Order Problem (DVOP), the solution of which provides the order needed by the DDGP. The computational results, given in Sect. 4, are obtained on graphs arising from proteins whose natural order does not satisfy the DDGP order requirements. Thus, we illustrate the necessity of the DVOP as a precondition for solving protein conformation problems with this discrete search technique.

1.1 Notation

Let $|V| = n$ and $|E| = m$. For all $v \in V$, let $\delta(v) = \{u \in V \mid \{u, v\} \in E\}$ be the star of vertices around v (also called the *adjacents* of v); for an order $<$ on V, let $\gamma(<, v) = \{u \in V \mid u < v\}$ be the set of predecessors of v in the order $<$ and $\rho(<, v) = |\gamma(v)| + 1$ the rank of v in $<$ (the order is total because V is finite). If $<$ is clear from the context, we write $\gamma(<, v)$ (resp. $\rho(<, v)$) as $\gamma(v)$ (resp. $\rho(v)$). For $V' \subseteq V$, we denote by $G[V']$ the subgraph of G induced by V'. We call an embedding x of G *valid* if (1) holds for G. For a positive integer K and $\mathcal{U} \subseteq \mathbb{R}^K$, we let $\mathsf{aff}\,\mathcal{U}$ be the affine closure of $\mathcal{U}$ (the smallest affine space containing all vectors in $\mathcal{U}$) and $\mathsf{conv}\,\mathcal{U}$ be the convex hull of $\mathcal{U}$ (the smallest convex set containing all vectors in $\mathcal{U}$). If $\mathcal{U}$ is an affine subspace of $\mathbb{R}^K$, we let $\mathsf{dim}\,\mathcal{U}$ be its dimension. If $x : V \to \mathbb{R}^K$ is an embedding, $U \subseteq V$ and $x' : U \to \mathbb{R}^K$ is a partial embedding, then x is an *extension* of x'.

2 The Discretizable Distance Geometry Problem

By "discretizable," we mean the existence of an order on V whereby vertices can be placed one at a time, following the order, in finitely many positions in space, exploiting the positions of the vertices that are already placed.

The DMDGP is discretizable in this same sense, but fails to be completely abstract in two practical regards: (i) it is restricted to three dimensions, and (ii) in order to place the vertex of rank i it requires distances to the three *immediate* adjacent predecessors. The first restriction evidently comes from proteins being naturally embedded in real physical space; the second restriction yields intersections of spheres having radii of similar orders of magnitude, thereby reducing numerical errors (this restriction can also be exploited to derive interesting symmetry properties (see [23])). We remove these restrictions by defining an abstract discretizable DGP.

We consider the basic (sub)problem of placing a vertex $v \in V$ in $\mathbb{R}^K$ (for a positive integer K) when the position of all vertices in a certain subset $U_v \subseteq V \smallsetminus \{v\}$ is known, and U_v is such that, for all $u \in U_v$, we have $\{u, v\} \in E$. In this case the quadratic system (1) reduces to:

$$\forall u \in U_v \quad \|x_u - x_v\| = d_{uv}. \tag{2}$$

Assuming $|U_v| > 1$ and choosing a specific vertex $w \in U_v$ we can, as in [5,34,3], subtract the w-th equation from the others in (2) in order to obtain the following (equivalent) system:

$$\forall u \in U_v \smallsetminus \{w\} \quad 2(x_u - x_w) \cdot x_v = (\|x_u\|^2 - d_{uv}^2) - (\|x_w\|^2 - d_{wv}^2) \tag{3}$$

$$\|x_v\|^2 - 2x_w \cdot x_v + \|x_w\|^2 = d_{wv}^2. \tag{4}$$

We remark that (3) is a linear system in the indeterminate vector x_v whereas (4) is a single quadratic equation in x_v. Let r be the rank of the linear system (3), which we write $2Ax_v = b$ with the u-th rows of A and b being respectively $x_u - x_w$ and $(\|x_u\|^2 - d_{uv}^2) - (\|x_w\|^2 - d_{wv}^2)$.

Lemma 1 *If $r = K$, then (2) has at most one solution.*

Proof If $r = K$, then (3) is equivalent to a system $2Ax_v = b$ where A is a square invertible $K \times K$ matrix; thus $x_v^* = A^{-1}\frac{b}{2}$ is the only solution of (3). If x_v^* satisfies (4), then (2) has x_v^* as unique solution; otherwise (2) has no solution. □

For a vector $x \in \mathbb{R}^K$, let x^{K-1} denote the $(K-1)$-vector consisting of the first $K-1$ components of x.

Lemma 2 *If $r = K-1$, then (2) has at most two solutions.*

Proof If $r = K-1$, then (3) is equivalent to a system $2Ax_v = b$, where A is a rectangular $(K-1) \times K$ matrix of rank $K-1$. We can therefore choose any column of A (choose, say, the K-th column of A and call it N, a $(K-1) \times 1$ matrix) as nonbasic, and rewrite $2Ax_v = b$ as $Bx_v^{K-1} + Nx_{vK} = \frac{b}{2}$, with B a basis of A, whence $x_v^{K-1} = B^{-1}(\frac{b}{2} - Nx_{vK})$. We can therefore replace x_v^{K-1} in (4) with $B^{-1}(\frac{b}{2} - Nx_{vK})$ to obtain a quadratic equation:

$$\alpha x_{vK}^2 - \beta x_{vK} + \eta = 0 \tag{5}$$

in x_{vK} only, where:

$$\alpha = \|B^{-1}N\|^2 + 1 \tag{6}$$

$$\beta = B^{-1}N \cdot (B^{-1}\frac{b}{2} - 2x_w^{K-1}) \tag{7}$$

$$\eta = \|B^{-1}\frac{b}{2}\|^2 + x_{wK}^2 - d_{wv}^2. \tag{8}$$

If $\beta^2 - 4\alpha\eta < 0$, then (5) has no real solutions, which implies that (2) has no solutions. Assuming $\beta^2 - 4\alpha\eta \geq 0$, let x_{vK}^+, x_{vK}^- be the two solutions of (5), and $x_v^+ = (B^{-1}(\frac{b}{2} - Nx_{vK}^+), x_{vK}^+)$ and $x_v^- = (B^{-1}(\frac{b}{2} - Nx_{vK}^-), x_{vK}^-)$ the corresponding vectors. Either $x_v^+ = x_v^-$, and (2) has exactly one solution, or $x_v^+ \neq x_v^-$, and (2) has exactly two solutions. □

If the rank is smaller than $K-1$, then there may be infinitely many placements for v, which means that U_v does not allow an appropriate order to be defined on V.

Lemma 3 *If $r < K-1$ and (2) has a solution, then it has infinitely many solutions.*

Proof If $r < K-1$, then, as in the proof of Lemma 2, we can choose a partition B, N of the column indices of A and express the components of x_v indexed by B as a function of the components indexed by N. We then obtain a quadratic equation as a function of all components indexed by N, which either has no solutions or has infinitely many solutions as $r < K-1$ implies $|N| > 1$. □

By Lemmata 1-3, we need only consider v such that the set of adjacent predecessors includes a set U_v whose linear system (3) rank is either K or $K-1$. The condition $|U_v| \geq K$ is necessary, but not sufficient: let $\mathcal{U}_v = \{x_u \mid u \in U_v\}$; if $|U_v| = K+1$, then the rank of (3) is exactly K only if $\mathsf{dim\ aff}\, \mathcal{U}_v = K$, i.e. if $\mathsf{conv}\, \mathcal{U}_v$ has nonzero volume in $\mathbb{R}^K$. Because $|U_v| = K+1$, $\mathsf{conv}\, \mathcal{U}_v$ is a K-simplex. The volume of a K-simplex in $\mathbb{R}^K$ is given by the Cayley-Menger formula:

$$\Delta_K(\mathcal{U}_v) = \sqrt{\frac{(-1)^{K+1}}{2^K (K!)^2} \begin{vmatrix} 0 & 1 & 1 & \dots & 1 \\ 1 & 0 & d_{12}^2 & \dots & d_{1,K+1}^2 \\ 1 & d_{12}^2 & 0 & \dots & d_{2,K+1}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & d_{1,K+1}^2 & d_{2,K+1}^2 & \dots & 0 \end{vmatrix}}. \tag{9}$$

(see [2], for example). The inequalities $\Delta_K(\mathcal{U}_v) > 0$ are called *strict simplex inequalities* (see [13]). We have thus established the following.

Lemma 4 *If $|U_v| = K+1$ and $\Delta_K(\mathcal{U}_v) > 0$, then $r = K$.*

If $|U_v| = K$, then the rank of (3) is at most $K-1$; it is exactly $K-1$ if $\mathsf{dim\ aff}\, \mathcal{U}_v = K-1$. In other words, we require the $(K-1)$-dimensional volume of $\mathsf{conv}\, \mathcal{U}_v$ to be nonzero. Because $|U_v| = K$, $\mathsf{conv}\, \mathcal{U}_v$ is a $K-1$ simplex in $\mathbb{R}^K$. As above, we use the Cayley-Menger formula for the volume.

Lemma 5 *If $|U_v| = K$ and $\Delta_{K-1}(\mathcal{U}_v) > 0$, then $r = K-1$.*

Proof We compute distances d_{uw} for all $u, w \in U_v$ by using the coordinates in $\mathcal{U}_v$. Because $|U_v| = K$, $\mathsf{dim\ aff}\, \mathcal{U}_v \leq K-1$. The Cayley-Menger determinant is well defined because $x_u \in \mathsf{aff}\, \mathcal{U}_v$ for all $u \in U_v$ and $d_{uw} = \|x_u - x_w\|$; therefore, the projection of d_{uw} on $\mathsf{aff}\, \mathcal{U}_v$ is d_{uw} itself. Because $\Delta_{K-1}(\mathcal{U}_v) > 0$, then $\mathsf{dim\ aff}\, \mathcal{U}_v \geq K-1$, concluding the proof. □

Putting together all the Lemmata in this section, we obtain a proof of the following.

Theorem 1 *Let $v \in V$. If $|U_v| \geq K$ and $\exists U' \subseteq U_v$ with $|U'| = K$ such that U' defines a $(K-1)$-simplex where strict simplex inequalities hold, then (2) has at most two solutions.*

Theorem 1 allows us to properly define the class of DGP instances that can be discretized. We remark that, by (9), $\Delta_K(\mathcal{U}_v)$ only depends on the distances d, which must be known for every pair of vertices in U_v: this means that $G[U_v]$ must be the K-clique $\mathbf{K}_K$. Under this hypothesis, we slightly abuse notation by writing $\Delta_K(U_v)$ to mean $\Delta_K(\mathcal{U}_v)$.

Discretizable Distance Geometry Problem (DDGP). Given a simple undirected graph $G = (V, E)$, an edge weight function $d : E \to \mathbb{R}_+$, an integer $K > 0$, a total order $<$ on V such that:

$$\forall v \in V \ (\rho(v) > K \to |\delta(v) \cap \gamma(v)| \geq K) \tag{10}$$

$$\forall v \in V \ \exists U_v \subseteq \delta(v) \cap \gamma(v) \ (G[U_v] = \mathbf{K}_K \ \wedge \ \Delta_{K-1}(U_v) > 0), \tag{11}$$

and a partial embedding $\bar{x} : V_0 = \{v \in V \mid \rho(v) \leq K\} \to \mathbb{R}^K$ valid on $G[V_0]$, decide whether there is a valid extension $x : V \to \mathbb{R}^K$ of $\bar{x}$.

Because the DDGP contains the DMDGP (see [26] for the definition of the DDGP in 3D and for a comparison between DDGP and DMDGP), the proof of **NP**-hardness of the DMDGP given in [15,17] also holds for the DDGP.

The DMDGP and DDGP can both be solved (approximately for a given $\varepsilon > 0$) using a recursive binary exploration following the order $<$ on V: at each rank i, use the already known positions of the adjacent predecessors in U_v to find at most two positions for the i-th vertex, and recurse the search over each of them. Such an algorithm, called Branch-and-Prune (BP), was described in [21], further discussed in [17], and used in several papers [27,19,18,28,29,25,20] to solve different DMDGP variants. Similar algorithms were proposed in [35,3]. Discrete search algorithms such as BP solve DDGP instances much faster than continuous (and SDP-based) searches (see [22,17]); most importantly, and uniquely among other distance geometry methods, they can be configured to find all incongruent solutions to a given instance.

An important remark about the restriction $G[U_v] = \mathbf{K}_K$ in (11) is in order. Because the BP is iterative in nature, it places vertex v in $\mathbb{R}^K$ after having placed its predecessors. Thus, even though $G[U_v]$ might not be the entire K-clique, at the time of placing v the whole of $\mathcal{U}_v$ is known, and the strict simplex inequalities can be verified for U_v. In other words, from a practical point of view, we can replace the condition $G[U_v] = \mathbf{K}_K$ by the much less restrictive $|U_v| = K$. Formally, this would not work, as the decision problem would be ill-defined: given a set of data, we would only be able to decide whether it is a valid DDGP instance or not by attempting to solve it. Because the DDGP is **NP**-hard, this would be impossible to do in polynomial time unless **P**=**NP**. An alternative way to get around this issue is to notice that the set of DDGP instances not satisfying the strict simplex inequalities have Lebesgue measure zero in the set of all DDGP instances: any given DDGP instance would therefore satisfy (11) in practice with probability 1. Essentially, this would allow one to ignore (11) altogether.

3 The Discretization Vertex Order Problem

This paper mainly discusses the following decision problem.

Discretization Vertex Order Problem (DVOP). Given a simple undirected graph $G = (V, E)$ and a positive integer K, establish whether there is an order $<$ on V such that: (a) $\{v \in V \mid \rho(v) \leq K\}$ is a K-clique in G, and (b) for each $v \in V$ with rank $\rho(v) > K$, we have $|\delta(v) \cap \gamma(v)| \geq K$.

We note that the DVOP does not verify whether the order satisfies the strict simplex inequalities mentioned in the DDGP. This is because, as mentioned above, the set of distance matrices yielding Cayley-Menger determinant having value *exactly* zero has

measure zero within the set of all possible (real) distance matrices. **NP**-completeness of the DVOP follows trivially from **NP**-completeness of the K-clique problem, for finding a DVOP order implies finding K vertices forming a clique in G.

Graphs corresponding to YES instances of DVOP are identified with $(K-1)$-*trilateration graphs* (see [6]), introduced in the sensor network community. We observe the close relation of the DVOP in dimension K to Henneberg graphs (see [9]) from classical rigidity theory with the following definition.

Definition 1 Given an integer $K > 0$, a simple, undirected graph $G = (V, E)$ is a *Henneberg I(K) graph* if (1) G is the complete graph on K vertices or (2) G is constructed from a Henneberg I(K) graph H with $|V| - 1$ vertices by adding a new vertex adjacent to exactly K vertices of H.

It is clear that the DVOP is YES if and only if the graph contains a spanning Henneberg I(K) subgraph. In particular, induction on $|V|$ trivially proves that the DVOP is YES on all Henneberg I(K) graphs. The converse does not necessarily hold because the Henneberg graph definition requires new vertices being adjacent to *exactly* K existing vertices. Thus, Henneberg I(K) graphs are somehow the minimal graphs whose associated DVOP problem is YES. For our purposes, however, the larger the sets $\delta(v) \cap \gamma(v)$ (for v of rank exceeding K), the faster BP will perform: $|\delta(v) \cap \gamma(v)| = K$ ensures that the search tree is at most binary, but more edges to v might make the current position for v infeasible, thereby pruning the current branch and speeding up the search. We therefore also consider the optimization version of the DVOP:

> Optimal Discretization Vertex Ordering Problem (ODVOP). Given a simple undirected graph $G = (V, E)$ and a positive integer K, find an order $<$ on V such that: (a) $\{v \in V \mid \rho(v) \leq K\}$ is a K-clique in G and (b) for each $v \in V$ of rank greater than K, the number of adjacent predecessors of v is maximum and is $\geq K$.

The ODVOP is a multi-objective maximization problem, whose objective function vector is $(|\delta(v) \cap \gamma(v)| \ : \ v \in V \ (\rho(v) > K))$. For all $v \in V$, let $\Gamma(v)$ be the set of edges in $G[\delta(v) \cap \gamma(v)]$.

Lemma 6 $\bigcup_{v \in V} \Gamma(v) = E$.

Proof Let $\{u, v\} \in E$ and assume without loss of generality that $u < v$. Then $u \in \delta(v) \cap \gamma(v)$, which implies $\{u, v\} \in \Gamma(v)$. □

Lemma 7 $\forall u \neq v \in V \ (\Gamma(u) \cap \Gamma(v) = \emptyset)$.

Proof Suppose $\{w, w'\} \in \Gamma(u) \cap \Gamma(v)$; then $((w \in \delta(u) \wedge w' = u) \vee (w' \in \delta(u) \wedge w = u)) \wedge ((w \in \delta(v) \wedge w' = v) \vee (w' \in \delta(v) \wedge w = v))$. Because $u \neq v$, we can assume without loss of generality that $(w \in \delta(u) \wedge w' = u) \wedge (w' \in \delta(v) \wedge w = v)$, which implies $u \in \delta(v) \wedge v \in \delta(u)$. Furthermore, because G is simple, $\{w, w'\} = \{u, v\}$. Because the order relation $<$ is not symmetric, if $u \in \gamma(v)$, then $v \notin \gamma(u)$ and vice versa; hence, $\{u, v\}$ is either in $\Gamma(u)$ or in $\Gamma(v)$, but not in both. Thus $\{w, w'\} \notin \Gamma(u) \cap \Gamma(v)$. □

Proposition 1 $\sum_{v \in V} |\delta(v) \cap \gamma(v)| = |E|$.

Proof We have:

$$\sum_{v \in V} |\delta(v) \cap \gamma(v)| = \sum_{v \in V} |\Gamma(v)| = \left| \bigcup_{v \in V} \Gamma(v) \right| + |S| = |E| + |S|,$$

where the last equality follows by Lemma 6 and S is a set formed by unions and set differences of intersections of the sets $\Gamma(v)$. By Lemma 7, all these intersections are empty, therefore $S = \emptyset$ and the result follows. □

Theorem 2 *For a given simple weighted undirected graph $G = (V, E, d)$, all DVOP-feasible solutions are in the Pareto set of the ODVOP.*

Proof Let $<, <'$ be solutions of the DVOP instance G. Let $<$ be in the Pareto set and suppose $<$ strictly dominates $<'$. Then:

$$\forall v \in V \quad |\delta(v) \cap \gamma(<', v)| \leq |\delta(v) \cap \gamma(<, v)| \tag{12}$$

$$\exists u \in V \quad |\delta(u) \cap \gamma(<', u)| < |\delta(u) \cap \gamma(<, u)|. \tag{13}$$

By Prop. 1 we have:

$$\sum_{v \in V} |\delta(v) \cap \gamma(<', v)| = \sum_{v \in V} |\delta(v) \cap \gamma(<, v)| = |E|. \tag{14}$$

We remark that (13)-(14) imply $\exists w \in V$ $(|\delta(v) \cap \gamma(<', v)| > |\delta(v) \cap \gamma(<, v)|)$, contradicting (12). □

Thm. 2 shows that the ODVOP and the DVOP are in some sense equivalent. The significance of the ODVOP is mostly algorithmic: because our proposed search method is based on a binary tree, it pays to keep the tree breadth limited, specially at the early nodes. We therefore use the ODVOP maximality requirements to influence the choice of the next vertex in the order in case of a draw. In other words, if there exist two or more candidate next vertices whose set of adjacent predecessors is greater than K, we choose one among the vertices yielding the largest such set.

3.1 Mathematical programming formulation

Different mathematical programming formulations for the DVOP can be conceived, based on two sets of decision variables: precedence variables $x : V \times V \to \{0, 1\}$ (for all $u, v \in V$ we let $x_{uv} = 1$ only if $u < v$) and rank variables $y : V \times \{1, \ldots, |V|\} \to \{0, 1\}$ (for $v \in V$ and $k \in \{1, \ldots, |V|\}$ we let $y_{vk} = 1$ only if $\rho(v) = k$). We tested a few of these formulations, using AMPL [7] and CPLEX [12], on a small set of randomly generated DVOP instances. We only present here the formulation that performed best. Our largest tested instance was a random graph with 50 vertices and 0.5 edge creation probability solved at the root node in 29.94s by CPLEX 11 running on a 1.4GHz CPU with 3GB RAM (no other tested formulation could even solve this instance to optimality, let alone at the root node).

– Sets.
  1. $G = (V, E)$: the graph;
  2. $R = \{1, \ldots, |V|\}$: the set of rank values.

– Parameters: $K \in \mathbb{N}$.
– Decision variables:

$$\forall v \in V, k \in R \quad y_{vk} = \begin{cases} 1 & \text{if } v \text{ is } k\text{-th in the order} \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

– Objective function: none.
– Constraints:
1. each rank k has a unique vertex assigned to it:

$$\forall k \in R \quad \sum_{v \in V} y_{vk} = 1; \tag{16}$$

2. each vertex has a unique rank assigned to it:

$$\forall v \in V \quad \sum_{k \in R} y_{vk} = 1; \tag{17}$$

3. each vertex with rank $> K$ has at least K adjacent predecessors:

$$\forall v \in V, k \in R \smallsetminus \{1, \ldots, K\} \quad \sum_{\substack{u \in V \\ \{u,v\} \in E}} \sum_{\substack{i \in R \\ i < k}} y_{ui} \geq K y_{vk}. \tag{18}$$

By (16)-(17), the rank is well defined. By (18), if $\rho(v) = k$ then at least K vertices adjacent to v have smaller rank. Thus the formulation above correctly models the DVOP.

3.2 The DVOP in fixed dimension

If K is fixed, the problem becomes polynomially solvable. For completeness, we provide an extension of the proof from [6] for trilateration graphs.

Proposition 2 *DVOP with fixed K is in* **P**.

Proof Finding all K-cliques in G can be carried out by simply testing all subsets of V of cardinality K: this requires $O(\binom{n}{K}K^2) = O(n^K)$ steps. For each K-clique, we build vertex orders on the remaining $n - K$ vertices by a greedy algorithm that chooses the vertex with highest number of adjacent predecessors as next in the order. The choice of each next vertex can be carried out trivially in $O(n^2)$ time, yielding a worst-case $O(n^{K+3})$ polynomial algorithm. The instance is a NO if no clique yields a subsequent order where at least one vertex has fewer than K adjacent predecessors, and YES if there is a clique that yields an order where all vertices have at least K adjacent predecessors. □

The interest of the algorithm in the above proof is that usually K is much smaller than n (typically $K \in \{2, 3\}$), so that the DVOP is definitely a "fixed-K" type of problem. Finding all 2-cliques simply amounts to listing all edges, and finding all 3-cliques to listing all triangles [32]. For a given clique C, we can improve on the greedy part to complete C to a valid DVOP order on V (or certify such an order does not exist) in $O(n^2)$ as shown in Alg. 1. A further (practical) improvement can be obtained by keeping $V \smallsetminus B$ ordered by descending α; because the α update at Step 11 is minor, one can use insertion sort as a practically fast re-sorting algorithm [4].

Algorithm 1 DVOP order completion.

Require: A clique $C \subseteq V$ with $|C| = K$
Ensure: Whether $\exists$ DVOP order from C
1: **for** $v \in V$ with $\rho(v) > K$ **do**
2: $\quad \alpha(v) \leftarrow |\delta(v) \cap C|$
3: **end for**
4: Initialize $B \leftarrow C$
5: **while** $|B| < n$ **do**
6: $\quad$ Let $v = \text{argmax}\{\alpha(u) \mid u \in V \setminus B\}$
7: $\quad$ **if** $\alpha(v) < K$ **then**
8: $\quad\quad$ Stop: no extension of C can result in a DVOP order
9: $\quad$ **end if**
10: $\quad$ **for** $u \in \delta(v) \setminus B$ **do**
11: $\quad\quad \alpha(u) \leftarrow \alpha(u) + 1$
12: $\quad$ **end for**
13: $\quad B \leftarrow B \cup \{v\}$
14: **end while**

The exactness of Alg. 1 follows by contradiction. Supposing it reaches Line 8 when C could in fact be extended to a DVOP order $<$, then there must be a rank where the vertex v chosen by the algorithm at Line 6 was not the one with the highest number of adjacent predecessors (for $<$ would provide at least K of them), against the maximal choice in Line 6.

4 Computational results

We performed computational experiments on a GNU C 4.1.2 implementation of Alg. 1 (compiled with `-O3` flag) running on an Intel Core2 2.13GHz CPU with 4GB RAM running Linux.

We consider instances generated from proteins having known conformations in the PDB [1]. Each PDB record consists in a set of atomic coordinates for a given protein: we generate instances by computing the distances between all the possible pairs of hydrogens in the molecule, and by keeping only the ones smaller than a predefined threshold δ. This procedure simulates data obtained from experiments of Nuclear Magnetic Resonance (NMR), because all the distances are between hydrogens, and only short-range distances are considered. The threshold δ usually ranges between 5Å and 6Å: we set $\delta = 5.5$Å because with this value the discretization assumptions (10)-(11) do not hold if the hydrogen atoms are ordered as in the PDB files: in other words, solving a DVOP becomes a necessary precondition.

For each protein, we record in Table 1 the number n of hydrogens it contains, the number $|E|$ of available distances, and the seconds of user CPU time taken to solve the DVOP. Two columns of Table 1 refer to the BP algorithm [21] applied to the reordered proteins. We report the solution quality in terms of Largest Distance Error (LDE):

$$\text{LDE}(\{x_1, x_2, \ldots, x_n\}) = \frac{1}{|E|} \sum_{\{i,j\}} \frac{|\,||x_i - x_j|| - d_{ij}\,|}{d_{ij}},$$

and the user CPU time. In all the experiments, the tolerance ε used when comparing known and computed distances is set to 0.001. The last two columns refer to DGSOL [24], a software for distance geometry based on a continuous formulation of the problem.

	Alg. 1			BP algorithm		DGSOL	
Instance	n	$\|E\|$	time	LDE	time	LDE	time
1brv	90	729	0.00	3.36e-11	0.01	4.14e-01	1.90
1a11	144	1192	0.00	2.43e-12	0.01	1.07e-05	5.27
1erp	209	1969	0.00	3.63e-11	0.05	3.95e-01	7.21
1aqr	214	1690	0.00	3.45e-11	0.02	6.19e-01	8.34
1bbl	221	1690	0.00	2.19e-08	0.05	9.29e-01	9.81
1ed7	261	2591	0.00	3.91e-11	0.05	8.34e-01	8.04
1h1j	261	2489	0.00	3.16e-11	0.03	3.41e-01	13.08
1ahl	268	2508	0.00	4.33e-11	0.02	6.46e-01	15.03
1dv0	275	2669	0.00	4.08e-10	0.05	9.20e-01	14.47
1k1v	277	2600	0.00	4.25e-11	0.06	7.42e-01	12.66
1ccq	389	3888	0.00	5.97e-11	0.10	7.47e-01	20.46
1a2s	480	4723	0.00	5.71e-08	0.77	7.72e-01	24.75
1acz	589	6067	0.01	5.36e-08	1.97	7.42e-01	44.15
2hsy	620	5935	0.01	8.23e-11	0.66	8.10e-01	32.66
1b4c	1152	11044	0.05	7.62e-08	1.81	9.22e-01	117.51
1a23	1157	11628	0.05	9.08e-11	2.38	8.79e-01	110.00
2ron	1501	15101	0.08	1.09e-06	4.15	8.47e-01	148.61
1ezo	2259	21049	0.17	4.89e-07	7.91	9.09e-01	308.90

Table 1 Performances of Alg. 1 and BP.

Because DGSOL accepts a set of lower and upper bounds on the available distances as input (and therefore solves a different problem than ours), the comparison is not wholly fair. Notwithstanding, DGSOL is the only well-known continuous optimization-based algorithm with publicly available code that we could use as a reference to compare against. We provided DGSOL with the set of intervals $[d-\varepsilon, d+\varepsilon]$, where d is the generic distance given to BP. The obtained values for the LDE function and the user CPU time show that Alg. 1 and BP together are faster (by around 2 orders of magnitude) and able to find better-quality (by around 10 orders of magnitude) solutions. Moreover, we point out that obtaining a good order by Alg. 1 only takes a fraction of the CPU time taken to solve the instance by BP.

Acknowledgments

We thank R. Andonov and S. Cafieri for useful comments. CL is grateful to FAPESP and CNPq for financial support.

References

1. Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I., Bourne, P.: The protein data bank. Nucleic Acid Research **28**, 235–242 (2000)
2. Blumenthal, L.: Theory and Applications of Distance Geometry. Oxford University Press, Oxford (1953)
3. Carvalho, R., Lavor, C., Protti, F.: Extending the geometric build-up algorithm for the molecular distance geometry problem. Information Processing Letters **108**, 234–237 (2008)
4. Cook, C., Kim, D.: Best sorting algorithm for nearly sorted lists. Communications of the ACM **23**(11), 620–624 (1980)
5. Dong, Q., Wu, Z.: A geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data. Journal of Global Optimization **26**, 321–333 (2003)

6. Eren, T., Goldenberg, D., Whiteley, W., Yang, Y., Morse, A., Anderson, B., Belhumeur, P.: Rigidity, computation, and randomization in network localization. IEEE Infocom Proceedings pp. 2673–2684 (2004)
7. Fourer, R., Gay, D.: The AMPL Book. Duxbury Press, Pacific Grove (2002)
8. Gu, J., Du, B., Pardalos, P.: Multispace search for protein folding. In: L. Biegler, T. Coleman, A. Conn, F. Santosa (eds.) Large-scale Optimization with Applications, Part III: Molecular Structure and Optimization, pp. 47–68. Springer (1997)
9. Henneberg, L.: Die graphische Statik der starren Systeme. B.G. Teubner, Leipzig (1911)
10. Huang, H., Pardalos, P.: A multivariate partition approach to optimization problems. Cybernetics and Systems Analysis **38**, 265–275 (2002)
11. Huang, H.X., Liang, Z.A., Pardalos, P.: Some properties for the Euclidean distance matrix and positive semidefinite matrix completion problems. Journal of Global Optimization **25**, 3–21 (2003)
12. ILOG: ILOG CPLEX 11.0 User's Manual. ILOG S.A., Gentilly, France (2008)
13. Jiao, Y., Stillinger, F., Torquato, S.: Geometrical ambiguity of pair statistics I. point configurations. Tech. Rep. 0908.1366v1, arXiv (2009)
14. Lavor, C., Liberti, L., Maculan, N.: Computational experience with the molecular distance geometry problem. In: J. Pintér (ed.) Global Optimization: Scientific and Engineering Case Studies, pp. 213–225. Springer, Berlin (2006)
15. Lavor, C., Liberti, L., Maculan, N.: The discretizable molecular distance geometry problem. Tech. Rep. q-bio/0608012, arXiv (2006)
16. Lavor, C., Liberti, L., Maculan, N.: Molecular distance geometry problem. In: C. Floudas, P. Pardalos (eds.) Encyclopedia of Optimization, second edn., pp. 2305–2311. Springer, New York (2009)
17. Lavor, C., Liberti, L., Maculan, N., Mucherino, A.: The discretizable molecular distance geometry problem. Computational Optimization and Applications (in revision)
18. Lavor, C., Liberti, L., Mucherino, A., Maculan, N.: On a discretizable subclass of instances of the molecular distance geometry problem. In: D. Shin (ed.) Proceedings of the 24th Annual ACM Symposium on Applied Computing, pp. 804–805. ACM (2009)
19. Lavor, C., Mucherino, A., Liberti, L., Maculan, N.: Computing artificial backbones of hydrogen atoms in order to discover protein backbones. In: Proceedings of the International Multiconference on Computer Science and Information Technology, pp. 751–756. IEEE, Mragowo, Poland (2009)
20. Lavor, C., Mucherino, A., Liberti, L., Maculan, N.: On the solution of molecular distance geometry problems with interval data. In: Proceedings of the International Workshop on Computational Proteomics. IEEE, Hong Kong (2010)
21. Liberti, L., Lavor, C., Maculan, N.: A branch-and-prune algorithm for the molecular distance geometry problem. International Transactions in Operational Research **15**, 1–17 (2008)
22. Liberti, L., Lavor, C., Mucherino, A., Maculan, N.: Molecular distance geometry methods: from continuous to discrete. International Transactions in Operational Research **18**, 33–51 (2010)
23. Liberti, L., Masson, B., Lavor, C., Lee, J., Mucherino, A.: On the number of solutions of the discretizable molecular distance geometry problem. Tech. Rep. 1010.1834v1[cs.DM], arXiv (2010)
24. Moré, J., Wu, Z.: Global continuation for distance geometry problems. SIAM Journal of Optimization **7**(3), 814–846 (1997)
25. Mucherino, A., Lavor, C.: The branch and prune algorithm for the molecular distance geometry problem with inexact distances. In: Proceedings of the International Conference on Computational Biology, vol. 58, pp. 349–353. World Academy of Science, Engineering and Technology (2009)
26. Mucherino, A., Lavor, C., Liberti, L.: The discretizable distance geometry problem. Optimization Letters (in revision)
27. Mucherino, A., Lavor, C., Liberti, L., Maculan, N.: On the definition of artificial backbones for the discretizable molecular distance geometry problem. Mathematica Balkanica **23**, 289–302 (2009)
28. Mucherino, A., Lavor, C., Maculan, N.: The molecular distance geometry problem applied to protein conformations. In: S. Cafieri, A. Mucherino, G. Nannicini, F. Tarissan, L. Liberti (eds.) Proceedings of the 8^{th} Cologne-Twente Workshop on Graphs and Combinatorial Optimization, pp. 337–340. École Polytechnique, Paris (2009)

29. Mucherino, A., Liberti, L., Lavor, C., Maculan, N.: Comparisons between an exact and a metaheuristic algorithm for the molecular distance geometry problem. In: F. Rothlauf (ed.) Proceedings of the Genetic and Evolutionary Computation Conference, pp. 333–340. ACM, Montreal (2009)
30. Pardalos, P., Liu, X.: A tabu based pattern search method for the distance geometry problem. In: F. Giannessi, T. Rapcsák, S. Komlósi (eds.) New Trends in Mathematical Programming. Kluwer, Dordrecht (1998)
31. Pardalos, P., Shalloway, D., Xue, G. (eds.): Global Minimization of Nonconvex Energy Functions: Molecular Conformation and Protein Folding, vol. 23. American Mathematical Society (1996)
32. Schank, T., Wagner, D.: Finding, counting and listing all triangles in large graphs, an experimental study. In: S. Nikoletseas (ed.) Workshop on Experimental Algorithms, *LNCS*, vol. 3503, pp. 606–609. Springer, Berlin (2005)
33. So, M.C., Ye, Y.: Theory of semidefinite programming for sensor network localization. Mathematical Programming **109**, 367–384 (2007)
34. Wu, D., Wu, Z.: An updated geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data. Journal of Global Optimization **37**, 661–673 (2007)
35. Wu, D., Wu, Z., Yuan, Y.: Rigid versus unique determination of protein structures with geometric buildup. Optimization Letters **2**(3), 319–331 (2008)