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

The Discretizable Distance Geometry Problem

A. Mucherino · C. Lavor · L. Liberti

Received: date / Accepted: date

Abstract We introduce the Discretizable Distance Geometry Problem in $\mathbb{R}^3$ (DDGP_3), which consists in a subclass of instances of the Distance Geometry Problem for which an embedding in $\mathbb{R}^3$ can be found by means of a discrete search. We show that the DDGP_3 is a generalization of the Discretizable Molecular Distance Geometry Problem (DMDGP), and we discuss the main differences between the two problems. We prove that the DDGP_3 is NP-hard and we extend the Branch & Prune (BP) algorithm, previously used for the DMDGP, for solving instances of the DDGP_3. Protein graphs may or may not be in DMDGP and/or DDGP_3 depending on vertex orders and edge density. We show experimentally that as distance thresholds decrease, PDB protein graphs which fail to be in the DMDGP still belong to DDGP_3, which means that they can still be solved using a discrete search.

Keywords distance geometry · DDGP_3 · DMDGP · combinatorial reformulations · branch and prune

1 Introduction

The Distance Geometry Problem (DGP) consists in finding the coordinates of a given set of points $\{x_1, x_2, \ldots, x_n\}$ in a three-dimensional space when some of the distances between pairs of such points are known [7]. Let $G = (V, E, d)$ be a weighted undirected graph, where each vertex in V corresponds to an x_i, and there is an edge between two vertices if and only if their relative distance is known (the weight associated to the edge). The graph G represents an instance of the DGP. We give the following formal definition of the DGP.

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

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

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

Definition 1 Let $G = (V, E, d)$ be a weighted undirected graph. The DGP is the problem of finding a function

$$x : V \longrightarrow \mathbb{R}^3$$

such that

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

where $x_u = x(u)$ and $x_v = x(v)$.

In its basic form, the DGP is a constraint satisfaction problem, because a set of coordinates x_v must be found that satisfies the constraints (1). In the definition, the symbol $|| \cdot ||$ represents the computed distance between x_u and x_v, whereas d_{uv} refers to the known distances. Once a solution function x has been identified, the final conformation X can be obtained by

$$X = \{x_v : v \in V\}.$$

Different approaches for solving the DGP have been proposed in the literature, and surveys can be found in [10,17]. The most common approach is the one in which the DGP is formulated as a continuous global optimization problem. The set of constraints (1) is replaced by a penalty function that measures how much computed and known distances differ. An example of penalty function which is often used is the Largest Distance Error (LDE):

$$LDE(\{x_1, x_2, \ldots, x_n\}) = \frac{1}{m} \sum_{\{u,v\}} \frac{| \, ||x_u - x_v|| - d_{uv} \, |}{d_{uv}}, \tag{2}$$

where m is the number of known distances. Solutions to the DGP can be found by minimizing this function. This is not trivial, because the function (2) (and even other proposed penalty functions) is not convex and contains many local minima. In fact, a well-known approach to the DGP is based on a method in which the penalty function is approximated by a sequence of smoother functions converging to the original one [18,19]. Note that, if the subset of known distances is feasible, then a set X is solution to the DGP if and only if the value of the penalty function in X is exactly 0.

The DGP has interesting applications. The problem of localizing sensors in wireless networks is an example of DGP [6,27]. Distances between sensors can be estimated by measuring the power used for a two-way communication, and the aim is to identify the positions of all the sensors. The main difficulty stands in the fact that not all the possible distances between sensors are known, because sensors that are too far from each other cannot communicate. However, sensor networks always include a wired backbone of sensors whose positions are known a priori. Such sensors are called *anchors*, and their positions can be exploited for solving the localization problem.

The DGP has also applications in the field of biology [5]. A molecule can be represented in a three-dimensional space by a set X, where each point x_i represents one of its atoms. There are experimental techniques, such as the Nuclear Magnetic Resonance (NMR), that are able to estimate distances between some pairs of atoms. Such distances can then be exploited for finding the coordinates of the atoms of the molecule by solving the corresponding DGP. Differently from the wireless sensor localization problem, there are no anchors, and the set of known distances is usually limited to distances smaller than 6Å. When X represents a molecule, the DGP is usually referred to as the Molecular Distance Geometry Problem (MDGP) [10,17].

The Discretizable Molecular Distance Geometry Problem (DMDGP) has been proposed in [9,11]. It consists of a subclass of instances for the MDGP whose 3D embeddings can be computed by a discrete search algorithm (we also say that these instances have the *combinatorial property*). The conception of this problem was inspired by the structure of particular molecules: the *proteins*. Proteins are formed by smaller molecules called amino acids, that bind to each other by forming one or more chains. As a consequence, a particular sequence of atoms can be identified in proteins, where each atom is bound to the preceding and to the following ones. This sequence of atoms is referred to as *backbone* of the protein. Since NMR experiments are able to detect short-range distances, distances between atoms of the protein backbones that are consecutive or separated by few atoms can be found by NMR. This information has been exploited for defining instances satisfying the combinatorial property [11,16, 21].

In order for MDGP instances to have the combinatorial property, two assumptions need to be satisfied (see Section 2 for more details). In particular, it is required that, for each vertex v, the distances between v and its three preceding vertices $v-1$, $v-2$ and $v-3$ must be known. In this paper, we weaken this requirement, and we introduce a new combinatorial property. Since we consider a weaker assumption, a larger number of instances of the DGP satisfy the new combinatorial property. We show experimentally that protein graphs from the PDB [2] may or may not have the combinatorial property according to the distance threshold allowed for defining edges, and that our weakened assumption allows instances to have the combinatorial property with lower distance thresholds. We also discuss the importance of the vertex orders for the protein graphs, and briefly describe an algorithm for identifying orders for which the combinatorial property is satisfied.

The rest of the paper is organized as follows. In Section 2, we will give a brief outline of the Discretizable Molecular Distance Geometry Problem (DMDGP) [9,11]. In Section 3, we will introduce the new combinatorial property for the DGP and the corresponding combinatorial optimization problem, to which we refer as the Discretizable Distance Geometry Problem (DDGP). Since the DDGP can be, in theory, extended to any dimension, and since we will limit the discussion in this paper to the case $n = 3$ only, we will refer to this problem as the DDGP$_3$. Properties of this new problem will be analyzed and discussed. In Section 4, possible strategies for solving the DDGP$_3$ will be discussed. An exact algorithm will be presented and some computational experiments will be shown in Section 5. Conclusions will be given in Section 6.

2 The Discretizable Molecular Distance Geometry Problem (DMDGP)

Proteins are molecules having particular geometric properties. They are formed by smaller molecules called *amino acids*, that are bound together forming a sort of chain. Along this chain, atoms that are common to all the amino acids form the so-called *protein backbone*, and atoms of the protein backbone which are close in sequence are also close in the three-dimensional conformation of the protein. Therefore, an instance of the MDGP related to protein backbones is such that atoms corresponding to close vertices are also close in distance. As a consequence, the relative distances between atoms represented by close vertices are known, because experimental techniques, such as NMR, are able to detect short range distances. This intuition brought to the definition of the Discretizable Molecular Distance Geometry Problem (DMDGP).

Definition 2 Let $G = (V, E, d)$ be a weighted undirected graph associated to an instance of the DGP. Let us suppose that there is a *total order relation* on the vertices of V. The DMDGP consists in all the instances of the DGP satisfying the following two assumptions:

A1 E contains all cliques on quadruplets of consecutive vertices;
A2 the following strict triangular inequality must hold:

$$\forall v \in \{1, \ldots, n-2\} \qquad d_{v,v+2} < d_{v,v+1} + d_{v+1,v+2},$$

where n is the number of vertices in V.

Assumption **A2** is satisfied in most of the cases. If, for a certain triplet of consecutive vertices, $d_{v,v+2}$ were perfectly equal to $d_{v,v+1} + d_{v+1,v+2}$, then the corresponding three atoms would be perfectly aligned. The Lebesgue measure of the subset not satisfying Assumption **A2** is zero, and so the probability of Assumption **A2** not being satisfied is zero in a purely technical sense. Assumption **A1** may be instead harder to be satisfied. When protein conformations are considered, there are many cases in which it is satisfied because of the particular structure of these molecules. In general, if some of the distances in quadruplets of consecutive atoms are not known, then the quadruplet cannot be a clique.

There are equivalent formulations of the DMDGP. The following theoretical result will be exploited in Section 3 when comparing the DMDGP to the DDGP$_3$.

Proposition 1 *Let $G = (V, E, d)$ be a weighted undirected graph associated to an instance of the DGP. Given a predefined ordering on V, assumption* **A1** *is equivalent to the following two assumptions:*

A3 $V_1 = \{1, 2, 3, 4\} \subset V$ *is a clique;*
A4 $\forall v \leq |V| - 3 \qquad \{(v, v+3), (v+1, v+3), (v+2, v+3)\} \subseteq E.$

Proof Let us start by proving that, if **A1** is satisfied, then **A3** and **A4** are also satisfied. The proof is trivial, because, if all the quadruplets of consecutive vertices are cliques, then V_1 is in particular a clique, and all the edges $(v, v+3)$, $(v+1, v+3)$ and $(v+2, v+3)$ must be in E, for all $v \leq |V| - 3$.

Let us consider now the two following quadruplets of consecutive vertices for some $v \in \{2, \ldots, |V| - 3\}$:

$$V_{v-1} = \{v - 1, v, v + 1, v + 2\}$$

and

$$V_v = \{v, v + 1, v + 2, v + 3\}.$$

Let us suppose that V_{v-1} is a clique. By this hypothesis on V_{v-1}, it follows that the distances between all the possible pairs of vertices in $\{v, v + 1, v + 2\}$ are known. Moreover, all the distances between the vertices in $\{v, v+1, v+2\}$ and $v+3$ are known because of **A4**, and, as a consequence, V_v is also a clique. Thus, by induction, we conclude that all the quadruplets of consecutive vertices in V are cliques. This proves that **A3** and **A4** imply **A1**. □

Note that the assumptions of the DMDGP (**A1** and **A2** or, equivalently, **A3**, **A4** and **A2**) strongly depend on the ordering of the vertices in V. Consider, as an example, an instance containing 5 vertices v_i, $i \in \{1, \ldots, 5\}$, such that $\{v_1, v_2, v_3, v_4\}$

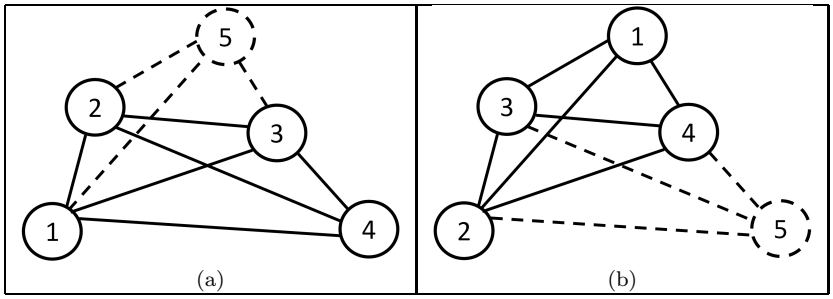

Fig. 1 An example of instance that satisfies the assumptions for the DMDGP if the original order of its vertices is modified: (a) the vertex 2 does not satisfy **A4**, because the edge $(v+2, v+3) = (4, 5)$ is not in E; (b) **A3**, **A4** and **A2** are all satisfied.

is a clique, and moreover $\{(v_1, v_5), (v_2, v_5), (v_3, v_5)\} \in E$ (see Figure 1). Assumption **A3** is satisfied, but **A4** is not satisfied, because one of the needed edges is absent. However, the ordering of the vertices can be changed in $\{v_5, v_1, v_2, v_3, v_4\}$. In this case, Assumption **A3** is still satisfied and even Assumption **A4** is satisfied, because v_5 (or v_4 in the previous order) is adjacent to the previous three vertices. Thus, given an instance of the DGP which is not an instance of the DMDGP, there might be an order (or more than one) that could allow the assumptions of the DMDGP to be satisfied.

When the assumptions of the DMDGP are satisfied, then, if the vertices are placed into a position by following the same order given to the vertices of V, only two possible positions can be chosen for the generic x_k (see Section 4 for more details). This combinatorial property leads to the definition of a binary tree of possible positions, where solutions to the DMDGP can be searched. As a consequence, the DMDGP can be seen as a combinatorial optimization problem [9,11]. As the DGP [26], the DMDGP is NP-hard [11].

3 The Discretizable Distance Geometry Problem in $\mathbb{R}^3$ ($\mathbf{DDGP_3}$)

We introduce in this section the Discretizable Distance Geometry Problem in $\mathbb{R}^3$ (DDGP_3). This is a combinatorial problem based on assumptions that are weaker with respect to the ones of the DMDGP. We give the following formal definition of the problem.

Definition 3 Let $G = (V, E, d)$ be a weighted undirected graph associated to an instance of the DGP. Let us suppose that there is a *partial order relation* on the vertices of V. The DDGP_3 consists in all the instances of the DGP satisfying the following two assumptions:

B1 there exists a subset V_1 of V such that
- $|V_1| = 4$;
- the order relation on V_1 is total;
- V_1 is a clique;
- $\forall v_0 \in V_1 \quad \forall v \in V \setminus V_1, \quad v_0 < v$.

B2 $\forall v \in V \setminus V_1$, $\exists\, u^1, u^2, u^3 \in V$ such that:
- $u^1 < v$, $u^2 < v$, $u^3 < v$;

– $\{(u^1, v), (u^2, v), (u^3, v)\} \in E$;
– $d_{u^1,u^3} < d_{u^1,u^2} + d_{u^2,u^3}$.

From the definition of the DDGP$_3$, only a partial order relation is required on the vertices of G. However, note that every partial order can be extended to a total order. Because of Assumption **B2**, for each vertex v, there must be at least 3 vertices u^1, u^2 and u^3 which precede v and such that the distances $d(u^1, v)$, $d(u^2, v)$, and $d(u^3, v)$ are known. This requirement is weaker than the analogous requirement in the assumptions of the DMDGP (indeed, in the DMDGP, it is also required that the four vertices u^1, u^2, u^3 and v are consecutive). This requirement can be satisfied in real applications. For the sensor network localization problem, for example, sensors should interact with at least other two sensors (we are in the two-dimensional space in this case). When this is not the case, however, sensors having only one neighboring sensor could be initially removed from the network, and the localization problem may be solved for a subnetwork. Then, the other sensors may be appended in a second phase in any position compatible with their single distance. As concerns proteins, the hypothesis for which each atom has at least 3 neighboring atoms is very realistic. Protein molecules are compact objects, and each atom should have several atoms in its surroundings which can be detected by NMR. Finally, note that the strict triangular inequality in Assumption **B2** is always satisfied in practice, because, as already remarked, the Lebesgue measure of the subset not satisfying it is zero.

The following results help understanding the main differences between the DMDGP and the DDGP$_3$.

Theorem 1 *Any instance of the DMDGP is also an instance of the DDGP$_3$.*

Proof We need to prove that, if an instance of the DGP satisfies **A1** and **A2** (or equivalently **A3**, **A4** and **A2**, see Proposition 1), then this instance also satisfies **B1** and **B2**.

A generic instance of the DMDGP is such that a total order relation is defined for the vertices of G. Therefore, the hypothesis for the DDGP$_3$ that there is at least a partial order is satisfied.

Let V_1 be equal to $\{1, 2, 3, 4\}$. It is easy to see that V_1 satisfies **B1**. Indeed, the cardinality of V_1 is 4, there is a total order relation for the vertices of V_1, V_1 is a clique (because of **A3**), and all the vertices in V_1 precede in rank all the others in V.

Let v be the generic vertex in $V \setminus V_1$ and $V_{v-3} = \{v-3, v-2, v-1, v\}$. Because of **A4**, the distances between v and all the other vertices in V_{v-3} are known, and the vertices in $\{v-3, v-2, v-1\}$ satisfy the strict inequality because of **A2**. Therefore, if we define $u^1 = v-1$, $u^2 = v-2$ and $u^3 = v-3$, then we have three vertices u^1, u^2 and u^3 that satisfy **B2**. Indeed, the vertices u^1, u^2 and u^3 precede v in order, the edges (u^1, v), (u^2, v) and (u^3, v) are in E because these relative distances are known, and the strict triangular inequality holds. □

Notice that the inverse of Theorem 1 is not true in general. Indeed, we can prove the following:

Proposition 2 *There exist instances of the DDGP$_3$ that are not instances of the DMDGP, for any possible ordering given to the vertices.*

Proof Let us consider an instance with 6 vertices v_i, $i \in \{1, \ldots, 6\}$, satisfying the following properties (see Figure 2):

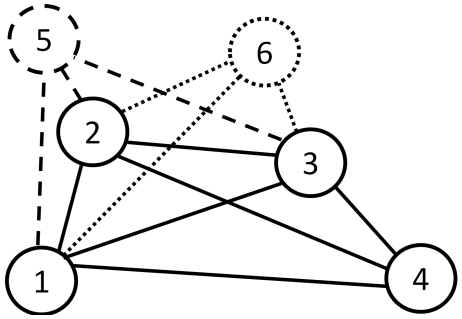

Fig. 2 An example of instance that does not satisfy the assumptions for the DMDGP, for any possible ordering given to the vertices.

- $\{v_1, v_2, v_3, v_4\}$ is a clique;
- v_5 is adjacent to v_1, v_2 and v_3;
- v_6 is adjacent to v_1, v_2 and v_3;
- the strict triangular inequality holds for all the possible triplets of vertices.

The assumptions for the DDGP$_3$ are satisfied. Indeed, **B1** is trivially satisfied. Moreover, **B2** is satisfied because, for both v_5 and v_6, we can set $u^1 = v_1$, $u^2 = v_2$ and $u^3 = v_3$.

On the other hand, the assumptions for the DMDGP can never be satisfied, for any ordering given to the vertices. At first, let us consider the original ordering. **A3** is satisfied because V_1 is a clique. Then, if the consider v_5, we can see that the three preceding vertices v_2, v_3 and v_4 are not all adjacent to v_5. We can observe the same for the vertex v_6. Therefore, the instance, with this ordering for the vertices, is not an instance of the DMDGP.

Let us try now to modify the ordering of the vertices with the aim of finding a particular one for which the assumptions for the DMDGP are satisfied. Let us divide the vertices of the instance in two parts: $I_1 = \{v_1, v_2, v_3\}$ and $I_2 = \{v_4, v_5, v_6\}$. Note that, from the properties of this instance, it follows that the vertices in I_1 are adjacent to all the others in the instance, whereas the vertices in I_2 are adjacent to v_1, v_2 and v_3 only. If we permute the vertices of I_1 or the vertices of I_2 without exchanging vertices between the two parts, then the assumptions will never be satisfied, because all the vertices in I_2 are not adjacent to the other vertices of I_2 (and this is needed, because, for example, the last vertex in the ordering should be adjacent to the previous two).

Finally, let us consider permutations where vertices in I_1 and I_2 are exchanged. In the original order, V_1 contains three vertices from I_1 and one vertex from I_2. Let us consider an order in which V_1 contains two vertices from I_1 and two vertices from I_2. In such a case, two vertices (the ones belonging to I_2) are not adjacent to each other, and therefore V_1 cannot be a clique. Then, in all the other possible permutations, V_1 contains two vertices from I_2 *at least*. It follows that there are no possible orderings for which the assumptions for the DMDGP are satisfied. □

The DDGP$_3$ is therefore a generalization of the DMDGP. The assumptions for the DDGP$_3$ can be satisfied, in general, independently from the fact that the instances are related to proteins, generic molecules or sensor networks. This allows to discretize a wider range of DGP problems arising in real-life applications. Both the DMDGP and

the DDGP$_3$ are combinatorial optimization problems. They allow to focus the search for solutions to DGPs on a discrete domain. As already mentioned, the DGP and the DMDGP are both NP-hard [11,26]. In both the cases, the NP-hardness of the two problems has been proved by reduction from the SubSet-Sum problem (in dimension 1 for the DGP, and in dimension 3 for the DMDGP).

Corollary 1 *The DDGP$_3$ is NP-hard.*

Proof By inclusion: instances of the DMDGP form a subclass of instances of the DDGP$_3$, and the DMDGP is NP-hard. □

4 Solving instances of the DMDGP and the DDGP$_3$

4.1 Building the binary tree of solutions

Both the assumptions of the DMDGP and of the DDGP$_3$ allow to discretize a general DGP. Let us suppose that the positions for the vertices $i \in \{1, \ldots, k-1\}$ of a solution to the problem are already placed in a fixed location, and that a position for the vertex k is searched. By the assumptions, there exist three vertices u^1, u^2 and u^3 such that the distances between k and u^1, u^2, u^3 are known. In the case of the DMDGP, the three vertices u^1, u^2, u^3 are the ones that precede k. In the case of the DDGP$_3$, u^1, u^2 or u^3 can be any vertex with a rank smaller than k. In both the cases, the distances between k and three other vertices (whose positions are known) can be used for computing the possible positions for k.

Let us consider three spheres, centered in x_{u^1}, x_{u^2} and x_{u^3}, and with radius $d(x_k, x_{u^1})$, $d(x_k, x_{u^2})$ and $d(x_k, x_{u^3})$, respectively. The intersection of these three spheres provides a set of positions that are feasible for the x_k (i.e. positions that respect the three distances from u^1, u^2 and u^3). Intersections among three spheres can be a circle, two points or one point only. The circle is obtained in the hypothesis that the three vertices u^1, u^2, u^3 are aligned, which is impossible because it is supposed that the strict triangular inequality (see **A2** for MDGP and **B2** for DDGP$_3$) must hold. Therefore, in all the cases, there are at most two positions for each generic k.

All possible positions for the vertices of a conformation X are used for defining a binary tree of solutions for the DMDGP and the DDGP$_3$. Since the intersection of the three spheres is rarely one point only (especially on the floating-point arithmetic of a computer machine), we suppose that, for each k, there are two possible positions. As a consequence, the binary tree contains 2^n positions for a conformation related to n vertices. However, in order to avoid considering equivalent solutions that can be obtained from a given solution by translations or rotations, the first three points can be fixed, so that the final binary tree has 2^{n-3} positions. Solutions to the DMDGP and to the DDGP$_3$ can be found by exploring this tree. The only difference between the two approaches is given by the distances and vertices used in the definition of each position in the tree.

The Branch & Prune (BP) algorithm [11,16] can be used for an efficient exploration of this binary tree. The binary tree is not constructed a priori, but it is rather built as the search proceeds. At each step of the algorithm, two new positions are computed for the current vertex k. They are added to the tree only if they pass some tests for feasibility. Indeed, the two positions are computed in a way that they satisfy the known distances between k and the three vertices u^1, u^2, u^3. However, there could be other

Algorithm 1 BP algorithm

```
BP(k, n, d)
for (i = 1, 2) do
  compute the i^th position for the vertex k: x_k^(i);
  check the feasibility of the position x_k^(i):
  if (the position x_k^(i) is feasible) then
    if (k = n) then
      one solution is found;
    else
      BP(k + 1,n,d);
    end if
  else
    the current branch is pruned;
  end if
end for
```

known distances that can be used for checking the feasibility of the found positions. The most simple and natural pruning test is the one in which the known distances and the distances obtained from the computed positions for the vertex k are compared. If they coincide (for a given tolerance), then the position being checked is feasible, otherwise it is not. In the latter case, the position is not added to the tree at all, and all the positions along the same branch of the tree are not considered, because they cannot be part of a feasible solution. This pruning phase in the BP algorithm allows to reduce the binary tree very quickly, so that an exhaustive search on the remaining branches is not too expensive. Algorithm 1 is a sketch of the BP algorithm.

In previous publications [12–15,20–22], we showed how the BP algorithm can efficiently solve instances of the DMDGP related to protein conformations. The software we developed, `MD-jeep` [25], which is an implementation in C of the BP algorithm, can be freely downloaded from the Internet. We compared `MD-jeep` to other two publicly available software tools for distance geometry, and showed that the BP algorithm is able to provide more accurate solutions in a shorter amount of time [11]. Apart from the procedure used for building the binary tree, the BP algorithm can be applied almost unchanged to the DDGP_3. As a consequence, all the results we obtained so far for the DMDGP can be considered as applicable to the DDGP_3 as well.

4.2 Generation of candidate atomic positions

The subproblem that needs to be solved at each iteration of the BP algorithm is the one of finding the intersection of three spheres. This subproblem needs to be solved every time the two positions for a given vertex k must be computed, and it is equivalent to the problem of finding the two solutions of the following system of quadratic equations:

$$\begin{cases} ||x_k - x_{u^1}|| = d(x_k, x_{u^1}) \\ ||x_k - x_{u^2}|| = d(x_k, x_{u^2}) \\ ||x_k - x_{u^3}|| = d(x_k, x_{u^3}). \end{cases} \tag{3}$$

Methods for finding solutions to the system (3) can be found, for example, in [4]. Note that, whatever method is used, it is very important that the found solutions are very accurate. Indeed, they represent the possible positions for the conformation, which have

to pass some tests for feasibility before being inserted in the binary tree. Therefore, if the found solutions for (3) are not accurate enough, then the pruning tests might reject them all, and no solutions are found.

In the case of DMDGP, the problem of intersecting the three spheres can be replaced by the problem of finding the possible torsion angles along a backbone of atoms of a molecule related to the total order relation of the DMDGP. Under the assumptions of the DMDGP, it can be proved that there are two possible torsion angles only for each quadruplet of consecutive atoms. Two torsion angles correspond to two possible positions for the last atom of the quadruplet. Properties of the DMDGP have been proved in [11] by exploiting properties of these torsion angles. We also remark that the employment of the torsion angles allows for a sufficiently low propagation of round-off errors during the execution of the BP algorithm, whereas the resolutions of the quadratic systems (3) may lead to instability issues.

While, in the case of the DMDGP, the choice of considering torsion angles is evident, this is not possible anymore when considering generic DDGP$_3$ instances. A sequence of quadratic systems need instead to be computed and, at each iteration of BP, we must be aware that some errors may be introduced in the computed coordinates. In order to keep the propagation of these errors as low as possible, we implement the following strategy. At each iteration of BP, the two possible positions for the current vertex x_k need to be computed. By Assumption **B2**, there are at least 3 vertices u^1, u^2 and u^3 which can be used for defining the quadratic system (3). If other vertices u^4, u^5, ..., u^l are also available, they can be used for checking the feasibility of the two computed positions. However, since the consecutivity property of the vertices u^1, u^2, u^3 and k is lost in the DDGP$_3$, we can also choose to use, for example, the vertices u^{l-2}, u^{l-1} and u^l for defining the quadratic system and to use the others in the pruning test. In our strategy for keeping the propagation of errors low, we try all the possible triples of vertices in $\{u^1, u^2, \ldots, u^l\}$ and we choose the triplet corresponding to the quadratic system with the most accurate solutions. The accuracy of the solutions can be evaluated by the pruning tests, by measuring the difference ε between $||x_i - x_j||$ and d_{ij}, for all the available distances d_{ij}.

We implemented two versions of the BP algorithm. The first one solves DMDGPs and the binary tree is built by computing the cosines of the torsion angles. The second one solves instead DDGP$_3$ instances, where the binary tree is built by solving the quadratic systems and the above strategy for the round-off errors is implemented.

In [3,29] there are similar approaches for solving the quadratic system (3), but we remark that the formal definition of the DDGP$_3$ introduces an ordering on V as an essential part of the input data, marking a fundamental difference between the present work and the ones presented in [3,29]. In fact, these papers propose a generalization of the geometric build-up algorithm [28], which computes the Cartesian coordinates for the current vertex k only if one can find at least four vertices with known Cartesian coordinates and known distances to k. However, depending on the instance, the geometric build-up algorithm may fail to solve the DDGP$_3$. In addition to this, the given order may produce numerical instabilities in the geometric build-up algorithm (for more details, see [28]). We also remark that the first work providing an iterative discrete search algorithm for the MDGP that only requires three (rather than four) previously embedded adjacent vertices is [9], accepted for publication in [11].

Algorithm 2 A reordering algorithm

```
reorder(G)
while (a valid ordering is not found) do
  find a 3-clique C in G;
  place the vertices of C at the beginning of new order: B = C;
  while (V \ B ≠ ∅) do
    find the vertex v in V \ B with the largest number l of adjacent vertices in B;
    if (l < 3) then
      break the while loop: there are no possible orderings for this choice of C;
    end if
    B = B + {v};
  end while
end while
```

4.3 Finding discretizable vertex orders

As discussed in Section 3, instances can satisfy the assumptions of the DDGP$_3$ if a suitable reordering for its atoms is found. The problem of finding vertex orders for a given graph G for which the assumptions are satisfied has been widely discussed in [8]. In this paper, we only point out that, given an instance that does not belong to the class of the DDGP$_3$, its vertex reordering may generate an instance for which the necessary assumptions are instead satisfied.

In order to verify if a suitable reordering exists for a given instance, we employ Algorithm 2 [8]. The basic idea is to find a 3-clique C in G and to consider their vertices as first vertices of the new ordering. In this way, assumption **B1** is satisfied. Then, all other vertices are positioned in the new ordering by looking for the ones with the largest number of adjacent vertices. If this number of adjacent vertices is always greater or equal to 3 for a certain clique C, then an ordering satisfying assumption **B2** exists. Otherwise, if, for all possible cliques C, there is at least one vertex for which the number of adjacent vertices is smaller than 3, then an ordering satisfying assumption **B2** does not exist. More details about this algorithm are given in [8].

Some computational experiments are presented in the next section. We point out that the instances considered in the experiments are artificially generated because we are not able to deal yet with noisy data and experimental errors. However, preliminary studies [20,21] proved that our approach to the problem can be extended for considering real-life instances. Recent efforts in this direction have been detailed in [23,24].

5 Computational experiments

We present in this section some computational experiments related to protein conformations. All the codes were written in C programming language and all the experiments were carried out on an Intel Core 2 CPU 6400 @ 2.13 GHz with 4GB RAM, running Linux. The codes have been compiled by the GNU C compiler v.4.1.2 with the `-O3` flag.

Two versions of the BP algorithm are considered, one for solving instances of the DMDGP, and the other one for solving instances of the DDGP$_3$. The second one is based on the solution of the quadratic systems for finding the intersection among 3 spheres. In order to solve the quadratic system, we consider the strategy in [4], for which two linear systems need to be solved. Since, in the DDGP$_3$, the distances

		$\Delta = 8$				$\Delta = 7$			
instance		DMDGP		$DDGP_3$		DMDGP		$DDGP_3$	
name	n	#Sol	LDE	#Sol	LDE	#Sol	LDE	#Sol	LDE
1erp	107	2	3.86e-15	2	1.61e-13	-	-	2	*3.25e-13*
1aqr	113	2	3.87e-15	2	1.55e-11	-	-	2	1.34e-11
1k1v	121	2	1.40e-15	2	8.95e-13	-	-	2	7.10e-13
1brz	157	2	6.22e-14	2	1.27e-11	-	-	2	1.55e-11
1ccq	173	-	-	2	3.32e-11	-	-	2	*1.15e-11*
1bqx	222	2	9.81e-15	2	4.18e-12	-	-	2	1.08e-11
1b4c	542	-	-	2	7.59e-12	-	-	2	*4.20e-11*
1a23	546	2	1.00e-14	2	2.94e-11	-	-	2	1.98e-10
1la3	548	2	6.76e-15	2	7.56e-11	-	-	2	1.03e-10
1d8v	770	2	2.70e-14	2	5.06e-11	-	-	2	1.99e-11

Table 1 Some experiments with the two versions of the BP algorithm on a set of 10 protein graphs.

between pairs of vertices in $\{u^1, u^2, u^3\}$ can be large, the coordinates related to such vertices may have distinct orders of magnitude. This can cause the occurrence of badly-scaled matrices for the two systems to be solved. Therefore, in our implementation, we employ the function `dgesvx` of the LAPACK library [1], which automatically scales the coefficient matrices before solving the linear systems.

Distances between the atoms of a molecule can be found by experimental techniques such as NMR. These experiments are able to provide distances between pairs of atoms which are shorter than a certain threshold Δ. Since we are not able yet to consider real data from NMR, we artificially generated the instances considered in this paper. We downloaded a subset of protein conformations from the Protein Data Bank (PDB) [2], we computed all the possible distances between pairs of atoms of the molecule, and we kept only the distances smaller than Δ. This is the same technique used for the computational experiments presented in [28], and, as in the quoted paper, we used different values for the threshold Δ to analyze how it influences the necessary assumptions and the BP algorithm. These experiments have as aim to compare the two considered versions of the BP algorithm. In the case of the $DDGP_3$, we verify if the necessary assumptions are satisfied, and, if not, we apply Algorithm 2 for finding a vertex ordering allowing for the discretization. We only consider the hydrogen atoms of the protein backbones.

Table 1 shows some computational experiments for a subset of protein conformations. The name given to the instance corresponds to the label for the considered protein in the PDB. n is the number of hydrogens on the backbone of the protein. For different values of Δ, both the versions of the BP algorithm have been considered, the one related to the DMDGP and the one related to the $DDGP_3$. For each experiment, the number #Sol of found solutions and the best LDE function value are given. In some cases the assumptions for the DMDGP were not satisfied and the BP algorithm (DMDGP version) could not be applied. When the assumptions for the $DDGP_3$ were not satisfied, instead, we used Algorithm 2 for finding a suitable ordering for the atoms for the instance, so that we could apply the BP algorithm ($DDGP_3$ version) in all cases. This is specified in the table with the italic style for the LDE function value.

We can see that, when $\Delta = 8$, there are only 2 instances over 10 in which the assumptions for the DMDGP are not satisfied. The assumptions for the $DDGP_3$ are instead always satisfied. Both versions of the BP algorithm are able to find accurate

solutions in a short amount of time (all experiments do not last more than one second). We only remark that the best LDE function values are a little higher when the solutions are found by the $DDGP_3$ version of BP. This is due to the propagation of errors in the solution of the linear systems, which are kept low by the implemented strategy (we point out that, without such a strategy, the propagation of errors would be so high that no solutions could be found by BP). When $\Delta = 7$, there are no instances that satisfy the assumptions for the DMDGP, and 7 instances out of 10 satisfy the assumptions for the $DDGP_3$ without modifying the vertex ordering. We were also able to solve the other 3 instances by the BP algorithm after a suitable reordering of its atoms.

6 Conclusions

We introduced the Discretizable Distance Geometry Problem in $\mathbb{R}^3$ ($DDGP_3$) as a subclass of instances of the DGP for which some particular assumptions are satisfied. Such assumptions allow to reformulate the problem as a combinatorial optimization problem, and hence to reduce the search space from a continuous to a discrete set. We showed that the $DDGP_3$ is an NP-hard problem, and we presented an exact algorithm, the Branch & Prune (BP) algorithm, for solving instances of this problem.

The $DDGP_3$ is a generalization of the Discretizable Molecular Distance Geometry Problem (DMDGP), because the assumptions for the DMDGP are more restrictive than the assumptions for the $DDGP_3$. We formally proved that each instance of the DMDGP is also an instance of the $DDGP_3$, and we showed that there are instances of the $DDGP_3$ that are not instances of the DMDGP. We also showed the importance of the ordering given to the vertices of the instances (the necessary assumptions could be satisfied or not depending on the ordering given to its vertices). In our computational experiments, we considered some instances for which the DMDGP assumptions were not satisfied, while the assumptions for the $DDGP_3$ were always satisfied, in some cases after a suitable reordering of its vertices.

The $DDGP_3$ includes a wider range of instances with respect to the previously studied DMDGP. Instances of the $DDGP_3$ are not necessarily related to molecules or, in particular, to proteins. Therefore, the $DDGP_3$ has a larger applicability, including the problem of localizing wireless sensors. We plan to investigate in future publications the application of the presented BP algorithm for the solution of real-life problems that can be formulated as a $DDGP_3$. To this aim, we will study possible extensions of this work to instances affected by noise and experimental errors. Preliminary studies in this direction, for the DMDGP, were published in [20, 21, 23, 24].

Acknowledgments

The authors would like to thank the Brazilian research agencies FAPESP and CNPq, the French research agency CNRS and École Polytechnique, for financial support. The authors also wish to thank Audrey Lee-St. John and Sonia Cafieri for their fruitful comments on this work.

References

1. E. Anderson, Z. Bai, J. Dongarra, A. Greenbaum, A. McKenney, J. Du Croz, S. Hammerling, J. Demmel, C. Bischof, D. Sorensen, *LAPACK: a Portable Linear Algebra Library for High-Performance Computers*, Supercomputing '90: Proceedings of the 1990 ACM/IEEE conference on Supercomputing, IEEE Computer Society Press, 2–11, 1990.
2. H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, *The Protein Data Bank*, Nucleic Acids Research **28**, 235–242, 2000.
3. R.S. Carvalho, C. Lavor, and F. Protti, *Extending the Geometric Buildup Algorithm for the Molecular Distance Geometry Problem*, Information Processing Letters **108**, 234-237, 2008.
4. I.D. Coope, *Reliable Computation of the Points of Intersection of n Spheres in n-space*, ANZIAM Journal **42**, 461–477, 2000.
5. G.M. Crippen and T.F. Havel, *Distance Geometry and Molecular Conformation*, John Wiley & Sons, New York, 1988.
6. 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, 2673–2684, 2004.
7. T.F. Havel, *Distance Geometry*, D.M. Grant and R.K. Harris (Eds.), Encyclopedia of Nuclear Magnetic Resonance, Wiley, New York, 1701-1710, 1995.
8. C. Lavor, J. Lee, A. Lee-St.John, L. Liberti, A. Mucherino, M. Sviridenko, *Discretization Orders for Distance Geometry Problems*, to appear in Optimization Letters, 2011.
9. C. Lavor, L. Liberti, and N. Maculan, *Discretizable Molecular Distance Geometry Problem*, Tech. Rep. q-bio.BM/0608012, arXiv, 2006.
10. C. Lavor, L. Liberti, and N. Maculan, *Molecular Distance Geometry Problem*, In: Encyclopedia of Optimization, C. Floudas and P. Pardalos (Eds.), 2^{nd} edition, Springer, New York, 2305–2311, 2009.
11. C. Lavor, L. Liberti, N. Maculan, A. Mucherino, *The Discretizable Molecular Distance Geometry Problem*, to appear in Computational Optimization and Applications, 2011.
12. C. Lavor, A. Mucherino, L. Liberti, and N. Maculan, *Computing Artificial Backbones of Hydrogen Atoms in order to Discover Protein Backbones*, IEEE Conference Proceedings, International Multiconference on Computer Science and Information Technology (IMCSIT09), Workshop on Computational Optimization (WCO09), Mragowo, Poland, 751–756, 2009.
13. C. Lavor, A. Mucherino, L. Liberti, and N. Maculan, *An Artificial Backbone of Hydrogens for Finding the Conformation of Protein Molecules*, Proceedings of the Computational Structural Bioinformatics Workshop (CSBW09), Washington D.C., USA, 152–155, 2009.
14. C. Lavor, A. Mucherino, L. Liberti, N. Maculan, *On the Computation of Protein Backbones by using Artificial Backbones of Hydrogens*, to appear in Journal of Global Optimization, 2011. Available online from July 24, 2010.
15. C. Lavor, A. Mucherino, L. Liberti, and N. Maculan, *Discrete Approaches for Solving Molecular Distance Geometry Problems using NMR Data*, International Journal of Computational Biosciences **1**(1), 88–94, 2010.
16. L. Liberti, C. Lavor, and N. Maculan, *A Branch-and-Prune Algorithm for the Molecular Distance Geometry Problem*, International Transactions in Operational Research **15** (1), 1–17, 2008.
17. L. Liberti, C. Lavor, A. Mucherino, N. Maculan, *Molecular Distance Geometry Methods: from Continuous to Discrete*, International Transactions in Operational Research **18**(1), 33–51, 2010.
18. J.J. Moré and Z. Wu, *Global Continuation for Distance Geometry Problems*, SIAM Journal on Optimization **7**, 814–836, 1997.
19. J.J. Moré and Z. Wu, *Distance Geometry Optimization for Protein Structures*, Journal of Global Optimization **15**, 219–223, 1999.
20. A. Mucherino, C. Lavor, *The Branch and Prune Algorithm for the Molecular Distance Geometry Problem with Inexact Distances*, Proceedings of World Academy of Science, Engineering and Technology (WASET), International Conference on Bioinformatics and Biomedicine (ICBB09), Venice, Italy, 349-353, 2009.
21. A. Mucherino, L. Liberti, C. Lavor, and N. Maculan, *Comparisons between an Exact and a MetaHeuristic Algorithm for the Molecular Distance Geometry Problem*, ACM Conference Proceedings, Genetic and Evolutionary Computation Conference (GECCO09), Montréal, Canada, 333–340, 2009.

22. A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, *On the Definition of Artificial Backbones for the Discretizable Molecular Distance Geometry Problem*, Mathematica Balkanica **23**(3-4), 289-302, 2009.
23. A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, *Strategies for Solving Distance Geometry Problems with Inexact Distances by Discrete Approaches*, Proceedings of Toulouse Global Optimization 2010 (TOGO10), Toulouse, France, 93–96, 2010.
24. A. Mucherino, C. Lavor, T. Malliavin, L. Liberti, M. Nilges, M. Maculan, *Influence of Pruning Devices on the Solution of Molecular Distance Geometry Problems*, Lecture Notes in Computer Science **6630**, P.M. Pardalos and S. Rebennack (Eds.), Proceedings of the 10th International Symposium on Experimental Algorithms (SEA11), Crete, Greece, 206–217, 2011.
25. A. Mucherino, L. Liberti, C. Lavor, `MD-jeep`*: an Implementation of a Branch & Prune Algorithm for Distance Geometry Problems*, Lectures Notes in Computer Science **6327**, K. Fukuda et al. (Eds.), Proceedings of the Third International Congress on Mathematical Software (ICMS10), Kobe, Japan, 186–197, 2010.
26. J.B. Saxe, *Embeddability of Weighted Graphs in k-space is Strongly NP-hard*, Proceedings of 17^{th} Allerton Conference in Communications, Control, and Computing, Monticello, IL, 480–489, 1979.
27. M-C. So and Y. Ye, *Theory of Semidefinite Programming for Sensor Network Localization*, Mathematical Programming, **109**, 367-384, 2007.
28. 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.
29. D. Wu, Z. Wu, Y. Yuan, *Rigid Versus Unique Determination of Protein Structures with Geometric Buildup*, Optimization Letters **2**, 319-331, 2008.