



**HAL**  
open science

## Recent advances on the interval distance geometry problem

Douglas Gonçalves, Antonio Mucherino, Carlile Lavor, Leo Liberti

► **To cite this version:**

Douglas Gonçalves, Antonio Mucherino, Carlile Lavor, Leo Liberti. Recent advances on the interval distance geometry problem. *Journal of Global Optimization*, 2017, 69 (3), pp.525-545. 10.1007/s10898-016-0493-6 . hal-02105295

**HAL Id: hal-02105295**

**<https://hal.science/hal-02105295>**

Submitted on 20 Apr 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Recent advances on the interval distance geometry problem

Douglas S. Gonçalves<sup>1</sup> · Antonio Mucherino<sup>2</sup> · Carlile Lavor<sup>3</sup> · Leo Liberti<sup>4</sup> 

Received: 28 January 2016 / Accepted: 20 December 2016  
© Springer Science+Business Media New York 2016

**Abstract** We discuss a discretization-based solution approach for a classic problem in global optimization, namely the distance geometry problem (DGP). We focus our attention on a particular class of the DGP which is concerned with the identification of the conformation of biological molecules. Among the many relevant ideas for the discretization of the DGP in the literature, we identify the most promising ones and address their inherent limitations to application to this class of problems. The result is an improved method for estimating 3D structures of small proteins based only on the knowledge of some distance restraints between pairs of atoms. We present computational results showcasing the usefulness of the new proposed approach. Proteins act on living cells according to their geometric and chemical properties: finding protein conformations can be very useful within the pharmaceutical industry in order to synthesize new drugs.

**Keywords** Distance geometry · Discretization · Molecular conformation

✉ Leo Liberti  
liberti@lix.polytechnique.fr

Douglas S. Gonçalves  
douglas.goncalves@ufsc.br

Antonio Mucherino  
antonio.mucherino@irisa.fr

Carlile Lavor  
clavor@ime.unicamp.br

<sup>1</sup> CFM, Universidade Federal de Santa Catarina, Florianópolis, Brazil

<sup>2</sup> IRISA, Université de Rennes 1, Rennes, France

<sup>3</sup> IMECC, Universidade Estadual de Campinas, Campinas, Brazil

<sup>4</sup> CNRS LIX, École Polytechnique, 91128 Palaiseau, France

# 1 Introduction

Given a positive integer  $K$  and a simple weighted undirected graph  $G = (V, E, d)$ , where  $d$  maps edges  $\{u, v\} \in E$  to positive interval weights  $[\underline{d}(\{u, v\}), \bar{d}(\{u, v\})]$ , the Distance Geometry Problem (DGP) [38] is the problem of finding a realization of the graph  $G$  in a  $K$ -dimensional Euclidean space. In other words, the DGP requires the identification of a map  $x : V \rightarrow \mathbb{R}^K$ , satisfying the distance constraints:

$$\underline{d}(\{u, v\}) \leq \|x(u) - x(v)\| \leq \bar{d}(\{u, v\}), \quad \forall \{u, v\} \in E, \quad (1)$$

where  $\|\cdot\|$  denotes the Euclidean norm.

A solution for (1) is called a *realization* or an *embedding*. In order to simplify the notation, we will use  $x_u := x(u)$  and  $d_{uv} := d(u, v) := d(\{u, v\})$  hereafter.

In structural biology, the problem of identifying molecular conformations from a given list of distance restraints between atom pairs is a DGP in dimension  $K = 3$ . This problem is also known in the scientific literature as the Molecular Distance Geometry Problem (MDGP). In this particular application, the distances may be exact (i.e.  $\underline{d}_{uv} = \bar{d}_{uv}$ ) or represented by a positive real-valued interval (i.e.  $\bar{d}_{uv} > \underline{d}_{uv} > 0$ ).

Exact distances are related to the chemical bonds whereas interval ones can be provided by experimental techniques. Such techniques include Nuclear Magnetic Resonance (NMR) [3], Förster resonance energy transfer (FRET) [7] and mass spectrometry (MS) cross-linking [10].

The DGP is NP-hard [56] and there exist several approaches to this problem (see [38, 52] and Sect. 1.1), where the DGP is reformulated as a global optimization problem on a continuous search domain, whose objective function is generally a penalty function of the distance constraints. More recently, a discrete approach to the DGP was proposed [39], where the continuous domain of the optimization problem is transformed into a discrete domain.

## 1.1 Literature review

Distance Geometry (DG) has played a prominent part in Global Optimization (GO) insofar as it has important applications to science (e.g. protein conformation) and engineering (localization of sensor networks, structural rigidity, control of unmanned underwater vehicles and robotic arms), and it is naturally cast as a system of nonconvex constraints (Eq. (1)) in terms of continuous decision variables. In general, DGPs are reformulated as a minimization of constraint violations. Such reformulations have the property that the optimal objective function value is zero for feasible instances, and strictly positive for infeasible ones. Various approaches have been proposed in this journal for the general case [15, 16, 26, 27, 34, 37, 47, 64, 68], and many others on the application to protein conformation [11, 17, 21, 43, 46, 54]. In this paper we focus on the case where the input graph is rigid, which implies that the search process has an inherently combinatorial side.

Over the years, the solution to MDGPs (DGPs arising in structural biology) have been typically attempted by employing tools such as ARIA [42], CYANA [23] and UNIO [22], which are all based on the Simulated Annealing (SA) meta-heuristic [28].

While molecular conformations are generally obtained by the above methods and successively stored in databases such as the Protein Data Bank (PDB) [4], a second class of methods based on Nonlinear Programming (NLP) solution techniques has emerged in the last decades. A well-known example is the DGSOL algorithm [48], which employs a homotopy method based on locally solving progressively finer Gaussian smoothings of the original problem.

57 A third class of methods is based on Euclidean Distance Matrix Completion [1, 14]. This is  
58 the case for the EMBED algorithm [12], which aims to fill in the missing distance bounds by  
59 constraint propagation of triangle and tetrangle inequalities. Thereafter, a candidate distance  
60 matrix (named dissimilarity matrix) is sampled from the completed interval distance matrix,  
61 and atom coordinates are obtained by matrix decomposition [13, 58]. Since the dissimilarity  
62 matrix is not guaranteed to be a Euclidean Distance matrix, some of the original constraints  
63 might be violated. The last phase therefore consists in minimizing the constraint violation  
64 by local minimization, using the obtained embedding as an initial point.

65 A fourth class is centered around the so-called Build-Up algorithm [15, 64]. These methods  
66 are based on the ancient idea of triangulation, used by humankind ever since navigation  
67 existed. In the context of distance geometry, where a point position is determined by the  
68 distances to it rather than the angles they subtend, this is known as “trilateration”. Build-Up  
69 algorithms in dimension  $K = 3$  attempt to place an unknown point  $x_i$  (for some  $i \leq n$ ) by  
70 identifying at least four other points with known positions, and having known distances to  
71  $x_i$ . When dealing with proteins and experimental data, the assumption of having four known  
72 exact distances to any given point may be excessively strong [44]. We point out, however,  
73 that some variants of the Build-Up algorithm overcome this limitation. For example, in order  
74 to address the uncertainty of the given distance values, the extension presented in [60] takes  
75 into account atomic coordinates and an unknown radius representing the uncertainty. Another  
76 variant [65] partly addresses the requirement of unknown vertices having at least four adjacent  
77 vertices with known positions. This variant can find multiple valid realizations, but appears  
78 to lack the ability to finding *all* possible incongruent realizations.

79 A fifth and very important class of methods is based on solving a Semidefinite Program-  
80 ming (SDP) relaxation of the DGP [6, 29, 45]. In particular, [29] exploits the cliques in the  
81 graph to reduce the size of the SDP formulation (also see [2]). This method was shown to  
82 be able to solve NMR instances containing real data and to reconstruct conformation models  
83 that are very close to the ones available on the PDB.

84 The authors of this paper are among the researchers who proposed and worked on a sixth  
85 class of methods based on a combinatorial algorithm called *Branch & Prune* (BP) [36].  
86 Protein graphs share some common properties: for example, they can be decomposed into a  
87 backbone subgraph and many side chains subgraphs [57] (these can be realized separately  
88 and then put together [55]). The backbone subgraph is larger than the subgraphs related to  
89 side chains, and hence most difficult to realize. However, it also defines an order on the atoms  
90 with certain topological properties, which we formally discuss below (informally, we can say  
91 that every atom in this order has at least three predecessors which are also adjacent in the  
92 graph structure). Under this assumption, the search domain of the underlying optimization  
93 problem can be reduced to a discrete set with a tree structure [32, 51], which can be searched  
94 by the BP algorithm [36]. If the distances are exact, BP can find all realizations of a given  
95 protein backbone graph. Although an exhaustive search in the conformation tree is worst  
96 case exponential [32], numerical experiments have shown that BP behaves polynomially in  
97 protein-like instances [40]. In fact, it can be proved that, in such cases, the problem is Fixed  
98 Parameter Tractable (FPT) [41]. In computational experiments, the parameter value could  
99 always be fixed at a single constant, which explains the polytime behaviour. For protein  
100 backbone instances with exact distances, BP is one of the fastest available methods, one of  
101 the most reliable, and the only one which can certifiably find all incongruent realizations.

102 An adaptation of the BP to the interval distance setting was proposed in [34], where  
103 intervals are replaced with a finite set of discrete points. We refer to this BP adaptation as  
104 the *interval BP* (*iBP*). This algorithm was tested on real protein instances in [8]. Although

105 this BP variant shows promise, its practical applicability is currently limited by the choice  
 106 of discretization points.

107 Several other approaches for solving DGPs can be found in the scientific literature. The  
 108 interested reader can refer to [33,38,52].

## 109 1.2 Aim of this paper

110 Our main motivation in this work is to improve the *iBP* algorithm proposed in [34] for  
 111 solving MDGPs with interval data. For this purpose, we identify the main limitations of this  
 112 discrete approach in presence of interval distances and propose a new variant of *iBP* to find  
 113 approximate solutions for interval MDGPs.

114 The identification of the barriers against the successful application of *iBP* in real set-  
 115 tings represents an important step towards a combinatorial methodology with the following  
 116 properties:

- 117 – it is specifically suitable for solving the protein conformation problem from distance  
 118 restraint data;
- 119 – it can work with uncertain data, specified as interval distances provided by experimental  
 120 techniques;
- 121 – it can potentially find all incongruent realizations of a given instance.

122 The paper is organized as follows. In Sect. 2, we define the subclass of DGP instances  
 123 describing protein backbone graphs: we discuss assumptions, discretization orders, the *iBP*  
 124 algorithm variant, pruning devices, and the parameterization of the coordinates. Section 3  
 125 presents a method, based on some interval distance constraints, which is able to reduce the  
 126 set of candidate positions for certain vertices before the *iBP* branching phase. Section 4  
 127 addresses the main limitations in handling larger molecules in presence of interval data and  
 128 presents a heuristic for finding approximate realizations. The computational results in Sect. 5  
 129 illustrate the improvements due to the proposed approaches.

## 130 2 A combinatorial approach

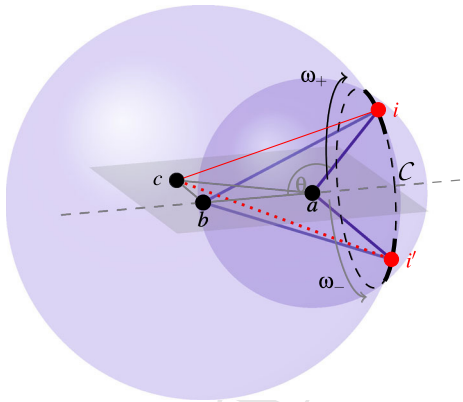
131 Let  $G = (V, E, d)$  be a simple weighted undirected graph representing an instance of the  
 132 MDGP. In the following, vertices of  $V$  will represent atoms of the given molecule and  $\{u, v\} \in$   
 133  $E$  if the distance between the atoms  $u$  and  $v$  is available. The map  $d$  relates each edge  $\{u, v\} \in$   
 134  $E$  to a positive interval weight  $[\underline{d}(u, v), \bar{d}(u, v)]$ . The MDGP asks to find a realization  
 135  $x : V \rightarrow \mathbb{R}^3$  (see Introduction), i.e. a molecular conformation in three-dimensional space  
 136 such that:

$$137 \quad \underline{d}(u, v) \leq \|x_u - x_v\| \leq \bar{d}(u, v), \quad \forall \{u, v\} \in E. \quad (2)$$

138 Recall that  $\underline{d}(u, v)$  and  $\bar{d}(u, v)$  denote, respectively, the lower and upper bounds for the  
 139 distance  $d(u, v)$  (with  $\underline{d}(u, v) = \bar{d}(u, v)$  if  $d(u, v)$  is exact). We also suppose that the given  
 140 set of distances is realizable in  $\mathbb{R}^3$ .

141 In order to discretize the search domain, MDGP instances need to satisfy some particular  
 142 assumptions. The main requirement is that the atoms need to be sorted in a way that there  
 143 are at least three *reference atoms* for each of them (aside, obviously, from the first three). We  
 144 say that an atom  $u$  is a reference for another atom  $v$  when  $u$  precedes  $v$  in the given atomic  
 145 order, and the distance  $d(u, v)$  is known. In such a case, candidate positions for  $v$  belong to  
 146 the sphere centered in  $u$  and having radius  $d(u, v)$ . When the *reference distance*  $d(u, v)$  is  
 147 given through an interval, the sphere becomes a spherical shell, namely, the region between

**Fig. 1** The two feasible arcs obtained by intersecting two spheres and one spherical shell



148 an inner sphere of radius  $\underline{d}$  and an outer sphere of radius  $\bar{d}$  with the same center. If three  
 149 reference atoms are available for  $v$ , then candidate positions (for  $v$ ) belong to the intersection  
 150 of three spherical shells. The easiest situation is the one where the three available distances  
 151 are exact and the intersection gives, in general, two possible positions for  $v$  [32]. However,  
 152 if only one of the three distances is allowed to take values into a certain interval, then the  
 153 intersection gives two arcs of a circle, generally disjoint, where sample points can be chosen  
 154 [34]. In both last situations, the discretization can be performed. More details are given in  
 155 the next section.

156 **2.1 *i*BP algorithm and discretization orders**

157 Let  $G = (V, E, d)$  be an instance of the MDGP and let us suppose that there exist an order  
 158 for the atoms  $v \in V$ , so that we can assign a numerical label  $i \in \{1, 2, \dots, |V|\}$  to each of  
 159 them. At each recursive call of the *i*BP algorithm, candidate positions for the current atom  $i$   
 160 are computed using the positions of the previously placed reference atoms and their distances  
 161 to the atom  $i$ .

162 When the distances between  $i$  and its references are exact, the intersection of three spheres  
 163 needs to be computed. If the reference atoms  $\{a, b, c\}$  are not collinear, then such an inter-  
 164 section results in at most two points. When this situation is verified for all atoms  $i > 3$ , then  
 165 the search domain has the structure of a binary tree [32].

166 However, if one of the three reference distances, say  $d_{ci}$ , is an interval, then the two spheres  
 167 centered at  $x_a$  and  $x_b$  need to be intersected with a spherical shell centered at  $x_c$ . As a result,  
 168 the intersection gives two candidate arcs (see Fig. 1). These arcs are over the dashed circle  
 169  $C$  defined by the intersection of the two spheres. When the intersection consists of two arcs,  
 170 a finite number  $D$  of sample positions should be selected from each of them [34]. This way,  
 171 we still have a discrete set of possible positions for the atom  $i$ .

172 Therefore, the discretization strongly depends on an *order* for the vertices (atoms) of  $G$   
 173 satisfying specific properties. Definition 1 formalizes the assumptions mentioned above.

174 **Definition 1** The *interval* Discretizable DGP in dimension 3 (*i*DDGP<sub>3</sub>)

175 Given a simple weighted undirected graph  $G = (V, E, d)$ , where  $E' \subset E$  is the subset of  
 176 edges for which their weights are exact distances, we say that  $G$  represents an instance of the  
 177 *i*DDGP<sub>3</sub> if there exists a total order on the vertices of  $V$  verifying the following conditions:

- 178 (a)  $G_C = (V_C, E_C) \equiv G[\{1, 2, 3\}]$  is a clique and  $E_C \subset E'$ ;
- 179 (b)  $\forall i \in \{4, \dots, |V|\}$ , there exists  $\{a, b, c\}$  such that

**Algorithm 1** The *i*BP algorithm.

---

```

1: iBP(i, n, d, D)
2: if (i > n) then
3:   // one solution is found
4:   print current conformation;
5: else
6:   // coordinate computation
7:   if ( $d_{ci}$  is an interval) then
8:     compute the two candidate arcs and add them to the list L
9:   else
10:    compute the two candidate positions and add them to the list L
11:   end if
12:   for  $j = 1, \dots, |L|$  do
13:     if ( $L(j)$  is an arc) then
14:       take D samples from the arc; set  $N = D$ ;
15:       else
16:         set  $N = 1$ ;
17:       end if
18:       // verifying the feasibility of the computed positions
19:       for  $k = 1, \dots, N$  do
20:         if ( $x_i^{j,k}$  is feasible) then
21:           iBP(i + 1, n, d, D);
22:         end if
23:       end for
24:     end for
25:   end if

```

---

- 180 1.  $a < i, b < i, c < i$ ;  
 181 2.  $\{\{b, i\}, \{c, i\}\} \subset E'$  and  $\{a, i\} \in E$ ;  
 182 3.  $\Delta_S(a, b, c) > 0$ ,

183 where  $\Delta_S(a, b, c)$  stands for the area of the triangle formed by  $\{a, b, c\}$ . Assumption (a)  
 184 allows us to place the first 3 atoms uniquely and fixes the realization with respect to rotation  
 185 and translations. Assumptions (b.1) ensures the existence of three reference atoms for every  
 186  $i > 3$ , and assumption (b.2) ensures that at most one of the three reference distances may  
 187 be represented by an interval. Finally, assumption (b.3) requires that the area  $\Delta_S(a, b, c)$  is  
 188 strictly positive, which prevents the references from being collinear. Under these assumptions,  
 189 the MDGP can be discretized.

190 Algorithm 1 is a sketch of the *i*BP algorithm for solving *i*DDGP<sub>3</sub> instances. In the algo-  
 191 rithm call, *i* is the current atom for which the candidate positions are searched, *n* is the total  
 192 number of atoms forming the considered molecule, *d* is the list of available distances (exact  
 193 or interval distances), and *D* is the discretization factor, i.e. the number of sample points that  
 194 are taken from the arcs in case the distance  $d_{ci}$  is represented by an interval. In the algorithm  
 195 (see lines 8 and 10), we make use of a list *L* of positions and arcs, from which candidate  
 196 positions are extracted.

197 Given an order for the vertices in *V* satisfying the assumptions in Definition 1, the algo-  
 198 rithm calls itself recursively in order to explore the tree of candidate positions. Every time  
 199 a new atomic position is computed, it defines a new branch of the tree. This phase in *i*BP  
 200 is named *branching phase*. For every computed atomic position, its feasibility is verified by  
 201 checking the constraints (2), up to the current tree layer, or other additional feasibility criteria  
 202 based on properties of the molecule, e.g. van der Waals' separation distance (VdW), chirality  
 203 constraints, and others [8, 53]. This phase in *i*BP is named *pruning phase*, and the criteria  
 204 are called *pruning devices* (see line 20 of Algorithm 1).



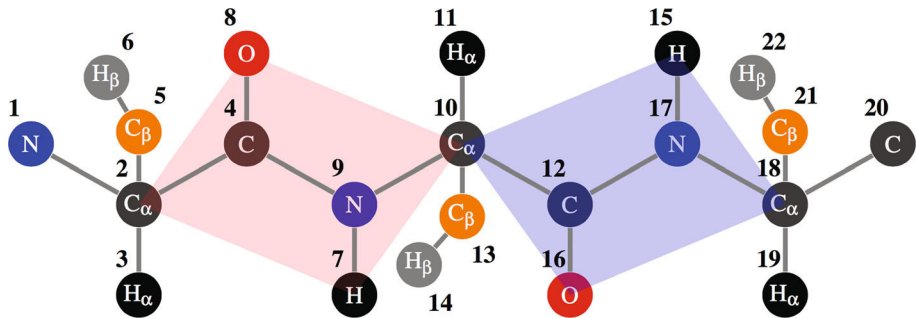


Fig. 2 A model for the protein backbone and a possible discretization order

205 Even if the tree grows exponentially (in the worst case scenario), the pruning devices allow  
 206 *i*BP to focus the search on the feasible parts of the tree. The easiest and most efficient pruning  
 207 device is the Direct Distance Feasibility (DDF) criterion [32], which consists in verifying  
 208 the  $\epsilon$ -feasibility of constraints involving distances between the current atom *i* and previously  
 209 placed atoms:

$$210 \quad \underline{d}(h, i) - \epsilon \leq \|x_h - x_i\| \leq \bar{d}(h, i) + \epsilon, \quad \forall \{h, i\} \in E, \text{ with } h < i \text{ and } h \notin \{a, b, c\}. \quad (3)$$

211 The distances involved in the above constraints are called *pruning distances*.

212 **2.2 Protein backbone model: discretization orders and pruning devices**

213 A necessary preprocessing step for solving MDGPs by this discrete approach consists in  
 214 finding suitable atomic orders allowing each atom *v* to have at least three reference atoms.  
 215 We name such orders *discretization orders* [9]. In previous works, discretization orders have  
 216 been either handcrafted [34], or obtained by looking for paths on *pseudo* de Bruijn graphs  
 217 consisting of cliques of *G* [50], or even automatically detected by a greedy algorithm [31,49].  
 218 In fact, if we consider distances defined by bond lengths and bond angles as exact, along with  
 219 the peptide plane geometry, it is possible to find orders for the protein backbone (and also for  
 220 side chains [11]) satisfying the assumptions required for the discretization. This preprocessing  
 221 step can be performed efficiently, in polynomial time [49], so that the necessary assumptions  
 222 can be fulfilled by graphs related to proteins. However, we point out that when some additional  
 223 assumptions are imposed to the searched orders, such as the *consecutivity* of the reference  
 224 vertices, this problem becomes NP-hard [9]. In this work, we consider a model for the protein  
 225 backbone as depicted in Fig. 2.

226 With the backbone atoms N,  $C_\alpha$ , and C, we also considered the attached H and  $H_\alpha$ , and we  
 227 have included only the  $C_\beta$  and  $H_\beta$  atoms to represent the side chain of each amino acid. There  
 228 are two exceptions for this amino acid model: the glycine, where  $C_\beta$  and  $H_\beta$  are replaced  
 229 by one H, and the proline, which has a missing hydrogen. The backbone model in Fig. 2  
 230 only considers 3 amino-acids, but it can be repeated for all amino-acids in a longer protein  
 231 sequence, because of the regular pattern defining the protein backbone.

232 The order depicted in Fig. 2 is the one used in our numerical experiments.

233 The first three atoms, N– $C_\alpha$ – $H_\alpha$ , of the first amino-acid can be used as initial clique (see  
 234 assumption (a) in Definition 1) for the discretization order because the involved distances are  
 235 defined by bond lengths and angles, that can be considered as exact [12]. Analogously, taking  
 236 into account the peptide plane distances and the distances between hydrogens provided by



237 NMR, it is not hard to verify that assumptions (b.1)–(b.2) of Definition 1 are satisfied by the  
 238 order given in Fig. 2.

239 On the basis of the model in Fig. 2 for the protein backbone geometry, it is possible to  
 240 conceive other pruning devices [8,53] to be integrated with DDF (see, Eq. 3), based on the  
 241 following considerations:

- 242 – Helices in proteins can be either right or left-handed. The former situation is statistically  
 243 more common, because of side chains steric constraints. In this work, we do not consider  
 244 side chains explicitly, but we suppose that it is possible to understand, from an analysis  
 245 of the protein sequence, whether right-handed or left-handed helices are expected to be  
 246 present. We call this pruning device as the *chirality-based device*: in some situations, it can  
 247 allow for placing uniquely some atoms during the execution of the search. For the carbon  
 248 C, in fact, we can get only one (instead of two) possible positions by using  $N-C_\alpha-H_\alpha$   
 249 as reference atoms. An analogous reasoning can be applied to  $C_\beta$ . The chirality defines  
 250 the orientation of the tetrahedron formed by C,  $C_\beta$ , N,  $C_\alpha$ ,  $H_\alpha$ , where  $C_\alpha$  is the chiral  
 251 center, and it can be used to avoid unnecessary branching;
- 252 – The tetrahedron around  $C_\alpha$  forms a clique as well as the peptide planes [2]. Such local  
 253 structures define rigid regions of the protein backbone. Using the peptide plane clique,  
 254 it is possible to find a unique position for  $C_\alpha$ . It is also possible to place N uniquely,  
 255 because its relative orientation with respect to H, C and  $C_\alpha$  of the same peptide plane can  
 256 be computed by taking into account the van der Waals minimum distance;
- 257 – The oxygen atoms in Fig. 2 are included in the model because they participate in hydrogen  
 258 bonds. Each oxygen can be placed uniquely by using the exact distances with the other  
 259 atoms of the peptide plane.

### 260 2.3 Computing coordinates for candidate positions

261 The method employed to compute the candidate positions at each recursive call of Algorithm 1  
 262 has a fundamental importance. While looking for candidate atomic positions for the atom  $i$ , it  
 263 is supposed that the reference atoms  $\{a, b, c\}$  are already positioned. These reference atoms  
 264 define a local coordinate system centered at  $a$  [19,62]. This coordinate system is illustrated  
 265 in Fig. 3.

266 Let  $v_1$  be the vector from  $b$  to  $a$  and  $v_2$  be the vector from  $b$  to  $c$ . The  $x$ -axis for the system  
 267 in  $a$  can be defined by  $v_1$ , and the unit vector in this direction is  $\hat{x} = v_1 / \|v_1\|$ . Moreover, the  
 268 vectorial product  $v_1 \times v_2$  gives another vector that defines the  $z$ -axis, whose corresponding  
 269 unit vector is  $\hat{z}$ . Finally, the vectorial product  $\hat{x} \times \hat{z}$  provides the vector that defines the  $y$ -axis  
 270 (let the unit vector be  $\hat{y}$ ).

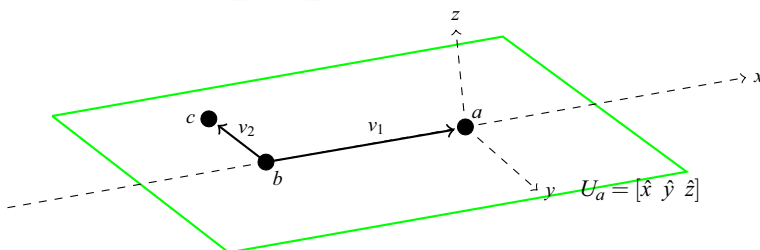


Fig. 3 The reference vertices  $a, b$  and  $c$  induce a system of coordinates

271 These three unit vectors are the columns of a matrix  $U_a = [\hat{x} \ \hat{y} \ \hat{z}]$ , whose role is to  
 272 convert directly vertex positions from the coordinate system defined in  $a$  to the canonical  
 273 system.

274 Once the matrix  $U_a$  has been computed, the canonical Cartesian coordinates for a candidate  
 275 position for the vertex  $i$  can be obtained by:

$$276 \quad x_i(\omega_i) = x_a + U_a \begin{bmatrix} -d_{ai} \cos \theta_i \\ d_{ai} \sin \theta_i \cos \omega_i \\ d_{ai} \sin \theta_i \sin \omega_i \end{bmatrix}, \quad (4)$$

277 where  $\theta_i$  and  $\omega_i$  are the angles related to the spherical coordinates of vertex  $i$ .

278 We will use the symbol  $\theta_i$  in order to refer to the angle formed by the two segments  $(i, a)$   
 279 and  $(a, b)$ , and we will use the symbol  $\omega_i$  to refer to the angle formed by the two planes  
 280 defined by the triplets  $(a, b, c)$  and  $(b, a, i)$  (see Fig. 1). The cosine of the angles  $\theta_i$  and  $\omega_i$   
 281 can be computed by exploiting the positions of the reference vertices  $a, b$  and  $c$ , as well as  
 282 the available distances  $d_{ai}, d_{bi}$  and  $d_{ci}$ . Thus,

$$283 \quad \cos \omega_i = \frac{\cos \theta_{c,b,i} - \cos \theta_{a,b,i} \cos \theta_{a,b,c}}{\sin \theta_{a,b,i} \sin \theta_{a,b,c}},$$

284 where we consider the positive values for the sines, and

$$285 \quad \cos \theta_i = \cos \theta_{b,a,i} = \frac{d_{ab}^2 + d_{ai}^2 - d_{bi}^2}{2 d_{ab} d_{ai}}.$$

286 Recall from Sect. 2.1 that if the three reference distances are all exact, then the three  
 287 spherical shells are in fact three spheres, whose intersection gives 2 points, with probability  
 288 1 [32]. The two points  $x_i^+$  and  $x_i^-$  correspond to two possible opposite values,  $\omega_i^+$  and  $\omega_i^-$ ,  
 289 for the angle  $\omega_i$ . When one of the three distances is instead represented by an interval (see  
 290 Definition 1), the third sphere becomes a spherical shell, and the intersection provides two  
 291 curves (see Fig. 1). These two curves correspond to two intervals,  $[\underline{\omega}_i^+, \bar{\omega}_i^+]$  and  $[\underline{\omega}_i^-, \bar{\omega}_i^-]$ ,  
 292 for the angle  $\omega_i$ . In order to discretize these intervals, a certain number of points, say  $D$ , can  
 293 be chosen from the two curves.

294 As shown in [19], the generalized procedure for the computation of atomic coordinates in  
 295 Algorithm 1, based on equation (4), is very stable when working on MDGP instances related  
 296 to real proteins. Moreover, equation (4) is also at the basis of an important technique that can  
 297 be used to reduce the feasible arcs obtained by sphere intersection. This technique for arc  
 298 reduction was firstly proposed in [20]. Another approach for arc reduction, based on Clifford  
 299 algebra, is presented in [30].

### 300 3 Pruning distances and arc reduction

301 When candidate atomic positions, at each recursive call of the  $iBP$  algorithm (see Algo-  
 302 rithm 1), are computed by intersecting two spheres with one spherical shell, a continuous  
 303 set of positions is obtained, which generally corresponds to two disjoint arcs, related to two  
 304 intervals for the corresponding torsion angle values.

305 During a typical run of Algorithm 1, every time the reference distance  $d_{ci}$  is represented  
 306 by an interval,  $D$  equidistant samples are taken from each arc [34]. As a consequence,  $2D$   
 307 atomic positions are generated in total, and  $2D$  new branches are added to the tree, at the  
 308 current layer, for every branch at the upper level. After their computation, the feasibility  
 309 of each atomic position is verified. On the one hand, too large  $D$  values can drastically

310 increase the width of the tree; on the other hand, too small values can generate trees where  
 311 no solutions can be found (all branches are pruned, because all positions, at a certain layer,  
 312 are not compatible with pruning distances).

313 In [20], an adaptive scheme was proposed for tailoring the branching phase of the *i*BP  
 314 algorithm so that all computed candidate positions are feasible at the current layer. The idea  
 315 is to identify, before the branching phase of the algorithm, the subset of positions on the  
 316 two candidate arcs that is feasible with respect to all pruning distances to be verified on the  
 317 current layer.

318 Let us suppose that, at the current layer *i*, the distance  $d_{ci}$  is represented by the interval  
 319  $[\underline{d}_{ci}, \bar{d}_{ci}]$ . By using Equation (4), two intervals for the angle  $\omega_i$  can be identified:  $[\underline{\omega}_i^+, \bar{\omega}_i^+] \subset$   
 320  $[0, \pi]$  and  $[\underline{\omega}_i^-, \bar{\omega}_i^-] \subset [\pi, 2\pi]$ , such that the distance constraints

$$\begin{aligned}
 & \|x_a - x_i(\omega_i)\| = d_{ai}, \\
 & \|x_b - x_i(\omega_i)\| = d_{bi}, \\
 & \underline{d}_{ci} \leq \|x_c - x_i(\omega_i)\| \leq \bar{d}_{ci},
 \end{aligned} \tag{5}$$

322 are satisfied.

323 However, there may be pruning distances, at layer *i*, that could be exploited for tightening  
 324 these two arcs. Let us suppose there is an  $h \in \{j < i \mid j \notin \{a, b, c\}\}$ , such that the distance  
 325  $d_{hi}$  is known and lies in the interval  $[\underline{d}_{hi}, \bar{d}_{hi}]$ . The solution set of the inequalities

$$\underline{d}_{hi} \leq \|x_h - x_i(\omega_i)\| \leq \bar{d}_{hi} \tag{6}$$

327 consists of intervals for  $\omega_i$  that are compatible with the distance  $d_{hi}$ .

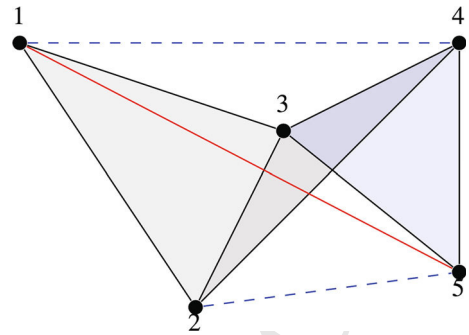
328 A discussion about how to solve the inequalities (6), by using Eq. (4), is presented in  
 329 details in [20].

330 The feasible positions for the atom *i* can be therefore obtained by intersecting the two  
 331 previously computed arcs (in bold in Fig. 1), and several spherical shells, each of them defined  
 332 by considering one pruning distance between *i* and  $h < i$ . For each available pruning distance,  
 333 other inequalities (6) can be defined and new arcs on the circle  $\mathcal{C}$  may be identified. The final  
 334 subset of  $\mathcal{C}$  which is compatible with all available distances can be found by intersecting the  
 335 arcs obtained for each pruning distance with the two initial disjoint arcs, given by Eq. (5).

336 After considering all pruning distances, i.e., after performing all intersections, the final  
 337 result provides a list of arcs on  $\mathcal{C}$  that are feasible with all the distances that can be verified  
 338 at the current layer. All positions that can be taken from these arcs are feasible at the current  
 339 layer: all of them generate a new branch and may serve as a reference for computing new  
 340 candidate positions on deeper layers of the tree. In order to integrate the *i*BP algorithm with  
 341 this adaptive scheme, there are two main changes to be performed on Algorithm 1. On line 8,  
 342 the adaptive scheme needs to be invoked for taking into consideration the information about  
 343 the pruning distances. Moreover, the use of the DDF pruning device has become unnecessary,  
 344 and it should not be considered at line 20 of Algorithm 1.

345 It is important to remark that this adaptive scheme is not supposed to speed up the execution  
 346 of the search, but rather to help in defining search trees that can actually contain solutions.  
 347 Without the use of this adaptive scheme, all sample positions selected from the two arcs  
 348 obtained with the discretization may be discovered to be infeasible as soon as the DDF  
 349 pruning device is invoked. The other extreme situation is instead the one where the adaptive  
 350 scheme can allow us to select a subset of sample positions that all bring to the definition of a  
 351 solution. Naturally, the second situation is desirable, even if, in terms of complexity, it tends  
 352 to increase the total computational cost.

**Fig. 4** Realization of five points in  $\mathbb{R}^3$



**4 Limitations of the current approach: finding approximate realizations**

For DGP instances where all available distances are exact, the presented discrete approach is extremely efficient, allowing for example to realize graphs having thousands of vertices in few seconds with a standard computer [32].

However, for  $iDDGP_3$  instances, there are some difficulties encountered by the  $iBP$  algorithm [34], even for finding one solution. Such limitations, related to the presence of interval data in both discretization and pruning distances, are discussed in this section and a heuristic to overcome such barriers is proposed.

**4.1 Sampled distances and embeddability**

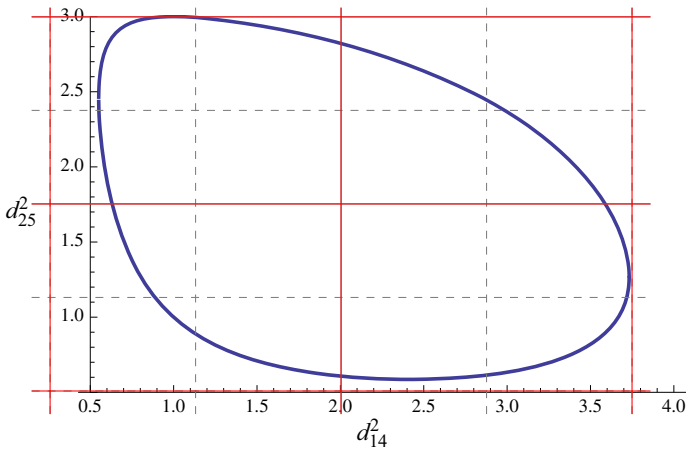
Recent computational experiments have shown that taking equidistant sample points on the feasible arcs (or equidistant samples from the interval distance, see Algorithm 1 in Sect. 2.1), even after the intersection with the available pruning distances (see Sect. 3), is not enough to allow the  $iBP$  algorithm to solve some MDGPs within a predefined precision. The sampled distances are taken independently in each layer of the tree and, in particular for small  $D$  values, it is *not* likely that they are compatible with each other and with other pruning distances available at deeper layers.

The underlying issue is related to the embeddability of a given set of distances. Suppose that we are positioning the atom  $i$  and that the interval distance  $[\underline{d}_{ci}, \bar{d}_{ci}]$  is used in the discretization. Even if we assume that there exists a distance value  $d_{ci}^*$  which is compatible with the other distances in  $E$  leading to a solution, we cannot ensure that, with a finite number  $D$  of samples taken from  $[\underline{d}_{ci}, \bar{d}_{ci}]$ , the compatible distance  $d_{ci}^*$  is actually sampled.

In order to illustrate this fact, consider the following example where five points in  $\mathbb{R}^3$  are embedded (Fig. 4).

Suppose that the straight lines represent exact distances, and let the black lines be the exact distances used in the discretization. The dashed blue lines are the interval distances (used to compute the possible positions of atoms 4 and 5) and the red straight line represents one pruning distance (that can be used to validate the possible positions for the atom 5). The associated distances are the following:  $d_{12} = d_{23} = d_{24} = d_{34} = d_{35} = d_{45} = 1$ ,  $d_{13} = \sqrt{2}$ ,  $d_{14} = \sqrt{x} \in [0.5, 2]$ ,  $d_{15} = \sqrt{3}$ ,  $d_{25} = \sqrt{y} \in [0.5, 2]$ .

According to the Cayley–Menger conditions [38,59], for this set of distances to be realizable in  $\mathbb{R}^3$ , it is necessary that



**Fig. 5** Solution set for the five point Cayley–Menger determinant with  $d_{14}^2$  and  $d_{25}^2$  as missing distances

$$\begin{vmatrix} 0 & 1 & 2 & x & 3 & 1 \\ 1 & 0 & 1 & 1 & y & 1 \\ 2 & 1 & 0 & 1 & 1 & 1 \\ x & 1 & 1 & 0 & 1 & 1 \\ 3 & y & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{vmatrix} = 0,$$

385

386 where the above matrix is a bordered distance matrix and  $|\cdot|$  denotes its determinant. The  
 387 solution set of this equation (the values for the missing (interval) squared distances  $x = d_{14}^2$   
 388 and  $y = d_{25}^2$ ) is represented by the blue curve in Fig. 5.

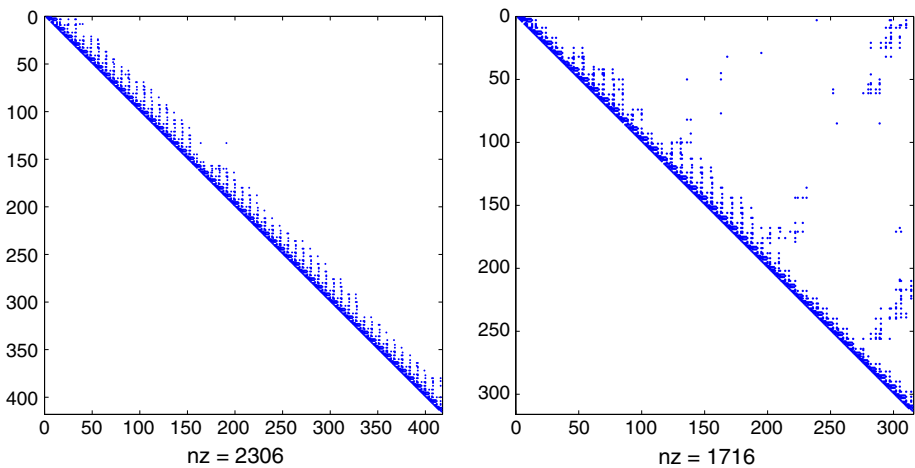
389 It is easy to see that, unless the grid is sufficient refined (number of samples  $D$  is sufficient  
 390 large), a valid pair of distances  $(d_{14}^2, d_{25}^2)$  can be sampled with probability zero.

#### 391 4.2 Long-range distance restraints

392 Long-range distance restraints are related to atoms that are at least four amino-acids apart in  
 393 the protein sequence. Even if far in the protein sequence, some atom pairs may be in condition  
 394 to be detected by an experimental technique. For example, if we consider NMR, it is typical  
 395 to detect distances between atoms that are very far in the sequence, but quite close in space  
 396 ( $\leq 5 \text{ \AA}$ ).

397 In case of all available distances are exact, the pruning distances efficiently guide the  
 398 search in the binary tree corresponding to the discretized search space [32,40]. However,  
 399 when interval distances are present, the search tree is no longer binary, because  $D$  samples  
 400 are taken from each feasible arc. Moreover, the DDF pruning criterion (3) becomes much  
 401 less effective when the bounds  $[\underline{d}, \bar{d}]$  are loose, resulting in a large number of active nodes  
 402 in the tree, which increases exponentially the cost of exploring a whole subtree.

403 Furthermore, since other interval distances are also employed in the discretization, the  
 404 sampled positions in the feasible arcs for previous atoms are only approximations for their  
 405 true positions, and such a sequence of approximate positions may lead to an infeasibility at  
 406 a further layer. For this reason, the longest-range pruning distances may fail to be verified  
 407 (even if they are represented by an interval).



**Fig. 6** Available distances for the instances 1FJK (left) and 2E2F (right)

408 To illustrate this fact, we depict in Fig. 6 the available entries of the upper triangular part of the distance matrices for two instances belonging to our set: 1FJK and 2E2F.  
 409 Notice that the almost-band structure close to the main diagonal is a consequence of the assumptions concerning the discretization. In fact, the distances between pairs  $(i - 1, i)$   
 410 and  $(i - 2, i)$  are generally derived from the bond lengths and angles, while the distances  $(i - 3, i)$  can be generally obtained from the analysis of the torsion angle among the  
 411 quadruplet of atoms  $(i - 3, i - 2, i - 1, i)$ . Moreover, the distance  $(i - 3, i)$  may be also estimated by applying an experimental technique such as NMR. Other distances that are far  
 412 from the main diagonal of the matrix can be obtained applying an experimental technique.  
 413

414 Although 1FJK has more atoms than 2E2F, the former instance can be easily solved by *i*BP in a few seconds, whereas the latter cannot be solved in less than one minute  
 415 (using  $D \leq 20$ ). The difficulty in solving 2E2F is related to the presence of long-range pruning distances: there are several entries in its distance matrix that are far from the diagonal.  
 416  
 417

### 423 4.3 Approximate realizations

424 The presence of interval distances implies uncertainty on the atomic positions obtained by sampling points in the intersection between spheres and spherical shells: even a small error  
 425 introduced at tree layer  $i$  can have a relevant propagation until the layer  $j \gg i$  and, when the pruning distance is finally tested, it is likely that the propagated error leads to infeasible  
 426 positions for atom  $j$ .  
 427

428 Thus, an error introduced during the intersection discretization in a certain tree layer, might make every sampled candidate position infeasible with pruning distances in a further  
 429 layer. This phenomenon is more evident when considering long-range distance restraints. One possibility to avoid pruning out all branches of the search tree, in order to obtain approximate  
 430 solutions to the problem, is to relax the distance constraints related to long-range distances.  
 431 We define the set  
 432

$$433 \mathcal{L} = \{\{i, j\} \in E \mid |i - j| \geq M\}, \quad (7)$$

where  $M$  is a positive integer used to identify long-range distance restraints. Our relaxation consists in avoiding the application of the DDF feasibility test (Eq. 3), as well as the intersection scheme (Sect. 3), to pruning distances in  $\mathcal{L}$ .

Naturally, when such pruning distances are neglected, some information is lost and this can have an impact on the found solutions. In fact, long-range distance restraints are the main responsible for the global fold. Thus, in order to mitigate this effect, we introduce another pruning criterion based on the partial Mean Distance Error (MDE) at the current layer  $k$ :

$$PMDE_k(X) = \frac{1}{|J_k|} \sum_{\{i,j\} \in J_k} \left[ \frac{\max \{ \underline{d}_{i,j} - \|x_i - x_j\|, 0 \}}{\underline{d}_{i,j}} + \frac{\max \{ \|x_i - x_j\| - \bar{d}_{i,j}, 0 \}}{\bar{d}_{i,j}} \right], \quad (8)$$

where

$$J_k = \{\{i, j\} \in E \mid i \leq k \wedge j \leq k\}.$$

Let  $n = |V|$  and note that  $J_n = E$ . It is common to measure the quality of a realization by the Mean Distance Error measure:

$$MDE(X) = PMDE_n(X).$$

Thus, by monitoring the  $PMDE_k(X)$  for  $k < n$ , we can control the quality of partial realizations. This suggests the *PMDE pruning device*: if at layer  $k$ ,  $PMDE_k(X) > \hat{\varepsilon}$ , then the candidate partial realization should be pruned. We set  $\hat{\varepsilon} > \varepsilon$ , where  $\varepsilon$  is the tolerance used in DDF (Eq. 3).

When this new pruning device is introduced, a solution found by Algorithm 1 is actually an approximate solution in the sense that it satisfies all distances in  $E \setminus \mathcal{L}$  (with tolerance  $\varepsilon$ ), while some distances in  $\mathcal{L}$  can be violated. However, the total MDE value for such a solution remains relatively small, because of the new pruning device based on (8). By applying this scheme, together with the chirality and peptide plane constraints (see Sect. 2.2), we expect that the fold of the obtained conformation mimics the fold of the true protein. This is the case for the set of instances used in the computational experiments.

## 5 Computational experiments

In this section we present some computational results on a set of artificially generated MDGP instances. Our aim is to assess the improvements on *iBP* (Algorithm 1) due to the integration of a set of recently proposed techniques: the pruning devices based on chirality and peptide plane geometry, described in Sect. 2.2; the arc reduction technique presented in Sect. 3; and the partial MDE pruning device introduced in Sect. 4.3.

The instances that we consider in our experiments were generated as it follows. The protein conformations were extracted from the PDB: by using the coordinates of a known conformation, all pairwise distances between atom pairs of the backbone were computed. Then, only a small subset of all distance pairs is kept for defining an instance. The distances related to bond lengths and those that can be obtained from bond angles are considered as exact, as well as distances between atoms belonging to the same peptide plane (see Fig 2). Torsion angles on the protein backbones give rise to the definition of interval distances, related to the minimal and maximal extension of such torsion angles. Distances between pairs of hydrogens are also included, as specified in the next subsection.



## 475 5.1 Assumptions concerning distances between hydrogens

476 During the generation of our instances, we rely on the premise that experimental techniques,  
 477 such as NMR spectroscopy, are able to give information about all distances between hydrogen  
 478 atoms that are close in space [35]. Moreover, these distances can be supposed to be more  
 479 precise than other ones. Statistics on such distances [5, 66], with the geometry of consecutive  
 480 peptide planes, validate this assumption.

481 We will consider therefore that all distances between hydrogens belonging to the same  
 482 or to two consecutive amino-acids are available, and we suppose that they lie in an interval  
 483 having width  $0.1\text{\AA}$ . Besides these distances, responsible for the local geometry, we also  
 484 consider distances between hydrogens that belong to amino acids that are far in the protein  
 485 sequence, but close in space. These distances are responsible for the global fold. We include  
 486 these distances in our generated instances whenever they are smaller than  $5\text{\AA}$  and consider  
 487 that imprecisions lead to an interval of width  $1\text{\AA}$ .

488 Hydrogen bonds  $H-O$ , responsible for stabilizing  $\alpha$ -helices and  $\beta$ -strands, are also con-  
 489 sidered. If the distance between  $H$  and  $O$  of distinct amino-acids is greater than  $1.3\text{\AA}$  and  
 490 less than  $3.5\text{\AA}$ , such a distance is included in our instances as an interval of width  $1\text{\AA}$ . All  
 491 intervals have a predefined width and their extremes are randomly generated in a way that  
 492 the interval contains the true distance.

## 493 5.2 Numerical results

494 Let us refer to the algorithm presented in [34] as *iBP*, while we will name “New *iBP*” the  
 495 algorithm integrated with the new method for the computation of candidate positions (see  
 496 Sect. 2.3), with the technique for arc reduction (see Sect. 3), with the chirality and peptide  
 497 plane pruning devices (see Sect. 2.2), and with the pruning device introduced in Sect. 4.3.

498 In both *iBP* variants, the tolerance used in the experiments for the DDF criterion (Eq. 3)  
 499 is  $\varepsilon = 0.001$ . In new *iBP*, for the PMDE-based pruning device, we used  $\hat{\varepsilon} = 0.01$  and set  
 500  $M = 40$  in definition of  $\mathcal{L}$  (see Eq. 7). We gradually increased the number of samples  $D$   
 501 taken from the feasible arcs until the first solution is found in less than 60 s (timeout).

502 The numerical experiments were run in a Intel MacBook Pro, 2Ghz, 2GB RAM, and the  
 503 Algorithm 1 was implemented in C programming language, compiled using GNU GCC with  
 504 flag -O3.

505 Table 1 shows a comparison between *iBP* and New *iBP*. The number of amino acids ( $aa$ ),  
 506 atoms ( $|V|$ ) and available distances ( $|E|$ ) are given for each instance. For the two versions  
 507 of *iBP*, the performance is evaluated by the minimum number of samples  $D$  (taken from  
 508 interval arcs) necessary to find one solution, the number of recursive calls and the CPU time  
 509 in seconds. The quality of the realization is assessed through the MDE. The character “\*”  
 510 means that the instance could not be solved in less than one minute for any value of  $D \leq 20$ .

511 We can notice that the arc reduction technique presented in Sect. 3 is very effective in  
 512 reducing the number of sample positions that we need to extract from the arcs in order to  
 513 obtain at least one solution. This is an important improvement because it is not known a  
 514 priori how many samples are sufficient to allow *iBP* to find a conformation. We can observe  
 515 that the number of calls and CPU time were reduced in 9 out of 11 instances. It is also worth  
 516 to mention that the pruning criteria based on peptide plane geometry and chirality helped  
 517 the new version of *iBP* in reducing the number of calls in some instances and improving the  
 518 global fold as well.

519 Concerning the MDE, the original *iBP* seems to be more stable, although it fails to  
 520 solve four of the instances (within the specified timeout). On the other hand, since the New

**Table 1** Numerical results on artificially generated instances from the PDB

PDB ID	Instance			<i>i</i> BP from [34]				New <i>i</i> BP			
	<i>aa</i>	$ V $	$ E $	<i>D</i>	Calls	Time	MDE	<i>D</i>	Calls	Time	MDE
2JMY	15	120	660	13	37,658	0.13	3e−06	5	2983	0.01	1e−16
2KXA	24	177	973	10	215,669	0.92	5e−06	3	5064	0.01	6e−03
1DSK	28	222	1210	14	31,309	0.13	4e−06	4	53,890	0.14	1e−06
2PPZ	36	287	1522	9	2,372,242	11.34	2e−06	3	442,965	1.87	4e−08
1AQR	40	310	1596	*	*	*	*	4	114,671	0.20	6e−03
2ERL	40	324	1792	14	1,495,282	6.14	4e−06	3	10,410	0.03	1e−03
2E2F	41	315	1716	*	*	*	*	3	19916	0.06	9e−03
1FJK	52	417	2306	12	115,426	0.73	4e−06	4	925,090	3.07	2e−06
2JWU	56	448	2416	*	*	*	*	4	226870	0.81	1e−02
2KIQ	57	455	2452	20	1,217,945	12.79	6e−06	4	317,136	1.12	7e−04
2LOW	64	497	2650	*	*	*	*	3	3,738,152	8.79	2e−07

*i*BP uses the relaxed pruning criterion PMDE, it cannot ensure a better MDE for all the instances. For some of them we observe a better MDE which is a consequence of the other considered pruning devices. Although we relaxed some distance constraints by using PMDE, the chirality constraints helped in improving the local geometry, resulting in a better MDE. For those instances with a worse MDE, like 2KXA, 2E2F or 2KIQ, the PMDE relaxation was, in some sense, the way to “pass-through” the long-range distance constraints and find a realization in a affordable time. Additionally, we remark that an MDE value around  $10^{-3}$  is able to guarantee a sufficient detection of the global fold of the protein. In fact, the realizations found by the New *i*BP are not so far from the true ones. The quality of such realizations is discussed in the next subsection.

### 5.3 Quality of a realization and practical usage

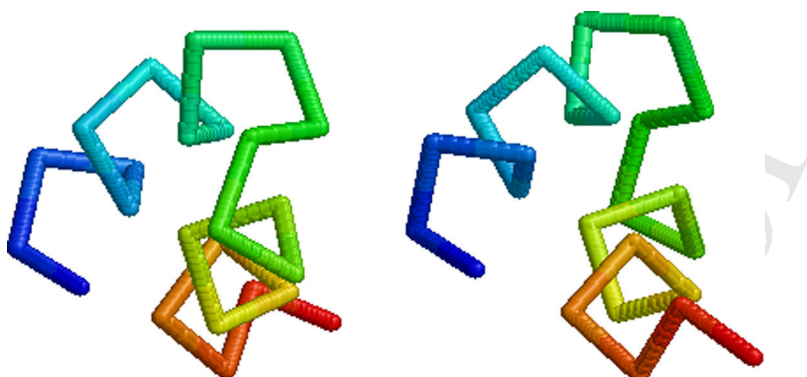
While looking at Table 1, a natural question emerges: how good are the realizations  $X$  with  $MDE(X) \approx 10^{-3}$  when compared to the “true” protein ?

Since we have relaxed the distance constraints related to long-range distances, in principle, we cannot ensure that the underlying molecule is recovered. However, we will illustrate that the realization found by the New *i*BP gives a very good approximation of the true conformation.

First, let us take a look at the instance 2KXA. Figure 7 shows the realizations found (first found solutions) by the original *i*BP and the New *i*BP. Although the MDE of the first is smaller than the second,  $10^{-6}$  against  $10^{-3}$ , we can see that the conformations are roughly the same, except by partial reflections. The New *i*BP produced a right-handed helix because it contains, in its list of pruning devices, the chirality-based device.

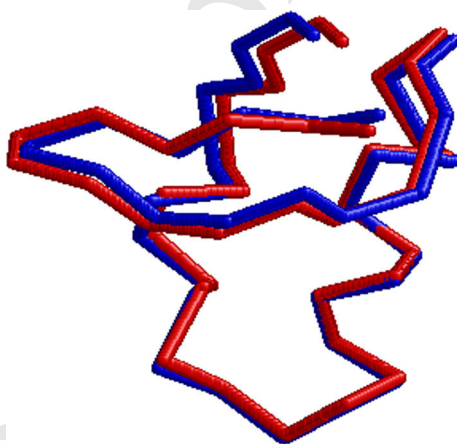
Now, let us consider the instance 2E2F. According to Table 1, the MDE for the realization found by the New *i*BP is approximately  $10^{-2}$ . By superimposing the realizations found with the true one from the PDB (first model), see Fig. 8, we obtain a RMSD value equal to  $0.7 \text{ \AA}$  (according to TM-align [67]).

Therefore, although the realization found by New *i*BP does not fit perfectly with the true conformation, it is close enough to identify its global fold, and it also can be used as a smart



**Fig. 7** Realization for 2KXA found by the original *iBP* (left) and the new *iBP* (right)

**Fig. 8** Superimposition of solution found by the New *iBP* (red) and the original PDB file (blue) for 2E2F



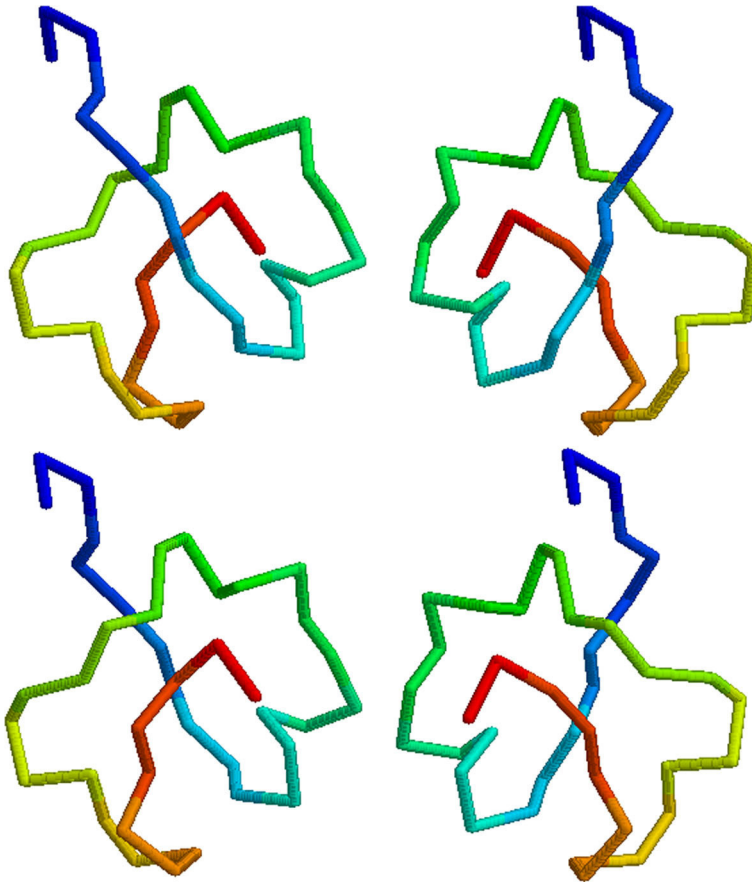
549 starting point for a local optimization intended to minimize the MDE and enforce VdW  
550 constraints [61].

551 Once the first solution  $X_1$  is found by the “New *iBP*”, a set of feasible exact distances  
552 for the distances that were originally represented by intervals can be selected. This set of  
553 distances defines a DGP instance with exact distances which contains  $X_1$  in its finite solution  
554 set. Moreover, by solving such an instance with the basic BP algorithm (for exact distances),  
555 we can compute all other feasible conformations that can be obtained from  $X_1$  by partial  
556 reflections [32,38]. This procedure gets rid of the flexings<sup>1</sup> in the molecule, but only in this  
557 case the solution set is finite.

558 Applying this scheme to a modified 1AQR instance, where hydrogen distances between  
559 consecutive amino-acids were removed and the threshold was lowered to 4.5 Å, four incon-  
560 gruent conformations are obtained, as depicted in Fig. 9.

561 We claim that, even though interval distances pose some difficulties to the extension of  
562 this combinatorial approach, it is still possible to explore all the (discrete) conformational  
563 space obtained with discretization. Henceforth, we propose our New *iBP* as an exploratory  
564 tool to enumerate protein conformations that satisfy most of the given distance restraints,  
565 that can be further improved by local minimization procedures.

<sup>1</sup> Continuous motions of part of the structure preserving all distance restraints.



**Fig. 9** Four incongruent realizations for 1AQR. The conformations in the *bottom* are reflections of the *top* ones. All four conformations differ by partial reflections

## 6 Conclusion and future work

We collected in this paper the most recent and promising advances in solving the MDGP with our combinatorial approach. The main contributions presented in this paper can be summarized as follows:

1. identification of the main barriers against the successful application of the discrete approach to real MDGPs with interval data;
2. another pruning devices based on chirality and peptide planes, whose easy implementation is allowed by the model and discretization order presented in Sect. 2.2;
3. a new pruning device that “relaxes” long-range distance constraints (Sect. 4.3) which allows us to obtain approximate realizations.

Computational experiments on artificially generated instances showed the effectiveness of all above mentioned points, when integrated in the *i*BP algorithm. We are in fact able to find approximate realizations for protein backbones up to 64 amino acids in an affordable time,

579 and with reasonable precision, that can be further improved by using our solutions as starting  
580 points for a local minimization solver.

581 In the presented experiments, the two compared versions of the *iBP* algorithm were  
582 both used for identifying only one solution to the problem. However, as remarked before  
583 and illustrated in Sect. 5.3, the *iBP* algorithm can potentially enumerate the entire solution  
584 set of a discretizable MDGP. Research is currently focused on efficiently enumerating all  
585 conformations belonging to the search tree. Due to the presence of interval distances, many  
586 solutions may belong to the same cluster/ensemble of conformations. Hence, the next step is  
587 to define a method to classify the solutions in equivalence classes and integrate *iBP* with an  
588 scheme able to pick only one representative conformation from each incongruent ensemble.

589 **Acknowledgements** DG and CL are thankful to the Brazilian research agencies FAPESP and CNPq for  
590 partial financial support. LL was partially supported by the “Bip:Bip” project within the ANR “Investissement  
591 d’Avenir” program. AM thanks University of Rennes 1 for financial support. AM and DG also acknowledge  
592 Brittany Region (France) for partial financial support.

## 593 References

- 594 1. Alfakih, A.Y., Khandani, A., Wolkowicz, H.: Solving Euclidean distance matrix completion problems via  
595 semidefinite programming. *Comput. Optim. Appl.* **12**, 13–30 (1999)
- 596 2. Alipanahi, B., Krislock, N., Ghodsi, A., Wolkowicz, H., Donaldson, L., Li, M.: Determining protein  
597 structures from NOESY distance constraints by semidefinite programming. *J. Comput. Biol.* **20**, 296–310  
598 (2013)
- 599 3. Almeida, F.C.L., Moraes, A.H., Gomes-Neto, F.: An overview on protein structure determination by  
600 NMR: historical and future perspectives of the use of distance geometry methods. In: Mucherino et al.  
601 [52], pp. 377–412
- 602 4. Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I., Bourne, P.: The  
603 Protein Data Bank. *Nucl. Acids Res.* **28**, 235–242 (2000)
- 604 5. Billeter, M., Braun, W., Wüthrich, K.: Sequential resonance assignments in protein <sup>1</sup>H nuclear magnetic  
605 resonance spectra. Computation of sterically allowed proton-proton distances and statistical analysis of  
606 proton-proton distances in single crystal protein conformations. *J. Mol. Biol.* **155**, 321–346 (1982)
- 607 6. Biswas, P., Lian, T., Wang, T., Ye, Y.: Semidefinite programming based algorithms for sensor network  
608 localization. *ACM Trans. Sens. Netw.* **2**, 188–220 (2006)
- 609 7. Bizien, T., Durand, D., Roblina, P., Thureau, A., Vachette, P., Pérez, J.: A brief Survey of State-of-the-Art  
610 BioSAXS. *Protein Pept. Lett.* **23**, 217–231 (2016)
- 611 8. Cassioli, A., Bordeaux, B., Bouvier, G., Mucherino, A., Alves, R., Liberti, L., Nilges, M., Lavor, C.,  
612 Malliavin, T.: An algorithm to enumerate all possible protein conformations verifying a set of distance  
613 constraints. *BMC Bioinform.* **16**, 16–23 (2015)
- 614 9. Cassioli, A., Gunluk, O., Lavor, C., Liberti, L.: Discretization vertex orders in distance geometry. *Discrete*  
615 *Appl. Math.* **197**, 27–41 (2015)
- 616 10. Chen, Z.A., Jawhari, A., Fischer, L., Buchen, C., Tahir, S., Kamenski, T., Rasmussen, M., Lariviere,  
617 L., Bukowski-Wills, J.-C., Nilges, M., Cramer, P., Rappsilber, J.: Architecture of the RNA polymerase  
618 II-TFIIF complex revealed by cross-linking and mass spectrometry. *EMBO J.* **29**, 717–726 (2010)
- 619 11. Costa, V., Mucherino, A., Lavor, C., Cassioli, A., Carvalho, L.M., Maculan, N.: Discretization orders for  
620 protein side chains. *J. Glob. Optim.* **60**, 333–349 (2014)
- 621 12. Crippen, G., Havel, T.: *Distance Geometry and Molecular Conformation*. Wiley, New York (1988)
- 622 13. Dattorro, J.: *Convex Optimization and Euclidean Distance Geometry*. *Μεθοο*, Palo Alto (2005)
- 623 14. Dokmanic, I., Parhizkar, R., Ranieri, J., Vetterli, M.: Euclidean distance matrices: essential theory, algo-  
624 rithms, and applications. *Sig. Process. Mag. IEEE* **32**(6), 12–30 (2015)
- 625 15. Dong, Q., Wu, Z.: A linear-time algorithm for solving the molecular distance geometry problem with  
626 exact inter-atomic distances. *J. Glob. Optim.* **22**, 365–375 (2002)
- 627 16. Dong, Q., Wu, Z.: A geometric build-up algorithm for solving the molecular distance geometry problem  
628 with sparse distance data. *J. Glob. Optim.* **26**(3), 321–333 (2003). doi:[10.1023/A:1023221624213](https://doi.org/10.1023/A:1023221624213)
- 629 17. Ferguson, D., Marsh, A., Metzger, T., Garrett, D., Kastella, K.: Conformational searches for the global  
630 minimum of protein models. *J. Glob. Optim.* **4**, 209–227 (1994)

- 631 18. Fiorioto, F., Damberger, F., Herrmann, T., Wüthrich, K.: Automated amino acid side-chain NMR assign-  
632 ment of proteins using 13C- and 15N-resolved 3D [1H,1H]-NOESY. *J. Biomol. NMR* **42**, 23–33 (2008)
- 633 19. Gonçalves, D.S., Mucherino, A.: Discretization orders and efficient computation of cartesian coordinates  
634 for distance geometry. *Optim. Lett.* **8**, 2111–2125 (2014)
- 635 20. Gonçalves, D.S., Mucherino, A., Lavor, C.: An adaptive branching scheme for the branch & prune  
636 algorithm applied to distance geometry. In: *IEEE Conference Proceedings*, pp. 463–469. Workshop on  
637 Computational Optimization (WCO14), FedCSIS14, Warsaw, Poland (2014)
- 638 21. Grand, S.L., Merz, K.: The application of the genetic algorithm to the minimization of potential energy  
639 functions. *J. Glob. Optim.* **3**, 49–66 (1993)
- 640 22. Guerry, P., Duong, V.D., Herrmann, T.: CASD-NMR 2: robust and accurate unsupervised analysis of raw  
641 NOESY spectra and protein structure determination with UNIO. *J. Biomol. NMR* **62**, 473–480 (2015)
- 642 23. Güntert, P.: Automated NMR structure calculation with CYANA. *Methods Mol. Biol.* **278**, 353–378  
643 (2004)
- 644 24. Herrmann, T., Güntert, P., Wüthrich, K.: Protein NMR structure determination with automated NOE  
645 assignment using the new software CANDID and the torsion angle dynamics algorithm DYANA. *J. Mol.  
646 Biol.* **319**, 209–227 (2002)
- 647 25. Herrmann, T., Güntert, P., Wüthrich, K.: Protein NMR structure determination with automated NOE-  
648 identification in the NOESY spectra using the new software ATNOS. *J. Biomol. NMR* **24**, 171–189  
649 (2002)
- 650 26. L, Hoai An: Solving large scale molecular distance geometry problems by a smoothing technique via the  
651 Gaussian transform and d.c. programming. *J. Glob. Optim.* **27**, 375–397 (2003)
- 652 27. Huang, H.X., Liang, Z.A., Pardalos, P.: Some properties for the Euclidean distance matrix and positive  
653 semidefinite matrix completion problems. *J. Glob. Optim.* **25**, 3–21 (2003)
- 654 28. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**, 671–680  
655 (1983)
- 656 29. Krislock, N., Wolkowicz, H.: Explicit sensor network localization using semidefinite representations and  
657 facial reductions. *SIAM J. Optim.* **20**, 2679–2708 (2010)
- 658 30. Lavor, C., Alves, R., Figueiredo, W., Petraglia, A., Maculan, N.: Clifford algebra and the discretizable  
659 molecular distance geometry problem. *Adv. Appl. Clifford Algebr.* **25**, 925–942 (2015)
- 660 31. Lavor, C., Lee, J., John, A.L.S., Liberti, L., Mucherino, A., Sviridenko, M.: Discretization orders for  
661 distance geometry problems. *Optim. Lett.* **6**, 783–796 (2012)
- 662 32. Lavor, C., Liberti, L., Maculan, N., Mucherino, A.: The discretizable molecular distance geometry prob-  
663 lem. *Comput. Optim. Appl.* **52**, 115–146 (2012)
- 664 33. Lavor, C., Liberti, L., Maculan, N., Mucherino, A.: Recent advances on the discretizable molecular  
665 distance geometry problem. *Eur. J. Oper. Res.* **219**, 698–706 (2012)
- 666 34. Lavor, C., Liberti, L., Mucherino, A.: The interval branch-and-prune algorithm for the discretizable  
667 molecular distance geometry problem with inexact distances. *J. Glob. Optim.* **56**, 855–871 (2013)
- 668 35. Lavor, C., Mucherino, A., Liberti, L., Maculan, N.: On the computation of protein backbones by using  
669 artificial backbones of hydrogens. *J. Glob. Optim.* **50**, 329–344 (2011)
- 670 36. Liberti, L., Lavor, C., Maculan, N.: A branch-and-prune algorithm for the molecular distance geometry  
671 problem. *Int. Trans. Oper. Res.* **15**, 1–17 (2008)
- 672 37. Liberti, L., Lavor, C., Maculan, N., Marinelli, F.: Double variable neighbourhood search with smoothing  
673 for the molecular distance geometry problem. *J. Glob. Optim.* **43**, 207–218 (2009)
- 674 38. Liberti, L., Lavor, C., Maculan, N., Mucherino, A.: Euclidean distance geometry and applications. *SIAM  
675 Rev.* **56**, 3–69 (2014)
- 676 39. Liberti, L., Lavor, C., Mucherino, A., Maculan, N.: Molecular Distance Geometry Methods: from Con-  
677 tinuous to Discrete. *Int. Trans. Oper. Res.* **18**, 33–51 (2011)
- 678 40. Liberti, L., Lavor, C., Mucherino, A.: The discretizable molecular distance geometry problem seems  
679 easier on proteins. In: Mucherino, A., Lavor, C., Liberti, L., Maculan, N. (eds.) *Distance Geometry*, pp.  
680 47–60. Springer, New York (2013)
- 681 41. Liberti, L., Masson, B., Lee, J., Lavor, C., Mucherino, A.: On the number of realizations of certain  
682 Henneberg graphs arising in protein conformation. *Discrete Appl. Math.* **165**, 213–232 (2014)
- 683 42. Linge, J.P., Habeck, M., Rieping, W., Nilges, M.: ARIA: automated NOE assignment and NMR structure  
684 calculation. *Bioinformatics* **19**, 315–316 (2003)
- 685 43. Locatelli, M., Schoen, F.: Minimal interatomic distance in morse clusters. *J. Glob. Optim.* **22**(1), 175–190  
686 (2002). doi:[10.1023/A:1013811230753](https://doi.org/10.1023/A:1013811230753)
- 687 44. Malliavin, T., Mucherino, A., Nilges, M.: Distance geometry in structural biology: new perspectives. In:  
688 Mucherino et al. [52], pp. 329–350
- 689 45. Man-Cho So, A., Ye, Y.: Theory of semidefinite programming for sensor network localization. *Math.  
690 Program. B* **109**, 367–384 (2007)



- 691 46. Maranas, C., Floudas, C.: Global minimum potential energy conformations of small molecules. *J. Glob.*  
692 *Optim.* **4**, 135–170 (1994)
- 693 47. Moré, J., Wu, Z.: Distance geometry optimization for protein structures. *J. Glob. Optim.* **15**(3), 219–234  
694 (1999). doi:[10.1023/A:1008380219900](https://doi.org/10.1023/A:1008380219900)
- 695 48. Moré, J., Wu, Z.: Distance geometry optimization for protein structures. *J. Glob. Optim.* **15**, 219–223  
696 (1999)
- 697 49. Mucherino, A.: On the identification of discretization orders for distance geometry with intervals. In:  
698 Proceedings of Geometric Science of Information (GSI13), pp. 231–238. Lecture Notes in Computer  
699 Science 8085, Paris, France (2013)
- 700 50. Mucherino, A.: A pseudo De Bruijn graph representation for discretization orders for distance geometry.  
701 In: Proceedings of the 3rd International Work-Conference on Bioinformatics and Biomedical Engineering  
702 (IWBBIO15), Part I, Lecture Notes in Bioinformatics, vol. 9043, pp. 514–523. Granada, Spain (2015)
- 703 51. Mucherino, A., Lavor, C., Liberti, L.: The discretizable distance geometry problem. *Optim. Lett.* **6**,  
704 1671–1686 (2012)
- 705 52. Mucherino, A., Lavor, C., Liberti, L., Maculan, N. (eds.): Distance Geometry: Theory, Methods and  
706 Applications. Springer, New York (2013)
- 707 53. Mucherino, A., Lavor, C., Malliavin, T., Liberti, L., Nilges, M., Maculan, N.: Influence of pruning devices  
708 on the solution of molecular distance geometry problems. In: Pardalos, P.M., Rebennack, S. (eds.) Pro-  
709 ceedings of the 10th International Symposium on Experimental Algorithms (SEA11), Lecture Notes in  
710 Computer Science, vol. 6630, pp. 206–217. Crete, Greece (2011)
- 711 54. Ryu, J., Kim, D.S.: Protein structure optimization by side-chain positioning via beta-complex. *J. Glob.*  
712 *Optim.* **57**(1), 217–250 (2013). doi:[10.1007/s10898-012-9886-3](https://doi.org/10.1007/s10898-012-9886-3)
- 713 55. Santana, R., Larrañaga, P., Lozano, J.: Side chain placement using estimation of distribution algorithms.  
714 *Artif. Intell. Med.* **39**, 49–63 (2007)
- 715 56. Saxe, J.B.: Embeddability of weighted graphs in  $k$ -space is strongly NP-hard. In: Proceedings of 17th  
716 Allerton Conference in Communications, Control and Computing, pp. 480–489. Monticello, IL (1979)
- 717 57. Schlick, T.: Molecular Modelling and Simulation: An Interdisciplinary Guide. Springer, New York (2002)
- 718 58. Schoenberg, I.: Remarks to Maurice Fréchet's article "Sur la définition axiomatique d'une classe d'espaces  
719 distanciés vectoriellement applicable sur l'espace de Hilbert". *Ann. Math.* **36**, 724–732 (1935)
- 720 59. Sippl, M., Scheraga, H.: Cayley–Menger coordinates. *Proc. Natl. Acad. Sci. USA* **83**, 2283–2287 (1986)
- 721 60. Sit, A., Wu, Z.: Solving a generalized distance geometry problem for protein structure determination.  
722 *Bull. Math. Biol.* **73**, 2809–2836 (2011)
- 723 61. Souza, M., Lavor, C., Murtitba, A., Maculan, N.: Solving the molecular distance geometry problem with  
724 inaccurate distance data. *BMC Bioinform.* **14**(Suppl. 9):S7, 1–6 (2013)
- 725 62. Thompson, H.: Calculation of cartesian coordinates and their derivatives from internal molecular coordi-  
726 nates. *J. Chem. Phys.* **47**, 3407–3410 (1967)
- 727 63. Volk, J., Herrmann, T., Wüthrich, K.: Automated sequence-specific protein NMR assignment using memetic  
728 algorithm MATCH. *J. Biomol. NMR* **41**, 127–138 (2008)
- 729 64. Wu, D., Wu, Z.: An updated geometric build-up algorithm for solving the molecular distance geometry  
730 problem with sparse distance data. *J. Glob. Optim.* **37**, 661–673 (2007)
- 731 65. Wu, D., Wu, Z., Yuan, Y.: Rigid versus unique determination of protein structures with geometric buildup.  
732 *Optim. Lett.* **2**, 319–331 (2008)
- 733 66. Wüthrich, K., Billeter, M., Braun, W.: Pseudo-structures for the 20 common amino acids for use in  
734 studies of protein conformations by measurements of intramolecular proton-proton distance constraints  
735 with Nuclear Magnetic Resonance. *J. Mol. Biol.* **169**, 949–961 (1983)
- 736 67. Zhang, Y., Skolnick, J.: TM-align: a protein structure alignment algorithm based on TM-score. *Nucl.*  
737 *Acids Res.* **33**, 2302–2309 (2005)
- 738 68. Zou, Z., Bird, R., Schnabel, R.: A stochastic/perturbation global optimization algorithm for distance  
739 geometry problems. *J. Glob. Optim.* **11**(1), 91–105 (1997). doi:[10.1023/A:1008244930007](https://doi.org/10.1023/A:1008244930007)