The Branch and Prune Algorithm for the Molecular Distance Geometry Problem with Inexact Distances

Antonio Mucherino, Carlile Lavor

Abstract—The Discretizable Molecular Distance Geometry Problem (DMDGP) consists in a subclass of distance geometry instances (related to molecules) that can be solved by combinatorial optimization. A modified version of the Branch and Prune (BP) algorithm, previously proposed for solving these instances, is presented, where it is supposed that exact distances are not known, but rather intervals where the actual distances are contained. This assumption is realistic, because instances of this problem can be defined by applying experimental techniques, such as the Nuclear Magnetic Resonance (NMR), that are subject to errors. Computational experiences show how inaccurate data can affect the quality and the number of solutions which are found by the modified version of the BP algorithm. Depending on the errors introduced in the data, less accurate solutions or a larger number of solutions is found. In the latter case, clusters containing the most similar conformations can be identified, and the cardinality of the solution set can be reduced.

Keywords—distance geometry, protein molecules, branch and prune, inexact distances.

I. Introduction

The Molecular Distance Geometry Problem (MDGP) is the problem of identifying the atomic positions of a molecular conformation by exploiting some known distance between pairs of atoms. This problem can also be seen as the problem of finding an immersion in $\Re^3$ of a given undirected and nonnegatively weighted graph $G = (V, E, d)$. In the graph, the set V of vertices represents the set of atoms of the conformation, the set E of edges indicates the pairs of atoms whose distance is known, and the weights d correspond to the known distances. The MDGP is NP-complete [13], although the problem is solvable in linear time when all the inter-atomic distances are known [3].

There are several methods proposed in the literature for the MDGP (the reader is referred to [6] for a survey). The most common approach is the one in which the MDGP is formulated as a continuous nonconvex optimization problem:

$$\min f(X),$$

where $f(X)$ is an error function which evaluates how much a given conformation $X = \{x_1, x_2, \ldots, x_n\}$ satisfies the known distances d in G. If $i, j \in V$ such that $(i, j) \in E$, then the distance $||x_i - x_j||$ between the two atoms x_i and x_j of X can be computed, and it can be compared to the known distance d_{ij}. One of the possible choices for $f(X)$ is the so-called Largest Distance Error (LDE):

$$f(X) = \frac{1}{|E|} \sum_{\{i,j\}} \frac{||x_i - x_j|| - d_{ij}}{d_{ij}}. \quad (1)$$

Naturally, if the set of known distances is feasible, X solves the problem if and only if $f(X) = 0$.

The Discretizable Molecular Distance Geometry Problem (DMDGP) [9], [10] consists of a certain subset of MDGP instances for which a discrete formulation can be supplied when two particular assumptions are satisfied. Our attention is focused on protein molecules, because their particular structure allows us to consider the discrete reformulation in most of the cases. A Branch and Prune (BP) algorithm has been proposed in [9] for the solution of this discrete problem.

In this paper, the management of possible experimental errors that can affect the distances corresponding to a given instance is investigated. Indeed, instances of the DMDGP can be generated from data obtained from experimental techniques, such as the Nuclear Magnetic Resonance (NMR) [4], and it is well-known that these data can be affected by experimental errors. In particular, NMR experiments cannot provide accurate distances between pairs of atoms, but rather an estimate of such distances. Therefore, instead of single and accurate values for the distances, intervals in which the actual distances are contained are usually provided.

A modified version of the BP algorithm is introduced in this work, which is able to handle intervals instead of exact distances. The computational experiments show that this new version of the algorithm is able to provide, in some cases, solutions having the same quality as exact distances were considered. However, the total number of found solutions increases, and clusters of similar solutions can be identified.

The paper is organized as follows. In Section II, the experimental errors that can affect real data are discussed, and a method for generating instances of the DMDGP affected by errors is shown. In Section III, the new version of the BP algorithm is presented, where the differences between the original and the modified version are pointed out. Computational results are presented in Section IV and conclusions are drawn in Section V.

II. Influence of experimental errors

Experiments of Nuclear Magnetic Resonance (NMR) provide information that can be used for estimating some of the

Antonio Mucherino is with LIX, École Polytechnique, Palaiseau, France. Email: mucherino@lix.polytechnique.fr

Carlile Lavor is with the Department of Applied Mathematics, State University of Campinas, Campinas-SP, Brazil. Email: clavor@ime.unicamp.br

distance between pairs of atoms of a molecule. The set of all such distances defines an instance of the MDGP. Moreover, if some assumptions are satisfied (see Section III), the generated instance is also an instance for the DMDGP. The needed assumptions are usually satisfied when dealing with instances related to proteins, and, in particular, protein backbones [7], [8].

In this work, real data from NMR are not considered, but our instances are rather generated from known protein conformations in order to validate the results by comparing the obtained conformations to the original ones. Conformations of proteins can be downloaded from a public database, the Protein Data Bank (PDB) [1], [12], where they are stored in `pdb` format. This is a text format where, among other information, there are the coordinates in the space of the atoms forming the molecule.

Proteins are formed by smaller molecules called *amino acids* that are bound to each other by defining a sort of chain. A sequence of bound atoms can therefore be identified along the protein conformation that goes amino acid per amino acid: this sequence of atoms is usually referred to as *protein backbone* and it is much studied. The focus of this work is on protein backbones only.

The procedure which is used for generating instances of the DMDGP is the following. Once a certain protein conformation is downloaded from the PDB, its backbone is extracted. In practice, for each amino acid of the molecule, only the atoms N, C_α and C are considered. Then, all the possible distances between the pairs of considered atoms are computed. An instance is defined by all the computed distances which are smaller than 6Å, in order to simulate data from NMR. In fact, NMR experiments are able to provide distances that are smaller than the threshold of 6Å.

This procedure is able to generate instances of the DMDGP in which the distances have a precision which is close to the maximum precision available on a computer machine. However, NMR experiments can provide distances that are not so accurate, and usually only an interval in which a given distance is contained is available. Moreover, there is a low probability that some distances are actually not contained into the provided interval. This kind of experimental error has been already investigated in [11], and it is supposed in the following that all the intervals are correct. Future works can be aimed at the combination of the modified version of the BP algorithm proposed in this paper and the strategy implemented in [11].

Since a realistic instance of the DMDGP is expected to contain a list of lower and upper bounds on a subset of distances between atoms, the procedure described above is used for generating exact distances between atoms, and then the method discussed in [2] is applied in order to introduce errors on such distances. Once the exact distances d_{ij} (< 6Å) have been computed from a given protein conformation, the lower and the upper bound of the distances d_{ij} are obtained, respectively, by

$$\begin{cases} l_{ij} = d_{ij} \max(0, 1 - |\underline{\varepsilon}_{ij}|) \\ u_{ij} = d_{ij}(1 + |\overline{\varepsilon}_{ij}|), \end{cases} \quad (2)$$

where $\underline{\varepsilon}_{ij}$ and $\overline{\varepsilon}_{ij}$ are random numbers in a normal distribution with center in 0 and variance σ^2. The chosen variance provides the corresponding noise introduced in the data. The version of the BP algorithm that is presented in the next section is able to handle instances generated in this way.

III. A modified version of the Branch and Prune algorithm

Let us consider an instance of the MDGP and let $G = (V, E, d)$ be the associated weighted undirected graph. If the following two assumptions are satisfied, then the considered instance is also an instance for the DMDGP:

Assumption 1: E contains all cliques on quadruplets of consecutive vertices, i.e. $\forall i \in \{4, \ldots, n\}$ and $\forall j, k \in \{i - 3, \ldots, i\}$:

$$(j, k) \in E;$$

Assumption 2: the following strict triangular inequality must hold $\forall i = 2, \ldots, n - 1$:

$$d_{i-1,i+1} < d_{i-1,i} + d_{i,i+1}.$$

In practice, Assumption 1 ensures that the distances between each pair of atoms in each quadruplet of consecutive atoms are all known. If Assumption 2 holds, moreover, triplets of consecutive atoms cannot be perfectly aligned. When both assumptions are satisfied, there exist only two possible positions where the generic atom x_i can be placed, if the three preceding atoms are already placed into a fixed location. This leads to the definition of a binary tree of possible molecular conformations, where the solutions of the DMDGP can be searched [9].

Given an instance of the DMDGP, the corresponding set E of edges can be divided into two subsets:

- H, which refers to all the distances between atoms i and j such that $j \leq i + 3$,
- $F = E - H$.

It is very important to note that the binary tree of solutions of the DMDGP can be completely defined by using the distances in the subset H (all these distances are all known because of the first assumption). Instead, the distances in F can be exploited for looking for solutions to the problem on the binary tree.

The basic idea behind the BP algorithm is as follows. At each step, two possible atomic positions are computed for the current atom i. This is equivalent to adding two new nodes on the binary tree of possible solutions. Once the two positions are computed, their feasibility is evaluated by employing *pruning tests*. When infeasible positions are discovered, the corresponding branch on the binary tree is pruned, because no solutions can be found on that branch. Algorithm 1 is a sketch of the BP algorithm. Other details on the algorithm can be found in [9], [10].

As already mentioned, it is very important to make a distinction between the distances in the subset H and the distances in the subset F. In the case in which the provided distances are exact, the two atomic positions for the atom i can be computed by intersecting the three spheres having centers in x_{i-3}, x_{i-2} and x_{i-1} and radius, respectively, $d_{i-3,i}$, $d_{i-2,i}$ and $d_{i-1,i}$. The problem of finding these intersections can be

Algorithm 1 The BP algorithm.

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

efficiently solved as described in [9]. If the assumptions for the DMDGP are satisfied, such intersections always define two atomic positions only.

If, instead of exact distances, intervals are provided, then this strategy cannot be applied. Therefore, an estimation of the actual distances from the available intervals are needed. One possible way to do this is to consider the average distances defined by the intervals. This obviously brings to approximations, because it is known that the correct distances are in the intervals, but they may not be in the middle of such intervals. However, the average distances are, in general, the best possible choices.

During the execution of the algorithm, couples of new positions are computed. In order to check the feasibility of such positions, pruning tests are used, which are based on the distances in the subset F. Different pruning tests can be used [7], and the most natural one is the following. Supposing that exact distances are provided, for each x_i and $j \in V$ such that $(j, i) \in F$, if computed and known distances match:

$$| \, ||x_i - x_j|| - d_{ij} \, | < \varepsilon,$$

for a given tolerance $\varepsilon > 0$, then the atomic position x_i is feasible, otherwise it is not. Branches of the tree containing infeasible positions can be pruned, because it is sure they do not contain solutions to the problem.

If, instead of exact distances d_{ij}, intervals $[l_{ij}, u_{ij}]$ are provided, then the pruning test, as described above, cannot be used. The immediate variant is the following. For each x_i and $j \in V$ such that $(j, i) \in F$, the atomic position x_i is infeasible if the following two inequalities do not hold:

$$l_{ij} < ||x_i - x_j|| < u_{ij}.$$

Note that, in this case, there is no need to use any tolerance ε, but the length of the interval $[l_{ij}, u_{ij}]$ plays the rule of ε. As a consequence, the actual tolerance used in this pruning test depends on the noise in the data.

IV. Computational experiences

The experiments presented in this section have been carried out on one core of an Intel Core 2 CPU 6400 @ 2.13 GHz with 4GB RAM, running Linux. Computational experiments have been performed on a set of instances generated from proteins having known conformation, as described in Section II.

instance	n	$\|E\|$	σ	*#Sol*	LDE
2erl	120	1136	0.02	2	1.33e-14
			0.05	2	1.33e-14
			0.08	2	1.33e-14
			0.10	2	1.33e-14
			0.15	4	1.33e-14
			0.20	4	1.33e-14
			0.30	32	1.33e-14

TABLE I
EXPERIMENTS ON THE PROTEIN 2erl, WHERE DIFFERENT NOISES ARE APPLIED TO THE DISTANCES RELATED TO F (IN ALL CASES, THE CPU TIME WAS 0.00).

Before presenting the computational results, only the protein 2erl is considered in Table I. In the table, n is the number of atoms of the backbone of the considered protein, $|E|$ is the number of given distances, σ is the noise defined in Section II, and *#Sol* is the number of found solutions. LDE represents the quality of the best found solution (see equation (1)): the distances d_{ij} used in the formula are the exact ones, the ones used in (2) for computing the intervals $[l_{ij}, u_{ij}]$. The CPU time is given in seconds, if not otherwise specified.

In these experiments, the noise was applied only to the distances in F, i.e. to the distances used in the pruning test. Therefore, the distances used for generating the binary tree of solutions are accurate. Table I shows that, for any noise σ applied to the distances, the BP algorithm is always able to find the same best solution. What changes is the total number of found solutions. The higher is the noise σ, the more solutions are found. This phenomenon is due to the fact that the lengths of the intervals $[l_{ij}, u_{ij}]$ used in the pruning tests increase as the noise on the data increases, and larger intervals allow to detect less infeasible atomic positions.

However, some deeper analysis on the solution set revealed that, even though the total number of solutions increases with noise, subgroups of solutions which are very similar to each other can be identified. In other words, the solution set can be partitioned in clusters, with each cluster containing solutions which are very similar, but different from the solutions in other clusters. For example, in the experiment shown in Table I with $\sigma = 0.30$, two separate clusters can be found, each of them containing 16 solutions.

Figure 1 shows two representatives of these clusters. The Root Mean Square Deviation (RMSD) [14] between all the possible pairs of solutions belonging to the same cluster, and also belonging to different clusters, have been computed. As Figure 1 shows, the mean RMSD value among the solutions of the same cluster is quite small, and this proves that such solutions are very similar to each other. Moreover, they are very different from the solutions in the other cluster, as the mean RMSD value among solutions belonging to different clusters shows.

Table II shows experiments on a subset of instances related to some of the proteins used by Biswas, Toh and Ye [2], Wu and Wu [14], and Hendrickson [5] in their computational experiments. All the experiments were limited to 3 hours of

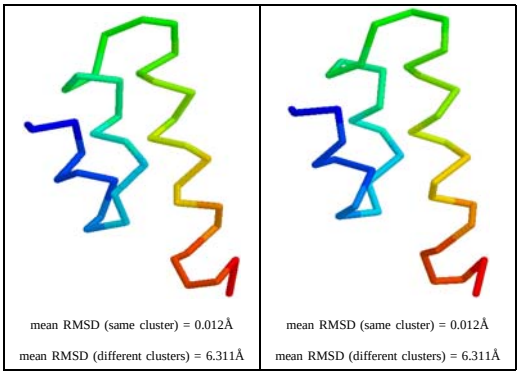

Fig. 1. Two representatives of the two clusters of solutions related to the experiment in Table I with $\sigma = 0.30$.

CPU time. In only three cases, corresponding to large proteins and to higher noise, the algorithm was still running after 3 hours. In one case (`1epw`, $\sigma = 0.20$), the algorithm found 16 solutions, but failed to terminate within the time limit. In other two cases (`1acz`, `1rgs`, $\sigma = 0.20$), when the algorithm was forced to stop after 3 hours, no solutions were found yet, probably because the pruning test was not able to prune, by that time, a sufficient number of branches in order to reach a leaf node of the search tree. In general, the experiments show that a larger noise σ does not affect the quality of the best found solution, evaluated through the LDE function. However, as the number of solutions increases, and the computational time for carrying out the experiments is larger, clusters of solutions can be found by comparisons by RMSDs. As a consequence, the cardinality of the solution set can be reduced to the number of clusters that can be identified: each solution can be defined as the average among all the conformations belonging to the same cluster.

In Table III, some experiments show how errors on the distances related to the subset H can affect the results of the BP algorithm. In these experiments, intervals for the distances in H with a certain noise σ are not generated, but the accuracy of these distances is directly decreased by modifying the number of decimal digits used for their representation. Only the accuracy of the distances $d_{i-3,i}$ is modified, because the distances $d_{i-2,i}$ and $d_{i-1,i}$ are known in proteins (bond lengths and bond angles are known *a priori*). In Table III, only one protein is considered, but similar results can be obtained with the other proteins of the considered set. The experiments show that the quality of the best solution found by the algorithm decreases with the accuracy of the distances $d_{i-3,i}$, whereas the number of found solutions increases as the noise on the distances in F increases. In this case, the tree of possible solutions built during the execution of the algorithm changes with the accuracy of the distances $d_{i-3,i}$. Therefore, the less is the accuracy of such distances, the less is the quality of the best found solution. The errors introduced on the distances in F do not have instead this effect. Such errors, as the previous experiments show, are able to interfere only on the total number of solutions found by the algorithm. Note that, when the accuracy on the distances $d_{i-3,i}$ goes below the 5^{th} decimal digit, the positions on the tree and the bounds $[l_{ij}, u_{ij}]$ on the distances in F become incompatible, and no solutions are found.

instance	n	$\|E\|$	σ	*#Sol*	LDE	time
`1brv`	57	476	0.05	2	1.39e-14	0.00e+00
			0.10	2	1.39e-14	0.00e+00
			0.20	2	1.39e-14	0.00e+00
`1aqr`	120	929	0.05	8	3.10e-13	1.00e-02
			0.10	128	3.10e-13	3.00e-02
			0.20	1024	3.10e-13	2.10e-01
`1crn`	138	1250	0.05	2	2.24e-13	0.00e+00
			0.10	2	2.24e-13	0.00e+00
			0.20	2	2.24e-13	0.00e+00
`1ahl`	147	1205	0.05	16	9.86e-13	0.00e+00
			0.10	16	9.86e-13	3.00e-02
			0.20	128	9.86e-13	6.10e-01
`1ptq`	150	1263	0.05	2	2.30e-13	0.00e+00
			0.10	4	2.30e-13	0.00e+00
			0.20	12	2.30e-13	3.00e-02
`1brz`	159	1394	0.05	4	4.48e-13	1.00e-02
			0.10	32	4.48e-13	2.86e+00
			0.20	512	4.48e-13	9.35e+03
`1hoe`	222	1995	0.05	2	3.18e-13	0.00e+00
			0.10	2	3.18e-13	0.00e+00
			0.20	4	3.18e-13	0.00e+00
`1lfb`	232	2137	0.05	2	5.31e-14	0.00e+00
			0.10	2	5.31e-14	0.00e+00
			0.20	16	5.31e-14	1.00e-02
`1pht`	249	2283	0.05	4	2.73e-12	0.00e+00
			0.10	8	2.73e-12	1.00e-02
			0.20	16	2.73e-12	1.46e+00
`1jk2`	270	2574	0.05	2	2.09e-13	0.00e+00
			0.10	6	2.09e-13	1.00e-02
			0.20	384	2.09e-13	3.00e-01
`1f39a`	303	2660	0.05	2	1.88e-08	1.00e-02
			0.10	8	1.88e-08	2.00e-02
			0.20	192	1.88e-08	9.79e+01
`1acz`	324	3060	0.05	8	2.75e-12	9.00e-02
			0.10	32	2.75e-12	8.85e+01
			0.20	0	-	**3h**
`1poa`	354	3193	0.05	2	1.36e-13	0.00e+00
			0.10	2	1.36e-13	0.00e+00
			0.20	2	1.36e-13	2.00e-02
`1fs3`	372	3443	0.05	4	8.08e-13	0.00e+00
			0.10	4	8.08e-13	1.00e-02
			0.20	16	8.08e-13	7.60e-01
`1mbn`	459	4599	0.05	2	1.36e-09	1.00e-02
			0.10	4	1.36e-09	0.00e+00
			0.20	32	1.36e-09	2.00e-02
`1rgs`	792	7626	0.05	2	4.22e-13	2.00e-02
			0.10	8	4.22e-13	9.50e+01
			0.20	0	-	**3h**
`1m40`	1224	20382	0.05	2	1.00e-12	2.00e-02
			0.10	2	1.00e-12	2.00e-02
			0.20	2	1.00e-12	3.00e-02
`1bpm`	1443	14292	0.05	2	2.85e-13	2.00e-02
			0.10	2	2.85e-13	4.00e-02
			0.20	4	2.85e-13	5.03e+00
`1n4w`	1610	16940	0.05	2	1.19e-12	3.00e-02
			0.10	4	1.19e-12	4.00e-02
			0.20	8	1.19e-12	4.90e-01
`1mqq`	2032	19564	0.05	2	9.89e-08	4.00e-02
			0.10	2	9.89e-08	9.00e-02
			0.20	4	9.89e-08	1.19e+01
`1rwh`	2265	21666	0.05	2	2.08e-13	6.00e-02
			0.10	2	2.08e-13	8.00e-02
			0.20	8	2.08e-13	2.53e+00
`3b34`	2790	29188	0.05	2	1.17e-11	1.00e-01
			0.10	2	1.17e-11	1.20e-01
			0.20	16	1.17e-11	4.24e+01
`2e7z`	2907	42098	0.05	2	4.26e-12	1.00e-01
			0.10	4	4.26e-12	2.30e-01
			0.20	8	4.26e-12	6.57e+00
`1epw`	3861	35028	0.05	16	3.44e-12	1.54e+00
			0.10	32	3.44e-12	3.40e+01
			0.20	16	3.44e-12	**3h**

TABLE II
EXPERIMENTS WITH DIFFERENT NOISES APPLIED TO ALL THE DISTANCES RELATED TO THE SUBSET F.

σ	0.05	0.08	0.10	0.15	0.20
# digits ($d_{i-3,i}$)	LDE (*#Sol*)	LDE (*#Sol*)	LDE (*#Sol*)	LDE (*#Sol*)	LDE (*#Sol*)
15	1.19e-12 (2)	1.19e-12 (2)	1.19e-12 (4)	1.19e-12 (4)	1.19e-12 (8)
14	1.19e-12 (2)	1.19e-12 (2)	1.19e-12 (4)	1.19e-12 (4)	1.19e-12 (8)
13	1.21e-12 (2)	1.21e-12 (2)	1.21e-12 (4)	1.21e-12 (4)	1.21e-12 (8)
12	1.49e-12 (2)	1.49e-12 (2)	1.49e-12 (4)	1.49e-12 (4)	1.49e-12 (8)
11	4.99e-12 (2)	4.99e-12 (2)	4.99e-12 (4)	4.99e-12 (4)	4.99e-12 (8)
10	4.33e-11 (2)	4.33e-11 (2)	4.33e-11 (4)	4.33e-11 (4)	4.33e-11 (8)
9	4.50e-10 (2)	4.50e-10 (2)	4.50e-10 (4)	4.50e-10 (4)	4.50e-10 (8)
8	4.29e-09 (2)	4.29e-09 (2)	4.29e-09 (4)	4.29e-09 (4)	4.29e-09 (8)
7	4.33e-08 (2)	4.33e-08 (2)	4.33e-08 (4)	4.33e-08 (4)	4.33e-08 (8)
6	4.02e-07 (2)	4.02e-07 (2)	4.02e-07 (4)	4.02e-07 (4)	4.02e-07 (8)
5	4.20e-06 (2)	4.20e-06 (2)	4.20e-06 (4)	4.20e-06 (4)	4.20e-06 (8)

TABLE III
EXPERIMENTS ON ONE PROTEIN (`1n4w`), IN WHICH DIFFERENT NOISES ARE APPLIED TO THE DISTANCES IN F AND A DIFFERENT NUMBER OF DECIMAL DIGITS ARE USED FOR REPRESENTING THE DISTANCES $d_{i-3,i}$.

V. CONCLUSIONS

The DMDGP aims at finding the three-dimensional conformations of molecules by combinatorial optimization. It is supposed that only some of the distances between pairs of atoms are known. Such distances can be obtained by experimental techniques such as NMR, where the distances between atoms closer than around 6Å can be estimated. The accuracy of these data is usually low, and therefore intervals where the actual distances are contained are usually provided, instead of exact distances.

A modified version of the BP algorithm is presented in this paper, that was previously proposed for solving the combinatorial optimization problem related to DMDGP instances. The modified version of the algorithm is able to overcome some of the problems arising when the accuracy of the known distances is not high, as in the case in which data from NMR experiments are used.

Starting from the known conformations of some proteins, a set of instances, in which different degrees of noise were introduced, was generated. The aim of the presented computational experiments was to study how these errors affect the results found by the BP algorithm. The experiments showed that less accurate distances can lead to the identification of more solutions. However, if particular distances are accurate enough (the distances in H), then the best solution found is the same as exact distances were considered. Moreover, another important result is that the solution set can be partitioned in clusters, and one unique solution can be generated as the representer of each cluster.

ACKNOWLEDGMENTS

The authors would like to thank the Brazilian research agencies FAPESP and CNPq, the French research agency CNRS and the École Polytechnique, for financial support. The authors also wish to thank Leo Liberti for his suggestions.

REFERENCES

[1] 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, vol. 28, pp. 235–242, 2000.

[2] P. Biswas, K.-C. Toh, Y. Ye, "A Distributed SDP Approach for Large-Scale Noisy Anchor-free Graph Realization with Applications to Molecular Conformation", SIAM Journal on Scientific Computing, vol. 30, pp. 1251–1277, 2008.

[3] Q. Dong, Z. Wu, "A Linear-time Algorithm for Solving the Molecular Distance Geometry Problem with Exact Inter-atomic Distances", Journal of Global Optimization, vol. 22, pp. 365–375, 2002.

[4] T.F. Havel, "Distance Geometry". In: D.M. Grant, R.K. Harris (Eds.), Encyclopedia of Nuclear Magnetic Resonance, Wiley, New York, pp. 1701–1710, 1995.

[5] B.A. Hendrickson, "The Molecule Problem: Exploiting Structure in Global Optimization", SIAM Journal on Optimization, vol. 5, pp. 835–857, 1995.

[6] C. Lavor, L. Liberti, N. Maculan,"Molecular Distance Geometry Problem". In: C. Floudas, P.M. Pardalos (Eds.), Encyclopedia of Optimization, 2^{nd} Edition, Springer, New York, pp. 2305–2311, 2009.

[7] C. Lavor, L. Liberti, A. Mucherino, and N. Maculan, "On a Discretizable Subclass of Instances of the Molecular Distance Geometry Problem", ACM Conference Proceedings, 24^{th} Annual ACM Symposium on Applied Computing (SAC09), Hawaii USA, pp. 804–805, 2009.

[8] 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 Conference IMCSIT09, Workshop on Combinatorial Optimization (WCO09), Poland, October 2009.

[9] L. Liberti, C. Lavor, N. Maculan, "A Branch-and-Prune Algorithm for the Molecular Distance Geometry Problem", International Transactions in Operational Research, vol. 15, pp. 1–17, 2008.

[10] L. Liberti, C. Lavor, N. Maculan, "Discretizable Molecular Distance Geometry Problem", Tech. Rep. q-bio.BM/0608012, arXiv, 2006.

[11] A. Mucherino, L. Liberti, C. Lavor, and N. Maculan, "Comparisons between an Exact and a Meta-heuristic Algorithm for the Molecular Distance Geometry Problem", ACM Conference Proceedings. Conference GECCO09, Montréal, Canada, pp. 333–340, 2009.

[12] Protein Data Bank (PDB), `http://www.rcsb.org/`

[13] 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, pp. 480–489, 1979.

[14] D. Wu, Z. Wu, "An Updated Geometric Build-up Algorithm for Solving the Molecular Distance Geometry Problem with Sparse Distance Data", Journal of Global Optimization, vol. 37, pp. 661–673, 2007.