

On a relationship between graph realizability and distance matrix completion

Leo Liberti and Carlile Lavor

Abstract We consider a certain subclass of Henneberg-type edge-weighted graphs which is related to protein structure, and discuss an algorithmic relationship between the DISTANCE GEOMETRY PROBLEM and the EUCLIDEAN DISTANCE MATRIX COMPLETION PROBLEM.

1 Introduction

The structure of proteins is strongly related to its function. Efforts for finding the three-dimensional structure of proteins include minimization of a potential energy function and exploitation of known chemical properties such as inter-atomic distances [29]. Such distances may be known because they refer to covalent bonds and angles, or because they can be found using Nuclear Magnetic Resonance (NMR) [4]. In this paper we focus on finding the protein structure using distance information only.

2 The Distance Geometry Problem

We employ an abstract generalized model of this problem, whereby we look for the realization in \mathbb{R}^K of a weighted simple undirected graph $G = (V, E, d)$, where we assume, to avoid the trivial case, that $n = |V| > K$. In the case of proteins, $K = 3$, V

L. Liberti
LIX, Ecole Polytechnique, 91128 Palaiseau, France; e-mail: liberti@lix.polytechnique.fr

C. Lavor
Dept. of Applied Math. (IMECC-UNICAMP) State University of Campinas, 13081-970, Campinas - SP Brazil; e-mail: clavor@ime.unicamp.br

represents the set of atoms, E represents the set of atom pairs for which a distance is available, and $d : E \rightarrow \mathbb{R}_+$ is the edge weight function encoding the distances. Given a positive integer K and a weighted simple undirected graph $G = (V, E, d)$, the DISTANCE GEOMETRY PROBLEM (DGP) asks whether there exists a realization $x : V \rightarrow \mathbb{R}^K$ such that:

$$\forall \{u, v\} \in E \quad \|x_u - x_v\|^2 = d_{uv}^2, \quad (1)$$

where the norm is assumed to be the Euclidean norm. In order to fix an orthogonal frame of reference and to avoid translations and rotations, we assume that a subset $V_0 = \{v_1, \dots, v_K\} \subseteq V$ and a partial realization $x' : V_0 \rightarrow \mathbb{R}^K$ are also given as part of the input. The DGP is also called the *graph realization problem*. Realizations satisfying Eq. (1) are called *valid*. Once a valid realization is found, distances between *all* pairs of vertices (not just those in E) can be determined. Formally, this extends $d : E \rightarrow \mathbb{R}_+$ to a function $\bar{d} : V \times V \rightarrow \mathbb{R}_+$. The values of the function \bar{d} can be arranged into a square *Euclidean distance matrix* on the point set $\mathcal{X} = \{x_v \mid v \in V\} \subseteq \mathbb{R}^K$. The pair (\mathcal{X}, \bar{d}) is known as a *distance space* [1].

3 The Euclidean Distance Matrix Completion Problem

In the EUCLIDEAN DISTANCE MATRIX COMPLETION PROBLEM (EDMCP) [8], the input is a partial square symmetric matrix A (i.e. a symmetric matrix where certain entries are missing) and the output is a pair (\bar{A}, K) where \bar{A} is a symmetric completion of A , and $K \in \mathbb{N}$ such that: (a) \bar{A} is a Euclidean distance matrix in \mathbb{R}^K and (b) K is minimum possible. We consider here a variant of the EDMCP, which we call EDMCP $_K$, where K is actually given as part of the input and the output certificate for YES instances only consists of the completion matrix \bar{A} of the partial matrix A as a Euclidean distance matrix (\bar{A} is also called a *valid completion* of A). It is easy to see that the EDMCP $_K$ is strongly related to the DGP: if x is a valid realization of G , then the partial distance matrix can be completed in polynomial time, and if \bar{A} is a valid completion of A , then the corresponding DGP graph is a clique, whose realization in \mathbb{R}^K can be found in polynomial time [3].

This mapping in the output parallels a mapping in the input data. A partial square symmetric matrix $A = (a_{uv})$ with missing components indexed by the set \bar{E} of unordered index pairs $\{u, v\}$ encodes the weighted simple undirected graph $G = (V, E, a)$ where V is the set of row/column indices, E is the complement of \bar{E} with respect to the set of all unordered pairs of V , and the edge weight a maps $\{u, v\}$ to a_{uv} . Conversely, a weighted simple undirected graph $G = (V, E, a)$ can be encoded in a partial square symmetric matrix A where the $\{u, v\}$ -th component is a_{uv} for all $\{u, v\} \in E$ and the other components are missing. We formalize this correspondence by setting $\mathcal{M}(G) = A$ for a graph G and its corresponding partial matrix A , and $\mathcal{G}(A) = G$ for a partial matrix A and its corresponding graph G . It is trivial to see that \mathcal{M} and \mathcal{G} are inverse operators.

4 Rigidity and Henneberg type graphs

The DGP refers to a field of study which is known as Distance Geometry (DG). DG was formally started in the 1930s, when Menger found how to decide whether a given square matrix encodes a distance matrix using Cayley determinants [19]. Blumenthal then extended Menger’s findings to a well-developed theory [1], and re-christened Cayley determinants “Cayley-Menger determinants”. The study of realizations of graphs in the plane and in space, however, dates much further back. The ancient Greeks were concerned with finding all polyhedra in space, for example. Statics, which is necessary to ensure that buildings will not collapse under the action of external forces, has existed ever since man got tired of being rained on and decided to build himself a roof. Realizations of graphs in space from the point of view of statics are known as “bar-and-joint frameworks”. Several important results on the rigidity of such frameworks date from the end of the XIX century [2,28]. Henneberg [6] was the first to formalize an iterative procedure for verifying whether such frameworks are rigid. In particular, one of his two “steps” (known as *Henneberg type I* step [7,30]) can be paraphrased (and generalized) as follows: if there is an order on V such that the first K vertices have a known realization, and such that every subsequent vertex is adjacent to at least K predecessors, then the graph almost certainly has a rigid realization in \mathbb{R}^K . This idea was already present in the works of Saviotti [27], as testified by the two-dimensional case shown in Fig. 1. The

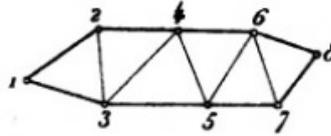


Fig. 1 Fig. 30 in [27].

set of Henneberg type I graphs gives rise to a subset of DGP instances known as the DISCRETIZABLE DGP (DDGP) [21].

In the above paragraph we used the term “rigid framework” and “almost certainly” intuitively. These can be formally defined as follows. A framework is a pair (G, x) where G is a graph and x is a valid realization; a flexing of a framework is a continuous map p from $[0, 1]$ to the set of all realizations of V such that $p(0) = x$, $p(t)$ satisfies Eq. (1) for all $t \in [0, 1]$, and $p(t)$ is not an isometry of x for all $t \in (0, 1]$. A framework is rigid if it has no flexing. As for “almost certainly rigid”, this means that the set of realizations which are not rigid has Lebesgue measure 0 in the set of all possible realizations.

If the Henneberg type I order is not explicitly given, it may not be immediately obvious how to find one. The problem of finding a Henneberg type I order is defined in [9] as the DISCRETIZATION VERTEX ORDER PROBLEM (DVOP). There is an exponential algorithm $O(n^{K+3})$ for solving the DVOP, which is polynomial for

fixed K . Implementations for $K = 3$ are very fast and can successfully be used as a preprocessing step to solving the DDGP.

5 Branch-and-Prune

For almost all edge weightings, Henneberg type I graphs can have finitely many different valid realizations whose corresponding distance spaces are incongruent. With a slight abuse of notation, we call two different valid realizations with incongruent distance spaces *incongruent realizations*. If vertex v has *exactly* K adjacent predecessors, then x_v is at the intersection P of K spheres in \mathbb{R}^K . The cardinality of P is in $\{0, 1, 2\}$ as long as the position of the K adjacent predecessors of v affinely spans a $(K - 1)$ -dimensional subspace of \mathbb{R}^K . The case $|P| = 0$ occurs when the edge weighting is such that G has no realization in \mathbb{R}^K . The case $|P| = 1$ only occurs when the subgraph induced by v and its K adjacent predecessors defines a flat simplex in \mathbb{R}^K . Since the set of flat simplices have Lebesgue measure 0 in the set of all simplices, this is a case which can be ignored almost all the time. The remaining case is $|P| = 2$, shown for $K = 3$ in Fig. 2. Thus, one can find all incongruent

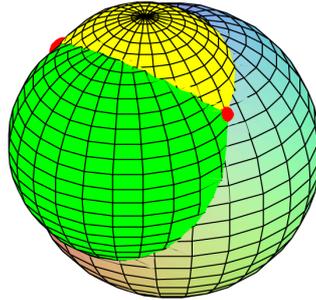


Fig. 2 General case for the intersection P of three spheres in \mathbb{R}^3 .

realizations of a Henneberg type I graph G by the following method:

- (a) place the first K vertices arbitrarily (this essentially fixes the reference system up to reflection);
- (b) place the v -th vertex in the Henneberg type I order in one of the points in P ;
- (c) for each position x_v for v in P , recursively call Step (b) with v replaced by $v + 1$.

In the worst case (i.e. whenever $|P|$ is always 2 and there are no other edges but those that define the Henneberg type I step), this gives rise to a full binary search

tree after level K , which amounts to 2^{n-K} different realizations, 2^{n-K-1} of which are incongruent, the other 2^{n-K-1} being their reflection through the first K vertices [11, Thm. 2]. We let X be the set of all realizations found by this method.

We remark that the recursive call at Step (c) may occur fewer than twice whenever vertex v has more than K adjacent predecessors, as the intersection of more than K spheres in \mathbb{R}^K almost always has either 0 or 1 point. Thus, vertices with more than K adjacent predecessors are used to “prune out” certain branches of the binary search tree. This is why the corresponding algorithm is called *Branch-and-Prune* (BP) [10, 17]. The BP algorithm was originally only defined for immediate predecessors [11], but was henceforth extended to work in several different situations: for Henneberg type I graphs [21], for certain types of interval-weighted graphs related to proteins [12, 15, 20], and for the purpose of overcoming a technical limitation of NMR machinery, which generally only provides reliable distance measures for pairs of hydrogen atoms [13, 14, 16, 22]. A publically available BP implementation is described in [25]. The current computational state-of-the-art for the BP algorithm is attained with a parallel BP implementation [23, 24], which can realize a protein backbone of 10^4 atoms in \mathbb{R}^3 in just over 10s of CPU time on a cluster of 8 nodes.

Step (b) of the BP algorithm is formalized in Alg. 1. It takes as input a vertex v of rank $\rho(v) > K$, a partial realization \bar{x} on the predecessors of v and a set X which will contain all the valid realizations at the end of the execution. We identify for convenience V with the set $[n] = \{1, \dots, n\}$ of vertex ranks, we denote the set of vertices adjacent to v by $N(v)$, and we use $S_{uv}^{\bar{x}}$ to denote the sphere centered at \bar{x}_u with radius d_{uv} . The BP algorithm starts with the call $\text{BP}(K+1, x', \emptyset)$, where x' is the (given) realization of the first K vertices. It was shown in [18, Lemma 3.4] that

Algorithm 1 $\text{BP}(v, \bar{x}, X)$

Require: A vertex $v \in V \setminus [K]$, a partial realization $\bar{x} = (x_1, \dots, x_{v-1})$, a set X .

```

1:  $P = \bigcap_{\substack{u \in N(v) \\ u < v}} S_{uv}^{\bar{x}}$ 
2: for  $p \in P$  do
3:    $x \leftarrow (\bar{x}, p)$ 
4:   if  $\rho(v) = n$  then
5:      $X \leftarrow X \cup \{x\}$ 
6:   else
7:      $\text{BP}(v+1, x, X)$ 
8:   end if
9: end for

```

the BP algorithm finds all incongruent solutions of the DDGP, and in [18, Prop. 3.5] that for almost all instances, no two distinct search tree nodes at a given level v will be such that one node has two subnodes and the other node only one.

5.1 Partial reflections

What do incongruent realizations of G look like? Partition the edges of G in those edges which ensure the existence of a Henneberg type I order (which we call *discretization edges*) and all the other edges (which we call *pruning edges*). Let \bar{X} be the set of all realizations of the subgraph of G defined by the discretization edges. Then, by definition, $|\bar{X}| = 2^{n-K}$. A *partial reflection* of $x \in \bar{X}$ with respect to a vertex $v > K$ is a map $\pi : (V \setminus [K]) \times \bar{X} \rightarrow \bar{X}$ such that $\pi_v = I^v \times R_{v,x}^{n-v}$, where $R_{v,x}$ is the reflection operator through the hyperplane defined by x_{v-K}, \dots, x_{v-1} ; in other words, $\pi_v(x) = (x_1, \dots, x_{v-1}, R_{v,x}(x_v), \dots, R_{v,x}(x_n))$. As remarked in [5, Sect. 2.1], partial reflections are also maps $X \rightarrow X$ (a proof of this is found in [18, Thm. 4.9]). A strong converse is also true for DDGP instances where each vertex is adjacent to at least K *immediate* predecessors (such instances are collectively known as the DISCRETIZABLE MOLECULAR DGP in general dimensions, or K DMDGP, see [11] for the case with $K = 3$), namely, that for any distinct $x, y \in X$ there is a composition ρ of partial reflections such that $y = \rho(x)$ [18, Thm. 5.4].

6 BP in distance space

As remarked in [26], the completion in \mathbb{R}^3 of a distance (sub)matrix with the following structure:

$$\begin{pmatrix} 0 & d_{12} & d_{13} & d_{14} & \boxed{\delta} \\ d_{21} & 0 & d_{23} & d_{24} & d_{25} \\ d_{31} & d_{32} & 0 & d_{34} & d_{35} \\ d_{41} & d_{42} & d_{43} & 0 & d_{45} \\ \boxed{\delta} & d_{52} & d_{53} & d_{54} & 0 \end{pmatrix} \quad (2)$$

can be carried out in constant time by solving a quadratic system in the unknown δ derived from setting the Cayley-Menger determinant [9, Eq. (9)] of the distance space (\mathcal{X}, d) to zero, where $\mathcal{X} = \{x_1, \dots, x_5\}$ and d is given by Eq. (2). This is because, for general K , the Cayley-Menger determinant is proportional to the K -volume of the simplex on $K + 1$ points, which is the (unique, up to rotations and translations) realization of the weighted 5-clique defined by a full distance matrix. Since a simplex on 5 points embedded in \mathbb{R}^3 necessarily has 4-volume equal to zero, it suffices to set the Cayley-Menger determinant of (2) to zero to obtain a quadratic equation in δ . We denote the pair $\{u, v\}$ indexing the unknown distance δ by $U(D)$, the Cayley-Menger determinant of a matrix D by $\text{CM}(D)$, and the corresponding quadratic equation in δ by $\text{CM}(D, \delta) = 0$. This equation has real solutions only if (2) is a Euclidean distance matrix. Furthermore, if it has real solutions at all, it almost always has two distinct solutions δ^1, δ^2 . These are two valid values for the missing distance d_{15} . This observation trivially extends to general K , where we consider a $K + 2$ point simplex realization of a weighted near-clique on $K + 2$ vertices with one missing edge.

6.1 The main idea

We consider a Henneberg type I graph G and a partial embedding \bar{x} for the subgraph $G[[K]]$ of G induced by the set $[K]$ of the first K vertices. We can find the two possible positions for v_{K+1} with just one call to $\text{BP}(K+1, \bar{x}, \emptyset)$, preventing recursion. By the BP root node symmetry defined in [11, Thm. 2], these two positions will define exactly one distance value $d_{1, K+1}$, so that we can add the edge $\{1, K+1\}$ to E with weight $d_{1, K+1}$, yielding a new graph G' . The Henneberg type I order guarantees that the distances d_{uv} are known for all $u, v \in [K+1]$, so that $G'[[K+1]]$ is a full $(K+1)$ -clique. Consider now vertex v_{K+2} : the Henneberg type I order guarantees that v_{K+2} has at least K adjacent predecessors. If it has $K+1$, then $G'[[K+2]]$ is the full $(K+2)$ -clique. Otherwise $G'[[K+2]]$ is a near-clique on $K+2$ vertices with one missing edge (say $\{u, K+2\}$ for some $u \in [K+1]$). We can therefore use the Cayley-Menger determinant to compute two possible values for $d_{u, K+2}$, as discussed above. Because the Henneberg type I order always guarantees at least K adjacent predecessors, this procedure can be generalized to vertices of any rank v in $V \setminus [K]$, and so it defines a recursive algorithm which branches whenever a distance can be assigned two different values, simply continues to the next rank whenever the subgraph induced by the current $K+2$ vertices is a full clique, and prunes all branches whenever the partial distance matrix defined on the current $K+2$ vertices has no Euclidean completion.

In general, this procedure holds for realizations in \mathbb{R}^K whenever there is a vertex order such that each next vertex v is adjacent to K predecessors: thus we can define a subgraph containing v and $K+1$ predecessors consisting of two $(K+1)$ cliques whose intersection is a K -clique (i.e. a near-clique with one missing edge). There are in general two possible realizations in \mathbb{R}^K for such subgraphs, as shown in Fig. 3.

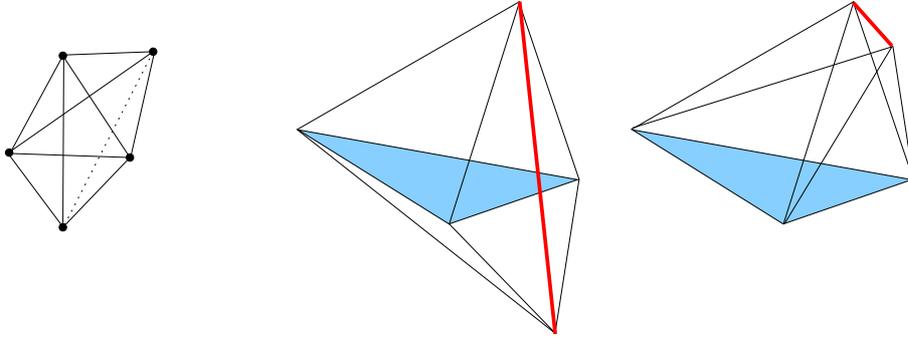


Fig. 3 On the left: a near clique on 5 vertices with one missing edge (dotted line). On the right: its two possible realizations in \mathbb{R}^3 for a given feasible edge weighting (distance values for the missing edge shown in red).

6.2 Formalization and properties

Alg. 2 formalizes such a recursive algorithm. It takes as input a vertex v of rank greater than $K + 1$, a partial matrix A and a set \mathcal{A} which will eventually contain all the possible completion of the partial matrix given as the problem input. For a given partial matrix A , a vertex v of $\mathcal{G}(A)$ and an integer $\ell \leq K$, let A_v^ℓ be the $\ell \times \ell$ symmetric submatrix of A including row and column v that has fewest missing components. Whenever A_v^{K+2} has no missing elements, the equation $\text{CM}(A_v^{K+2}, \delta) = 0$ is either a tautology if A_v^{K+2} is a Euclidean distance matrix, or unsatisfiable in \mathbb{R} otherwise. In the first case, we define it to have $\delta = d_{uv}$ as a solution, where u is the smallest row/column index of A_v^{K+2} . In the second case, we define it to have no solutions.

Algorithm 2 $\text{dBP}(v, A, \mathcal{A})$

Require: A vertex $v \in V \setminus [K + 1]$, a partial matrix A , a set \mathcal{A} .

```

1:  $P = \{\delta \mid \text{CM}(A_v^{K+2}, \delta) = 0\}$ 
2: for  $\delta \in P$  do
3:    $\{u, v\} \leftarrow \text{U}(A_v^{K+2})$ 
4:    $d_{uv} \leftarrow \delta$ 
5:   if  $A$  is complete then
6:      $\mathcal{A} \leftarrow \mathcal{A} \cup \{A\}$ 
7:   else
8:      $\text{dBP}(v+1, A, \mathcal{A})$ 
9:   end if
10: end for

```

Lemma 1. *In Step 1 of Alg. 2, A_v^{K+2} always has at most one missing distance involving v .*

Proof. At level v of Alg. 2, all distances for $\{u, w\}$ for $u, w < v$ are known by the induction hypothesis. The induction starts because either $d_{1, K+1}$ is part of the input partial matrix, or, if not, by calling $\text{BP}(K + 1, \bar{x}, \emptyset)$ just for level $K + 1$, without recursion: then the distance $d_{1, K+1}$ can be computed. By the Henneberg type I order, v is adjacent to at least K predecessors, so that the densest $(K + 2) \times (K + 2)$ symmetric submatrix of A involving row and column v must be such that all other rows/columns are indexed by as many adjacent predecessors of v as possible. Since there are at most $K + 1$ such adjacent predecessors, there is at most one missing distance in A_v^{K+2} , and it involves v . If A can be completed to a Euclidean distance matrix, then the missing distance is assigned a feasible value in Step 4. This completes the induction step.

Corollary 1. *In Step 3 of Alg. 2, U is well defined.*

Theorem 1. *At the end of Alg. 2, \mathcal{A} contains all possible completions of the input partial matrix.*

Proof. By contradiction, if not then there must be a recursive call when there is a $\gamma \in \mathbb{R}_+$ such that $d_{uv} = \gamma$ yields a partial matrix which can be completed to a Euclidean distance matrix, but $\gamma \notin P$. But by Lemma 1 this would mean that the quadratic equation $\text{CM}(A_v^{K+2}, \delta) = 0$ in δ has more than two solutions, which is impossible.

6.3 A dual Branch-and-Prune

The resemblance of Alg. 1 and 2 is such that it is very easy to assign dual meanings to the original (otherwise known as *primal*) BP algorithms. As was made clear in Sect. 3, weighted graphs and partial symmetric matrices are dual to each other through the inverse mappings \mathcal{M} and \mathcal{G} . Whereas in the primal BP we decide realizations of the graph, in the dual BP we decide the completions of partial matrices, so realizations and distance matrix completions are dual to each other. The primal BP decides on points $x_v \in \mathbb{R}^K$ to assign to the next vertex v , whereas the dual BP decides on distances δ to assign to the next missing distance incident to v and to a predecessor of v ; there are at most two choices of x_v as there are at most two choices for δ ; only one choice of x_v is available whenever v is adjacent to strictly more than K predecessor, and the same happens for δ ; finally, no choices for x_v are available in case the current partial realization cannot be extended to a full realization of the graph, as well as no choices for δ are available in case the current partial matrix cannot be completed to a Euclidean distance matrix. This means that weighted edges and points in Euclidean space are dual to each other. The same vertex order can be used by both the primal and the dual BP (so the order is self-dual).

There is one clear difference between primal and dual BP: namely, that the dual BP needs an initial $(K + 1)$ -clique, whereas the primal BP only needs an initial K -clique. This difference also has a dual interpretation: a complete Euclidean distance matrix corresponds to two (rather than one) realizations, one the reflection of the other through the hyperplane defined by the first K points.

References

1. Blumenthal, L.: Theory and Applications of Distance Geometry. Oxford University Press, Oxford (1953)
2. Cremona, L.: Le figure reciproche nella statica grafica. G. Bernardoni, Milano (1872)
3. Dong, Q., Wu, Z.: A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances. *Journal of Global Optimization* **22**, 365–375 (2002)
4. Gunther, H.: NMR Spectroscopy: Basic Principles, Concepts, and Applications in Chemistry. Wiley, New York (1995)
5. Hendrickson, B.: Conditions for unique graph realizations. *SIAM Journal on Computing* **21**(1), 65–84 (1992)
6. Henneberg, L.: Die Graphische Statik der starren Systeme. Teubner, Leipzig (1911)

7. John, A.L.S.: Geometric constraint systems with applications in cad and biology. Ph.D. thesis, University of Massachusetts at Amherst (2008)
8. Laurent, M.: Matrix completion problems. In: C. Floudas, P. Pardalos (eds.) *Encyclopedia of Optimization*, second edn., pp. 1967–1975. Springer, New York (2009)
9. Lavor, C., Lee, J., John, A.L.S., Liberti, L., Mucherino, A., Sviridenko, M.: Discretization orders for distance geometry problems. *Optimization Letters* (DOI: 10.1007/s11590-011-0302-6). DOI DOI:10.1007/s11590-011-0302-6
10. Lavor, C., Liberti, L., Maculan, N.: The discretizable molecular distance geometry problem. Tech. Rep. q-bio/0608012, arXiv (2006)
11. Lavor, C., Liberti, L., Maculan, N., Mucherino, A.: The discretizable molecular distance geometry problem. *Computational Optimization and Applications* (DOI: 10.1007/s10589-011-9402-6). DOI DOI:10.1007/s10589-011-9402-6
12. Lavor, C., Liberti, L., Mucherino, A.: The *i*Branch-and-Prune algorithm for the discretizable molecular distance geometry problem with interval data. *Journal of Global Optimization* (accepted)
13. Lavor, C., Mucherino, A., Liberti, L., Maculan, N.: An artificial backbone of hydrogens for finding the conformation of protein molecules. In: *Proceedings of the Computational Structural Bioinformatics Workshop*, pp. 152–155. IEEE, Washington D.C., USA (2009)
14. Lavor, C., Mucherino, A., Liberti, L., Maculan, N.: Computing artificial backbones of hydrogen atoms in order to discover protein backbones. In: *Proceedings of the International Multi-conference on Computer Science and Information Technology*, pp. 751–756. IEEE, Mragowo, Poland (2009)
15. Lavor, C., Mucherino, A., Liberti, L., Maculan, N.: Discrete approaches for solving molecular distance geometry problems using NMR data. *International Journal of Computational Biosciences* **1**, 88–94 (2010)
16. Lavor, C., Mucherino, A., Liberti, L., Maculan, N.: On the computation of protein backbones by using artificial backbones of hydrogens. *Journal of Global Optimization* **50**, 329–344 (2011)
17. Liberti, L., Lavor, C., Maculan, N.: A branch-and-prune algorithm for the molecular distance geometry problem. *International Transactions in Operational Research* **15**, 1–17 (2008)
18. Liberti, L., Masson, B., Lee, J., Lavor, C., Mucherino, A.: On the number of solutions of the discretizable molecular distance geometry problem. In: *Combinatorial Optimization, Constraints and Applications (COCO11)*, *LNCS*, vol. 6831, pp. 322–342. Springer, New York (2011)
19. Menger, K.: Untersuchungen über allgemeine metrik. *Mathematische Annalen* **103**, 466–501 (1930). DOI 10.1007/BF01455705
20. Mucherino, A., Lavor, C.: The branch and prune algorithm for the molecular distance geometry problem with inexact distances. In: *Proceedings of the International Conference on Computational Biology*, vol. 58, pp. 349–353. World Academy of Science, Engineering and Technology (2009)
21. Mucherino, A., Lavor, C., Liberti, L.: The discretizable distance geometry problem. *Optimization Letters* (accepted)
22. Mucherino, A., Lavor, C., Liberti, L., Maculan, N.: On the definition of artificial backbones for the discretizable molecular distance geometry problem. *Mathematica Balkanica* **23**, 289–302 (2009)
23. Mucherino, A., Lavor, C., Liberti, L., Talbi, E.G.: On suitable parallel implementations of the branch & prune algorithm for distance geometry. In: *Proceedings of the Grid5000 Spring School*. Lille, France (2010)
24. Mucherino, A., Lavor, C., Liberti, L., Talbi, E.G.: A parallel version of the branch & prune algorithm for the molecular distance geometry problem. In: *ACS/IEEE International Conference on Computer Systems and Applications (AICCSA10)*. IEEE conference proceedings, Hammamet, Tunisia (2010)
25. Mucherino, A., Liberti, L., Lavor, C.: MD-jeep: an implementation of a branch-and-prune algorithm for distance geometry problems. In: K. Fukuda, J. van der Hoeven, M. Joswig,

- N. Takayama (eds.) *Mathematical Software, LNCS*, vol. 6327, pp. 186–197. Springer, New York (2010)
26. Porta, J., Ros, L., Thomas, F.: Inverse kinematics by distance matrix completion. In: *Proceedings of the 12th International Workshop on Computational Kinematics*, pp. 1–9 (2005)
 27. Saviotti, C.: Nouvelles méthodes pour le calcul des travures réticulaires. In: Appendix to L. Cremona, “Les figures réciproques en statique graphique”, pp. 37–100. Gauthier-Villars, Paris (1885)
 28. Saviotti, C.: *La statica grafica: Lezioni*. U. Hoepli, Milano (1888)
 29. Schlick, T.: *Molecular modelling and simulation: an interdisciplinary guide*. Springer, New York (2002)
 30. Tay, T.S., Whiteley, W.: Generating isostatic frameworks. *Structural Topology* **11**, 21–69 (1985)