Recent advances on the discretizable molecular distance geometry problem

Carlile Lavor[a], Leo Liberti[b,*], Nelson Maculan[c], Antonio Mucherino[d]

[a]*Dept. of Applied Math. (IMECC-UNICAMP), State University of Campinas, 13081-970, Campinas - SP, Brazil*
[b]*LIX, École Polytechnique, 91128 Palaiseau, France*
[c]*Federal University of Rio de Janeiro (COPPE–UFRJ), C.P. 68511, 21945-970, Rio de Janeiro - RJ, Brazil*
[d]*IRISA, University of Rennes 1, Rennes, France*

Abstract

The Molecular Distance Geometry Problem (MDGP) consists in finding an embedding in $\mathbb{R}^3$ of a nonnegatively weighted simple undirected graph with the property that the Euclidean distances between embedded adjacent vertices must be the same as the corresponding edge weights. The Discretizable Molecular Distance Geometry Problem (DMDGP) is a particular subset of the MDGP which can be solved using a discrete search occurring in continuous space; its main application is to find three-dimensional arrangements of proteins using Nuclear Magnetic Resonance (NMR) data. The model provided by the DMDGP, however, is too abstract to be directly applicable in proteomics. In the last five years our efforts have been directed towards adapting the DMDGP to be an ever closer model of the actual difficulties posed by the problem of determining protein structures from NMR data. This survey lists recent developments on DMDGP related research.

Keywords: graph theory, bioinformatics, protein conformation, branch-and-prune

1. Introduction

The determination of the three-dimensional structure of a given protein is an all-important and formidable problem in biochemistry, mainly because the function of a protein is linked to its structure as well as to its atomic composition [39]. We consider here the subproblem of determining the protein structure with information arising from Nuclear Magnetic Resonance (NMR) data [10].

*Corresponding author
Email addresses: clavor@ime.unicamp.br (Carlile Lavor), liberti@lix.polytechnique.fr (Leo Liberti), maculan@cos.ufrj.br (Nelson Maculan), antonio.mucherino@irisa.fr (Antonio Mucherino)

We assume NMR data can be stored as a nonnegatively interval-weighted simple (i.e., without loops or parallel edges) undirected graph $G = (V, E, d)$ where V represents a subset of atoms of the molecule for which distance measurements can be obtained, $\{u, v\} \in E$ if a distance measurement is present between atoms u and v, and d associates an edge $\{u, v\} \in E$ with the respective interval measurement $[d^L_{uv}, d^U_{uv}]$ (since precise distances are also known for certain edges, such as for covalent bonds, some intervals might have $d^L_{uv} = d^U_{uv}$). The main problem is that of finding a set (alternatively, all sets) of Cartesian coordinates for the atoms that are consistent with all the distance information. We shall call this problem the Protein Structure from NMR Data (PSNMR).

Our survey will focus on several variants of the PSNMR. Specifically, we shall consider the cases when: (a) d maps E into nonnegative real numbers (instead of intervals); (b) V is the set of *all* atoms; (c) a particular order on V guarantees the existence of an iterative search for the position of $v \in V$ given the positions of its adjacent predecessors; (d) the Euclidean space used for the embedding has an arbitrary number of dimensions (this is useful for applications other than to molecular structure prediction). Each case gives rise to different theoretical results; we show how we combined them in order to derive a very efficient discrete search in continuous space that addresses the main problem.

This paper is organized as follows: in the rest of Sect. 1 we give a very short review of continuous search based methods and illustrate their weaknesses as a motivation to work towards a discrete search. In Sect. 2 we introduce our discrete approach. In Sect. 3 we generalize the discrete search method to Euclidean spaces of arbitrary dimensions. In Sect. 4 we discuss automatic methods to find "good" orders for V guaranteeing the existence of a discrete search method. In Sect. 5 we restrict V to only contain hydrogen atoms. Sect. 6 presents our implementation to serial and parallel architectures. Sect. 7 concludes the paper and discusses future work.

1.1. Problems solved by continuous methods

Given a simple undirected graph $G = (V, E)$ and a positive integer K, an *embedding* of G in $\mathbb{R}^K$ is a function $x : V \to \mathbb{R}^K$. Let $d : E \to \mathbb{R}_+$ be a given edge weight function defined on $G = (V, E, d)$. An embedding is *valid* for G if

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

where $\| \cdot \|$ is the Euclidean norm, $x_v = x(v)$ for all $v \in V$ and $d_{uv} = d(\{u, v\})$ for all $\{u, v\} \in E$. For any $U \subseteq V$ let $G[U]$ be the subgraph of G induced by U (i.e., $(U, \{\{u, v\} \in E \mid u, v \in U\})$). An embedding of $G[U]$ is a *partial embedding* of G. If x is an embedding of G and y is an embedding of $G \cup H$, for some simple undirected weighted graph H, such that $\forall u \in U \; x_u = y_u$ then y is an *extension* of x. With a slight abuse of notation, if $v \notin U$ and y is an embedding of $G[U \cup \{v\}]$, we write $y = (x, y_v)$; in this case we also say that the point y_v extends x.

The most basic model for the PSNMR problem is the following.

MOLECULAR DISTANCE GEOMETRY PROBLEM (MDGP). Given a nonnegatively weighted simple undirected graph $G = (V, E, d)$, is there a valid embedding of G in $\mathbb{R}^3$?

This is one of the foremost problems in distance geometry [2]; we shall call its generalization to $\mathbb{R}^K$ (with K being given as part of the input) the DISTANCE GEOMETRY PROBLEM (DGP), and denote the restriction of the DGP to a particular fixed dimension K by DGP$_K$. If d is an interval-valued function, i.e., $d(\{u, v\}) = [d^L_{uv}, d^U_{uv}]$ for all $\{u, v\} \in E$, we obtain a problem which is "closer" to the PSNMR.

INTERVAL MOLECULAR DISTANCE GEOMETRY PROBLEM (iMDGP). Given a nonnegatively interval-weighted simple undirected graph $G = (V, E, d)$, is there an embedding $x : V \to \mathbb{R}^3$ such that:

$$\forall \{u, v\} \in E \quad d^L_{uv} \leq \|x_u - x_v\| \leq d^U_{uv}? \tag{2}$$

In this case, an embedding is valid if it satisfies (2). Again, we consider the generalization to $\mathbb{R}^K$ and call it the INTERVAL DISTANCE GEOMETRY PROBLEM (iDGP).

1.2. Characterization of the solution set

Let $\bar{X} = \{x : V \to \mathbb{R}^K \mid x \text{ satisfies (2)}\}$ be the set of all solutions to an iDGP instance. Then, if T is a translation or rotation of $\mathbb{R}^K$, for all $x \in \bar{X}$ we also have $T(x) \in \bar{X}$. Because there are continuously many such transformations, it follows that $|\bar{X}| = 2^{\aleph_0}$. We define an equivalence relation $\sim$ on $\bar{X}$ such that $x \sim y$ if and only if there is a translation or rotation T such that $y = T(x)$. We then define $X = \bar{X}/\sim$ and identify the equivalence classes of X with one of their representatives $x \in \bar{X}$. We can now consider X as the "interesting" set of solutions of an iDGP instance. We remark that $|X|$ is not necessarily infinite. In fact, most of the iDGP variants considered in the sequel will have a finite $|X|$.

1.3. Problem complexity

A reduction from the SUBSET-SUM problem to the DGP$_1$ with unit weights was given in [38], showing that DGP$_1$ is **NP**-complete. For fixed values of K, [38] describes a reduction from 3-SAT to DGP$_1$ with integer weights and a reduction from DGP$_1$ with integer weights to DGP$_K$ with integer weights. In the same paper, Saxe also remarked that, since YES certificates for the DGP generally involve irrational numbers for $K > 1$, it is not clear whether the DGP belongs to the class **NP** or not. From this it follows that the MDGP is **NP**-hard, and the same holds for DGP$_K$ for each integer $K > 1$. Considering formal decision problems as sets of instances, it is clear that DGP$_3$ = MDGP $\subset$ iMDGP $\subset$ iDGP and DGP$_K \subset$ DGP for all $K \in \mathbb{N}$. Again, because singletons are also intervals, DGP $\subset$ iDGP. Thus, by restriction ([9], Sect. 3.2.1), the DGP, iMDGP and iDGP are also **NP**-hard.

1.4. Limitations of continuous methods

The problems listed above are naturally cast as nonlinear systems of equations and inequalities, and can therefore be reformulated to minimizing an objective function consisting of a sum of error terms, which is a Global Optimization (GO) problem. Some continuous methods for solving such problems are surveyed in [16, 25]. These methods often exhibit the following disadvantages.

- *Reliability.* All computations are floating-point; this yields inaccurate solutions. Moreover, it is well known that floating point errors often accumulate, which in the long run may invalidate the solution.
- *Efficiency.* GO methods often involve locally solving a (nonconvex) Nonlinear Programming (NLP) subproblem; local NLP solvers are complex pieces of software which may take a long time to converge.
- *Completeness.* To the best of our knowledge, there is no continuous method which is able to compute all solutions of an iDGP instance; and in fact most continuous methods are actually designed to compute at most one solution.

Of course these disadvantages are due to a trade-off against generality. In the rest of this paper we shall present mixed combinatorial methods for solving *subclasses* of the iDGP. It is this restriction that allows our methods to be more reliable, efficient and complete than continuous methods. Moreover, the subclasses for which our methods work are a good model for solving the iMDGP on proteins, which are in fact the main motivation for the PSNMR.

2. The Discretizable Molecular Distance Geometry Problem

Although the DGP implicitly requires a search in continuous space, if an appropriate order is given on V, we can show that the search space has a finite number of valid embeddings, up to translations and rotations. For an order $<$ on V and for each $v \in V$, let $\rho(v) = |\{u \in V \mid u \leq v\}|$ be the *rank* of v in V with respect to $<$. Since the rank defines a bijection between V and $\{1, \ldots, |V|\}$, we can identify v with its rank and extend arithmetic notation to V so that for some appropriate $i \in \mathbb{Z}$, $v + i$ denotes the vertex $u \in V$ with $\rho(u) = \rho(v) + i$.

We now outline an iterative algorithm for solving the DMDGP, a subset of the MDGP which will be defined below. We assume that an order is given on V. Suppose we want to embed a vertex $v \in V$ of rank greater than three in $\mathbb{R}^3$, and suppose also that: (a) we already know a valid embedding for all vertices preceding v; (b) the edges $\{v-3, v\}$, $\{v-2, v\}$, $\{v-1, v\}$ are in E. This means that the embedding of v, denoted by x_v, belongs to the three spheres centered at $x_{v-3}, x_{v-2}, x_{v-1}$ with respective radii $d_{v-3,v}, d_{v-2,v}, d_{v-1,v}$. The intersection of three spheres in $\mathbb{R}^3$ can either be empty, or consist of exactly one point, or of exactly two points (see Fig. 1), or of uncountably many points [4] (see Fig. 2). Because we assume all vertices preceding v are already embedded prior to v, we know all their mutual distances. In particular, we

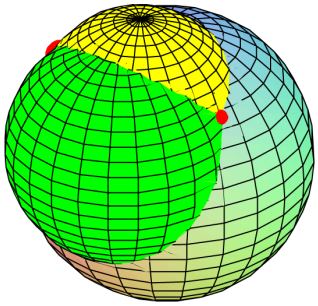

Figure 1: Three spheres intersect in exactly two points.

know $d_{v-3,v-1}, d_{v-3,v-2}, d_{v-2,v-1}$. As long as the strict triangular inequality $d_{v-3,v-1} < d_{v-3,v-2} + d_{v-2,v-1}$ holds, then the intersection can only have either one or two points, depending on whether the discriminant of a certain quadratic polynomial in x_v is zero or nonzero [4]: we call this the *finite sphere intersection property*. Because this discriminant can in general take any value in $\mathbb{R}_+$, and a singleton set has Lebesgue measure zero in $\mathbb{R}_+$, the sphere intersection has one point with probability 0 and two points with probability 1. We remark that the strict triangular inequality condition can only be checked once the predecessors of v have been embedded; this prevents us from recognizing aprioristically whether an MDGP instance conforms to this condition or not. We address this limitation by requiring that all 4-cliques of consecutive vertices are subgraphs of G. Thus, each 3-(sub)clique $\mathbf{K}_3^v = \{v-3, v-2, v-1\}$ is used to verify the strict triangular inequality, and the edges from $\mathbf{K}_3^v$ to v guarantee the finite sphere intersection property. If we proceed by embedding vertices iteratively this way we end up with a tree of possibilities where each embedded vertex gives rise to either one or two new positions for the embedding of the next vertex in the order. Since the first vertex triplet has only one possible embedding up to translations and rotations (because E contains a clique on the first four vertices), $|X|$ is finite with probability 1 [15, 17].

Several existing works exploit the finite sphere intersection property, but considering *four* (rather than three, as in our case) spheres [6, 7, 8, 41, 40, 5]; in general, the non-empty intersection of four spheres in $\mathbb{R}^3$ contains exactly *one* point: this follows because the system $\forall j \in \{1,2,3,4\}\ \|x_{v-j} - x_v\|^2 = d_{v-j,v}^2$ can be reduced to a square 3×3 linear system which is nonsingular under simple geometric regularity conditions. This ensures that the worst-case running time of an iterative algorithm based on this idea is $O(|V|)$. In [6] G is assumed to be a clique. In [7] this requirement is weakened: the so-called *geometric build-up algorithm* can only find a valid embedding if, for the current vertex, one can find at least four previously embedded adjacent vertices; depending on the instance, however, the algorithm in [7] may fail to find a valid embedding even

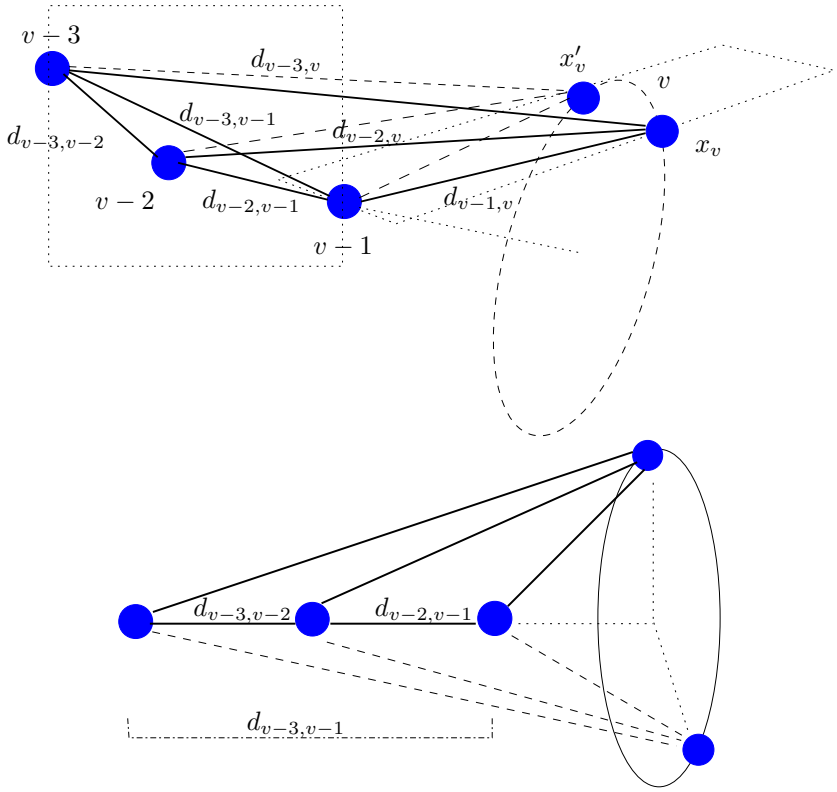

Figure 2: Locus of the intersection of three spheres: exactly two points (above) with $d_{v-3,v-1} < d_{v-3,v-2}+d_{v-2,v-1}$ and uncountably many (below) with $d_{v-3,v-1} = d_{v-3,v-2}+d_{v-2,v-1}$.

if one exists. In [40] the geometric build-up algorithm is modified to deal with some restricted types of measurement errors in the data. In [8] the finite sphere intersection property is introduced in the framework of wireless sensor networks.

Naturally, requiring known distances to four previously embedded adjacent vertices limits the extent of the iterative embedding algorithm to instances with relatively dense graphs. Because distances are usually hard to obtain (this is true for both molecules and sensor networks), an effort should be made in order to weaken this requirement. Although similar concepts were already known in rigidity [36], the first work providing an iterative discrete search algorithm for the MDGP that only requires *three* (rather than four) previously embedded adjacent vertices is [15, 17]. Other methods based on this weaker assumption are given in [23, 3, 42]. The following defines a subclass of MDGP instances conforming to these weaker assumptions [15, 17].

> Discretizable Molecular Distance Geometry Problem (DMDGP). Given a nonnegatively weighted simple undirected graph $G = (V, E, d)$,

an order $<$ on V and a mapping $x' : \{1, 2, 3\} \to \mathbb{R}^3$ such that:

1. x' is a valid embedding of $G[\{1, 2, 3\}]$ (START)
2. G contains all 4-cliques of $<$-consecutive vertices as induced subgraphs (DISCRETIZATION)
3. $\forall v \in V$ of rank greater than 3, $d_{v-3,v-1} < d_{v-3,v-2} + d_{v-2,v-1}$ (STRICT TRIANGULAR INEQUALITIES),

is there a valid embedding x of G in $\mathbb{R}^3$ extending x'?

We remark that the formal definition of the DMDGP introduces an order on V as an essential part of the input data; this marks a fundamental difference between [15, 17] and [23, 3, 42]. We shall discuss this further in Sect. 4.

2.1. Problem complexity

While it is clear that DMDGP $\subset$ MDGP, the DMDGP does not include any of the **NP**-hard classes described in Sect. 1.3, so restriction cannot be used to establish its **NP**-hardness. An explicit reduction from the SUBSET-SUM problem to the DMDGP was, however, provided in [15, 17].

2.2. Branch-and-Prune framework

We describe the Branch-and-Prune (BP) algorithm for solving the DMDGP [15, 17, 23]. The version given here is recursive (for clarity); it is also parameterized so that its variants, described in the rest of this paper, can be presented as configurations or simple modifications of Alg. 1. We recall that, given $G = (V, E)$ and $U \subseteq V$, $G[U]$ denotes the subgraph of G induced by U. For $v \in V$, $N(v) = \{u \in V \mid \{u, v\} \in E\}$ is set of vertices adjacent to v. We denote by $S^{K-1}(y, r)$ the sphere in $\mathbb{R}^K$ (where $K = 3$) centered at y with radius r.

The BRANCHANDPRUNE call has five arguments: the weighted simple undirected graph $G = (V, E, d)$ given as part of the DMDGP instance, a current vertex v being embedded, a subset $U \subseteq N(v)$ with $|U| = K$ (where K is the dimension of the embedding space), a valid embedding x' of a subgraph of G containing $G[U]$, and the set X of valid embeddings of G currently found. The recursion starts with the call BRANCHANDPRUNE$(G, 4, \{1, 2, 3\}, y, \emptyset)$ where y is the valid embedding of $\{1, 2, 3\}$ given as part of the DMDGP instance.

The BP algorithm shown in Alg. 1 builds a binary search tree whose nodes at level v represent possible spatial positions p for the vertex v. Whenever the test in Step 5 for validity of an embedding fails, the branch of p is pruned; pruning techniques are discussed in Sect. 2.2.4.

Theorem 1 ([15, 17]). *At termination of Alg. 1, X contains all valid embeddings of G extending x'.*

Algorithm 1 The Branch-and-Prune algorithmic framework.

BranchAndPrune(G, v, U, x', X):
Let P be the intersection of the K spheres $S^{K-1}(x'_u, d_{uv})$ for $u \in U$
for $p \in P$ **do**
  Extend the current embedding to $x = (x', p)$
  if x is a valid embedding of $G[\{1, \ldots, v\}]$ **then**
    if (v is the last vertex) **then**
      Append x to X
    else
      Let $U' = (U \setminus \{\min U\}) \cup \{v\}$
      BranchAndPrune(G, $v + 1$, U', x, X)
    end if
  end if
end for

2.2.1. Completeness

The BP algorithm generates a search tree. For each leaf node of this tree, the unique path to the root node encodes an embedding of G. By Thm. 1, the unique paths from each leaf node at level $|V|$ encode all valid embeddings of G extending x'. We remark that the BP can be stopped after the first valid embedding has been found when just one solution of the DMDGP is needed. It can also be allowed to proceed until all valid embeddings have been identified. This makes the BP algorithm complete.

2.2.2. Algorithmic complexity

In the worst case, when no pruning occurs, $|P| = 2$ at each iteration, which means that the search tree is a full binary tree. This makes the BP worst-case complexity exponential in $|V|$.

2.2.3. Performance

In order to assess the empirical behaviour of the BP algorithm we measure its efficiency in terms of seconds of user CPU time, and its reliability in terms of the Largest Distance Error (LDE):

$$\frac{1}{|E|} \sum_{\{u,v\} \in E} \frac{|\|x_u - x_v\| - d_{uv}|}{d_{uv}}. \tag{3}$$

The computational results shown in [15, 17] are markedly different from most continuous approaches: they scale up with instance size considerably better both in terms of CPU time and reliability. On the `1epw` PDB [1] instance, for example, which has 3861 atoms and 35028 distances, the BP took 0.25s to find all solutions, and yielded an LDE of 4×10^{-12}, which is very close to the zero LDE of an exactly valid embedding (we remark that all computations are carried out in floating point, so attaining an LDE of value *exactly* zero is practically impossible). By comparison, DGSOL [28] took 2038s and produced

an embedding with an LDE value around 0.5. This instance is not an isolated case: the BP consistently outperforms all continuous approaches we have tested [28, 14, 24].

In very recent work [27] we argue that on average protein instances the BP search tree has bounded width, thus yielding a polynomial-time algorithm; this makes the BP efficient.

As for the reliability, our implementation employs several devices in order to limit the propagation of floating point errors given by the computation of P, from the choice of appropriate vertex orders minimizing the range of values taken by the entries in the distance matrix (Sect. 4) to the exploitation of repeated vertices (Sect. 5), for which the zero distance between a vertex and its repetition is used as verification device for the embedding of nearby vertices. This makes the BP reliable.

2.2.4. Pruning the BP search tree

In this section we use the notation of Alg. 1. Pruning out infeasible branches of the BP search tree reduces the CPU time taken by the BP algorithm. Consider the point $p \in P \subseteq \mathbb{R}^K$ embedding vertex v (line 2 in Alg. 1): if p extends x' to a valid embedding $x = (x', p)$ of $G[\{1, \ldots, v\}]$ then p is *feasible*; otherwise it is *infeasible*. In the latter case, the whole sub-tree rooted at p can be pruned.

The most natural pruning test at the iteration when the BP places vertex v is to consider the vertex subset $\bar{U} = \{u \in V \mid u < v \wedge u \in N(v) \wedge u \notin U\}$. Vertices in $\bar{U}$ provide distances to v. Since $u < v$ for $u \in \bar{U}$, such vertices will already have been embedded in previous BP iterations; however, since $u \notin U$, these vertices have not been used to compute P. Thus their positions x_u can be matched against x_v to check for consistency of x_v. If $\|x_u - x_v\| \neq d_{uv}$, then (x', x_v) is not a valid embedding of G and the BP node encoding x_v can be pruned from the search tree. In practical implementation, the condition on which we prune a BP node is $|\|x_u - x_v\| - d_{uv}| > \varepsilon$, for a constant tolerance $\varepsilon > 0$ [15, 17, 23]. We shall call this pruning device *Direct Distance Feasibility* (DDF).

Another pruning device that can be used during the discrete search is based on the point-to-point Dijkstra shortest-path searches on Euclidean graphs [19]. Consider the vertices u, v, w with $u < v < w$ such that $\{u, w\} \in E$, i.e. the distance d_{uw} is known. Suppose that a position for the vertex u is already available, and that the feasibility of the node x_v needs to be verified. Let $D(v, w)$ be an upper bound to the distance $\|x_v - x_w\|$ for all possible valid embeddings. Then, if $\|x_u - x_v\| > d_{uw} + D(v, w)$ holds, the node x_v can be pruned [19] because the triangular inequality is negated. A valid upper bound $D(v, w)$ can be computed by finding the shortest path between the vertex v and the vertex w in G. We call this pruning device *Dijkstra Shortest Path* (DSP).

Computational experiments showed that the DSP detects infeasible embeddings sooner than the DDF, but it is also more computationally expensive. From a worst-case complexity point of view, the DDF is $O(1)$, whereas in a naive implementation the DSP is $O(|V|^2)$. Pre-computing all shortest paths in G in $O(|V|^3)$ reduces the DSP computation to a look-up operation on a table

of $|V|^2$ entries. Using a hash table, this yields an average $O(1)$ query time, but tests in this sense revealed that in practice the DDF strikes a better practical tradeoff between pruning efficacy and computational cost.

Other pruning devices could be conceived in the application of the DMDGP to protein structure determination using NMR data. Proteins are composed by a main chain of amino acids (the *backbone*) to which are attached some *side chains*. We can model atoms (i.e., vertices) using their van der Waals radii [39]: if the two atoms are not bound, they should not be embedded at points with shorter distance than the threshold given by van der Waals radii. Naturally, such thresholds depend on the kind of atoms involved. Moreover, when considering DMDGPs restricted to backbone atoms only, an auxiliary problem could be solved during the search. Every time a C_α carbon is placed, the conformation of the side chain attached to the carbon could be found by solving a Side Chain Placement Problem (SCPP) [37]. If such a problem has no solutions, then the atomic position for the C_α carbon is deemed infeasible.

2.3. Cardinality of the solution set

It was empirically observed that for most DMDGP instances, BP always finds a number of solutions that is a power of two [23]. Counterexamples to this conjecture are given in Lemma 5.1 in [15, 17] and in Sect. 6 in [26]. It was shown in [26] that, for the DMDGP, $|X|$ is a power of two with probability 1.

2.4. Overcoming practical limitations

Computational experiments (see for example [15, 17, 23, 19]) showed that the BP algorithm, when employing the pruning device DDF only, is very efficient in finding the whole set of solutions for DMDGPs. In at most a few seconds of user CPU time on a standard computer all the possible valid embeddings for G can be identified. The DMDGP, however, is an inaccurate model of the PSNMR, which is our main target application.

2.4.1. Interval distances

NMR experiments cannot provide exact distances, but only a lower and an upper bound to these distances. As a consequence, for each distance, an interval is generally available in which the actual distance value is contained. This makes the discretization process much more complex. While the pruning device DDF, for example, can be trivially adapted for interval data [29], the generation of the binary tree of solutions may require the computation of the intersection of three spherical shells [32]. In other words, interval distances cannot natively be used to satisfy the Discretization axiom, but they can be used effectively to prune the BP search tree.

Suppose we need to find the possible positions for the vertex v. If $d_{v-3,v}$, $d_{v-2,v}$ and $d_{v-1,v}$ are represented by the intervals $[d^L_{v-3,v}, d^U_{v-3,v}]$, $[d^L_{v-2,v}, d^U_{v-2,v}]$ and $[d^L_{v-1,v}, d^U_{v-1,v}]$, three spherical shells can be defined, which are centered in x_{v-3}, x_{v-2} and x_{v-1}, have inner radii $d^L_{v-3,v}$, $d^L_{v-2,v}$ and $d^L_{v-1,v}$, and outer radii $d^U_{v-3,v}$, $d^U_{v-2,v}$ and $d^U_{v-1,v}$, respectively. Representing *arbitrary* spherical shell

intersections in function of distance intervals by means of finite data structures does not seem an easy task. In [18] we propose a strategy to deal with this problem with a further (realistic) discretization assumption.

2.4.2. Distances between hydrogens

Another important issue is related to the enforcement of the DISCRETIZATION requirement in DMDGP instances arising from proteins. DISCRETIZATION requires the availability of a certain number of distances, whereas NMR experiments can usually only estimate short range distances (no larger than 4Å or 5Å, depending on the NMR machinery). Moreover, generally, only distances between hydrogen atoms are available from NMR experiments [39].

We address this problem from two points of view. In Sect. 4 we describe an automatic method to find best vertex orders to satisfy DISCRETIZATION. A second strategy, addressing the limitation in hydrogen-related distances posed by the NMR, is discussed in Sect. 5: a hand-crafted vertex order satisfying DISCRETIZATION is defined for the hydrogen atoms of the protein backbones, which are placed first; the other backbone atoms (mainly carbons and nitrogens) are placed in a second stage using an auxiliary DMDGP instance.

3. The Discretizable Distance Geometry Problem

Although our driving application is to embed proteins in 3D, other applications of graph embedding (wireless sensor networks, graph drawing) require embeddings in Euclidean spaces of varying dimensions. Since the finite sphere intersection property also holds in Euclidean spaces of arbitrary dimensions, we discuss two variants of the DMDGP requiring embeddings in $\mathbb{R}^K$.

The DGP, which generalizes the MDGP to a Euclidean space of arbitrary dimension K, asks for a valid embedding of G in $\mathbb{R}^K$. The generalization of the DMDGP to $\mathbb{R}^K$ replaces triplets of immediate adjacent predecessors with K-tuples of adjacent (but not necessarily immediate) predecessors. Furthermore, strict triangle inequalities are replaced with strict simplex inequalities [2]. Strict triangle inequalities ensure that the three predecessors in the DMDGP statement are not collinear; in other words, they ensure that the 2-simplex defined by the predecessors has nonzero volume. Strict simplex inequalities generalize this idea. For a set $U = \{x_i \in \mathbb{R}^{K-1} \mid i \leq K\}$ of points in $\mathbb{R}^{K-1}$, let D be the symmetric matrix whose (i,j)-th component is $\|x_i - x_j\|^2$ for all $i, j \leq K$ and let D' be D bordered by a left $(0, 1, \ldots, 1)^\top$ column and a top $(0, 1, \ldots, 1)$ row (both of size $K+1$). Then the Cayley-Menger formula [2, 12] states that the volume $\Delta_{K-1}(U)$ of the $(K-1)$-simplex on U is given by

$$\Delta_{K-1}(U) = \sqrt{\frac{(-1)^K}{2^{K-1}((K-1)!)^2}|D'|}. \tag{4}$$

The strict simplex inequalities are given by $\Delta_{K-1}(U) > 0$. For $K = 3$, these reduce to strict triangle inequalities. We remark that only the distances of the

simplex edges are necessary to compute $\Delta_{K-1}(U)$, rather than the actual points in U; the needed information can be encoded as a complete graph $\mathbf{K}_K$ on K vertices with edge weights as the distances. This implies that $\Delta_{K-1}(U)$ is well defined also if U is a set of *vertices* of V (instead of points in $\mathbb{R}^{K-1}$) as long as $G[U] = \mathbf{K}_K$. We also let $[K] = \{1, \ldots, K\}$.

3.1. From immediate to adjacent predecessors

For intersections of K appropriately defined spheres to yield at most 2 points, the centers need not necessarily be *immediate* predecessors, as the DMDGP require. To embed a vertex v using the embedding of vertices before v in the order, it suffices that there are at least K adjacent predecessors of v. Because of this, we can relax Discretization and define a larger class of discretizable instances [30]. For $v \in V$, if V is ordered let $\gamma(v)$ be the set of predecessors of v.

> Discretizable Distance Geometry Problem (DDGP). Given a positive integer K, a nonnegatively weighted simple undirected graph $G = (V, E, d)$, an order $<$ on V and a mapping $x' : [K] \to \mathbb{R}^K$ such that:
>
> 1. x' is a valid embedding of $G[[K]]$ (Start)
> 2. $\forall v \in V \setminus [K]$ $(|N(v) \cap \gamma(v)| \geq K)$ (Discretization)
> 3. $\forall v \in V \setminus [K]$ $\exists U_v \subset N(v) \cup \gamma(v)$ $(G[U_v] = \mathbf{K}_K \wedge \Delta_{K-1}(U_v) > 0)$ (Strict Simplex Inequalities),
>
> is there a valid embedding x of G in $\mathbb{R}^K$ extending x'?

Again, we denote DDGP instances with fixed K by DDGP_K. By Discretization and Strict Simplex Inequalities, the DDGP can be solved using BP (Alg. 1) — just replace Step 9 with "let $U' = U_{v+1}$". We remark that the results of Sect. 2.3 do not apply to the DDGP.

Requiring $G[U_v] = \mathbf{K}_K$ is a strong condition. In practice we usually relax $G[U_v] = \mathbf{K}_K \wedge \Delta_{K-1}(U_v) > 0$ to simply $|U_v| = K$. This does not necessarily ensure that the instance can be discretized. However, because BP is an iterative algorithm on the order of V, the positions of all vertices in U_v are known before embedding v, which implies that the Strict Simplex Inequalities condition can be verified by the BP algorithm itself.

3.2. Problem complexity

Because the DDGP contains all DMDGP instances as a subproblem, it is **NP**-hard by restriction. We remark that the Discretization condition makes this problem the "smallest" **NP**-hard problem with respect to K: replacing K by $K+1$ would yield instances having a K-trilateration order [8], for which the embedding problem is in **P**. This can be seen by restricting the set P in Alg. 1 such that $|P| \leq 1$: the BP search tree width would then be bounded by 1, which means that the BP would have a worst-case running time $O(L|V|)$, where L is the complexity of finding P.

4. Discretization orders

In the family of problems that the BP can solve, i.e., DMDGP and DDGP, an order $<$ on the vertex set V is always given, guaranteeing that the edges in E satisfy the DISCRETIZATION requirement. In practice, DMDGP instances coming from proteins are endowed with their natural *backbone order*, which may not satisfy DISCRETIZATION. In this section we discuss the problem of finding a good order or determining that one such order does not exist.

> 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 $|N(v) \cap \gamma(v)| \geq K$.

We note that the DVOP does not verify whether the order satisfies the STRICT SIMPLEX INEQUALITIES requirement. This is because the set of distance matrices yielding a Cayley-Menger determinant (see Eq. (4)) having value exactly zero has Lebesgue 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.

Intuitively, the larger the sets $N(v) \cap \gamma(v)$ (for v of rank exceeding K), the smaller the sets P in Alg. 1 for early ranks will be, and the better the BP will perform. Sets of adjacent predecessors of size exactly K ensure that $|P| \leq 2$, but more pruning distances 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, 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 $\rho(v) > K$, $|N(v) \cap \gamma(v)|$ is maximum and exceeds K.

The ODVOP is a multi-objective maximization problem, whose objective function vector is $(|N(v) \cap \gamma(v)| \mid v \in V \ (\rho(v) > K))$. We prove in [13] that all DVOP solutions are in the Pareto set of the ODVOP. In practice, however, we 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.

NP-completeness of the DVOP notwithstanding, when K is fixed, the DVOP is in **P**: for each possible K-clique of G, we greedily build the order on V by choosing large sets of adjacent predecessors earliest. Because K is typically much smaller than $|V|$, and in practical instances arising from proteins K is really fixed to 3, this algorithm performs fast enough to be able to determine useful orders as a pre-processing step to the BP.

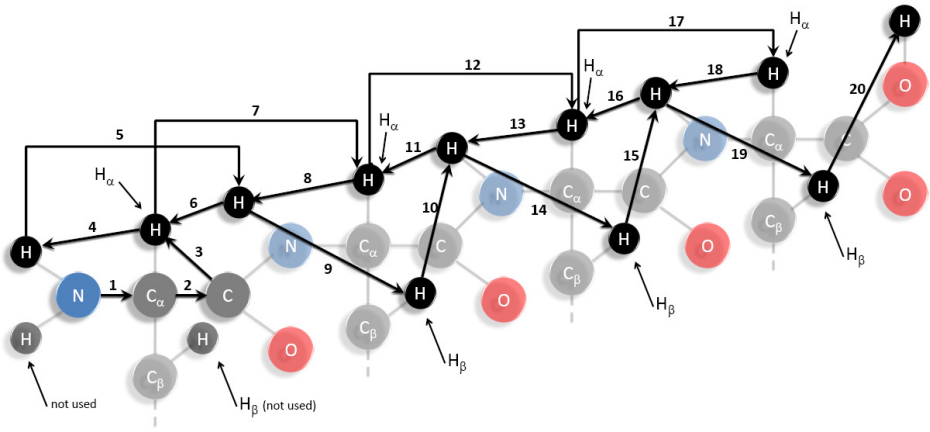

Figure 3: The artificial backbone of hydrogens satisfying the DMDGP requirements. The order is given in the arrow labels. Not all the edges of the graph are actually shown.

As a testbed for DVOP-based techniques, we considered a subset of $DDGP_3$ instances from the PDB [1] where we kept all inter-atomic distances up to 5.5Å. With such a low threshold, the backbone order is not valid w.r.t. DISCRETIZATION. Using the DVOP, we were able to embed all 18 protein graphs (from 90 to 2259 backbone atoms) in around 21 seconds of user CPU time for the whole test set (this includes solving the DVOP, which took 1/40th of the DDGP solution time on average), with average accuracy 10^{-10} measured in LDE (see Eq. (3)); this confirms the reliability of the BP. By comparison, DGSOL [28] in its standard configuration took 800s and yielded an average accuracy of 5×10^{-1}.

5. An artificial backbone of hydrogens

Our first attempt to consider NMR data, which usually provide distances between hydrogen atoms only if closer than a given threshold, has been presented in [20, 21, 31]. We defined an order for the hydrogens related to protein backbones which allows us to satisfy DISCRETIZATION. Figure 3 shows the proposed vertex order, indicated by the black arrows in the picture and by their labels (showing a progressive index), which we named *artificial backbone of hydrogens*. Note that this particular order considers the same atom more than once. Because of this, the relative distances between atoms farther in sequence are reduced, and a new kind of distance is introduced: the distance equal to zero existing between two copies of the same atom (obviously placed in the same spatial point).

The complexity of computing P in Alg. 1 does not change, because the second copy of an atom can only be placed in the same place as the first copy. Thus, no branching occurs in correspondence with duplicated atoms, and the worst-case complexity of the BP variant exploiting orders with repetitions in

still exponential in $|V|$. In [22], we showed that, because of steric constraints due to the particular structure of protein backbones, all distances necessary to guarantee DISCRETIZATION can be obtained by NMR. Once the problem is discretized and solved by the BP algorithm limited to hydrogen atoms, the remaining backbone atoms, and in particular the sequence of atoms (N, C_α, C) can be obtained by solving another MDGP. We proved that this MDGP is easy to solve, because assumptions stronger than the ones needed for the DMDGP are satisfied [20]. In particular, each other backbone atom N, C_α, C has at least 4 adjacent predecessors. As a consequence, the order is a trilateration order and the instance can be embedded in polynomial time [8].

6. Implementation and parallelization

`MD-jeep` is an implementation of Alg. 1 in the C programming language [35]. It is distributed under the GNU General Public License (v.2) and it can be downloaded from `http://www.antoniomucherino.it/en/mdjeep.php`. `MD-jeep` accepts as input a list of distances in a text file with a predefined format, and returns PDB files containing the solutions to the problem as output. The PDB is a standard format for storing molecular conformations [1], which is compatible with many other software packages for molecular management and visualization. For example, two views, obtained using `RasMol` (`http://www.rasmol.org/`), of one of the solutions found by `MD-jeep` are given in Fig. 4.

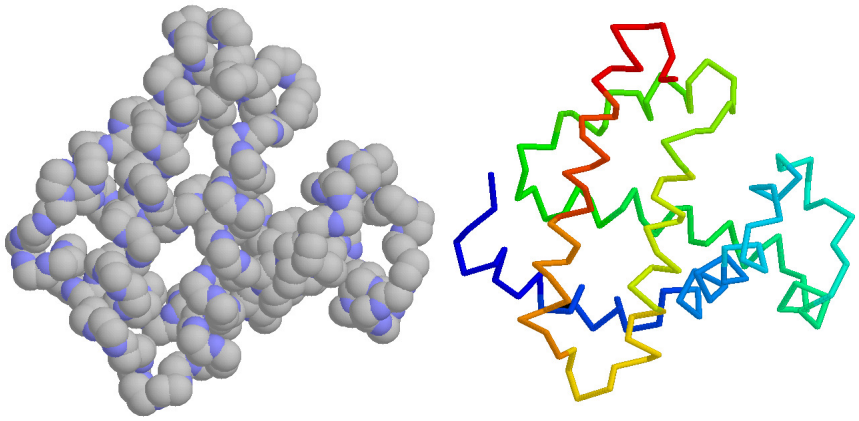

Figure 4: Two different graphical representations of the embeddings obtained by `MD-jeep`.

6.1. Parallel BP

We are also working on parallel implementations of the BP algorithm [33, 34] for the DMDGP. The basic idea is to exploit the DMDGP order to partition

the instance graph into subgraphs, whose embeddings can be found independently by separate processes and then recombined, similar to "decomposition-recombination" technique in Computer Aided Design (CAD) [11]. Embedding each subgraph requires a call to a sequential BP algorithm, and the recombination is carried out by the master process.

Let us suppose that the number n of vertices related to a given instance graph is divisible by the number p of processes involved in the parallel computation. Then p induced subgraphs $G_i = G[V_i]$ can be defined for all $i \leq p$ by setting:

$$V_i = \left\{1 + \frac{(i-1)n}{p}, \ldots, 3 + \frac{in}{p}\right\}.$$

For all $i \leq p$ we define $E_i = E[V_i] = \{\{u, v\} \in E \mid u, v \in V_i\}$. This partition of V guarantees that each G_i is a DMDGP instance if and only if G is.

We remark that $\bar{E} = \bigcup_{i \leq p} E_i$ does not cover E; in particular, edges $\{u, v\} \in E$ with $u \in V_i, v \in V_j$ for $i \neq j$ do not belong to $\bar{E}$. As a consequence, the corresponding distances d_{uv} are not used while the single processes work on the subgraphs G_i. However, they can be exploited later after the communication phase, when the local solutions found by the single processes are combined together in order to find the final set of solutions to the original instance.

The communication phase is implemented by following the classical cascade schema, so that only $\log_2 p$ communications are required to make p processes exchange the partial embeddings found by the sequential calls to BP (we suppose that p is a power of 2). Each partial embedding is coded by a sequence of binary variables. In order to reduce the time needed for the communication, each binary variable is stored in a single bit of an array of integer numbers. Each set of partial embeddings can be used for defining the local binary tree of solutions, which can be represented by graphs $T_k = (W_k, H_k)$, where vertices in W_k represent atomic positions, and edges in H_k connect vertices related to consecutive atomic positions. We employ the following procedure for combining the sets of partial embeddings found by two processes k_1 and $k_2 = k_1 + 1$. Let $T_{k_1,k_2} = (W_{k_1,k_2}, H_{k_1,k_2})$ be the graph which is the combination between T_{k_1} and T_{k_2}. The vertex set W_{k_1,k_2} is defined so that it contains all the vertices in W_{k_1} and W_{k_2}. The vertices in W_{k_2} are duplicated as many times as the number of leaf vertices in W_{k_1}, and new labels are assigned to them. The edge set H_{k_1,k_2} is computed similarly, and, for each leaf vertex v_l of W_{k_1}, a new edge is added between v_l and the various copies of the first vertex of W_{k_2}. If this procedure is performed recursively considering all the graphs T_k, then the final tree of solutions, representing the final set of embeddings, can be reconstructed. Distances related to atoms previously assigned to different processes can be used for pruning branches of the final tree for removing infeasible solutions.

Computational experiments (refer to [34] for more details) showed the efficiency of the parallel approach; the CPU time gain ratio between successive processor configurations (e.g., 1 against 2, 2 against 4 and so on) decreases as p increases (in a few cases, executions with more processes actually took slightly longer). This is due to the fact that, as p increases, the subgraphs assigned

to each process get smaller, whereas the number of edges in $E \setminus \bar{E}$ increases. As a consequence, the calls to the (sequential) BP process on each subgraph tend to be less expensive than the master process that builds the BP tree for the whole graph. Parallel implementations overcoming this issue are currently under study.

7. Conclusion and future work

This paper gives an overview of the Discretizable Molecular Distance Geometry Problem, which offers a good model for finding protein structures with NMR data. We discussed variants, complexity, solution algorithms and extensions to deal with protein-specific features, such as limitations on the type of atoms that NMR usually provides information on.

On a short term, future work concerns the following topics: treatment of errors in the NMR data; polynomiality of the BP in the average case; exploitation of the BP tree symmetries. Longer term future work includes: the integration of the side chain embeddings; discovering unknown protein structures from real NMR data; synthesizing a BP-based integrated method to solve the PSNMR problem; looking for more applications (notably in embedding whole molecular complexes).

One notable open theoretical question is whether the DGP_K is in **NP** for $K > 1$. The embedding that certifies a YES instance usually involves real numbers even though the instance data is rational (or even integer). As the embeddings solve a system of polynomials of second degree in several variables, it is easy to show that only algebraic numbers, rather than transcendental ones, are needed to express the components of each vector in the embedding. Thus, a finite precise symbolic representation for the embeddings is readily available, for example as the set of minimal polynomials having all the required algebraic numbers as roots. Whether all such numbers can be encoded by means of expressions whose length is polynomial in the instance size is as yet unclear.

Acknowledgments

We are grateful to Jon Lee, Audrey Lee-St. John, Benoît Masson and Maxim Sviridenko for co-authoring some of the papers we wrote on different facets of this topic. We also wish to thank Thérèse Malliavin, Leandro Martínez and Michael Nilges for useful discussions, and an anonymous referee for carefully checking the manuscript. This work was partially supported by the Brazilian research agencies FAPESP and CNPq.

References

[1] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The protein data bank. *Nucleic Acid Research*, 28:235–242, 2000.

[2] L. Blumenthal. *Theory and Applications of Distance Geometry*. Oxford University Press, Oxford, 1953.

[3] R.S. Carvalho, C. Lavor, and F. Protti. Extending the geometric build-up algorithm for the molecular distance geometry problem. *Information Processing Letters*, 108:234–237, 2008.

[4] I.D. Coope. Reliable computation of the points of intersection of n spheres in $\mathbb{R}^n$. *Australian and New Zealand Industrial and Applied Mathematics Journal*, 42:C461–C477, 2000.

[5] R. Davis, C. Ernst, and D. Wu. Protein structure determination via an efficient geometric build-up algorithm. *BMC Structural Biology*, 10(Suppl 1):S7, 2010.

[6] Q. Dong and Z. Wu. A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances. *Journal of Global Optimization*, 22:365–375, 2002.

[7] 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.

[8] T. Eren, D.K. Goldenberg, W. Whiteley, Y.R. Yang, A.S. Morse, B.D.O. Anderson, and P.N. Belhumeur. Rigidity, computation, and randomization in network localization. *IEEE Infocom Proceedings*, pages 2673–2684, 2004.

[9] M.R. Garey and D.S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-Completeness*. Freeman and Company, New York, 1979.

[10] H. Gunther. *NMR Spectroscopy: Basic Principles, Concepts, and Applications in Chemistry*. Wiley, New York, 1995.

[11] C. Jermann, G. Trombettoni, B. Neveu, and P. Mathis. Decomposition of geometric constraint systems: a survey. *International Journal of Computational Geometry and Applications*, 16:379–414, 2006.

[12] Y. Jiao, F.H. Stillinger, and S. Torquato. Geometrical ambiguity of pair statistics I. point configurations. Technical Report 0908.1366v1, arXiv, 2009.

[13] C. Lavor, J. Lee, A. Lee-St. John, L. Liberti, A. Mucherino, and M. Sviridenko. Discretization orders for distance geometry problems. *Optimization Letters*, DOI: 10.1007/s11590-011-0302-6.

[14] C. Lavor, L. Liberti, and N. Maculan. Computational experience with the molecular distance geometry problem. In J. Pintér, editor, *Global Optimization: Scientific and Engineering Case Studies*, pages 213–225. Springer, Berlin, 2006.

[15] C. Lavor, L. Liberti, and N. Maculan. The discretizable molecular distance geometry problem. Technical Report q-bio/0608012, arXiv, 2006.

[16] C. Lavor, L. Liberti, and N. Maculan. Molecular distance geometry problem. In C. Floudas and P. Pardalos, editors, *Encyclopedia of Optimization*, pages 2305–2311. Springer, New York, second edition, 2009.

[17] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino. The discretizable molecular distance geometry problem. *Computational Optimization and Applications*, DOI: 10.1007/s10589-011-9402-6.

[18] C. Lavor, L. Liberti, and A. Mucherino. The *i*BP algorithm for the discretizable molecular distance geometry problem with interval data. *Journal of Global Optimization*, submitted.

[19] C. Lavor, L. Liberti, A. Mucherino, and N. Maculan. On a discretizable subclass of instances of the molecular distance geometry problem. In D. Shin, editor, *Proceedings of the 24th Annual ACM Symposium on Applied Computing*, pages 804–805. ACM, 2009.

[20] C. Lavor, A. Mucherino, L. Liberti, and N. Maculan. An artificial backbone of hydrogens for finding the conformation of protein molecules. In *Proceedings of the Computational Structural Bioinformatics Workshop*, pages 152–155, Washington D.C., USA, 2009. IEEE.

[21] C. Lavor, A. Mucherino, L. Liberti, and N. Maculan. Computing artificial backbones of hydrogen atoms in order to discover protein backbones. In *Proceedings of the International Multiconference on Computer Science and Information Technology*, pages 751–756, Mragowo, Poland, 2009. IEEE.

[22] C. Lavor, A. Mucherino, L. Liberti, and N. Maculan. On the computation of protein backbones by using artificial backbones of hydrogens. *Journal of Global Optimization*, 50:329–344, 2011.

[23] 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.

[24] L. Liberti, C. Lavor, N. Maculan, and F. Marinelli. Double variable neighbourhood search with smoothing for the molecular distance geometry problem. *Journal of Global Optimization*, 43:207–218, 2009.

[25] L. Liberti, C. Lavor, A. Mucherino, and N. Maculan. Molecular distance geometry methods: from continuous to discrete. *International Transactions in Operational Research*, 18:33–51, 2010.

[26] L. Liberti, B. Masson, C. Lavor, J. Lee, and A. Mucherino. On the number of solutions of the discretizable molecular distance geometry problem. Technical Report 1010.1834v1[cs.DM], arXiv, 2010.

[27] L. Liberti, B. Masson, C. Lavor, and A. Mucherino. Branch-and-prune trees with bounded width. In G. Nicosia and A. Pacifici, editors, *Proceedings of CTW 2011*, Rome, submitted.

[28] J.J. Moré and Z. Wu. Global continuation for distance geometry problems. *SIAM Journal of Optimization*, 7(3):814–846, 1997.

[29] A. Mucherino and C. Lavor. The branch and prune algorithm for the molecular distance geometry problem with inexact distances. In *Proceedings of the International Conference on Computational Biology*, volume 58, pages 349–353. World Academy of Science, Engineering and Technology, 2009.

[30] A. Mucherino, C. Lavor, and L. Liberti. The discretizable distance geometry problem. *Optimization Letters*, in revision.

[31] A. Mucherino, C. Lavor, L. Liberti, and N. Maculan. On the definition of artificial backbones for the discretizable molecular distance geometry problem. *Mathematica Balkanica*, 23:289–302, 2009.

[32] A. Mucherino, C. Lavor, L. Liberti, and N. Maculan. Strategies for solving distance geometry problems with inexact distances by discrete approaches. In S. Cafieri, E. Hendrix, L. Liberti, and F. Messine, editors, *Proceedings of the Toulouse Global Optimization workshop*, pages 93–96, Toulouse, 2010.

[33] A. Mucherino, C. Lavor, L. Liberti, and E-G. Talbi. On suitable parallel implementations of the branch & prune algorithm for distance geometry. In *Proceedings of the Grid5000 Spring School*, Lille, France, 2010.

[34] A. Mucherino, C. Lavor, L. Liberti, and E-G. Talbi. A parallel version of the branch & prune algorithm for the molecular distance geometry problem. In *ACS/IEEE International Conference on Computer Systems and Applications (AICCSA10)*, Hammamet, Tunisia, 2010. IEEE conference proceedings.

[35] A. Mucherino, L. Liberti, and C. Lavor. `MD-jeep`: an implementation of a branch-and-prune algorithm for distance geometry problems. In K. Fukuda, J. van der Hoeven, M. Joswig, and N. Takayama, editors, *Mathematical Software*, volume 6327 of *LNCS*, pages 186–197, New York, 2010. Springer.

[36] B. Roth. Rigid and flexible frameworks. *American Mathematical Monthly*, 88(1):6–21, 1981.

[37] R. Santana, P. Larrañaga, and J.A. Lozano. Combining variable neighbourhood search and estimation of distribution algorithms in the protein side chain placement problem. In *Proc. of Mini Euro Conference on Variable Neighbourhood Search, Tenerife, Spain*, 2005.

[38] J.B. Saxe. Embeddability of weighted graphs in k-space is strongly NP-hard. *Proceedings of 17th Allerton Conference in Communications, Control and Computing*, pages 480–489, 1979.

[39] T. Schlick. *Molecular modelling and simulation: an interdisciplinary guide*. Springer, New York, 2002.

[40] A. Sit, Z. Wu, and Y. Yuan. A geometric build-up algorithm for the solution of the distance geometry problem using least-squares approximation. *Bulletin of Mathematical Biology*, 71:1914–1933, 2009.

[41] D. Wu and Z. Wu. 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.

[42] D. Wu, Z. Wu, and Y. Yuan. Rigid versus unique determination of protein structures with geometric buildup. *Optimization Letters*, 2(3):319–331, 2008.