The *interval* Branch-and-Prune algorithm for the Discretizable Molecular Distance Geometry Problem with inexact distances

Carlile Lavor[1], Leo Liberti[2], Antonio Mucherino[3]

[1] *Dept. of Applied Math. (IMECC-UNICAMP), State University of Campinas, 13081–970, Campinas - SP, Brazil*
Email:clavor@ime.unicamp.br

[2] *LIX, École Polytechnique, F-91128 Palaiseau, France*
Email:liberti@lix.polytechnique.fr

[3] *IRISA, University of Rennes 1, F-35000 Rennes, France*
Email:antonio.mucherino@irisa.fr

October 9, 2011

Abstract

The Distance Geometry Problem in three dimensions consists in finding an embedding in $\mathbb{R}^3$ of a given nonnegatively weighted simple undirected graph such that edge weights are equal to the corresponding Euclidean distances in the embedding. This is a continuous search problem that can be discretized under some assumptions on the minimum degree of the vertices. In this paper we discuss the case where we consider the full-atom representation of the protein backbone and some of the edge weights are subject to uncertainty within a given nonnegative interval. We show that a discretization is still possible and propose the *i*BP algorithm to solve the problem. The approach is validated by some computational experiments on a set of artificially generated instances.

1 Introduction

We consider the problem of determining a Euclidean embedding of a simple weighted graph, the so-called Distance Geometry Problem (DGP). This problem has at least three important applications: finding the three-dimensional conformation (the coordinates of all the atoms) of a molecule from a subset of inter-atomic distances found using Nuclear Magnetic Resonance (NMR) [11, 17]; finding the position of wireless sensors given some of the distances (estimated by monitoring the power needing to communicate with each sensor's neighbours) [4, 28]; and graph drawing (`www.graphdrawing.org`).

In this paper we consider the application to finding the three-dimensional conformation of proteins. In this case, this problem is usually referred to as Molecular DGP (MDGP). Proteins are important molecules which perform several functions in living beings. If their three-dimensional conformations are discovered, they are able to reveal the specific function that each protein is supposed to perform. A web database named Protein Data Bank (PDB) [1] is collecting all the three-dimensional conformations of proteins that scientists in the world have been able to obtain. To date, a rather small percentange of conformations on the PDB have been obtained through NMR experiments, where the corresponding MDGP has been solved by general-purpose continuous approaches for global optimization. The meta-heuristic Simulated Annealing [7, 23] is employed in most of the cases. However, various approaches for solving the MDGP have been proposed in the literature, and recent surveys can be found in [11, 17]. Other interesting works include, for example, [18] and [8], and others can be found in the edited book [25].

Let $G = (V, E, d)$ be a nonnegatively weighted simple undirected graph representing an instance of the MDGP. Vertices of G correspond to the atoms forming the molecule, and edges indicate if the

distance between the respective atoms is known or not. Since we focus our attention on proteins and on NMR experiments for obtaining estimates of inter-atomic distances, we are able to make the following assumptions, which will allow us to discretize the problem:

1. Inter-atomic distances corresponding to the set $E' \subset E$ of all (unordered) pairs of atoms separated by at most two covalent bonds will be represented by positive rational numbers, since bond lengths and bond angles can be considered fixed at their equilibrium values in protein molecules [27].
2. There exists a set $E'' \subset E$ of pairs of atoms separated by exactly three covalent bonds, for which it is possible to compute tight lower and upper bounds to the corresponding distances; these distances will be represented by intervals of rational numbers and the possible values will be represented by a discretized set of D values within this interval [24].
3. There exists a set $F \subset E$ of inter-atomic distances which can be estimated using NMR measurements. Note that NMR does not provide distances to all possible pairs of atoms: the atoms must be closer than a given distance threshold (usually set between 4Å and 5Å), and they are usually hydrogen atoms [27]. In addition to this, since these distances are not precise, they will be represented by positive rational intervals.

Distances corresponding to edges in F might be affected by experimental errors: in this paper we describe a method which only uses edges in $E' \cup E''$ in order to discretize the search space. Distances corresponding to F will only be used to verify feasibility of partial embeddings.

In general, embedding a general graph in Euclidean space requires a continuous search [17]. In this paper, we discretize the problem and we propose the *interval* Branch-and-Prune (iBP) algorithm, which is an extension of the algorithm given in [16]. We use Assumptions 1-2 to discretize the problem and Assumption 3 to prune out unlikely configurations. Note that only distances from $E' \cup E''$ are used for the discretization. The distances from NMR (in the subset F of E) are only used for pruning purposes. As a consequence, the new discrete domain of the problem is completely independent from experimental NMR data. As already remarked in our previous publications, the advantages in considering a discrete search, with respect to a continuous one, are: increased efficiency, increased solution accuracy and completeness (in the sense that all embeddings can be found).

The discretization of the search space is based on the observation that, in general, three spheres in $\mathbb{R}^3$ intersect in at most two points. This observation is also used to find graph embeddings in [2, 10, 29]. A similar observation for the intersection of three circles in $\mathbb{R}^2$ (which in general consists of at most one point) leads to a polynomial-time algorithm for the Sensor Network Localization Problem [4]. A technique for reliably computing such intersection points is given in [3].

This work, which continues the sequence of papers [15, 14, 10, 16, 19, 20, 21], moves an important step towards taking into account the characteristics peculiar to NMR data. In order to discretize the search space, we assumed (irrealistically) in [10, 16, 20] that all distances for pairs in $E' \cup E'' \cup F$ are known precisely. In [21] we only consider NMR distances referring to hydrogen pairs; consequently, we compute a partial 3D structure including hydrogens only, neglecting other atoms. In [14, 15] this limitation is removed by considering a linear algebra based method [29] for computing non-hydrogen positions. We also developed a strategy for managing wrong distances in [22]. We remark that only precise distances (rather than intervals) are considered in [15, 14, 21, 22].

Our first attempt to consider interval data has been presented in [19]. We assumed that the distances in F are defined by a lower and an upper bound, and we modified the pruning phase of the Branch-and-Prune (BP) algorithm [16] from a "by value" form to a "by interval" form. However, the distances needed for discretization were still supposed to be exact. We further observed that even a very low uncertainty on these distances is able to spoil the discretization process, in which case no solutions can be found.

In the present work, we address the latter phenomenon. Even though interval data are used, we will be able to mantain the discretization process. We propose three concurrent improvements towards

considering real NMR data: (a) we only consider NMR distances referring to hydrogen atoms; (b) we represent them by intervals; (c) we compute the positions of non-hydrogen atoms together with the ones of hydrogen atoms, thereby avoiding the numerical instabilities due to solving linear systems. The proposed *i*BP algorithm relies on a carefully hand-crafted atom sequence which exploits repetitions in order to make sure that for each atom being placed there are distances to three previously placed atoms. These distances guarantee that the discretization process can be applied independently from the considered instance and of the presence or not of interval represented distances. Preliminary results in this direction have been presented in [13]. The idea of exploiting certain vertex sequences in order to prove that certain graphs have rigid embeddings first appeared in [5]; see [6] for an extensive discussion.

The rest of this paper is organized as follows. In Sect. 2 we introduce notation, some main concepts, and give some preliminary definitions. In Sect. 3 we construct the protein backbone graph and a vertex sequence that allows the discretization of the search space. In Sect. 4 we propose the *interval* Branch-and-Prune (*i*BP) algorithm for the protein backbone graph using the order in the vertex sequence. In Sect. 5 we present some computational results. Sect. 6 concludes the paper.

2 Preliminary notions

We formally define in this section the decision problems discussed in the paper and briefly recall the original BP algorithm.

> Distance Geometry Problem in 3 dimensions (DGP_3). Given a nonnegatively weighted simple undirected graph $G = (V, E, d)$ where $d : E \to \mathbb{R}_+$, is there an embedding $x : V \to \mathbb{R}^3$ such that
> $$\forall \{u, v\} \in E \quad \|x(u) - x(v)\| = d(u, v)\,? \tag{1}$$

Since DGP_1 (where x maps into $\mathbb{R}$) is **NP**-complete [26] and contained in DGP_3, and it is not known whether DGP_3 is itself in **NP**, it follows that DGP_3 is **NP**-hard.

Notation. For a graph $G = (V, E)$ and a subset $V_0 \subseteq V$ we let $G[V_0]$ be the subgraph of G induced by V_0; for $v \in V$ we let $\delta_E(v) = \{u \in V \mid \{u, v\} \in E\}$ be the set of vertices adjacent to v (if there is no ambiguity we omit the E index). For an order $<$ on V and $v \in V$ we let $\gamma_<(v) = \{u \in V \mid u < v\}$ be the set of predecessors of v in the order $<$ and $\rho_<(v) = |\gamma_<(v)| + 1$ be the rank of v in the order $<$ (if there is no ambiguity we omit the $<$ index).

In [10, 12] we introduced a subclass of DGP_3 whose instances can be solved using a discrete search algorithm.

> Discretizable Molecular Distance Geometry Problem (DMDGP). Given a nonnegatively weighted simple undirected graph $G = (V, E, d)$ where $d : E \to \mathbb{R}_+$, a subset $V_0 \subseteq V$ and an order $<$ on V such that:
>
> - $V_0 = \{1, 2, 3\}$ and $G[V_0]$ is a clique (Start)
> - for all $v \in V \setminus V_0$ we have
>   1. $v-3$, $v-2$, $v-1 \in \delta(v) \cap \gamma(v)$ (Discretization)
>   2. $d(v-3, v-2) + d(v-2, v-1) > d(v-3, v-1)$ (Strict Triangular Inequalities),
>
> is there an embedding $x : V \to \mathbb{R}^3$ such that (1) holds ?

As in the MDGP instances, vertices of G correspond to the atoms forming the molecule and edges indicate if the distance between the respective atoms is known or not.

The DMDGP is **NP**-hard [10, 12] and its instances can be solved using the BP algorithm [16]: the first 3 vertices in V_0 can be embedded by Start; inductively, any vertex v of rank greater than 3 can be placed at the intersection of three spheres centered at $v-3, v-2, v-1$ with respective radii $d(v-3,v)$, $d(v-2,v)$, $d(v-1,v)$ by Discretization; this intersection consists of at most 2 points x'_v, x''_v by Strict Triangular Inequalities. This gives rise to a binary tree search whose leaves represent valid embeddings of G. Branches can be pruned using distances from v to vertices in $\delta(v) \cap \gamma(v)$ (other than the ones used for the discretization) that are incompatible with either x'_v or x''_v or both. This yields an extremely fast algorithm [16] which is also able to find all embeddings for a given graph (modulo rotations and translations).

We recently also proposed a generalization of the DMDGP to dimensions higher than 3, the Discretizable Distance Geometry Problem (DDGP) [20]. In the three-dimensional case (DDGP_3), the main difference with the DMDGP stands in the Discretization assumption. Instead of the three immediate predecessors of v, any vertices u, w and z having rank smaller than v can be considered. It follows that the DDGP_3 relies on assumptions which are weaker than the ones of the DMDGP. In particular, the Discretization assumption of the DDGP_3 does not reflect any feature of molecules or proteins, and therefore the DDGP_3 can be considered as a more generic problem that can be employed in other applications arising in fields different from biology. Since the DDGP_3 contains the DMDGP, and the DMDGP is **NP**-hard [10, 12], the DDGP_3 is also **NP**-hard.

An interesting subproblem which arose from the DMDGP and the DDGP_3 is the following. Given a graph G representing an instance of the DGP_3 for which the assumptions for the DMDGP or the assumptions for the DDGP_3 are not satisfied, can we sort the vertices of G such that the assumptions become satisfied? Since the DDGP_3 relies on weaker assumptions, we investigated the possibility to reorder the vertices of G for having the DDGP_3 assumptions satisfied, and we found an efficient solution method [9]. In that work, however, all the distances are considered to be exact, and this does not reflect real experimental data. An immediate extension of this problem could be: can we sort the vertices of G in order to have the assumptions satisfied (and so to discretize the problem) using only distances from E' and E''?

This is a very interesting problem that we will not try (at least in this work) to solve in general. We rather carefully hand-craft a sequence of atoms, related to protein backbones, having the desired features. Even though the assumptions for the DDGP_3 are weaker, we consider here the DMDGP, because its assumptions can be verified in a easier way by checking any picture drawing the considered sequence of atoms.

In order to facilitate our task, we also allow for repeated atoms in the sequence. This trick allows us to consider distances between copies of the same atom, that are naturally equal to 0, thus increasing the number of exact distances that can be considered. Obviously, since the same atom can be duplicated several times, the final sequence of atoms could have a length which is much larger than the original sequence of atoms. However, this increase in length is not reflected on the tree obtained by the discretization, because copies of an atom which has been already placed somewhere can only take one position. In other words, there is no branching on the tree in correspondence with duplicated atoms.

We remark that we are making no claim as to the necessity of vertex repetition: we have not proved that there is no order (without repetitions) on V with the desired properties. In fact we conjecture that quite the reverse holds, i.e. the existence of an order with repetitions might imply the existence of an order without repetitions satisfying the first two requirements of the DMDGP (or of the DDGP_3); in this sense, vertex repetitions are to be considered as merely a tool to facilitate a complex task.

3 A vertex order for protein backbones

3.1 Vertex orders with repetitions

We present in this section the definition of *repetition order* (re-order), and we will show later that our hand-crafted vertex order is a re-order.

Let $G = (V, E, d)$ be the nonnegatively weighted simple undirected graph associated to an instance of the DGP$_3$. The set of edges E can be partitioned into those edges $\{u, v\} \in E'$ for which $d(u, v)$ is a real nonnegative number, and those edges $\{u, v\} \in E''$ for which $d(u, v)$ is a finite set of points belonging to a positive rational interval $[d^L_{uv}, d^U_{uv}]$. We remark that, in the practice, we consider nonnegative rational numbers, but we formally define each graph G with real nonnegative weights. For notational simplicity, we assume for the rest of this section that $d(E'')$ is a set of intervals — we shall exploit their discretization in Sect. 4. Let $V' = V \cup \{0\}$.

3.1 Definition
A repetition order (re-order) is a sequence $r : \mathbb{N} \to V'$ with length $|r| \in \mathbb{N}$ (for which $r_i = 0$ for all $i > |r|$) such that:

- *$G[\{r_1, r_2, r_3\}]$ is a clique*
- *for all $i \in \{4, \ldots, |r|\}$ the sets $\{r_{i-2}, r_i\}, \{r_{i-1}, r_i\}$ are edges in E'*
- *for all $i \in \{4, \ldots, |r|\}$ the set $\{r_{i-3}, r_i\}$ is either a singleton (i.e. $r_{i-3} = r_i$) or an edge in $E' \cup E''$.*

In practice, a re-order builds a longer (virtual) protein backbone whose edge structure is derived from the original graph G and it attempts to at least partially satisfy the requirements of a DMDGP instance. To this end, it exploits edges in E' (i.e. those corresponding to known precise distances). The first two properties ensure that there are at least two adjacent predecessors (namely r_{i-2} and r_{i-1}) corresponding to precise distances, for $i = \{3, 4, \ldots, |r|\}$. If $\{r_{i-3}, r_i\}$ is an edge in E' or $r_{i-3} = r_i$, then r_i can be placed in at most 2 points because of DISCRETIZATION. If $\{r_{i-3}, r_i\}$ is an edge in E'' the locus of r_i is the homeomorphic image of an interval in $\mathbb{R}^3$, as shown in Fig. 1. The search can then be made discrete (although possibly not binary) by considering Assumption 2 on E'' (see Sect. 1).

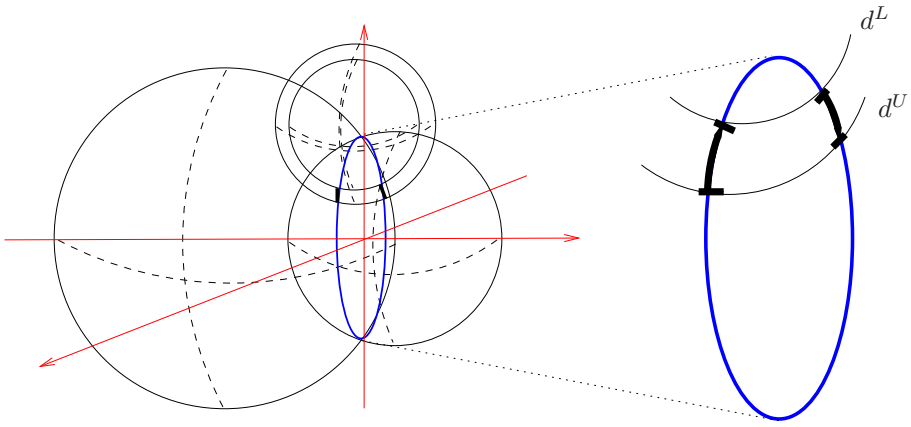


Figure 1: The intersection of two spheres with a spherical shell.

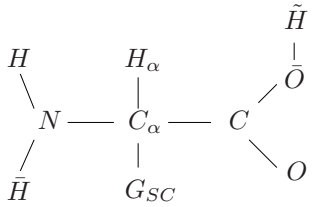
Thus, any re-order corresponds to an instance of the DMDGP, where some of the edges $\{r_{i-3}, r_i\}$ may not correspond to precise distances, but rather to intervals. In the following, we will introduce a graph representation of the protein backbones and we will show that their atoms, in their natural order, do not

define a re-order. Then, we will present our hand-crafted vertex order and prove that it is a re-order. Finally, we will discuss how to deal with edges $\{r_{i-3}, r_i\}$ which correspond to interval data.

In the rest of this section we construct the protein backbone graph G_{PB} and a vertex sequence with repetitions which happens to be a re-order for G_{PB}.

3.2 The protein backbone graph

Proteins are chains of smaller molecules called *amino acids*, which are chemically bound to each other. All amino acids share a common structure, which can be represented by the graph G_{AA} as shown below:



where H, N, C, O represent respectively hydrogen, nitrogen, carbon and oxygen atoms (we used tildes and bars to distinguish between atoms of the same type); C_α is a carbon atom bound to the *side chain* represented by the subgraph G_{SC} (we remark that amino acids only differ by the structures of their side chains); H_α is a hydrogen atom bound to the carbon C_α. All the edges in G_{AA} represent covalent bonds, whose associated weight is a real number.

During the protein synthesis, a sequence of amino acids bind together to form a chain. This operation can be described by the following graph operations. Let G'_1 be the graph associated to the first amino acid, and let G'_2 be the graph associated to the second amino acid. In order to obtain the graph $G_{12} = (V_{12}, E_{12})$, representing two bound amino acids, we need to make the following operations on the two graphs G'_1 and G'_2:

1. $G'_1[\{C, O, \bar{O}, \tilde{H}\}]$ is contracted to a vertex labelled C^1, yielding a modified graph $G_1 = (V_1, E_1)$;
2. $G'_2[\{\bar{H}, N\}]$ is contracted to a vertex labelled N^2, yielding a modified graph $G_2 = (V_2, E_2)$;
3. $V_{12} = V_1 \cup V_2$;
4. $E_{12} = E_1 \cup E_2 \cup \{C^1, N^2\}$.

The graph operations are shown graphically in Fig. 2. If we now replace G_1 by G_{12} it is clear that the same operation can be carried out again recursively any finite number $p \in \mathbb{N}$ of times. If we repeat this operation for all the amino acids forming a protein, then the resulting graph $G_{12\ldots p}$, with edge set $E_{12\ldots p}$ encoding the covalent bonds, represents the whole protein. Since all the edges of the graphs of the single amino acids represent covalent bonds, and therefore the weights associated to their edges are real numbers, all the edges of the protein graph $G_{12\ldots p}$ are weighted by real numbers. Note that all the graphs representing the single amino acids are connected to each other by the edge $\{C^i, N^{i+1}\}$ and that they all have the same structure, apart from the initial graph G_1 and the final graph G_p, because of the initial H in G_1 and a final $COOH$ group in G_p.

Our current focus is on protein backbones only. Then, we will not consider the amino acid side chains, but rather only the common part of each amino acid. The graph G_{PB} of a protein backbone can be obtained by removing all the graphs G_{SC} from each amino acid forming the protein. A well-known chemical property can help us to find known real distances between some pairs of atoms in V_{PB}: namely, all angles between consecutive covalent bonds are known real numbers [27]. This allows us to compute all the sides of the associated triangles exactly. For all $i \in \{1, \ldots, p\}$ we let $\bar{E}^i_T = \{\{H^i, C^i_\alpha\}, \{N^i, H^i_\alpha\}, \{N^i, C^i\},$

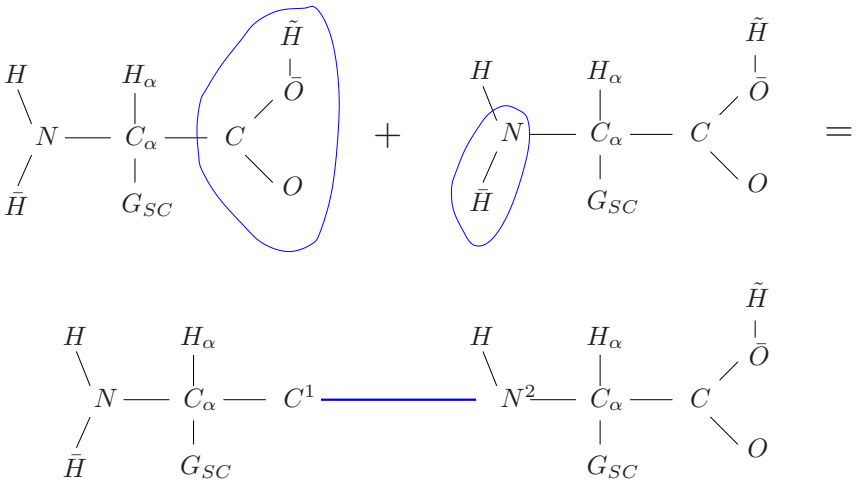


Figure 2: The binding of two amino acids.

$\{H^i_\alpha, C^i\}$. Furthermore, let $E^1_T = \bar{E}^1_T \cup \{\{H^0, H^1\}, \{H^0, C^1_\alpha\}, \{C^1_\alpha, N^2\}, \{C^1, H^2\}, \{C^1, C^2_\alpha\}\}$, for all $i \in \{2, \ldots, p-1\}$ let $E^i_T = \bar{E}^i_T \cup \{\{C^i_\alpha, N^{i+1}\}, \{C^i, H^{i+1}\}, \{C^i, C^{i+1}_\alpha\}\}$, and let $E^p_T = \bar{E}^p_T \cup \{\{C^p_\alpha, O^p\}, \{C^p_\alpha, O^{p+1}\}, \{C^p, H_f\}, \{O^p, O^{p+1}\}, \{O^{p+1}, H_f\}\}$. We let

$$V_{\mathsf{PB}} = \{H^0, H^1, N^1, C^1_\alpha, H^1_\alpha, C^1, \ldots, H^i, N^i, C^i_\alpha, H^i_\alpha, C^i, \ldots, H^p, N^p, C^p_\alpha, H^p_\alpha, C^p, O^p, O^{p+1}, H_f\},$$

$$E_{\mathsf{PB}} = E_{12\ldots p} \cup \bigcup_{i \le p} E^i_T,$$

and finally define the *protein backbone graph* to be $G_{\mathsf{PB}} = (V_{\mathsf{PB}}, E_{\mathsf{PB}})$, shown in Fig. 3. It is easy to verify

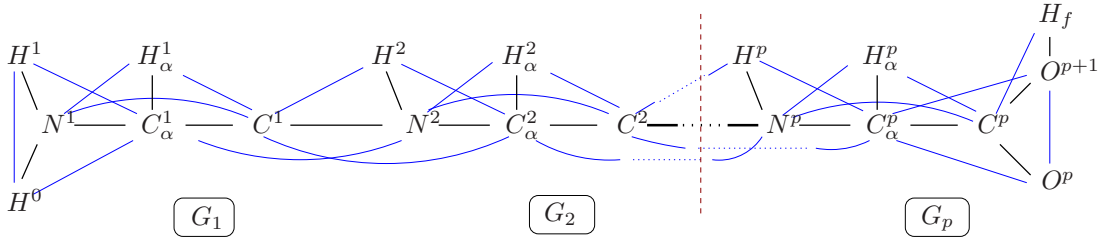


Figure 3: The protein backbone graph G_{PB}. Thick edges correspond to covalent bonds in E_{PB}.

that natural orders of the vertices of the graph G_{PB} (see Fig. 3), for example, $\{H^0, H^1, N^1, C^1_\alpha, H^1_\alpha, C^1, \ldots, H^i, N^i, C^i_\alpha, H^i_\alpha, C^i, \ldots, H^p, N^p, C^p_\alpha, H^p_\alpha, C^p, O^p, O^{p+1}, H_f\}$, are not re-orders. For this reason, we introduce a hand-crafted vertex order in the next section, which is a re-order.

3.3 A hand-crafted vertex order

The protein backbone graph G_{PB} has a similar repetitive structure given by the amino acids which compose it. Therefore, once a possible vertex order is identified for the generic amino acid, the same order can be duplicated for all the others. The only exception is given by the first and the last amino acids, that contain additional some atoms, because they are bound to only another amino acid.

Let us start then by assigning the following order to the atoms of the first amino acid of G_{PB}, as shown in Fig. 4:

$$r^1_{\mathsf{PB}} = \{N^1, H^1, H^0, C^1_\alpha, N^1, H^1_\alpha, C^1_\alpha, C^1\}.$$

One of the hydrogens bound to N^1 (in general, there is only one hydrogen) is indicated by the symbol

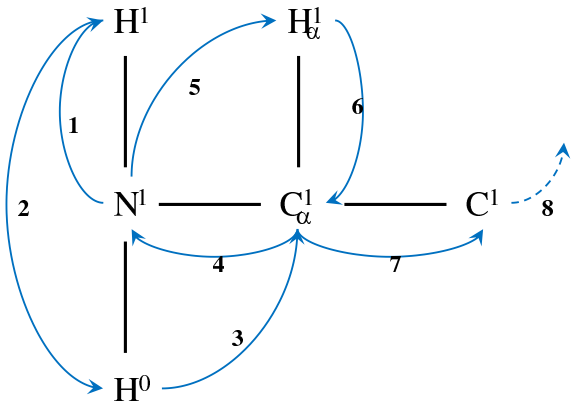


Figure 4: The order r^1_{PB}.

H^0. The carbon C^1_α and the nitrogen N^1 appear twice in the sequence. The other carbon of the first amino acid, the atom C^1, is considered, in this case, only once. Let us now assign the following order to the atoms of the second amino acid, as shown in Fig. 5:

$$r^2_{\mathsf{PB}} = \{N^2, C^2_\alpha, H^2, N^2, C^2_\alpha, H^2_\alpha, C^2, C^2_\alpha\}.$$

This sequence of atoms is used for building a *bridge* between the first amino acid and the third one,

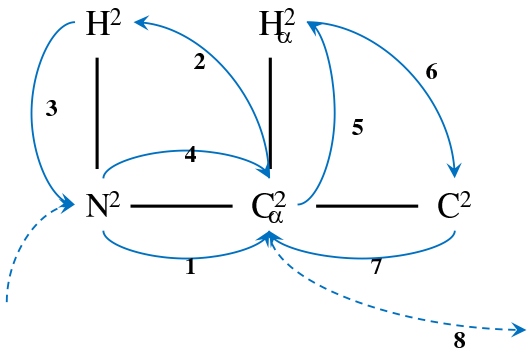


Figure 5: The order r^2_{PB}.

from which a generic order will be considered. In fact, the order defined on the second amino acid is quite similar to the generic one. Atoms are considered more than once, and, in particular, the carbon C^2_α appears in the sequence 3 times. This is the vertex order for the generic amino acid (from the third to last but one), as shown in Fig. 6:

$$r^i_{\mathsf{PB}} = \{N^i, C^{i-1}, C^i_\alpha, H^i, N^i, C^i_\alpha, H^i_\alpha, C^i, C^i_\alpha\}.$$

The nitrogen N^i is considered twice, the carbon C^i_α is considered 3 times, and the carbon C^{i-1} belonging to the previous amino acid is repeated among the atoms of the amino acid i. Note that hydrogen atoms are never duplicated, because the number of distances regarding these atoms in E' or E'' is quite limited:

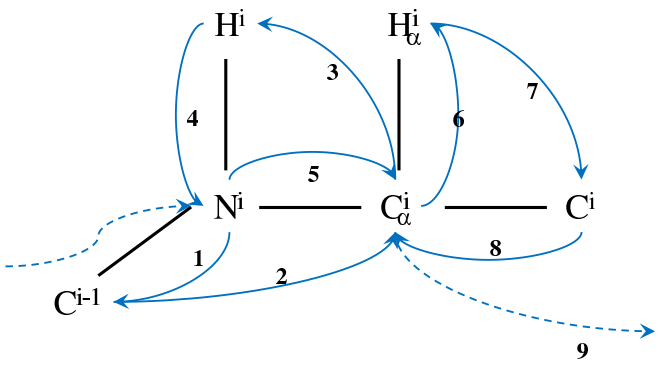


Figure 6: The generic order r^i_{PB}.

most of the distances regarding hydrogens belong to F. When the last amino acid is considered, we need to take into account that there are some atoms more in the final part of the amino acid. Therefore, the vertex order that we define for the last amino acid, as shown in Fig. 7, is:

$$r^p_{\mathsf{PB}} = \{N^p, C^{p-1}, C^p_\alpha, H^p, N^p, C^p_\alpha, H^p_\alpha, C^p, C^p_\alpha, O^p, C^p, O^{p+1}\}.$$

Note that this is the only case in which oxygen atoms appear. The last atom in r^p_{PB} is the oxygen O^{p+1},

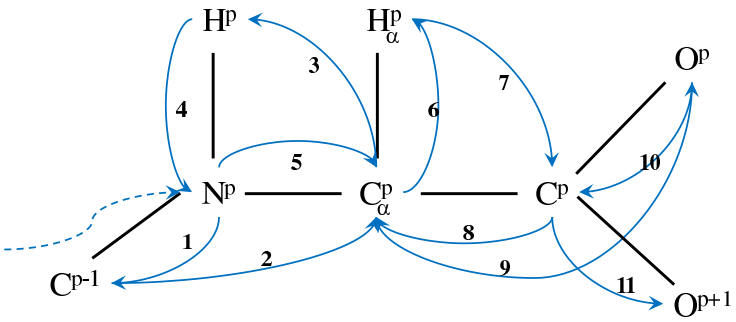


Figure 7: The order r^p_{PB}.

to which we assigned the superscript $p+1$ in order to distinguish it from the other oxygen O^p, even though there is no amino acid $p+1$. We suppose that the set of atoms COO^- ends the sequence, that is, the hydrogen H of the group COOH is lost and this makes the last part of the sequence negatively charged. Equivalently, we could have considered the set of atoms NH_3^+ at the beginning of the sequence (see Fig. 4) instead of group NH_2. To this aim, a third hydrogen H could be considered in r^1_{PB}.

Let us indicate by the symbol r_{PB} the defined vertex order on the whole protein backbone G_{PB}:

$$r_{\mathsf{PB}} = \bigcup_{i=1}^{p} r^i_{\mathsf{PB}}.$$

Fig. 8 shows the hand-crafted order for a small protein backbone containing 3 amino acids. It shows

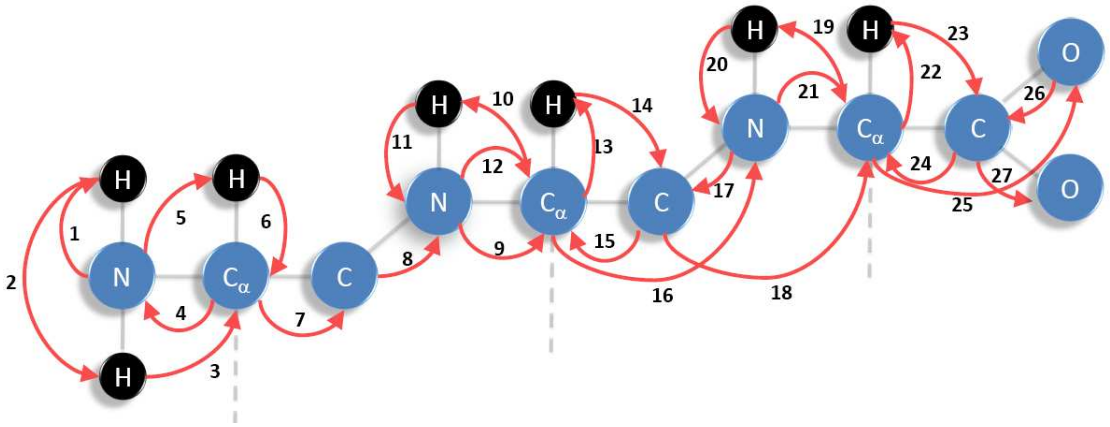


Figure 8: The hand-crafted re-order r_{PB}.

the vertex order for the first amino acid, for the second one, and for the generic amino acid, with the last three vertices concerning the last amino acid of the sequence. By comparing Fig. 3 and Fig. 8, we can see that only edges of the kind $\{r_{i-3}, r_i\}$ may correspond to discretizable intervals (in E''). As a consequence, we have the following lemma:

3.2 Lemma
The sequence r_{PB} is a re-order for G_{PB}.

Because of repeated atoms, many distances equal to 0 appear in the sequence r_{PB}. If we consider three consecutive vertices and suppose that two of these vertices refer to the same atom, then they will be perfectly aligned and this will go against the Strict Triangular Inequalities assumption. In fact, by definition of r_{PB} it is easy to verify by inspection that this does not happen.

3.3 Lemma
The sequence r_{PB} satisfies the Strict Triangular Inequalities *assumption of the DMDGP.*

Using these two lemmata, it is easy to prove the following.

3.4 Theorem
Any instance of the MDGP with interval data whose graph G_{PB} is a protein backbone graph has a finite number of incongruent embeddings.

We remark that vertices associated to edges in E'' might lead to an impractically high number of embeddings. For this reason, the number D of discrete values in intervals for edges in E'' plays a very important role. Setting $D = 1$ reduces the instance to an ordinary real-value weighted DMDGP. A previous work [19] showed that the average of the known interval is usually not a sufficiently accurate approximation of the actual distance. However, if D is too large, the number of embeddings could be huge.

By Lemma 3.3, the distances equal to 0 are never related to $\{r_{i-2}, r_i\}$ and $\{r_{i-1}, r_i\}$. However, we can have distances equal to 0 associated to other pairs of vertices. Some of the distances related to the edges $\{r_{i-3}, r_i\}$ can be 0, which agree with the definition of a re-order. For example, let us consider the duplicated atom C_α^1, which is in position 7. The first copy of this atom is instead in position 4. When discretizing at level 7, therefore, the distance between the two copies of C_α^1 is considered. The fact that a distance equal to 0 is used in the discretization process implies that only one possible position for the current atom is feasible. In this case, the distance equal to 0 is directly exploited for computing the unique atomic position of the duplicated atom.

Moreover, distances equal to 0 can also appear in correspondence with edges $\{r_i, r_j\}$, with $j > i + 3$. For example, let us consider the second copy of the nitrogen N^1 of the first amino acid. It is known that its distance from the first copy of N^1 is 0. The first copy appears in position 1 along the sequence, whereas the second copy appears in position 5. Since that distance (equal to 0) is not used in the discretization process, two positions for the atom can be computed, and one of the two can be pruned by employing the distance equal to 0. However, since we know this is a duplicated atom, we can directly place this atom in the same position as its previous copy.

When C_α atoms are considered, a property of the protein backbone can be exploited [27]. If the C_α atom is not duplicated, the edge $\{r_{i-3}, r_i\}$ is represented by an interval, and therefore multiple branching should be required. However, the distance corresponding to $\{r_{i-3}, r_i\}$ involves two carbon atoms C_α of two successive amino acids, and hence there are restrictions for their relative configurations, because the backbone atoms C_α, C, N, C_α are forced to be in the same plane by a peptide bond [27]. As a consequence, only two torsion angles can be defined for this quadruplet of atoms, and then only two possible positions can be computed for the considered C_α.

Finally, as in the DMDGP with real-weighted edges [10, 12], we need only concern ourselves with one half of the (finite) solution space: there is a reflection symmetry around the plane defined by the first three atoms of the sequence r_{PB} which allow us to fix the position of the fourth atom.

4 The interval Branch-and-Prune (iBP)

Algorithm 1, called *interval* BP (iBP) is an extension of the classic BP algorithm, previously proposed in [16], that addresses interval data in the sense explained above. It is the first exact algorithm for discrete MDGPs which consider interval data. The input arguments of Alg. 1 are: the index i of the re-order whose image r_j indexes the atom currently being placed, the re-order r, the edge weight function d and the integer D.

Algorithm 1 The iBP algorithm.

1: iBP(j, r, d, D)
2: **if** (r_j is a duplicated atom) **then**
3: copy coordinates of previous copy of r_j in $x^1_{r_j}$
4: iBP$(j + 1, r, d, D)$;
5: **else**
6: **if** ($d(r_j - 3, r_j)$ is exact) **then**
7: $b = 2$;
8: **else**
9: $b = 2D$;
10: **end if**
11: **for** $k \in \{1, \ldots, b\}$ **do**
12: compute the k-th atomic position $x^k_{r_j}$ for the r_j-th atom;
13: check the feasibility of position $x^k_{r_j}$ using edges in F;
14: **if** ($x^k_{r_j}$ is feasible) **then**
15: **if** ($j = |r|$) **then**
16: a solution x is found, print it;
17: **else**
18: iBP$(j + 1, r, d, D)$;
19: **end if**
20: **end if**
21: **end for**
22: **end if**

The correctness of the algorithm follows because at Step 11 it tests all possible positions for atom r_i

for feasibility. The algorithm terminates when j reaches $|r|$ if the instance is YES or before (when no $x^k_{r_i}$ is feasible at a given recursion level) if the instance is NO. The recursive calls of the iBP algorithm generate a search tree structure: each node has a number of subnodes equal to b. Leaf nodes at level $|r|$ correspond to embeddings.

4.1 A detailed test instance

We consider in this section a very simple instance with the same atoms and edges in E', E'' as in Fig. 8. The distance function d is constructed as follows. Given four numbers $\bar{d}_1, \bar{d}_2, l_3, u_3$ such that $2\bar{d}_1 > \bar{d}_2$ and $l_3 < u_3$, the distance between every pair $\{u, v\}$ of bound atoms is $d_{uv} = \bar{d}_1$; the distance between every pair $\{u, v\}$ of atoms two covalent bonds apart is $d_{uv} = \bar{d}_2$. The distance function d maps every pair $\{u, v\}$ of atoms three covalent bonds apart to a discrete set of D values in the interval $[l_3, u_3]$. As is, the obtained instance has no edges in F that can be used for pruning, so the iBP yields a full tree. We randomly choose a leaf node at level $|r|$ and consider the corresponding embedding: from this embedding we derive all new distances (we let F be the set of corresponding edges) and discard those that exceed 5Å. To each of the remaining edges $\{u, v\} \in F$ we assign a distance interval $[d_{uv} - \varepsilon, d_{uv} + \varepsilon]$. Fig. 9 shows the generated test instance.

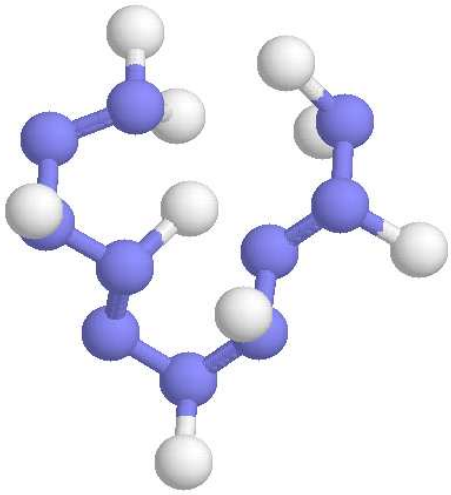


Figure 9: The test instance.

Fig. 10 shows the tree structure related to this instance. The positions of the first three atoms can be obtained using the known information on the distances in E'. The branching starts at level 4, in correspondence with the atom C^1_α. Due to the symmetry property of the DMDGP, we can discard one of the branches at level 4, and concentrate our researchers only on one of them. At level 5 we have the first duplicated atom, the nitrogen N^1 which already appeared at level 1. Therefore, we have no branching, because the new copy of N^1 can only be placed in the same position of its previous copy. The first hydrogen in the vertex order on which we need to branch appears at level 6. This is the hydrogen H^1_α. Since the distance between this atom and the previous H^1 is an interval, we need to discretize the interval and take from it D exact distances. As a consequence, $2D$ branches are added at level 6 on the binary tree. At level 7, we find another duplicated atom, and therefore, there is no branching. After this atom, we have a sequence of 3 atoms that are neither duplicated nor hydrogens: depending on the fact that an interval needs to be discretized or not, only two or $2D$ branches are added to the tree. The first hydrogen of the second amino acid is at level 11. Since the distance between C^1 and H^2 is in E', we have only two branches. The other cases are similar to the previous ones.

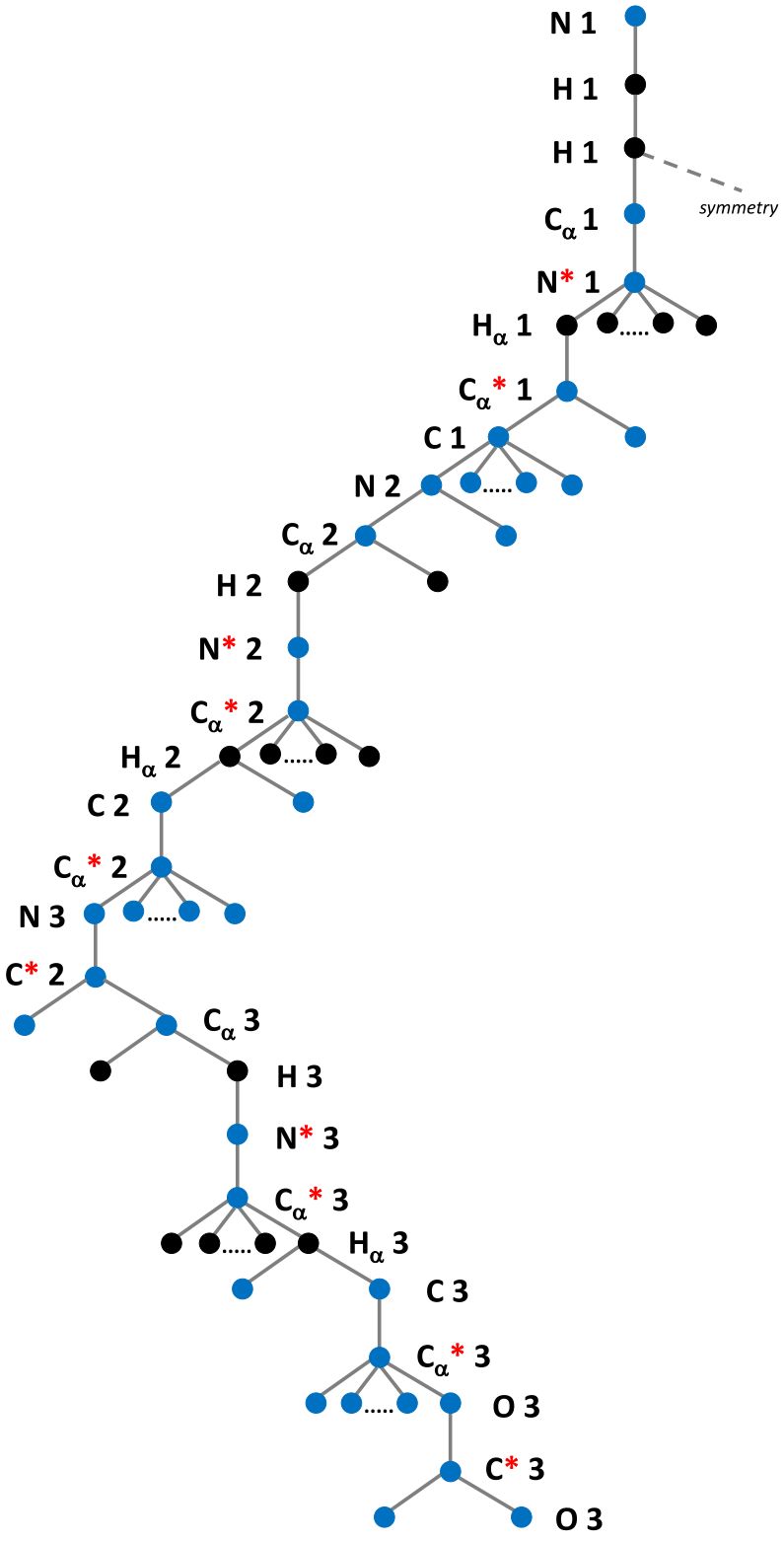


Figure 10: Part of the tree of the test instance.

Table 1 provides the number of branches on each layer of the tree in Fig. 10. We consider here the generated instance in Fig. 9, where $\varepsilon = 0.3$ and $D = 6$. In particular, the last but first column of Table 1 shows the number of branches of the full tree, where no pruning is applied. If r_i is a duplicated atom (Alg. 1, Step 4) just one subnode is created. If r_i is not duplicated, then: (1) if $b = 2$ (Step 7) two subnodes are created; (2) if $b = 2D$ (Step 9), $2D$ subnodes are created. Without pruning, the full tree has 1528823808 nodes at level 28. The last column shows the corresponding statistics when pruning is applied. Notice that sometimes, for chemical reasons, pruning is carried out when a distance is too short (less than 1Å).

layer	*atom*	*amino acid*	*duplicated?*	*branches w/out pruning*	*branches with pruning*
1	N	1	no	1	1
2	H	1	no	1	1
3	H	1	no	1	1
4	C_α	1	no	2	2
5	N	1	yes	2	2
6	H_α	1	no	24	**18**
7	C_α	1	yes	24	18
8	C	1	no	48	36
9	N	2	no	576	360
10	C_α	2	no	1152	720
11	H	2	no	2304	**10**
12	N	2	yes	2304	10
13	C_α	2	yes	2304	10
14	H_α	2	no	27648	**70**
15	C	2	no	55296	140
16	C_α	2	yes	55296	140
17	N	3	no	663552	1400
18	C	2	yes	663552	1400
19	C_α	3	no	1327104	2800
20	H	3	no	2654208	**4**
21	N	3	yes	2654208	4
22	C_α	3	yes	2654208	4
23	H_α	3	no	31850496	**9**
24	C	3	no	63700992	18
25	C_α	3	yes	63700992	18
26	O	3	no	764411904	52
27	C	3	yes	764411904	52
28	O	3	no	1528823808	**10**

Table 1: The number of branches, step by step, of the discrete domain with and without pruning.

5 Computational results

We show in this section some computational experiments on larger instances. All the experiments showed in the paper have been performed on an Intel Core 2 CPU 6400 @ 2.13 GHz with 4GB RAM, running Linux. The *i*BP algorithm (see Section 4) have been implemented in C programming language and compiled by the GNU C++ compiler v.4.1.2 with the `-O3` flag.

The procedure which has been employed for generating a set of instances is the same already detailed in Section 4.1 for the generation of the small test instance. The only difference is that the order for the generic amino acid (see Figure 6) is repeated as many times as needed, because the new instances

are composed by more than 3 amino acids. Naturally, these instances are not good representatives of real protein backbones for the global shape they have, but they are still useful for the purposes of the experiments. The paramenter ε is always set to 0.3 for all generated instances.

Table 2 shows the details of some experiments performed with the iBP algorithm. In this table, n_{aa}

n_{aa}	n	$\|E\|$	LDE	#Sol	D	time
10	91	676	3.03e-05	1	3	0.01
20	181	1398	1.45e-05	1	3	0.01
50	473	3587	4.47e-05	1	3	0.13
70	631	5038	2.79e-05	1	3	0.15
100	901	7223	2.22e-05	1	3	0.17
120	1081	8654	3.76e-05	1	6	0.99

Table 2: Some experiments with larger instances. Only one solution is required.

is the total number of amino acids, n is the total number (including the repetitions) of considered atoms and $|E|$ is the number of distances which are available (exact distances and intervals). We only require one solution, and therefore #Sol is always 1. D is the minimum number of sample distances to be taken from the intervals for obtaining at least one solution to the problem. Finally, the CPU time (in seconds) is given for each experiment. These experiments show that the discretization with interval data can also be applied to instances having a larger dimension.

6 Conclusion

This paper moves an important step towards bringing the very successful Branch-and-Prune algorithm towards distance geometry problems with interval data. Previously, discrete search in distance geometry was only possible with precise distances, a limitation which rules out the tackling of real protein conformation problems based on NMR distances obtained experimentally. Although for the moment we only limit our computational tests to artificially generated instances, we are already in touch with several biochemistry research teams in order to obtain raw NMR data. Future works will be aimed at the efficient solution of NMR instances by the presented technique.

Acknowledgments

The authors wish to thank Leandro Martínez, Thérèse Malliavin and Michael Nilges for their valuable comments on this work. The authors would also like to thank the Brazilian research agencies FAPESP and CNPq, the French research agencies CNRS and ANR ("Bip:Bip project"), and École Polytechnique, for financial support.

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] 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.

[3] 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.

[4] 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.

[5] L. Henneberg. *Die graphische Statik der starren Systeme*. B.G. Teubner, Leipzig, 1911.

[6] A. Lee-St. John. *Geometric Constraint Systems with Applications in CAD and Biology*. PhD thesis, University of Massachusetts at Amherst, 2008.

[7] S. Kirkpatrick, C.D.Jr. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[8] N. Krislock. *Semidefinite Facial Reduction for Low-Rank Euclidean Distance Matrix Completion*. PhD thesis, University of Waterloo, 2010.

[9] C. Lavor, J. Lee, A. Lee-St. John, L. Liberti, A. Mucherino, and M. Sviridenko. Discretization orders for distance geometry problems. *Optimization Letters*, to appear.

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

[11] 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.

[12] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino. The discretizable molecular distance geometry problem. *Computational Optimization and Applications*, to appear, 2011.

[13] C. Lavor, L. Liberti, and A. Mucherino. On the solution of molecular distance geometry problems with interval data. In *International Conference on Bioinformatics & Biomedicine*, Hong Kong, 2010. IEEE conference proceedings.

[14] C. Lavor, A. Mucherino, L. Liberti, and N. Maculan. Discrete approaches for solving molecular distance geometry problems using nmr data. *International Journal of Computational Biosciences*, 1:88–94, 2011.

[15] 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.

[16] 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.

[17] 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, 2011.

[18] X. Liu and P.M. Pardalos. A tabu based pattern search method for the distance geometry problem. In F. Giannessi et. al, editor, *New Trends in Mathematical Programming*, pages 223–234. Kluwer Academic Publishers, 1998.

[19] 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.

[20] A. Mucherino, C. Lavor, and L. Liberti. The discretizable distance geometry problem. *Optimization Letters*, to appear.

[21] 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.

[22] A. Mucherino, L. Liberti, C. Lavor, and N. Maculan. Comparisons between an exact and a metaheuristic algorithm for the molecular distance geometry problem. In F. Rothlauf, editor, *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 333–340, Montreal, 2009. ACM.

[23] M. Nilges, A.M. Gronenborn, A.T. Brunger, and G.M. Clore. Determination of three-dimensional structures of proteins by simulated annealing with interproton distance restraints. application to crambin, potato carboxypeptidase inhibitor and barley serine proteinase inhibitor 2. *Protein Engineering*, 2:27–38, 1988.

[24] M. Nilges, M.J. Macias, S.I. O'Donoghue, and H. Oschkinat. Automated noesy interpretation with ambiguous distance restraints: The refined nmr solution structure of the pleckstrin homology domain from β-spectrin. *Journal of Molecular Biology*, 269:408–422, 1997.

[25] P.M. Pardalos, D. Shalloway, and G. Xu, editors. *Global Minimization of Nonconvex Energy Functions: Molecular Conformation and Protein Folding*. AMS, DIMACS, 1996.

[26] 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.

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

[28] M-C. So and Y. Ye. Theory of semidefinite programming for sensor network localization. *Mathematical Programming*, 109:367–384, 2007.

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