Noname manuscript No.
(will be inserted by the editor)

The Discretizable Molecular Distance Geometry Problem

Carlile Lavor · Leo Liberti · Nelson Maculan · Antonio Mucherino

Received: Oct. 19, 2009 / Revised: Nov. 23, 2010

Abstract Given a simple weighted undirected graph $G = (V, E, d)$ with $d : E \to \mathbb{R}_+$, the Molecular Distance Geometry Problem (MDGP) consists in finding an embedding $x : V \to \mathbb{R}^3$ such that $||x_u - x_v|| = d_{uv}$ for each $\{u, v\} \in E$. We show that under a few assumptions usually satisfied in proteins, the MDGP can be formulated as a search in a discrete space. We call this MDGP subclass the Discretizable MDGP (DMDGP). We show that the DMDGP is **NP**-hard and we propose a solution algorithm called Branch-and-Prune (BP). The BP algorithm performs remarkably well in practice in terms of speed and solution accuracy, and can be easily modified to find all incongruent solutions to a given DMDGP instance. We show computational results on several artificial and real-life instances.

Keywords distance geometry · branch-and-prune · molecular conformation · protein · NMR

CR Subject Classification 92E10 · 90C26 · 90C27 · 65K05

1 Introduction

It is well known that the role and function of a molecule is determined by both its chemical structure (the atoms that compose it and the way they bond) and its three-dimensional structure [17]. Supposing the chemical structure is known, finding the

The present work is based on the preliminary 2006 technical report [39].

Carlile Lavor
Department of Applied Mathematics (IMECC-UNICAMP), State University of Campinas, 13081-970, Campinas - SP, Brazil. E-mail: clavor@ime.unicamp.br

Leo Liberti
LIX, École Polytechnique, 91128 Palaiseau, France. E-mail: liberti@lix.polytechnique.fr

Nelson Maculan
Federal University of Rio de Janeiro (COPPE–UFRJ), C.P. 68511, 21945-970, Rio de Janeiro - RJ, Brazil. E-mail: maculan@cos.ufrj.br

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

conformation of the atoms in $\mathbb{R}^3$ is usually tackled by a mixture of chemical analysis and mathematical methods. Some insight as to the molecular spatial conformation can be gained by employing Nuclear Magnetic Resonance (NMR) techniques [27], which are able to give a measure of the distance between (but not of the positions of) pairs of atoms closer than around 5 to 6Å [63], p. 19. The problem of finding the atomic positions given a subset of atomic distances can be formalized as follows.

> Molecular Distance Geometry Problem (MDGP): given a weighted simple undirected graph $G = (V, E, d)$, is there an embedding $x : V \to \mathbb{R}^3$ such that $||x_u - x_v|| = d_{uv}$ for each $\{u, v\} \in E$?

The set V represents the atoms, the set E are the atom pairs $\{u, v\}$ for which the distance d_{uv} is known. The MDGP has been shown to be **NP**-hard via a reduction from 3-Partition [62], although the problem is solvable in linear time when all the inter-atomic distances are known [20]. The MDGP is usually formulated as a continuous nonconvex optimization problem:

$$\min_x g(x),$$

where

$$g(x) = \sum_{\{u,v\}\in E} (||x_u - x_v||^2 - d_{uv}^2)^2. \tag{1}$$

Obviously, x is a solution if and only if $g(x) = 0$.

It should be noted that we refer to the MDGP as a precisely formalized decision problem, and not as a practical chemical problem. We therefore make three assumptions that in real life may be easily challenged: (a) a subset of exact distances (as opposed to approximate) is given as part of the input; (b) no measurement errors occur; (c) the optimal 3D embedding of the graph is not influenced by a potential energy minimization term on the objective function. Concerning points (a) and (b), there are two types of experimental errors arising from NMR distance measurements: systematic uncertainty on each measurement, and a certain (low) percentage of completely wrong measurements [5]. Errors of the first type are usually dealt with by introducing distance bounds [51], for which a suitable modification of the method described in this paper exists [52]. To the best of our knowledge, errors of the second type have only been tackled by the Error Correcting Code (ECC) proposed in [5]. Naturally, this ECC could be applied to the protein backbone distance matrix as a preprocessing step to our method.

The present work falls into three main application categories:

1. molecular conformation, and proteomics in praticular [18];
2. sensor network localization [3, 65, 8];
3. graph drawing (`www.graphdrawing.org`) and rigidity [59, 16].

The important common point of the above applications is Euclidean distance geometry [13], i.e. finding the geometrical locus in $\mathbb{R}^K$ of the vertices of weighted graphs so that Euclidean distances are consistent with the edge weights. In the rest of the paper, we follow proteomics as our main application theme. Many contributions to distance geometry, however, have been made in recent years from researchers working in sensor networks, graph drawing and rigidity. Sensor network localization differs from proteomics in that (a) the position of a subset of vertices (called *anchors*) is known a priori — these usually correspond to the fixed parts of the communication network; (b) the positions are mostly (but not necessarily) sought in $\mathbb{R}^2$ rather than $\mathbb{R}^3$. The complexity of sensor network localization has been discussed in [22]: the problem is NP-hard in general (this follows from [62]) but given a specific vertex order, called *trilateration* order, the problem becomes polynomially solvable. Several works on sensor networks formulate the MDGP as a Semidefinite Programming (SDP) problem, using the well-known relationship between MDGP, Euclidean Distance Matrices (EDM) and Semidefinite Programs (SDP) [64]; the solution of the SDP only provides a relaxation of the MDGP, which is subsequently refined in order to obtain a feasible embedding of the given graph [11,12,9,7]. Recently, SDP techniques for the MDGP were given a boost by the successful application of facial analysis of the semidefinite matrix cone [4]. The SDP-based facial reduction algorithm described in [32,33] exploits the fact that SDPs related to distance geometry problems are generally highly degenerate. In fact, the presence of cliques in the instance graphs implies that the corresponding semidefinite matrices may have a very low rank. The SDP cone faces are characterized and used for reducing the problem to several subproblems of smaller size. The main idea is to increase the size of the cone faces by finding the intersection of smaller faces, for which the corresponding SDP has already been solved. This allows the solution of the original problem in a finite number of steps.

In practice, the MDGP is usually solved via continuous optimization methods (see [40,48] for overviews). In [28], the molecule is decomposed into uniquely realizable maximally rigid clusters; each cluster's 3D structure is determined independently of the others. The graph minor resulting from contracting each cluster to a single vertex is then embedded using a multi-start continuous search approach; this last step makes this method into a heuristic one. In [10], the molecule is also decomposed into clusters and a semidefinite programming relaxation is used to localize each cluster. In [50, 51], a Gaussian smoothing of (1) is derived in a closed analytical form depending on a smoothing parameter λ. The proposed algorithm is called Global Continuation Algorithm (GCA): the smoothed problem is locally solved for iteratively increasing values of λ (this brings the smoothed problem closer and closer to the original problem), each local solution process starting from the solution of the previous smoothing. In [1, 2], the MDGP is formulated as D.C. (difference of convex functions) programming problems and solved using a variant of the D.C. Algorithm (DCA). In [38,45,47], three Variable Neighborhood Search-based algorithms are proposed. Other methods are the alternating projection algorithm [25], the multi-scaling algorithm [31,66], the geometric build-up algorithm [20,21,68,69], a stochastic/perturbation algorithm by Zou, Bird, and Schnabel [70] and a population-based metaheuristic [26]. Two completely different approaches to solving the MDGP are given in [37] (based on quantum computation) and [67] (based on algebraic geometry).

One of the most stringent limitations of MDGP algorithms is solution accuracy. Because there exist many different spatial conformations having objective function values very near zero, being able to discriminate between very small numbers is important. Compared with continuous methods, combinatorial methods are usually more suitable to produce extremely accurate values; this provides sufficient motivation to work on a combinatorial algorithm for (a subclass of) the MDGP.

A protein consists of a main backbone and several "dangling" side chains. The NMR technique can of course be applied to proteins in particular, and indeed many of the algorithms to solve the general MDGP have been tested on proteins. In particular, we consider in this paper the protein backbones, i.e. the sequences of bonded atoms defining a sort of chain in the protein conformations. For each amino acid, we consider the three atoms N, C_α and C. The particular structure of this chain of atoms makes it possible to formulate the MDGP applied to protein backbones as a discrete search problem: this has an enormous impact on speed and solution accuracy, as floating point arithmetics calculations are fewer than with continuous search methods. We formalize this by introducing the Discretizable Molecular Distance Geometry Problem (DMDGP), which consists of a certain subset of MDGP instances (to which most protein backbones belong) for which a discrete formulation can be supplied. The determination of the spatial position of the side chains is called the SIDE CHAIN PLACEMENT PROBLEM (SCPP) [60,61]. Although in this paper we only consider the determination of the protein backbone, it is clear that given a set of likely backbones, some of them can be discarded if the resulting SCPP instance turns out to be infeasible. In this sense, the DMDGP and the SCPP are largely complementary.

1.1 DMDGP definition

When a total order is explicitly given on V, and u, v are the i-th and j-th indices respectively, we also write d_{ij} for d_{uv}.

> DISCRETIZABLE MOLECULAR DISTANCE GEOMETRY PROBLEM (DMDGP): given a simple weighted undirected graph $G = (V, E, d)$ such that there exists an order $v_1, \ldots, v_n$ of V satisfying the following requirements:
> 1. E contains all cliques on quadruplets of consecutive vertices: $\forall i \in \{4, \ldots, n\}\ \forall j, k \in \{i-3, \ldots, i\}\ (\{j, k\} \in E)$;
> 2. the following strict triangular inequality holds: $\forall i \in \{2, \ldots, n-1\}\ d_{i-1,i+1} < d_{i-1,i} + d_{i,i+1}$,
>
> is there an embedding $x : V \to \mathbb{R}^3$ such that $||x_u - x_v|| = d_{uv}$ for each $\{u, v\} \in E$?

The distances $d_{i-1,i}$ are called *bond lengths*, for $i \in \{2, \ldots, n\}$, and the angles $\theta_{i-2,i}$ between atoms v_{i-2}, v_{i-1}, v_i are called *bond angles*, for $i \in \{3, \ldots, n\}$. The ordering on V is called the *backbone ordering*. Furthermore, we partition E in two sets H and F such that $H = \{\{i, j\} \in E \mid |i - j| \leq 3\}$ and $F = E \setminus H$. In this paper, the order on V is defined by a linear chain of atoms connected to each other by covalent bonds.

In practice, Assumption 1 requires that the bond lengths and angles, as well as the distances between atoms separated by three consecutive bond lengths are known. The distances between atoms separated by two consecutive bond lengths may of course be trivially computed from the bond lengths and angles. Assumption 2 says that no bond angle may be a multiple of π. Assumption 1 is applicable to many proteins as NMR is able to compute distances of atoms which are close together, and groups of four consecutive atoms in the backbone ordering are usually closer than the threshold value of 6Å [17,63]. Assumption 2 is also applicable to proteins as, to the best of our knowledge, no protein has bond angles of exactly π. Furthermore, the probability measure of a protein having a bond angle of exactly π is zero.

Given an MDGP instance, determining whether it is a DMDGP one involves finding an order that satisfies Assumptions 1-2. We discuss the problem of finding a vertex order satisfying Assumption 1 in [36], showing that for embeddings in $\mathbb{R}^3$ such orders can be found in polynomial time. On the other hand, 2 is satisfied with probability 1 when d is a partial distance function (i.e. a distance function defined on a proper subset E of the set of all index pairs $\{i, j\}$ for $i \neq j \in V$): the set of triplets of values satisfying the triangular inequality at equality has Lebesgue measure zero in the set of all such triplets.

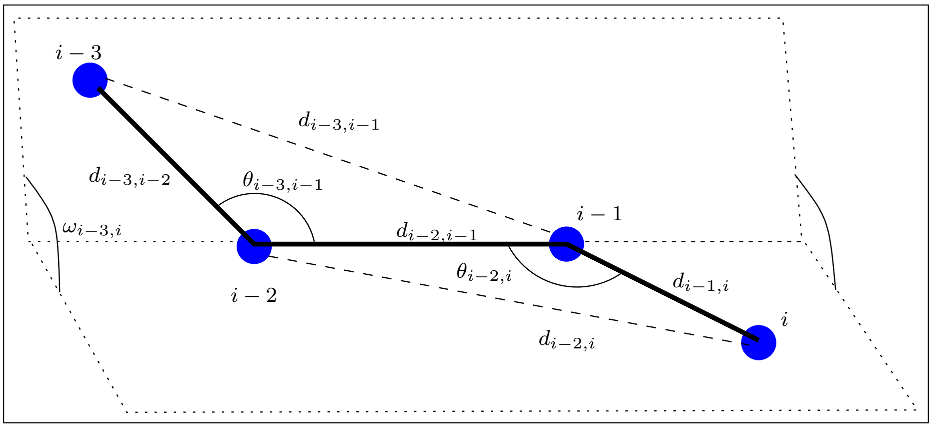

Fig. 1 Definitions of bond lengths, bond angles and torsion angles.

1.2 Original contributions

In [46], we proposed a discrete search algorithm, called Branch-and-Prune (BP), for finding solutions to certain MDGP instances, and tested it on artificial instances. In a sequence of recent conference papers [41,42,52,54,56] we explored several different aspects of the BP algorithm. Other solution methods having some relations with BP are [21,15]. In this paper we formalize an important set of MDGP instances that can be solved by a discrete search as a separate decision problem and investigate some of its theoretical properties. We then discuss comparative computational results on artificial as well as real instances.

The rest of this paper is organized as follows. In Section 2, we derive the discrete formulation of the DMDGP. In Section 3, we prove that the DMDGP is **NP**-hard. In Section 4, we discuss the BP algorithm to solve the DMDGP to optimality. Section 5 presents the computational results on some artificial and real-life instances. In Section 6 we discuss the relationship between the DMDGP and the Euclidean distance matrix completion problem. Section 7 concludes the paper and presents ongoing work.

2 Discrete formulation of the MDGP

In the following, we will restrict our attention to the DMDGP. In order to describe a molecule with n atoms, in addition to the bond lengths $d_{i-1,i}$, for $i = 2, \dots, n$, and the bond angles $\theta_{i-2,i}$, for $i = 3, \dots, n$, we also have to consider the *dihedral* or *torsion angles* $\omega_{i-3,i}$, for $i = 4, \dots, n$, which are the angles between the normals through the planes defined by the atoms $i-3, i-2, i-1$ and $i-2, i-1, i$ (see Fig. 1). Note that, in most molecular conformation calculations, all the bond lengths and bond angles are assumed to be known *a priori*. Thus, the first three atoms of the molecule can be fixed and the fourth atom can be determined by the torsion angle $\omega_{1,4}$. The fifth atom can be determined by the torsion angles $\omega_{1,4}$ and $\omega_{2,5}$, and so on.

The geometrical intuition behind the discrete formulation is that the i-th atom lies on the intersection of three spheres centered at atoms $i-3, i-2, i-1$ having respective

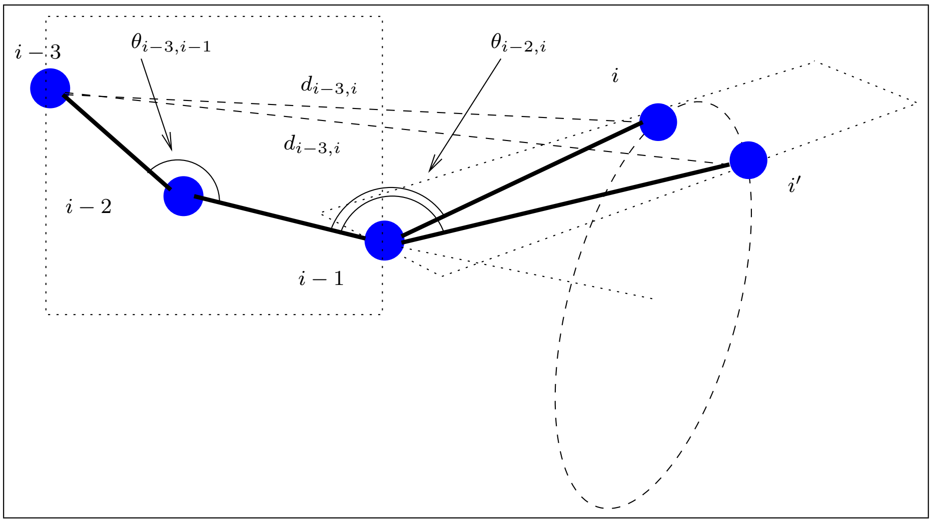

Fig. 2 Discretization of the problem. The atom i can only be in the two shown positions (i and i') in order to be feasible with the distance $d_{i-3,i}$.

radii $d_{i-3,i}, d_{i-2,i}, d_{i-1,i}$. By Assumptions 1 and 2, and the fact that no two atoms can ever take the same position in space, the intersection of the three spheres defines at most two points (labeled i and i' in Fig. 2). This allows us to express the position of the i-th atom in terms of the preceding three, giving us 2^{n-3} possible molecules. Of course some of these will be infeasible with respect to the distances in F (i.e. distances between atoms which are further apart than 4 units in the backbone ordering).

Given bond lengths $d_{1,2}, \ldots, d_{n-1,n}$, bond angles $\theta_{1,3}, \ldots, \theta_{n-2,n}$, and torsion angles $\omega_{1,4}, \ldots, \omega_{n-3,n}$ of a molecule with n atoms, it is well known [57] that the Cartesian coordinates $(x_{i_1}, x_{i_2}, x_{i_3})$ for each atom i in the molecule can be obtained as:

$$\begin{bmatrix} x_{i_1} \\ x_{i_2} \\ x_{i_3} \\ 1 \end{bmatrix} = B_1 B_2 B_3 \cdots B_i \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \ i \in \{4, \ldots, n\},$$

where the matrices are defined inductively as follows:

$$B_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$B_2 = \begin{bmatrix} -1 & 0 & 0 & -d_{1,2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, B_3 = \begin{bmatrix} -\cos\theta_{1,3} & -\sin\theta_{1,3} & 0 & -d_{2,3}\cos\theta_{1,3} \\ \sin\theta_{1,3} & -\cos\theta_{1,3} & 0 & d_{2,3}\sin\theta_{1,3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

$$B_i = \begin{bmatrix} -\cos\theta_{i-2,i} & -\sin\theta_{i-2,i} & 0 & -d_{i-1,i}\cos\theta_{i-2,i} \\ \sin\theta_{i-2,i}\cos\omega_{i-3,i} & -\cos\theta_{i-2,i}\cos\omega_{i-3,i} & -\sin\omega_{i-3,i} & d_{i-1,i}\sin\theta_{i-2,i}\cos\omega_{i-3,i} \\ \sin\theta_{i-2,i}\sin\omega_{i-3,i} & -\cos\theta_{i-2,i}\sin\omega_{i-3,i} & \cos\omega_{i-3,i} & d_{i-1,i}\sin\theta_{i-2,i}\sin\omega_{i-3,i} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

for $i \in \{4, \ldots, n\}$. Thus, the Cartesian coordinates of all atoms in the molecule are completely determined by $\cos \omega_{i-3,i}$ and $\sin \omega_{i-3,i}$ for $i \in \{4, \ldots, n\}$.

Lemma 1 *For DMDGP instances,* $\cos \omega_{i-3,i}$ *can be computed in* $O(1)$ *for all* $i \in \{4, \ldots, n\}$.

Proof This follows by the cosine law for torsion angles [58] (p. 278) and by the fact that all distances among the atoms $i-3, i-2, i-1, i$ are known. □

Theorem 1 *Given a DMDGP instance* $G = (V, E, d)$*, the number of embeddings* $x : V \to \mathbb{R}^3$ *such that* $||x_u - x_v|| = d_{uv}$ *for each* $\{u, v\} \in E$ *is finite, up to translations and rotations.*

Proof The proof is by induction. For a molecule with 4 atoms, we can use the bond lengths $d_{1,2}, d_{2,3}$ and the bond angle $\theta_{1,3}$ in order to determine the matrices B_2 and B_3, defined in (2), and obtain the partial embedding:

$$x_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

$$x_2 = \begin{pmatrix} -d_{1,2} \\ 0 \\ 0 \end{pmatrix},$$

$$x_3 = \begin{pmatrix} -d_{1,2} + d_{2,3} \cos \theta_{1,3} \\ d_{2,3} \sin \theta_{1,3} \\ 0 \end{pmatrix}$$

fixing the first three atoms. Since we also know the distance $d_{1,4}$, by Lemma 1 we can obtain the value of $\cos \omega_{1,4}$. Thus, $\sin \omega_{1,4}$ can take at most two possible values $\pm\sqrt{1 - \cos^2 \omega_{1,4}}$. Consequently, by (3), we obtain at most two possible positions x_4, x_4' for the fourth atom of the molecule, given by

$$x_4 = \begin{bmatrix} -d_{1,2} + d_{2,3} \cos \theta_{1,3} - d_{3,4} \cos \theta_{1,3} \cos \theta_{2,4} + d_{3,4} \sin \theta_{1,3} \sin \theta_{2,4} \cos \omega_{1,4} \\ d_{2,3} \sin \theta_{1,3} - d_{3,4} \sin \theta_{1,3} \cos \theta_{2,4} - d_{3,4} \cos \theta_{1,3} \sin \theta_{2,4} \cos \omega_{1,4} \\ d_{3,4} \sin \theta_{2,4} \left(\sqrt{1 - \cos^2 \omega_{1,4}} \right) \end{bmatrix},$$

$$x_4' = \begin{bmatrix} -d_{1,2} + d_{2,3} \cos \theta_{1,3} - d_{3,4} \cos \theta_{1,3} \cos \theta_{2,4} + d_{3,4} \sin \theta_{1,3} \sin \theta_{2,4} \cos \omega_{1,4} \\ d_{2,3} \sin \theta_{1,3} - d_{3,4} \sin \theta_{1,3} \cos \theta_{2,4} - d_{3,4} \cos \theta_{1,3} \sin \theta_{2,4} \cos \omega_{1,4} \\ d_{3,4} \sin \theta_{2,4} \left(-\sqrt{1 - \cos^2 \omega_{1,4}} \right) \end{bmatrix}.$$

We remark that the only difference in x_4 and x_4' is a sign change in the last component. Now assume that for $i \geq 4$ atoms we have a finite number of embeddings solving the DMDGP instance. Adding one more atom in the molecule and using Lemma 1 again, we can obtain the value of $\cos \omega_{i-2,i+1}$. From each partial embedding with i atoms, at most two extensions to the $(i+1)$-st atom can be obtained by using $\sin \omega_{i-2,i+1} = \pm\sqrt{1 - \cos^2 \omega_{i-2,i+1}}$ in matrix B_{i+1}, given in (3). □

Corollary 1 *For a DMDGP instance with* $n \geq 4$ *atoms, there are at most* 2^{n-3} *possible embeddings up to translations and rotations.*

Informally, a graph is rigid if it has no uncountable set of embeddings modulo translations and rotations. Precise definitions can be found in [59].

Corollary 2 *DMDGP instance graphs are rigid.*

A rigid graph is uniquely realizable if it only has one embedding modulo translations and rotations [28]. DMDGP instances are rigid graphs but may fail to be uniquely realizable. Moreover, for interesting DMDGP instances such as protein backbone graphs, the uniquely realizable subgraphs may be fairly small. This makes the application of heuristic methods such as ABBIE [28] possible but not promising.

Note that each possible embedding of the DMDGP is defined by a sequence of torsion angles $\omega_{1,4}, \ldots, \omega_{n-3,n}$. By using the matrices B_i (see (3)), this sequence can be converted to a sequence $x = (x_1, \ldots, x_n) \in \mathbb{R}^{3n}$ of Cartesian vectors and, using the objective function g defined in (1), the validity of an embedding can be established simply by testing if $g(x) = 0$.

2.1 Fourth level symmetry

In this section, we show that there is a symmetry around the plane defined by the first three atoms of the conformations which are embeddings solving the DMDGP. This allows us to reduce computational costs by half. First, we need two lemmata (whose proofs are in the appendix).

Lemma 2 *Let the matrix Q_i be defined by*

$$Q_i = B_4 \cdots B_i,$$

for $i \in \{4, \ldots, n\}$, where its elements are denoted by

$$Q_i = \begin{bmatrix} q^i_{11} & q^i_{12} & q^i_{13} & q^i_{14} \\ q^i_{21} & q^i_{22} & q^i_{23} & q^i_{24} \\ q^i_{31} & q^i_{32} & q^i_{33} & q^i_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

If we invert the sign of $\sin \omega_{i-3,i}$ in all the matrices B_i, for $i \in \{4, \ldots, n\}$, and denote the new matrices obtained by B'_i, then the elements of the matrix Q'_i, defined by

$$Q'_i = B'_4 \cdots B'_i,$$

are given by

$$Q'_i = \begin{bmatrix} q^i_{11} & q^i_{12} & -q^i_{13} & q^i_{14} \\ q^i_{21} & q^i_{22} & -q^i_{23} & q^i_{24} \\ -q^i_{31} & -q^i_{32} & q^i_{33} & -q^i_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

for $i \in \{4, \ldots, n\}$.

Lemma 3 *Let $x_1, \ldots, x_n \in \mathbb{R}^3$ be the Cartesian coordinates defined by the torsion angles $\omega_{1,4}, \ldots, \omega_{n-3,n}$. If we invert the sign of $\sin \omega_{i-3,i}$ in all the matrices B_i, for $i \in \{4, \ldots, n\}$, then the new Cartesian coordinates $x'_1, \ldots, x'_n \in \mathbb{R}^3$ are given by*

$$\begin{bmatrix} x'_{i_1} \\ x'_{i_2} \\ x'_{i_3} \end{bmatrix} = \begin{bmatrix} x_{i_1} \\ x_{i_2} \\ -x_{i_3} \end{bmatrix},$$

for $i \in \{1, \ldots, n\}$.

Theorem 2 *Let $x : V \to \mathbb{R}^3$ be an embedding solving a given DMDGP instance, defined by the torsion angles $\omega_{1,4}, \ldots, \omega_{n-3,n}$. If we invert the sign of $\sin \omega_{i-3,i}$ in all the matrices B_i, for $i \in \{4, \ldots, n\}$, we obtain another embedding $x' : V \to \mathbb{R}^3$ solving the same instance.*

Proof Let $X = \{x_1, \ldots, x_n\}$ be the conformation associated to an embedding $x : V \to \mathbb{R}^3$ solving the DMDGP, defined by the torsion angles $\omega_{1,4}, \ldots, \omega_{n-3,n}$, $X' = \{x'_1, \ldots, x'_n\}$ be the conformation obtained by inverting the sign of $\sin \omega_{i-3,i}$ in all the matrices B_i, for $i \in \{4, \ldots, n\}$, and $R : \mathbb{R}^3 \to \mathbb{R}^3$ be the function defined by

$$R(x_{i_1}, x_{i_2}, x_{i_3}) = (x_{i_1}, x_{i_2}, -x_{i_3}).$$

Since R is an orthogonal operator,

$$||x_i - x_j|| = ||R(x_i) - R(x_j)|| \;\; \forall (i, j). \tag{4}$$

By Lemma (3),

$$||R(x_i) - R(x_j)|| = ||x'_i - x'_j|| \;\; \forall (i, j). \tag{5}$$

Since x solves the given DMDGP instance,

$$||x_i - x_j|| = d_{i,j} \;\; \forall \{i, j\} \in E, \tag{6}$$

where E is the set of pairs of atoms (i, j) whose Euclidean distances d_{ij} are known. Thus, by (4), (5), and (6), we get

$$||x'_i - x'_j|| = d_{i,j} \;\; \forall \{i, j\} \in E,$$

stating that x' is also an embedding solving the given DMDGP instance. □

3 Complexity

In this section, we show that the DMDGP is **NP**-hard by reduction from the SUBSET-SUM problem:

> SUBSET-SUM. Given nonnegative integers $a_1, \ldots, a_n$, is there a partition into two sets, encoded by $s \in \{-1, +1\}^n$, such that each subset has the same sum, i.e. $\sum_{i=1}^{n} s(i) a_i = 0$?

The MDGP is shown to be **NP**-hard in [62] (a helpful sketch of the proof is given in [50]) by reducing SUBSET-SUM to a 1-dimensional MDGP with distance constraints between successive atoms (in an arbitrary atomic ordering) plus a single distance constraint between the first and the last atom, forcing this distance to be zero. As has been mentioned above, the MDGP in an arbitrary dimension K is **NP**-hard by reduction from 3-PARTITION [62].

For the special case of the DMDGP, we have to consider additional distance constraints between any pairs of atoms $1, 2$ or 3 indices apart in the atom sequence.

Theorem 3 *The DMDGP is **NP**-hard.*

Proof We reduce from SUBSET-SUM. Given an instance $a_1, \ldots, a_n$ of the latter, we define an instance of DMDGP on $3n+1$ points numbered 0 to $3n$, with the following distance constraints:

$$d_{i,i+1} = a_{\lfloor i/3 \rfloor} + 1 \qquad \forall i \in \{0, ..., 3n-1\}, \tag{7}$$

$$d_{i,i+2} = \sqrt{d_{i,i+1}^2 + d_{i+1,i+2}^2} \qquad \forall i \in \{0, ..., 3n-2\}, \tag{8}$$

$$d_{i,i+3} = \sqrt{d_{i,i+1}^2 + d_{i+1,i+2}^2 + d_{i+2,i+3}^2} \qquad \forall i \in \{0, ..., 3n-3\}, \tag{9}$$

$$d_{0,3n} = 0. \tag{10}$$

Now we claim that the SUBSET-SUM instance has a solution iff the DMDGP instance has a solution. First, let $s \in \{-1, +1\}^n$ be a solution to the SUBSET-SUM-problem. We define the $3n+1$ points as follows: $x_0 = (0,0,0)$ and for every $0 < i \leq 3n$ with $i = 3(k-1) + j$ we set $x_i = x_{i-1} + s_k a_k e_j$, where $e_0 = (1,0,0), e_1 = (0,1,0)$ and $e_2 = (0,0,1)$. By inspection, this is a solution to the DMDGP instance.

In the rest of the proof, we use the symbol x to mean both an embedding and the first coordinate of Euclidean 3-space — the meaning will be clear from the context. Assume that the DMDGP instance has a solution $X = \{x_{v_1}, \ldots, x_{v_n}\}$. Without loss of generality, we can assume that $x_{v_1} = (0,0,0)$ and that x_{v_2} lays on the x-axis. Now equation (8) implies that the bond angle between $x_{v_1}, x_{v_2}, x_{v_3}$ is $\frac{\pi}{2}$. Again, without loss of generality, assume that the second segment is parallel to the y-axis. By Eq. (9), there are only two possibilities for x_{v_4}, and they force the third bond to be parallel to the z-axis. The same arguments apply to all other bonds, which shows that the bond β between v_{i-1} and v_i is parallel to the (i mod 3)-th axis (where $x = 0, y = 1, z = 2$). Now give the bond β an orientation from v_{i-1} to v_i (which can either be in the same or in the opposite direction of this axis). We define a sign vector $s \in \{-1, +1\}^{3n}$, which encodes these orientations. In this setting, point $3n$ has coordinates (x, y, z) defined by

$$x = \sum_{i \bmod 3 = 0} s_i a_{\lfloor i/3 \rfloor},$$

$$y = \sum_{i \bmod 3 = 1} s_i a_{\lfloor i/3 \rfloor},$$

$$z = \sum_{i \bmod 3 = 2} s_i a_{\lfloor i/3 \rfloor}.$$

By equation (7), we actually have $(x, y, z) = (0, 0, 0)$. Now let s^0, s^1, s^2 be three vectors from $\{-1, +1\}^n$, which are s restricted to indices i mod $3 = 0$, i mod $3 = 1$ or i mod $3 = 2$, respectively. Then any of those is a solution to the original SUBSET-SUM problem by the previous equations. □

It is interesting to note that Assumption 1 in the definition of the DMDGP is, in a certain sense, the tightest possible for the problem to be **NP**-hard. Assumption 1 states that each quadruplet of consecutive vertices in the defined order is a clique in the distance graph. Tightening the assumption further, we might ask whether the problem would still be **NP**-hard if each *quintuplet* of consecutive vertices were a clique. This, however, fails to be the case. A *trilateration graph* [22] in $\mathbb{R}^D$ is a graph with an order $(v_1, \ldots, v_n)$ on the vertices such for all vertices v_i with $i > D+1$, $\{j, i\} \in E$ for all $j \in \{i - D - 2, \ldots, i - 1\}$ (i.e. each vertex is adjacent to the preceding $D+1$ vertices). In three-dimensional space, this implies having distances to at least 4 vertices earlier in the order, which means having a clique for each consecutive quintuplet. By [22] (Theorem 9), the MDGP problem associated to a trilateration graph can be solved in polynomial time.

4 Branch-and-Prune algorithm

In this section, we present a solution algorithm for the DMDGP called Branch-and-Prune (BP). At each step, we can place the i-th atom in two possible positions x_i, x'_i. However, either or both of these positions may be infeasible with respect to a number of constraints. The search is branched on all atomic positions that are feasible with respect to all constraints; by contrast, if a position is not feasible the search scope is pruned. In this context, we call the feasibility verifications *pruning tests*. We note in passing that BP is not an exact algorithm for the DMDGP insofar as it is not clear whether the DMDGP is in **NP** or not. The embeddings produced by BP are approximate solutions to the DMDGP.

The Direct Distance Feasibility (DDF) pruning test is as follows: for all distance pairs $\{j, i\} \in F$ (with $j < i$) we check that $|\|x_j - x_i\| - d_{ji}| < \varepsilon$, where $\varepsilon > 0$ is a given tolerance. If the inequality does not hold, we prune the search node. Even though this pruning test is very simple, it is very effective. The BP algorithm is therefore an algorithmic framework whose definition is completed by expliciting the pruning tests. These can be of geometrical or of physical-chemical nature. An important feature of BP is that, in exponential worst-case (but practically very short) time, it will find *all* incongruent solutions to a given instance.

4.1 Algorithmic framework

Let T be a graph representation of the search tree. Initially, T is initialized to the search nodes $1 \to 2 \to 3 \to 4$ (no branching), since the first three atoms can be fixed to feasible positions x_1, x_2, x_3 and the fourth atom x_4 can be fixed to any of its two possible positions by Theorem 2. By the current rank of the search tree, we mean the index of the atom being placed at the current node. At each search tree node of rank i we store:

- the position $x_i \in \mathbb{R}^3$ of the i-th atom;
- the cumulative product $Q_i = \prod_{j=1}^{i} B_j$ of the torsion matrices;
- a pointer to the parent node $P(i)$;
- pointers to the subnodes $L(i), R(i)$ (initialized to a dummy value PRUNED if infeasible).

Notice that the edge structure of the graph T is encoded in the operators $P(), L(), R()$ defined at each node. The recursive procedure at rank $i - 1$ is given in Algorithm 1. Let $y = (0, 0, 0, 1)^\top$, $\varepsilon > 0$ a given tolerance and v a node with rank $i - 1$ in the search tree T.

4.2 Euclidean bounds pruning tests

Step 8 in Alg. 1 can be enhanced in several ways. We describe here an improvement based on shortest path computations; in particular, we employ the fact that interatomic distances are assumed to be Euclidean. Much like the pruning of the search

Algorithm 1 BP algorithm.

```
BranchAndPrune($T$, $v$, $i$)
if ($i \leq n-1$) then
   // Compute the possible placements for $i$-th atom:
   calculate the torsion matrices $B_i, B'_i$ via Eq. (3);
   retrieve the cumulative torsion matrix $Q_{i-1}$ from the parent node $P(v)$;
   compute $Q_i = Q_{i-1}B_i$, $Q'_i = Q_{i-1}B'_i$ and $x_i, x'_i$ from $Q_i y, Q'_i y$;
   let $\lambda = 1, \rho = 1$;
   // Pruning tests:
   if ($x_i$ is feasible) then
      create a node $z$, store $Q_i$ and $x_i$ in $z$, let $P(z) = v$ and $L(v) = z$;
      set $T \leftarrow T \cup \{z\}$;
      BranchAndPrune($T$, $z$, $i+1$);
   else
      set $L(v) =$ PRUNED;
   end if
   if ($x'_i$ is feasible) then
      create a node $z'$, store $Q_i$ and $x_i$ in $z'$, let $P(z) = v$ and $R(v) = z'$;
      set $T \leftarrow T \cup \{z'\}$;
      BranchAndPrune($T$, $z'$, $i+1$);
   else
      set $R(v) =$ PRUNED;
   end if
else
   // Rank $n$ reached, a solution was found:
   solution stored in parent nodes ranked $n$ to 1, output by back-traversal;
end if
```

scope in point-to-point Dijkstra shortest-path searches on Euclidean graphs, we can prune away an atomic position i if it is too far with respect to the given distances. Consider atoms h, i, k with $h < i < k$ such that $\{h, k\} \in E$ (so that d_{hk} is known). Assume that the BP has already placed atom h and that we are now verifying feasibility for atom i. Let $D(i, k)$ be an upper bound to the distance $||x_i - x_k||$ for all possible embeddings $x : V \to \mathbb{R}^3$ which are feasible DMDGP solutions.

Lemma 4 *If $D(i, k) < ||x_h - x_i|| - d_{hk}$ for all feasible $x : V \to \mathbb{R}^3$, then the BP search node for atomic position x_i can be pruned.*

Proof Suppose, to get a contradiction, that position x_i is feasible for the DMDGP instance being solved. By definition, $D(i, k) \geq ||x_i - x_k||$. Since distances are Euclidean, $||x_i - x_k|| \geq ||x_h - x_i|| - ||x_h - x_k||$. Hence $D(i, k) \geq ||x_h - x_i|| - d_{hk} > D(i, k)$, which is a contradiction. □

By Lemma 4, every upper bound $D(i, k)$ to the distance $||x_i - x_k||$ provides a valid pruning test. Furthermore, in all Euclidean graphs the Euclidean distance between two vertices is a lower bound to the cost of all paths joining the two vertices in the graph. We therefore let $D(i, k)$ be the cost of the shortest path from i to k in G, which provides a valid pruning test. However, we will show in Section 5 that this pruning test is able to improve the overall performances of the BP algorithm in few examples only. This pruning test will be referred to as DSP (Dijkstra Shortest-Paths).

5 Computational results

In order to benchmark the proposed method, we test a class of artificial MDGP instances described in [35] and some real instances (proteins) from the *Protein Data Bank* (PDB) [6], which can be accessed at `http://www.rcsb.org/pdb/`. In all the tables, the instances are described by their names, their atomic sizes n and the number of given distances $|E|$. The results in Tables 1-7 refer to comparative results obtained from four methods: BP stopped after the first solution is found (BP-One), BP run to completion (BP-All), an implementation of the GCA algorithm called DGSOL [51] and SDP-based facial reduction [32].

We compare these methods using three measures: CPU time, Largest Distance Error (LDE), defined as

$$\mathrm{LDE} = \frac{1}{|E|} \sum_{\{i,j\} \in E} \frac{|\ ||x_i - x_j|| - d_{ij}|}{d_{ij}},$$

and Root Mean Square Deviation (RMSD) [14], which is a distance from a given conformation which is considered correct. Although the RMSD is often used to show how well algorithms perform in terms of distance to the “right” protein listed in the PDB, it has the disadvantage that it *requires* a solution to be known a priori. Not all measures are meaningful when applied to all methods: for example the RMSD is only meaningful for conformations whose LDE is zero or close to zero (i.e. if the LDE is large then obviously the conformation is far from the correct one).

For the BP-One method we report CPU time and LDE in Tables 1-2 and 4-5; the RMSD value is reported for BP-All (Table 3) but not for BP-One because its solution is also in the set of all solutions found by BP-All. For BP-All we also report the cardinality of the solution set (#Sol; the number of solutions is in fact 2 × #Sol, due to Theorem 2) in Tables 1-2 and 5. For DGSOL we report CPU and LDE in Tables 4-5; we do not report the RMSD because the LDE values are too large. For the SDP facial reduction algorithm we report CPU, LDE (Table 6) and RMSD (Table 7).

Benchmarking a single method for a new problem is difficult because of the lack of alternative methods. Our comparisons, based on DGSOL and the SDP-based facial reduction technique, are unfair from an algorithmic point of view: neither the SDP-based techniques nor DGSOL require a vertex order satisfying Assumptions 1-2, and both can also work with interval distances. On the other hand, no technique but BP-All can find all incongruent conformations satisfying the distance constraints.

Overall, as far as CPU time is concerned, we have the order BP-One < BP-All < SDP-based technique < DGSOL (this is contradicted by very few instances). As for the LDE, we have SDP-based technique < BP-One = BP-All < DGSOL: in particular, the LDEs found by SDP-based technique and BP-One/BP-All are virtually the same (in the interval $[10^{-14}, 10^{-9}]$), whereas DGSOL is closer to 10^{-1}, which essentially means that the conformation is wrong. Concerning the RMSD, the order is BP-All < BP-One = SDP-based techniques; we recall that we did not compute the RMSD for DGSOL because of the fairly large LDE scores. For practical applications to proteomics, the unique feature of BP-All of being able to compute all incongruent solutions is very valuable, as the practitioner will be able to choose the correct conformation based on biochemical criteria. Although the version of BP-All described in this paper cannot deal with interval distances (a necessary feature for working with proteins), an extension in this direction is currently under way [52].

All tests have been carried out on a single core of an Intel Core 2 CPU 6400 @ 2.13 GHz with 4GB RAM, running Linux. The code implementing the BP algorithm has been compiled by the GNU C++ compiler v.4.1.2 with the `-O3` flag.

5.1 Instances

We consider two types of instances: artificially generated "Lavor" instances [35] and a set of 25 protein backbones obtained from the PDB. The Lavor instances are based on the model proposed by [57], whereby a molecule is represented as a linear chain of atoms. Bond lengths and angles are kept fixed, and a set of likely torsion angles is generated randomly. Depending on the initial choice of bond lengths and angles, the Lavor instances give rather more realistic models of proteins than other randomly generated instances do (such as for example the instances described in [51]). We name the Lavor instances as `lavor`n, where n is the number of atoms in the molecule. We generate and test different Lavor instances for each size $n \in \{10, \dots, 70\} \cup \{100i | 1 \leq i \leq 10\}$.

The set of 25 protein backbones from the PDB includes some of the proteins used by Biswas, Toh and Ye in [10], by Wu and Wu in [68], and by Hendrickson in [28]. We do not consider proteins formed by more than one chain of amino acids; for each protein, only distances between atoms not greater than 6Å are considered as input for the algorithms.

5.2 Experiments on BP

Both BP-One and BP-All implement the two pruning tests DDF and DSP, where we set $\epsilon = 0.001$ when the pruning test DDF is used. We investigate the efficiency of the methods when only DDF is used or both of them are used. We always use DDF because it is very simple and it represents a very natural way for pruning atoms: if the known and obtained distances do not match, the computed atom position cannot be considered. The pruning test DSP is more complex: it requires that shortest paths in the graph G need to be computed. This task can be computationally demanding, and moreover the same shortest path may be needed more than once during the algorithm. For this reason, we compute all the possible shortest paths in G before the BP algorithm starts, using the well-known Floyd-Warshall algorithm [24]: this is the standard algorithm for computing all shortest paths in a graph; its complexity is $O(n^3)$. In the Tables 1 and 2, #DDF and #DSP represent, respectively, the number of times the pruning tests DDF and DSP pruned atom positions. Note that DDF is always applied before DSP and that, if DDF prunes an atom, there is no need to use DSP.

In Table 1, BP-One and BP-All only employ the DDF pruning test. BP-One is very fast in finding a solution with good accuracy. It never takes more than 10 seconds on large instances, with LDE values ranging from 10^{-6} to 10^{-16}. The time obviously increases when BP-All is used, due to the large number of solutions. We remark that a limit of 1h of user CPU time is enforced on BP-All.

In Table 2, the two methods employ both pruning tests (DDF and DSP). The quality is comparable to Table 1. The CPU time is different due to the use of both pruning tests. For all instances, the number of atoms pruned by DDF when it is used by itself (see Table 1) is never greater than the number of atoms pruned by DDF and

Instance			*BP-One*			*BP-All*		
Name	n	$\|E\|$	*CPU*	*#DDF*	*LDE*	*CPU*	*#DDF*	*#Sol*
lavor10	10	24	0.00	0	1.63e-16	0.00	0	64
lavor15	15	70	0.00	7	1.08e-09	0.00	11	1
lavor20	20	103	0.00	9	1.28e-09	0.00	16	1
lavor25	25	106	0.00	10	1.62e-09	0.00	49	2
lavor30	30	219	0.00	15	3.86e-09	0.00	381	2
lavor35	35	166	0.00	13	1.22e-09	0.00	169	16
lavor40	40	306	0.00	16	5.61e-06	0.00	136	2
lavor45	45	351	0.00	30	4.79e-09	0.00	58	1
lavor50	50	203	0.00	49	6.50e-10	0.07	23364	512
lavor55	55	224	0.00	26	1.43e-09	3.45	1304580	262144
lavor60	60	227	0.00	17	2.05e-09	0.62	262428	8192
lavor65	65	455	0.00	1165	7.89e-09	0.02	5184	8
lavor70	70	331	0.00	25	1.23e-08	16.30	2798220	4194304
lavor100	100	605	1.94	815010	5.58e-09	5.31	2230462	1
lavor200	200	1844	0.00	394	7.60e-08	0.05	17649	16
lavor300	300	2505	0.03	9265	1.45e-08	0.05	19182	2
lavor400	400	2600	0.02	2592	3.37e-09	1h	728896182	41600953
lavor500	500	4577	0.50	133833	1.62e-07	1h	617225487	1148416
lavor600	600	5473	0.00	1538	5.06e-08	1h	171487104	249589548
lavor700	700	4188	0.24	55579	3.62e-08	1h	423613358	1852485
lavor800	800	6850	9.79	1132948	1.67e-08	1h	371154974	307152
lavor900	900	7965	2.52	908990	6.30e-08	1h	228252250	52905984
lavor1000	1000	8229	4.04	182362080	2.00e-08	1h	250543954	399

Table 1 Lavor instances solved by the methods BP-One and BP-All, where only the pruning test DDF is used.

Instance			*BP-One*				*BP-All*			
Name	n	$\|E\|$	*CPU*	*#DDF*	*#DSP*	*LDE*	*CPU*	*#DDF*	*#DSP*	*#Sol*
lavor10	10	24	0.00	0	0	1.63e-16	0.00	0	0	64
lavor15	15	70	0.00	7	0	1.08e-09	0.00	11	0	1
lavor20	20	103	0.00	9	0	1.28e-09	0.00	16	0	1
lavor25	25	106	0.00	8	1	1.62e-09	0.00	24	5	2
lavor30	30	219	0.00	15	0	3.86e-09	0.00	277	10	2
lavor35	35	166	0.00	9	1	1.22e-09	0.00	145	8	16
lavor40	40	306	0.00	16	0	5.61e-06	0.00	64	4	2
lavor45	45	351	0.00	17	2	4.79e-09	0.00	39	3	1
lavor50	50	203	0.00	24	8	6.50e-10	0.05	5924	5648	512
lavor55	55	224	0.00	22	2	1.43e-09	7.02	1173508	65536	262144
lavor60	60	227	0.00	13	2	2.05e-09	1.22	221462	16387	8192
lavor65	65	455	0.00	720	115	7.89e-09	0.01	3200	528	8
lavor70	70	331	0.00	25	0	1.23e-08	38.50	2732664	32774	4194304
lavor100	100	605	0.44	39800	4560	5.58e-09	1.23	108864	12238	1
lavor200	200	1844	0.02	113	9	7.60e-08	0.10	3289	307	16
lavor300	300	2505	0.09	649	63	1.45e-08	0.16	2719	298	2
lavor400	400	2600	0.17	937	292	3.37e-09	1h	150759333	7451646	9626711
lavor500	500	4577	0.42	851	623	1.62e-07	1h	44790111	27674559	1784456
lavor600	600	5473	0.58	481	100	5.06e-08	1h	56466165	623012	39866479
lavor700	700	4188	2.35	4223	2143	3.62e-08	1h	51582169	14136943	406571
lavor800	800	6850	29.40	59181	38122	1.67e-08	1h	22143204	3924390	174294
lavor900	900	7965	21.80	223996	23440	6.30e-08	1h	57254854	454984	13480402
lavor1000	1000	8229	8.35	48482	4341	2.00e-08	1h	43128660	3833147	416

Table 2 Lavor instances solved by the methods BP-One and BP-All, where both the pruning tests DDF and DSP are used.

DSP in cooperation (see Table 2). This means that DSP is able to prune atoms that DDF does not prune. In practice, when DDF and DSP work together, more atomic position are marked as infeasible and pruned earlier on the search tree, so that fewer search nodes are explored. Unfortunately, this reduction in the number of search nodes is counterbalanced by longer processing times for each node.

In Table 3 we compare LDE and RMSD values. Only instances having no more than 4 solutions in total (including the ones that can be generated by the symmetry given by Theorem 2) are considered. Two symmetric solutions have exactly the same LDE values, but they can have different RMSD values. In all the cases, there is at least one solution with small RMSD value. Since the original PDB files we used for generating the instances are precise to the third decimal digit [6,14], the values we obtained (in the order of 10^{-7}) are enough to decide that the conformation is correct.

Name	n	$\|E\|$	solution	*LDE*	*RMSD*
`1brv`	57	476	1	1.39e-14	4.17e+00
			2	1.39e-14	**5.60e-08**
`1aqr`	120	929	1	7.42e-07	6.94e+00
			2	3.10e-13	6.94e+00
			3	3.10e-13	**7.10e-08**
			4	7.42e-07	8.08e-04
`2erl`	120	1136	1	1.33e-14	**3.69e-07**
			2	1.33e-14	6.26e+00
`1crn`	138	1250	1	2.24e-13	**3.86e-06**
			2	2.24e-13	6.50e+00
`1ahl`	147	1205	1	9.98e-13	**1.45e-06**
			2	9.98e-13	7.54e+00
`1ptq`	150	1263	1	2.30e-13	**2.40e-06**
			2	2.30e-13	7.47e+00
`1brz`	159	1394	1	4.48e-13	**1.95e-07**
			2	3.60e-07	6.86e-04
			3	3.60e-07	7.79e+00
			4	4.48e-13	7.79e+00
`1hoe`	222	1995	1	3.18e-13	**3.54e-06**
			2	3.18e-13	7.67e+00
`1lfb`	232	2137	1	5.31e-14	**2.50e-06**
			2	5.31e-14	9.44e+00
`1pht`	249	2283	1	2.73e-12	**8.93e-07**
			2	2.73e-12	1.06e+01
`1jk2`	270	2574	1	2.09e-13	**6.37e-06**
			2	2.09e-13	8.32e+00
`1f39a`	303	2660	1	1.88e-08	**4.13e-06**
			2	1.88e-08	1.04e+01
`1acz`	324	3060	1	2.75e-12	**1.69e-06**
			2	1.31e-07	2.22e-04
			3	1.40e-07	1.85e-04
			4	2.71e-07	2.88e-04
			5	2.71e-07	9.75e-00
			6	1.40e-07	9.75e-00
			7	1.31e-07	9.75e-00
			8	2.75e-12	9.75e-00
`1poa`	354	3193	1	1.36e-13	**5.83e-06**
			2	1.36e-13	8.73e+00
`1fs3`	378	3443	1	8.08e-13	**1.40e-06**
			2	8.08e-13	1.16e+01
`1mbn`	459	4599	1	1.36e-09	1.11e+01
			2	1.36e-09	**6.06e-07**
`1rgs`	792	7626	1	4.22e-13	**1.24e-06**
			2	4.22e-13	1.53e+01
`1m40`	1224	20382	1	1.00e-12	**5.20e-07**
			2	1.00e-12	1.52e+01
`1bpm`	1443	14292	1	2.85e-13	1.86e+01
			2	2.85e-13	**5.09e-06**
`1n4w`	1610	16940	1	1.19e-12	1.96e+01
			2	1.19e-12	**6.51e-07**
`1mqq`	2032	19564	1	4.90e-12	2.05e+01
			2	4.90e-12	**9.78e-06**
`1rwh`	2265	21666	1	2.08e-13	**9.44e-07**
			2	2.08e-13	2.16e+01
`3b34`	2790	29188	1	1.17e-11	**1.30e-06**
			2	1.17e-11	2.43e+01
`2e7z`	2907	42098	1	4.26e-12	**7.68e-07**
			2	4.26e-12	2.27e+01
`1epw`	3861	35028	1	3.44e-12	**4.98e-06**
			2	3.44e-12	2.02e+01

Table 3 Comparisons between the solutions found by BP and the original protein conformations. The solutions that can be obtained by symmetry are also included.

As explained above, at each step the BP algorithm checks the feasibility of two possible atom positions, corresponding with the two possible choices of the sign of the sine of the torsion angle ω: $+$ or $-$. The first three atoms always have a positive sign. The fourth sign is also fixed to $+$ in the BP algorithm: if BP finds #Sol solutions, then other #Sol solutions can be obtained by symmetry inverting all the signs from the fourth to the last one, as stated in Theorem 2. Note that two different solutions of the same instance can differ only in the sign of one torsion angle. A different torsion angle in only one point of the protein backbone can change the associated RMSD value dramatically. For example, if we consider the instance `1aqr`, we can see (Tables 5 and 3) that BP is able to find 2 solutions, and therefore we have 4 solutions in total. The solutions 3 and 4 correspond with small RMSD values, and, in particular, the solution

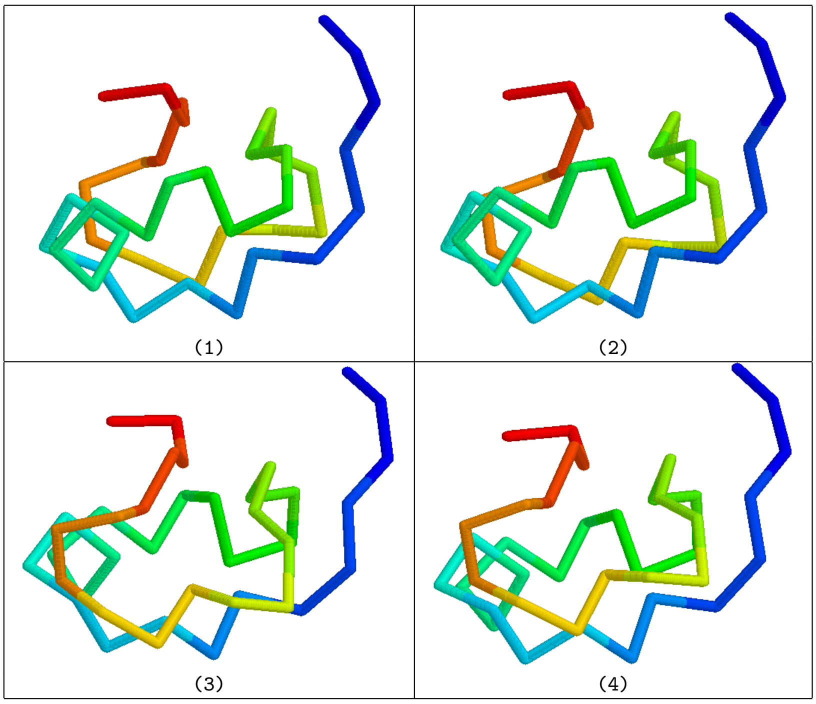

Fig. 3 The four solutions of the instance `1aqr`. Solutions `(1)` and `(2)` are found by BP-All. Solutions `(3)` and `(4)` are symmetric with respect to the other two.

3 corresponds with the smallest one. Solutions 3 and 4 only differ by one torsion angle, and this change is able to increase the associated RMSD value from about 10^{-8} to about 10^{-4}. For the same instance `1aqr`, the RMSD values for the solution 1 and 2 are apparently the same. However, when we computed the RMSDs with a higher precision, we found that there is a positive difference, caused by the only different torsion angle: the RMSD associated to the solution 1 is 6.938957, and the RMSD associated to the solution 2 is 6.938940.

Figure 3 shows the backbone of the four solutions related to the instance `1aqr`. Solutions (1) and (2) are found by BP-All. Solutions (3) and (4) are obtained by symmetry applying the Theorem 2. The colors used for representing the conformations range from blue (first amino acid) to red (last amino acid). Note that only the protein backbone is shown, which is usually represented as the set of segments connecting consecutive C_α Carbon atoms in the conformation. The only difference between the two solutions (1) and (2) is in only one torsion angle. The only difference between the two solutions (3) and (4) obtained by symmetry is in the same torsion angle.

We also performed more extensive tests on a much larger set of protein instances from the PDB: we queried the PDB website for monomeric proteins (i.e. with only one chain of amino acids) having a resolution ranging from 0 and 1.5Å. From this set, we eliminated all proteins having a sequence similarity greater than 30%. The resulting set contains about 700 protein conformations: for each of these, we generated a DMDGP instance by only keeping distances smaller than 6Å, and we only kept those whose

Instance			*DGSOL*	
Name	n	$\|E\|$	*CPU*	*LDE*
lavor10	10	24	0.03	3.01e+01
lavor15	15	70	0.05	0.00e+00
lavor20	20	103	0.08	0.00e+00
lavor25	25	106	0.24	2.62e-02
lavor30	30	219	1.02	4.43e-07
lavor35	35	166	1.38	1.00e-01
lavor40	40	306	0.57	1.94e-06
lavor45	45	351	1.33	9.47e-07
lavor50	50	203	1.55	7.43e-02
lavor55	55	224	2.06	2.31e-03
lavor60	60	227	0.41	1.50e+03
lavor65	65	455	2.94	1.27e-01
lavor70	70	331	5.10	9.60e-02
lavor100	100	605	4.40	1.67e-01
lavor200	200	1844	43.94	4.08e-01
lavor300	300	2505	58.81	3.49e-01
lavor400	400	2600	65.35	6.87e-01
lavor500	500	4577	239.97	8.06e-01
lavor600	600	5473	244.45	6.99e-01
lavor700	700	4188	223.52	4.99e-01
lavor800	800	6850	447.27	6.07e-01
lavor900	900	7965	440.30	6.41e-01
lavor1000	1000	8229	534.97	6.95e-01

Table 4 Lavor instances solved by DGSOL.

natural backbone order satisfied Assumptions 1-2 (see [36] for those orders that fail to satisfy the assumptions). On this large set, BP-All never took more than 0.1 seconds for finding all incongruent solutions. The quality of the solutions is very accurate in all the cases: the LDE value is, in most of the cases, about 10^{-13}. When BP-All is applied, the number of solutions is almost always 1; two solutions are found for 1hx0, 1itx, 1r6x, four solutions are found for 2p9w and thirty-two for 1iom.

5.3 Comparison with DGSOL

In Table 4, we show computational results obtained using DGSOL for solving the same instances used in Tables 1 and 2. Comparing the LDE values of the solutions, it is easy to see that BP is able to find much more accurate solutions than DGSOL in less time. Table 5 shows the results for the set of protein instances.

5.4 Comparison with the SDP-based facial reduction technique

The BP algorithm was compared to the SDP-based facial reduction technique on the set of protein instances. Table 6 reports the LDE comparison, whereas Table 7 reports the RMSD comparison. The average LDE found by BP is in the order of 10^{-10}, whereas the SDP-based technique yields 10^{-13}; if we eliminate the three outliers in the BP columns, the average score for BP is in the order of 10^{-12}. The RMSD comparison is in favour of the BP-All because it explores the set of all incongruent solutions, which must also contain the correct solution. The BP algorithm is faster than the SDP-based technique. We recall that the comparative tests are limited to protein backbones.

6 Relations with the Euclidean Distance Matrix Completion Problem

The MDGP is closely related to the following problem:

Instance			*BP-One*		*BP-All*		*DGSOL*	
Name	*n*	$\|E\|$	*CPU*	*LDE*	*CPU*	*#Sol*	*CPU*	*LDE*
1brv	57	476	0.00	1.54e-14	0.00	1	1.48	2.74e-01
1aqr	120	929	0.00	1.86e-09	0.00	2	7.77	4.88e-01
2erl	120	1136	0.00	1.33e-14	0.00	1	9.38	2.92e-01
1crn	138	1250	0.00	2.24e-13	0.00	1	9.47	2.24e-01
1ahl	147	1205	0.00	1.50e-09	0.00	2	6.95	1.46e-01
1ptq	150	1263	0.00	2.30e-13	0.00	1	9.16	1.21e-01
1brz	159	1394	0.00	3.53e-13	0.00	2	11.39	4.66e-01
1hoe	222	1995	0.00	3.18e-13	0.00	1	16.83	2.06e-01
1lfb	232	2137	0.00	5.31e-14	0.00	1	38.94	2.88e-01
1pht	249	2283	0.00	2.73e-12	0.00	1	42.50	2.00e-01
1jk2	270	2574	0.00	2.09e-13	0.00	1	86.98	4.05e-01
1f39a	303	2660	0.00	2.68e-12	0.00	1	37.24	2.80e-01
1acz	324	3060	0.00	3.15e-12	0.02	8	35.97	3.97e-01
1poa	354	3193	0.00	1.36e-13	0.00	1	64.03	4.67e-01
1fs3	378	3443	0.00	8.08e-13	0.01	1	54.68	2.69e-01
1mbn	459	4599	0.00	1.36e-09	0.00	1	124.24	4.46e-01
1rgs	792	7626	0.00	4.22e-13	0.01	1	237.93	4.69e-01
1m40	1224	20382	0.02	1.00e-12	5.26	1	1142.49	4.89e-01
1bpm	1443	14292	0.02	2.85e-13	0.02	1	398.29	5.06e-01
1n4w	1610	16940	0.02	1.19e-12	0.02	1	994.51	5.26e-01
1mqq	2032	19564	0.02	4.90e-12	0.06	1	451.58	5.40e-01
1rwh	2265	21666	0.02	2.08e-13	0.06	1	934.29	5.38e-01
3b34	2790	29188	0.07	1.17e-11	0.07	1	940.95	6.47e-01
2e7z	2907	42098	0.08	4.26e-12	0.09	1	915.39	6.40e-01
1epw	3861	35028	0.16	3.19e-12	0.25	1	2037.86	4.92e-01

Table 5 PDB instances solved by the methods BP-One, BP-All and DGSOL.

Instance			*BP-One*		*SDP-based*	
Name	*n*	$\|E\|$	*CPU*	*LDE*	*CPU*	*LDE*
1brv	57	476	0.00	1.54e-14	0.03	1.24e-14
1aqr	120	929	0.00	1.86e-09	0.06	2.54e-13
2erl	120	1136	0.00	1.33e-14	0.06	2.52e-13
1crn	138	1250	0.00	2.24e-13	0.06	2.24e-14
1ahl	147	1205	0.00	1.50e-09	0.07	2.41e-14
1ptq	150	1263	0.00	2.30e-13	0.08	2.54e-14
1brz	159	1394	0.00	3.53e-13	0.07	2.01e-13
1hoe	222	1995	0.00	3.18e-13	0.12	1.31e-13
1lfb	232	2137	0.00	5.31e-14	0.11	1.86e-14
1pht	249	2283	0.00	2.73e-12	0.10	9.54e-14
1jk2	270	2574	0.00	2.09e-13	0.15	2.74e-14
1f39a	303	2660	0.00	2.68e-12	0.12	3.91e-13
1acz	324	3060	0.00	3.15e-12	0.13	3.04e-13
1poa	354	3193	0.00	1.36e-13	0.20	2.53e-12
1fs3	378	3443	0.00	8.08e-13	0.17	2.27e-13
1mbn	459	4599	0.00	1.36e-09	0.22	9.67e-14
1rgs	792	7626	0.01	4.22e-13	0.42	1.58e-13
1m40	1224	20382	0.02	1.00e-12	0.71	1.08e-12
1bpm	1443	14292	0.03	2.85e-13	0.76	7.73e-13
1n4w	1610	16940	0.03	1.19e-12	0.86	5.44e-13
1mqq	2032	19564	0.03	4.90e-12	1.22	6.17e-13
1rwh	2265	21666	0.04	2.08e-13	1.38	3.01e-12
3b34	2790	29188	0.07	1.17e-11	1.68	3.00e-13
2e7z	2907	42098	0.24	4.26e-12	1.88	2.88e-13
1epw	3861	35028	0.33	3.19e-12	2.31	1.45e-12

Table 6 PDB instances solved by the BP-One and the SDP-based facial reduction algorithm.

> Euclidean Distance Matrix Completion Problem (EDMCP). Given a simple weighted undirected graph $G = (V, E, d)$ and a positive integer K', is there an integer $K \leq K'$ and an embedding $x : V \to \mathbb{R}^K$ such that $\forall \{u, v\} \in E\ (\|x_u - x_v\| = d_{uv})$?

The above decision problem is usually considered in its optimization version of minimizing K such that the embedding of G exists in $\mathbb{R}^K$. The EDMCP is usually stated in its matrix form, with the graph G replaced by its weighted adjacency matrix $D = (D_{ij})$, where $D_{ij} = d_{ij}$ for all $\{i, j\} \in E$ and D_{ij} unspecified otherwise [34,29,19]. The "partial matrix" is usually assumed to be a *pre-distance matrix*: i.e. a symmetric square matrix with 0 diagonal. There is a decevingly minor but fundamental difference between the MDGP and the EDMCP: the dimension of the embedding Euclidean space is given as part of the input (the constant 3) in the MDGP, whereas it is part of the

			RMSD		
Name	n	$\lvert E\rvert$	BP-One	BP-All	SDP-based
1brv	57	476	4.17e+00	5.60e-08	4.17e+00
1aqr	120	929	6.94e+00	7.10e-08	6.93e+00
2erl	120	1136	3.69e-07	3.69e-07	8.06e-05
1crn	138	1250	3.86e-06	3.86e-06	7.03e-05
1ahl	147	1205	1.45e-06	1.45e-06	8.81e-05
1ptq	150	1263	2.40e-06	2.40e-06	6.65e-05
1brz	159	1394	1.95e-07	1.95e-07	5.98e-05
1hoe	222	1995	3.54e-06	3.54e-06	4.02e-05
1lfb	232	2137	2.50e-06	2.50e-06	4.92e-05
1pht	249	2283	8.93e-07	8.93e-07	3.51e-05
1jk2	270	2574	6.37e-06	6.37e-06	2.35e-05
1f39a	303	2660	4.13e-06	4.13e-06	4.13e-05
1acz	324	3060	1.69e-06	1.69e-06	4.14e-05
1poa	354	3193	5.83e-06	5.83e-06	3.80e-05
1fs3	378	3443	1.40e-06	1.40e-06	3.59e-05
1mbn	459	4599	1.11e+01	6.06e-07	1.11e+01
1rgs	792	7626	1.24e-06	1.24e-06	1.39e-05
1m40	1224	20382	5.20e-07	5.20e-07	1.32e-05
1bpm	1443	14292	1.86e+01	5.09e-06	1.86e+01
1n4w	1610	16940	1.96e+01	6.51e-07	1.96e+01
1mqq	2032	19564	2.05e+01	9.78e-06	2.05e+01
1rwh	2265	21666	9.44e-07	9.44e-07	2.93e-05
3b34	2790	29188	1.30e-06	1.30e-06	1.08e-05
2e7z	2907	42098	7.68e-07	7.68e-07	1.01e-05
1epw	3861	35028	4.98e-06	4.98e-06	1.36e-05

Table 7 RMSD comparison between the BP algorithm and the SDP-based technique.

output in the EDMCP. Trivial deductions of complexity results by inclusion are therefore impossible. Indeed, whereas the MDGP is **NP**-hard, there is no known polynomial reduction from an **NP**-hard problem to the EDMCP.

By [64], if we border a pre-distance matrix with a left column and top row such that $D_{00} = 0$ and $\forall i \leq k \leq n \; D_{k0} = D_{0k} = (2n \sum_j D_{kj} - \sum_{ij} D_{ij})/(2n^2)$ and define $A = (A_{ij})$ where $A_{ij} = \frac{1}{2}(D_{0i} + D_{0j} - D_{ij})$ then D is a Euclidean Distance Matrix (EDM) if and only if A is a Positive SemiDefinite (PSD) matrix. Using SDP techniques (interior point methods) it is possible to solve the PSD Completion Problem (PSDCP) to any desired accuracy $\varepsilon > 0$ in polynomial time [30]. In order to obtain the equivalent EDM one must solve the linear system $\forall i, j \; 2A_{ij} = D_{0i} + D_{0j} - D_{ij}$ (this can be done in $O(|V|^3)$). Once D is known, an embedding in $\mathbb{R}^K$ with minimum K can be found using bisection on K ($O(\log |V|)$, since $|V|$ is an upper bound for the embedding space dimension) and solving an MDGP in $\mathbb{R}^K$ on a complete graph ($O(|V|)$ using [20]). This gives a polynomial time reduction from the PSDCP to the EDMCP. The practical limitation is that it is not clear how the desired accuracy ε varies with respect to this reduction.

If we consider a natural generalization of the DMDGP to embeddings in arbitrary dimension K', then cliques of quadruplets of consecutive vertices are replaced by cliques of K'-uples of consecutive vertices in Assumption 1 of the DMDGP definition, and the strict triangular inequalities of Assumption 2 are replaced by strict simplex inequalities [13]. The subclass of EDMCP instances that satisfy these generalized assumptions contains symmetric, zero-diagonal banded matrices whose semi-band height is equal to K', with scattered nonzero entries outside the bands. The nonzeros in the band correspond to the distances that ensure that the search space is discrete. The nonzeros outside the band correspond to distances used for DDF pruning tests. Naturally, we also include matrices for which there is a column permutation that transforms them in the desired banded form. One could then perform a bisection search to identify the minimum dimension of the embedding space K by running a BP algorithm for each tested $K \leq K'$. We shall develop this idea in further work.

7 Conclusion

In this paper, we formally define a subclass (called DMDGP) of the Molecular Distance Geometry Problem, related to proteins, for which a discrete formulation can be supplied. We prove that the DMDGP is **NP**-hard. We then present a solution algorithm for the DMDGP whose worst-case running time is exponential, but whose practical performance allows us to find all incongruent conformations of protein backbones of almost 4000 atoms in 0.25s with high accuracy, with respect to both LDE and RMSD error measures.

7.1 Ongoing work

Work is ongoing in several directions.

- Adapting the BP algorithm to work with data coming from NMR experiments. Such data present several limitations, the main ones being that: (a) the distance function maps into the set of real intervals (this accommodates measurement errors), (b) the most reliable distance measurements are between hydrogen atoms. We address limitation (a) in [43] and (b) in [44].
- BP parallelization [53]. We are currently able to solve DMDGP instances of 10000 vertices in 13.38s on 1 CPU and on 1.57s on 64 CPUs (computations carried out on the Grid 5000 — `www.grid5000.fr`).
- A study of the relation between the number of solutions of the DMDGP and the ε constant used in pruning tests [52].
- A study of symmetries in the DMDGP, which explains why the number of solutions is a power of 2 in all presented tables [49].
- An analysis of the average BP execution time attempting to explain the fact that the observed time increase look linear (instead of exponential) in function of the input size.

In future work, we plan to extend the BP so that it is able to deal with side chains. In order to do this we propose use the methods of [36] to find a vertex order that also includes the side chain atoms. In case such an order is not found, continuous optimization methods will be employed: once the side chain is embedded independently of the rest of the protein, it will be possible to establish whether it can be glued to the main backbone or not. This will provide a further pruning test for the BP.

Eventually, we believe that BP will serve as the primary algorithm in a practically efficient software to be used by NMR practitioners to study the conformation of new proteins. Such a software is already under way [55] and the current implementation is freely distributed `http://www.antoniomucherino.it/en/mdjeep.php`.

Acknowledgments

The authors are indebted to Dr. C. Dürr, to whom much of the material on complexity is due, and to one anonymous referee for insightful comments. The authors would also like to thank for financial support: the Brazilian research agencies FAPESP and CNPq, the French research agency CNRS and École Polytechnique.

References

1. L.T. Hoai An. Solving large scale molecular distance geometry problems by a smoothing technique via the gaussian transform and d.c. programming. *Journal of Global Optimization*, 27:375–397, 2003.
2. L.T. Hoai An and P.D. Tao. Large-scale molecular optimization from distance matrices by a d.c. optimization approach. *SIAM Journal on Optimization*, 14:77–114, 2003.
3. J. Bachrach and C. Taylor. Localization in sensor networks. In I. Stojmenović, editor, *Handbook of Sensor Networks*, pages 3627–3643. Wiley, 2005.
4. G. Barker. The lattice of faces of a finite dimensional cone. *Linear Algebra and its Applications*, 7(1):71–82, 1973.
5. B. Berger, J. Kleinberg, and T. Leighton. Reconstructing a three-dimensional model with arbitrary errors. *Journal of the ACM*, 46(2):212–235, 1999.
6. H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The protein data bank. *Nucleic Acid Research*, 28:235–242, 2000.
7. P. Biswas. *Semidefinite programming approaches to distance geometry problems.* PhD thesis, Stanford University, 2007.
8. P. Biswas, T. Lian, T. Wang, and Y. Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions in Sensor Networks*, 2:188–220, 2006.
9. P. Biswas, T-C. Liang, K-C. Toh, T-C. Wang, and Y. Ye. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE Transactions on Automation Science and Engineering*, 3:360–371, 2006.
10. P. Biswas, K.C. Toh, and Y. Ye. A distributed SDP approach for large-scale noisy anchor-free graph realization with applications to molecular conformation. *SIAM Journal on Scientific Computing*, 30(3):1251–1277, 2008.
11. P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd international symposium on Information processing in sensor networks (IPSN04)*, pages 46–54, New York, NY, USA, 2004. ACM.
12. P. Biswas and Y. Ye. A distributed method for solving semidefinite programs arising from ad hoc wireless sensor network localization. In *Multiscale Optimization Methods and Applications*, volume 82, pages 69–84. Springer, 2006.
13. L. Blumenthal. *Theory and Applications of Distance Geometry.* Oxford University Press, Oxford, 1953.
14. F. Burkowski. *Structural Bioinformatics: an Algorithmic Approach.* CRC Press, New York, 2009.
15. R.S. Carvalho, C. Lavor, and F. Protti. Extending the geometric build-up algorithm for the molecular distance geometry problem. *Information Processing Letters*, 108:234–237, 2008.
16. R. Connelly. Generic global rigidity. *Discrete Computational Geometry*, 33:549–563, 2005.
17. T. Creighton. *Proteins: Structures and Molecular Properties.* Freeman & C., New York, 1993.
18. G.M. Crippen and T.F. Havel. *Distance Geometry and Molecular Conformation.* Wiley, New York, 1988.
19. J. Dattorro. *Convex Optimization and Euclidean Distance Geometry.* $\mathcal{M}\epsilon\beta oo$, Palo Alto, 2005.
20. Q. Dong and Z. Wu. A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances. *Journal of Global Optimization*, 22:365–375, 2002.
21. Q. Dong and Z. Wu. A geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data. *Journal of Global Optimization*, 26:321–333, 2003.
22. T. Eren, D.K. Goldenberg, W. Whiteley, Y.R. Yang, A.S. Morse, B.D.O. Anderson, and P.N. Belhumeur. Rigidity, computation, and randomization in network localization. *IEEE Infocom Proceedings*, pages 2673–2684, 2004.
23. C. Floudas and P. Pardalos, editors. *Encyclopedia of Optimization.* Springer, New York, second edition, 2009.
24. R.W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345, 1962.
25. W. Glunt, T.H. Hayden, S. Hong, and J. Wells. An alternating projection algorithm for computing the nearest Euclidean distance matrix. *SIAM Journal on Matrix Analysis and Applications*, 11(4):589–600, 1990.

26. A. Grosso, M. Locatelli, and F. Schoen. Solving molecular distance geometry problems by global optimization algorithms. *Computational Optimization and Applications*, 43:23–27, 2009.
27. T. Havel. Distance geometry. In D. Grant and R. Harris, editors, *Encyclopedia of Nuclear Magnetic Resonance*, pages 1701–1710. Wiley, New York, 1995.
28. B.A. Hendrickson. The molecule problem: exploiting structure in global optimization. *SIAM Journal on Optimization*, 5:835–857, 1995.
29. H.-X. Huang, Z.-A. Liang, and P. Pardalos. Some properties for the Euclidean distance matrix and positive semidefinite matrix completion problems. *Journal of Global Optimization*, 25:3–21, 2003.
30. C. Johnson, B. Kroschel, and H. Wolkowicz. An interior-point method for approximate positive semidefinite completions. *Computational Optimization and Applications*, 9:175–190, 1998.
31. A. Kearsley, R. Tapia, and M. Trosset. The solution of the metric stress and sstress problems in multidimensional scaling by newton's method. *Computational Statistics*, 13:369–396, 1998.
32. N. Krislock. *Semidefinite Facial Reduction for Low-Rank Euclidean Distance Matrix Completion.* PhD thesis, University of Waterloo, 2010.
33. N. Krislock and H. Wolkowicz. Explicit sensor network localization using semidefinite representations and facial reductions. *SIAM Journal on Optimization*, 20:2679–2708, 2010.
34. M. Laurent. Matrix completion problems. In Floudas and Pardalos [23], pages 1967–1975.
35. C. Lavor. On generating instances for the molecular distance geometry problem. In L. Liberti and N. Maculan, editors, *Global Optimization: from Theory to Implementation*, pages 405–414. Springer, Berlin, 2006.
36. C. Lavor, J. Lee, A. Lee-St. John, L. Liberti, A. Mucherino, and M. Sviridenko. Discretization orders for distance geometry problems. *Optimization Letters*, accepted for publication.
37. C. Lavor, L. Liberti, and N. Maculan. Grover's algorithm applied to the molecular distance geometry problem. In *Proc. of VII Brazilian Congress of Neural Networks, Natal, Brazil*, 2005.
38. C. Lavor, L. Liberti, and N. Maculan. Computational experience with the molecular distance geometry problem. In J. Pintér, editor, *Global Optimization: Scientific and Engineering Case Studies*, pages 213–225. Springer, Berlin, 2006.
39. C. Lavor, L. Liberti, and N. Maculan. The discretizable molecular distance geometry problem. Technical Report q-bio/0608012, arXiv, 2006.
40. C. Lavor, L. Liberti, and N. Maculan. Molecular distance geometry problem. In Floudas and Pardalos [23], pages 2305–2311.
41. C. Lavor, L. Liberti, A. Mucherino, and N. Maculan. On a discretizable subclass of instances of the molecular distance geometry problem. In D. Shin, editor, *Proceedings of the 24th Annual ACM Symposium on Applied Computing*, pages 804–805. ACM, 2009.
42. C. Lavor, A. Mucherino, L. Liberti, and N. Maculan. Computing artificial backbones of hydrogen atoms in order to discover protein backbones. In *Proceedings of the International Multiconference on Computer Science and Information Technology*, pages 751–756, Mragowo, Poland, 2009. IEEE.
43. C. Lavor, A. Mucherino, L. Liberti, and N. Maculan. On the solution of molecular distance geometry problems with interval data. In *Proceedings of the International Workshop on Computational Proteomics*, Hong Kong, 2010. IEEE.
44. C. Lavor, A. Mucherino, L. Liberti, and N. Maculan. On the computation of protein backbones by using artificial backbones of hydrogens. *Journal of Global Optimization*, accepted.
45. L. Liberti, C. Lavor, and N. Maculan. Double VNS for the molecular distance geometry problem. In *Proc. of Mini Euro Conference on Variable Neighbourhood Search, Tenerife, Spain*, 2005.
46. L. Liberti, C. Lavor, and N. Maculan. A branch-and-prune algorithm for the molecular distance geometry problem. *International Transactions in Operational Research*, 15:1–17, 2008.
47. L. Liberti, C. Lavor, N. Maculan, and F. Marinelli. Double variable neighbourhood search with smoothing for the molecular distance geometry problem. *Journal of Global Optimization*, 43:207–218, 2009.
48. L. Liberti, C. Lavor, A. Mucherino, and N. Maculan. Molecular distance geometry methods: from continuous to discrete. *International Transactions in Operational Research*, 18:33–51, 2010.

49. L. Liberti, B. Masson, C. Lavor, J. Lee, and A. Mucherino. On the number of solutions of the discretizable molecular distance geometry problem. Technical Report 1010.1834v1[cs.DM], arXiv, 2010.
50. J.J. Moré and Z. Wu. Global continuation for distance geometry problems. *SIAM Journal of Optimization*, 7(3):814–846, 1997.
51. J.J. Moré and Z. Wu. Distance geometry optimization for protein structures. *Journal of Global Optimization*, 15:219–234, 1999.
52. A. Mucherino and C. Lavor. The branch and prune algorithm for the molecular distance geometry problem with inexact distances. In *Proceedings of the International Conference on Computational Biology*, volume 58, pages 349–353. World Academy of Science, Engineering and Technology, 2009.
53. A. Mucherino, C. Lavor, L. Liberti, and E-G. Talbi. A parallel version of the branch & prune algorithm for the molecular distance geometry problem. In *ACS/IEEE International Conference on Computer Systems and Applications (AICCSA10)*, Hammamet, Tunisia, 2010. IEEE conference proceedings.
54. A. Mucherino, C. Lavor, and N. Maculan. The molecular distance geometry problem applied to protein conformations. In S. Cafieri, A. Mucherino, G. Nannicini, F. Tarissan, and L. Liberti, editors, *Proceedings of the* 8^{th} *Cologne-Twente Workshop on Graphs and Combinatorial Optimization*, pages 337–340, Paris, 2009. École Polytechnique.
55. A. Mucherino, L. Liberti, and C. Lavor. `MD-jeep`: an implementation of a branch-and-prune algorithm for distance geometry problems. In K. Fukuda, J. van der Hoeven, M. Joswig, and N. Takayama, editors, *Mathematical Software*, volume 6327 of *LNCS*, pages 186–197, New York, 2010. Springer.
56. A. Mucherino, L. Liberti, C. Lavor, and N. Maculan. Comparisons between an exact and a metaheuristic algorithm for the molecular distance geometry problem. In F. Rothlauf, editor, *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 333–340, Montreal, 2009. ACM.
57. A.T. Phillips, J.B. Rosen, and V.H. Walke. Molecular structure determination by convex underestimation of local energy minima. In P.M. Pardalos, D. Shalloway, and G. Xue, editors, *Global Minimization of Nonconvex Energy Functions: Molecular Conformation and Protein Folding*, volume 23, pages 181–198. American Mathematical Society, 1996.
58. A. Pogorelov. *Geometry.* Mir Publishers, Moscow, 1987.
59. B. Roth. Rigid and flexible frameworks. *American Mathematical Monthly*, 88(1):6–21, 1981.
60. R. Santana, P. Larrañaga, and J.A. Lozano. Combining variable neighbourhood search and estimation of distribution algorithms in the protein side chain placement problem. In *Proc. of Mini Euro Conference on Variable Neighbourhood Search, Tenerife, Spain*, 2005.
61. R. Santana, P. Larra naga, and J.A. Lozano. Side chain placement using estimation of distribution algorithms. *Artificial Intelligence in Medicine*, 39:49–63, 2007.
62. J.B. Saxe. Embeddability of weighted graphs in k-space is strongly NP-hard. *Proceedings of 17th Allerton Conference in Communications, Control and Computing*, pages 480–489, 1979.
63. T. Schlick. *Molecular modelling and simulation: an interdisciplinary guide.* Springer, New York, 2002.
64. I.J. Schoenberg. Remarks to maurice fréchet's article "sur la définition axiomatique d'une classe d'espaces distanciés vectoriellement applicable sur l'espace de hilbert". *Annals of Mathematics*, 36(3):724–732, 1935.
65. M-C. So and Y. Ye. Theory of semidefinite programming for sensor network localization. *Mathematical Programming*, 109:367–384, 2007.
66. M. Trosset. Applications of multidimensional scaling to molecular conformation. *Computing Science and Statistics*, 29:148–152, 1998.
67. L. Wang, R. Mettu, and B.R. Donald. An algebraic geometry approach to protein structure determination from nmr data. In *Proceedings of the Computational Systems Bioinformatics Conference.* IEEE, 2005.
68. D. Wu and Z. Wu. An updated geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data. *Journal of Global Optimization*, 37:661–673, 2007.
69. D. Wu, Z. Wu, and Y. Yuan. Rigid versus unique determination of protein structures with geometric buildup. *Optimization Letters*, 2(3):319–331, 2008.
70. Z. Zou, R. Bird, and R. Schnabel. A stochastic/perturbation global optimization algorithm for distance geometry problems. *Journal of Global Optimization*, 11:91–105, 1997.

Appendix

Proof of LEMMA 2

The proof is by induction. For $n = 4$, we obtain:

$$Q_4 = \begin{bmatrix} -\cos\theta_{2,4} & -\sin\theta_{2,4} & 0 & -d_{3,4}\cos\theta_{2,4} \\ \sin\theta_{2,4}\cos\omega_{1,4} & -\cos\theta_{2,4}\cos\omega_{1,4} & -\sin\omega_{1,4} & d_{3,4}\sin\theta_{2,4}\cos\omega_{1,4} \\ \sin\theta_{2,4}\sin\omega_{1,4} & -\cos\theta_{2,4}\sin\omega_{1,4} & \cos\omega_{1,4} & d_{3,4}\sin\theta_{2,4}\sin\omega_{1,4} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$Q_4' = \begin{bmatrix} -\cos\theta_{2,4} & -\sin\theta_{2,4} & 0 & -d_{3,4}\cos\theta_{2,4} \\ \sin\theta_{2,4}\cos\omega_{1,4} & -\cos\theta_{2,4}\cos\omega_{1,4} & -\big(-\sin\omega_{1,4}\big) & d_{3,4}\sin\theta_{2,4}\cos\omega_{1,4} \\ \sin\theta_{2,4}\big(-\sin\omega_{1,4}\big) & -\cos\theta_{2,4}\big(-\sin\omega_{1,4}\big) & \cos\omega_{1,4} & d_{3,4}\sin\theta_{2,4}\big(-\sin\omega_{1,4}\big) \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Suppose now that the assertion is valid for $n = i - 1$. Rewritting Q_i, we get

$$\begin{aligned} Q_i &= (B_4 \cdots B_{i-1})B_i \\ &= Q_{i-1}B_i, \end{aligned}$$

where the elements of Q_{i-1} are denoted by

$$Q_{i-1} = \begin{bmatrix} q_{11}^{i-1} & q_{12}^{i-1} & q_{13}^{i-1} & q_{14}^{i-1} \\ q_{21}^{i-1} & q_{22}^{i-1} & q_{23}^{i-1} & q_{24}^{i-1} \\ q_{31}^{i-1} & q_{32}^{i-1} & q_{33}^{i-1} & q_{34}^{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$B_i = \begin{bmatrix} -\cos\theta_{i-2,i} & -\sin\theta_{i-2,i} & 0 & -d_{i-1,i}\cos\theta_{i-2,i} \\ \sin\theta_{i-2,i}\cos\omega_{i-3,i} & -\cos\theta_{i-2,i}\cos\omega_{i-3,i} & -\sin\omega_{i-3,i} & d_{i-1,i}\sin\theta_{i-2,i}\cos\omega_{i-3,i} \\ \sin\theta_{i-2,i}\sin\omega_{i-3,i} & -\cos\theta_{i-2,i}\sin\omega_{i-3,i} & \cos\omega_{i-3,i} & d_{i-1,i}\sin\theta_{i-2,i}\sin\omega_{i-3,i} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Considering the product $Q_{i-1}B_i$, we obtain

$$Q_{i-1}B_i = \begin{bmatrix} V & X & Y & Z \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where

$$\begin{aligned} V &= \begin{bmatrix} q_{11}^{i-1}(-b) + q_{12}^{i-1}(cd) + q_{13}^{i-1}(ce) \\ q_{21}^{i-1}(-b) + q_{22}^{i-1}(cd) + q_{23}^{i-1}(ce) \\ q_{31}^{i-1}(-b) + q_{32}^{i-1}(cd) + q_{33}^{i-1}(ce) \end{bmatrix}, \\ X &= \begin{bmatrix} q_{11}^{i-1}(-c) + q_{12}^{i-1}(-bd) + q_{13}^{i-1}(-be) \\ q_{21}^{i-1}(-c) + q_{22}^{i-1}(-bd) + q_{23}^{i-1}(-be) \\ q_{31}^{i-1}(-c) + q_{32}^{i-1}(-bd) + q_{33}^{i-1}(-be) \end{bmatrix}, \\ Y &= \begin{bmatrix} q_{12}^{i-1}(-e) + q_{13}^{i-1}(d) \\ q_{22}^{i-1}(-e) + q_{23}^{i-1}(d) \\ q_{32}^{i-1}(-e) + q_{33}^{i-1}(d) \end{bmatrix}, \\ Z &= \begin{bmatrix} q_{11}^{i-1}(-ab) + q_{12}^{i-1}(acd) + q_{13}^{i-1}(ace) + q_{14}^{i-1} \\ q_{21}^{i-1}(-ab) + q_{22}^{i-1}(acd) + q_{23}^{i-1}(ace) + q_{24}^{i-1} \\ q_{31}^{i-1}(-ab) + q_{32}^{i-1}(acd) + q_{33}^{i-1}(ace) + q_{34}^{i-1} \end{bmatrix}, \end{aligned}$$

and $a = d_{i-1,i}$, $b = \cos\theta_{i-2,i}$, $c = \sin\theta_{i-2,i}$, $d = \cos\omega_{i-3,i}$, and $e = \sin\omega_{i-3,i}$.

By induction hypothesis, we have

$$Q'_{i-1} = \begin{bmatrix} q_{11}^{i-1} & q_{12}^{i-1} & -q_{13}^{i-1} & q_{14}^{i-1} \\ q_{21}^{i-1} & q_{22}^{i-1} & -q_{23}^{i-1} & q_{24}^{i-1} \\ -q_{31}^{i-1} & -q_{32}^{i-1} & q_{33}^{i-1} & -q_{34}^{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Considering the product $Q'_{i-1}B'_i$, where

$$B'_i = \begin{bmatrix} -\cos\theta_{i-2,i} & -\sin\theta_{i-2,i} & 0 & -d_{i-1,i}\cos\theta_{i-2,i} \\ \sin\theta_{i-2,i}\cos\omega_{i-3,i} & -\cos\theta_{i-2,i}\cos\omega_{i-3,i} & -\big(-\sin\omega_{i-3,i}\big) & d_{i-1,i}\sin\theta_{i-2,i}\cos\omega_{i-3,i} \\ \sin\theta_{i-2,i}(-\sin\omega_{i-3,i}) & -\cos\theta_{i-2,i}(-\sin\omega_{i-3,i}) & \cos\omega_{i-3,i} & d_{i-1,i}\sin\theta_{i-2,i}(-\sin\omega_{i-3,i}) \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

we obtain

$$Q'_{i-1}B'_i = \begin{bmatrix} V' & X' & Y' & Z' \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where

$$\begin{aligned} V' &= \begin{bmatrix} q_{11}^{i-1}(-b) + q_{12}^{i-1}(cd) - q_{13}^{i-1}(c(-e)) \\ q_{21}^{i-1}(-b) + q_{22}^{i-1}(cd) - q_{23}^{i-1}(c(-e)) \\ -q_{31}^{i-1}(-b) - q_{32}^{i-1}(cd) + q_{33}^{i-1}(c(-e)) \end{bmatrix}, \\ X' &= \begin{bmatrix} q_{11}^{i-1}(-c) + q_{12}^{i-1}(-bd) - q_{13}^{i-1}(-b(-e)) \\ q_{21}^{i-1}(-c) + q_{22}^{i-1}(-bd) - q_{23}^{i-1}(-b(-e)) \\ -q_{31}^{i-1}(-c) - q_{32}^{i-1}(-bd) + q_{33}^{i-1}(-b(-e)) \end{bmatrix}, \\ Y' &= \begin{bmatrix} q_{12}^{i-1}(e) - q_{13}^{i-1}(d) \\ q_{22}^{i-1}(e) - q_{23}^{i-1}(d) \\ -q_{32}^{i-1}(e) + q_{33}^{i-1}(d) \end{bmatrix}, \\ Z' &= \begin{bmatrix} q_{11}^{i-1}(-ab) + q_{12}^{i-1}(acd) - q_{13}^{i-1}(ac(-e)) + q_{14}^{i-1} \\ q_{21}^{i-1}(-ab) + q_{22}^{i-1}(acd) - q_{23}^{i-1}(ac(-e)) + q_{24}^{i-1} \\ -q_{31}^{i-1}(-ab) - q_{32}^{i-1}(acd) + q_{33}^{i-1}(ac(-e)) - q_{34}^{i-1} \end{bmatrix}. \end{aligned}$$

Representing the matrix Q_i by

$$Q_i = Q_{i-1}B_i = \begin{bmatrix} q_{11}^{i} & q_{12}^{i} & q_{13}^{i} & q_{14}^{i} \\ q_{21}^{i} & q_{22}^{i} & q_{23}^{i} & q_{24}^{i} \\ q_{31}^{i} & q_{32}^{i} & q_{33}^{i} & q_{34}^{i} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and comparing the matrices $Q_{i-1}B_i$ and $Q'_{i-1}B'_i$ given above, we conclude that

$$Q'_i = Q'_{i-1}B'_i = \begin{bmatrix} q_{11}^{i} & q_{12}^{i} & -q_{13}^{i} & q_{14}^{i} \\ q_{21}^{i} & q_{22}^{i} & -q_{23}^{i} & q_{24}^{i} \\ -q_{31}^{i} & -q_{32}^{i} & q_{33}^{i} & -q_{34}^{i} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Proof of Lemma 3

For $n = 1, 2, 3$ the assertion is clearly true. By the previous lemma, we have

$$\begin{bmatrix} x_{i_1} \\ x_{i_2} \\ x_{i_3} \\ 1 \end{bmatrix} = B_1 B_2 B_3 Q_i \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = B_1 B_2 B_3 \begin{bmatrix} q^i_{14} \\ q^i_{24} \\ q^i_{34} \\ 1 \end{bmatrix}$$

and

$$\begin{bmatrix} x'_{i_1} \\ x'_{i_2} \\ x'_{i_3} \\ 1 \end{bmatrix} = B_1 B_2 B_3 Q'_i \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = B_1 B_2 B_3 \begin{bmatrix} q^i_{14} \\ q^i_{24} \\ -q^i_{34} \\ 1 \end{bmatrix},$$

for $i = 4, ..., n$, and calculating the product $B_1 B_2 B_3$, we obtain

$$B_1 B_2 B_3 = \begin{bmatrix} \cos\theta_{1,3} & \sin\theta_{1,3} & 0 & -d_{1,2} + d_{2,3}\cos\theta_{1,3} \\ \sin\theta_{1,3} & -\cos\theta_{1,3} & 0 & d_{2,3}\sin\theta_{1,3} \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Thus,

$$\begin{bmatrix} x_{i_1} \\ x_{i_2} \\ x_{i_3} \\ 1 \end{bmatrix} = B_1 B_2 B_3 \begin{bmatrix} q^i_{14} \\ q^i_{24} \\ q^i_{34} \\ 1 \end{bmatrix} = \begin{bmatrix} -d_{1,2} + d_{2,3}\cos\theta_{1,3} + q^i_{14}\cos\theta_{1,3} + q^i_{24}\sin\theta_{1,3} \\ d_{2,3}\sin\theta_{1,3} + q^i_{14}\sin\theta_{1,3} - q^i_{24}\cos\theta_{1,3} \\ -q^i_{34} \\ 1 \end{bmatrix}$$

and

$$\begin{bmatrix} x'_{i_1} \\ x'_{i_2} \\ x'_{i_3} \\ 1 \end{bmatrix} = B_1 B_2 B_3 \begin{bmatrix} q^i_{14} \\ q^i_{24} \\ -q^i_{34} \\ 1 \end{bmatrix} = \begin{bmatrix} -d_{1,2} + d_{2,3}\cos\theta_{1,3} + q^i_{14}\cos\theta_{1,3} + q^i_{24}\sin\theta_{1,3} \\ d_{2,3}\sin\theta_{1,3} + q^i_{14}\sin\theta_{1,3} - q^i_{24}\cos\theta_{1,3} \\ q^i_{34} \\ 1 \end{bmatrix},$$

for $i = 4, ..., n$. That is,

$$\begin{bmatrix} x'_{i_1} \\ x'_{i_2} \\ x'_{i_3} \end{bmatrix} = \begin{bmatrix} x_{i_1} \\ x_{i_2} \\ -x_{i_3} \end{bmatrix},$$

for $i = 1, ..., n$.