Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 379–384

ISBN 978-83-60810-51-4

On suitable orders for discretizing Molecular Distance Geometry Problems related to protein side chains

Virginia Costa*, Antonio Mucherino†, Carlile Lavor‡, Luiz M. Carvalho§ and Nelson Maculan*

*COPPE, Federal University of Rio de Janeiro, Rio de Janeiro-RJ, Brazil.
{virscosta,maculan}@cos.ufrj.br

†IRISA, University of Rennes 1, Rennes, France.
antonio.mucherino@irisa.fr

‡IMECC-UNICAMP, Campinas-SP, Brazil.
clavor@ime.unicamp.br

§IME, State University of Rio de Janeiro, Rio de Janeiro-RJ, Brazil.
luizmc@ime.uerj.br

***Abstract*—Proteins are important molecules that are widely studied in biology. Their three-dimensional conformations can give clues about their function, however an optimal methodology for the identification of such conformations has not been found yet. Experiments of Nuclear Magnetic Resonance (NMR) are able to estimate distances between some pairs of atoms forming the protein, and the problem of identifying the possible conformations satisfying the available distance constraints is known in the scientific literature as the Molecular Distance Geometry Problem (MDGP). Since some years, some of us have been working on a suitable discretization for the MDGP and on an efficient Branch & Prune (BP) algorithm which is based on a tree search. In order to perform this discretization, however, some assumptions need to be satisfied. We recently hand-crafted a special order for protein backbone atoms which allows us to discretize all MDGPs concerning backbones. In this paper, we do the same for the side chains of some amino acids. Our computational experiments show that the inclusion of the side chain information allows to improve the performances of the BP algorithm.**

I. Introduction

THE Molecular Distance Geometry Problem (MDGP) consists in finding the suitable conformations for a certain molecule which satisfy a set of constraints based on some distances between pairs of its atoms [2]. When the distance information is given through a list of lower and upper bounds on the distances, i.e. by a list of suitable real intervals, the problem is also referred to as *interval* MDGP (iMDGP) [10]. The iMDGP, by its nature, is a constraint satisfaction problem, which is NP-hard [12]. Over the years, its solution has been attempted by formulating global optimization problems in continuous spaces [7], where a penalty objective function is generally employed in order to measure the *satisfaction* of the distance constraints for given molecular conformations. More recently, some of the authors of this paper introduced a new class of iMDGP instances for which the search domain can be reduced to a discrete space having the structure of a tree [4]. We refer to this class of problems as the *interval* Discretizable MDGP (iDMDGP). Instances belonging to this class can be solved by employing an efficient *interval* Branch & Prune (iBP) algorithm [5], [6].

Experiments of Nuclear Magnetic Resonance (NMR) are able to estimate the distances between some pairs of atoms of a molecule. Nowadays, NMR is the second most used technique for the identification of protein conformations, which are very important molecules performing several fundamental functions in living beings. Due to this great interest, three-dimensional conformations of proteins found by the scientific community are generally stored in a web database named the Protein Data Bank (PDB) [1]. To date, however, and in most cases, iMDGPs corresponding to NMR experiments for a given protein are formulated as continuous global optimization problems, whose solution is attempted by the meta-heuristic Simulated Annealing (SA) [9].

As it is well-known, a meta-heuristic search can give no guarantees on the optimality of the obtained solutions. However, since meta-heuristics such as SA can be easily implemented, this kind of optimization methods is widely used by the biological and chemical community. This observation stimulated our research on the iDMDGP and on the iBP. The latter, indeed, is an exact algorithm that is potentially able to identify the complete set of solutions for a given instance. This point is crucial in domains such as biology, where the identification of one possible conformation for a protein which satisfies all distance constraints does not directly imply that the actual protein conformation was found. Our aim is to provide the biologist with a complete set of *mathematical* solutions (we can guarantee that all solutions were found and that there are no other solutions that escaped to the search); biologist's task is then the one of successively filtering the set of mathematical

solutions by selecting the conformations that have a more evident biological sense.

The application of iBP is possible when the instance at hand belongs to the iDMDGP class, i.e. when some particular assumptions are satisfied. The main requirement for the discretization is the presence of a suitable atomic order (that can be different from the *natural* ordering of the atoms in a molecule) such that, for each atom v, there are at least three preceding reference atoms u_1, u_2 and u_3 with known distance from v. This allows us to compute the possible positions for the atom v by intersecting three Euclidean objects in the three-dimensional space. When the distance between two atoms, say u_1 and v, is precise, the considered Euclidean object is a sphere centered in u_1 (this atom precedes in the order the atom v, so we suppose its position is already known) and having radius $d(u_1, v)$. If this distance is instead imprecise, i.e. it is represented by an interval $[\underline{d}(u_1, v), \bar{d}(u_1, v)]$, then the Euclidean object is a spherical shell centered in u_1 having minimum radius $\underline{d}(u_1, v)$ and maximum radius $\bar{d}(u_1, v)$. When, for a certain v, two reference distances are precise and only one is represented by an interval, then a finite subset of atomic positions for v can be computed by applying the procedure detailed in [5]. More details about the discretization process are given in Section II.

The satisfaction of the assumptions for the iDMDGP depends upon the order given to the atoms of the instance. The original order may not allow for the discretization, but other possible orders might be identified. We also point out that no discretization orders could be available, because of the lack of distance information. NMR experiments, generally, are able to give estimates of distances between pairs of hydrogen atoms whose relative distance is shorter than a certain threshold (approximately 4–5Å). When we consider molecules with a known chemical composition, such as proteins, a priori information on some interatomic distances can be obtained and included in the instance. Such an information, moreover, can be exploited for identifying an atomic order for which the iDMDGP assumptions are satisfied, independently on the additional information that NMR experiments can provide. In [5], one of such orders for the discretization of instances related to protein backbones (without side chains) has been presented.

In this paper, we propose suitable discretization orders for the side chains of some of the amino acids that can be involved in the synthesis of the proteins. This work is motivated by the fact that NMR instances mainly contain information related to such side chains: NMR data usually regard distances between pairs of hydrogen atoms, and some side chains are composed by several hydrogens. Moreover, some side chains are hydrophobic, which means that they do not react with the protein solvent (water), but they are rather buried inside the protein interior. As a consequence, the interatomic distances between pairs of hydrogens belonging to different hydrophobic side chains are likely to be close in space and, therefore, are likely to be detected through NMR experiments.

The rest of the paper is organized as follows. In Section II, we briefly present the iDMDGP and give a sketch of the iBP algorithm, which we employ for solving iDMDGPs with interval data. In Section III, we discuss about the discretization order for instances related to protein backbones, while, in Section IV, we present some new hand-crafted orders related to 8 side chains. Computational experiments, detailed in Section V, show that the inclusion of the information related to the side chains allows to improve the pruning capabilities of the iBP algorithm. Conclusions are drawn in Section VI.

Algorithm 1 The iBP algorithm.

```
1:  iBP(j, r, d, D)
2:  if (r_j is a duplicated atom) then
3:     copy coordinates of previous copy of r_j in x^1_{r_j}
4:     iBP(j + 1, r, d, D);
5:  else
6:     if (d(r_{j-3}, r_j) is exact) then
7:        b = 2;
8:     else
9:        b = 2D;
10:    end if
11:    for k ∈ {1, ..., b} do
12:       compute the k-th position x^k_{r_j} for the r_j-th atom;
13:       check the feasibility of position x^k_{r_j};
14:       if (x^k_{r_j} is feasible) then
15:          if (j = |r|) then
16:             a solution x is found, print it;
17:          else
18:             iBP(j + 1, r, d, D);
19:          end if
20:       end if
21:    end for
22: end if
```

II. The iDMDGP and the iBP algorithm

An instance of the iDMDGP can be represented by a weighted undirected graph $G = (V, E, d)$, where each vertex $v \in V$ represents an atom and each edge $(u, v) \in E$ represents the known distance between the vertices u and v. The weight $d(u, v)$ associated to an edge (u, v) can correspond either to a precise distance or to a suitable interval where the actual distance is supposed to be contained.

Let us suppose that there exists a total order relationship for the vertices of $V = \{1, \ldots, n\}$. Assumption 1 of the iDMDGP requires that the first 3 atoms in V form a clique (all their interatomic distances are known) and that, for each $v > 3$, the distances between v and the 3 immediate predecessors are known:

$$\textbf{Ass.1} \quad \forall v \in \{4, \ldots, n\}\ \forall j, k \in \{v-3, \ldots, v\} \quad (j, k) \in E.$$

Moreover, for any triplet of consecutive vertices, the strict triangular inequality must be satisfied (Assumption 2):

$$\textbf{Ass.2} \quad \forall v \in \{2, \ldots, n-1\}$$

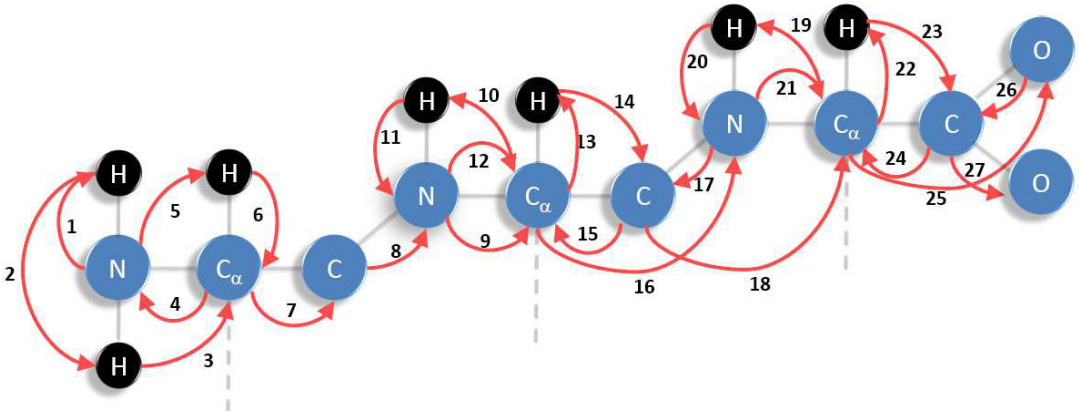

Fig. 1. The hand-crafted order r_{PB} for the protein backbone.

$$d(v-1,v+1) < d(v-1,v) + d(v,v+1).$$

In all atomic orders discussed in this paper, all distances related to edges (u,v), with $v-u \leq 2$, are precise. Therefore, only precise distances are concerned in Assumption 2.

Assumption 1 allows to compute the possible positions for the generic atom v as the intersection among three Euclidean objects, which are related to the three immediate preceding atoms $v-3$, $v-2$ and $v-1$. Each Euclidean object can be either a sphere (when the distance is precise) or a spherical shell (when the distance is represented by an interval). The intersection among three spheres consists of, with probability one, two points in the three-dimensional space [3]. The role of Assumption 2 is to prevent the case in which the sphere intersection gives infinitely many points.

If one of the distances is represented by an interval, one of the spheres is replaced by a spherical shell, so that the intersection consists of, with probability one, two disjoint curves. In order to guarantee the discretization in this case, we choose a certain number of sample distances from the available interval, and we identify a predetermined number of possible atomic positions on the two curves [5]. All orders discussed in this paper are constructed so that, for each v, at least two reference distances are exact. Therefore, our intersections can at most produce two disjoint curves that can be successively discretized.

More particularly, the iBP algorithm has been conceived in order to manage the three following situations (see Alg. 1) [5]. Let r be a discretization order related to a certain iDMDGP instance. We also refer to such an order as a *re-order*, because distinct vertices r_j of the ordering can represent the same atom of the considered molecule. This is done for *producing* precise distances: the distance between two copies of the same atom r_{j_1} and r_{j_2} is evidently 0. In such a case, the algorithm recognizes that the current vertex represents a duplicated atom, and it associates to this atom the position of its previous copy. Otherwise, there are other two possible situations to manage: either the three reference distances are precise, or only the distance $d(r_{j-3}, r_j)$ is represented by an interval. In the first case, the sphere intersection gives the only two possible positions for r_j. In the second case, the intersection provides two curves: we choose D sample distances from the interval distance, and we intersect the corresponding three spheres D times. As a consequence, $2 \times D$ possible atomic positions are determined for r_j.

iBP recursively calls itself until the search domain is exhaustively explored. In the algorithm call, j is the rank of the current vertex, r is the vertex order allowing for the discretization, d represents the distance information (precise distances and intervals), and D is the number of sample distances taken from intervals. As it can be deduced from the algorithm sketch, this search domain has the structure of a tree, and it can be organized in $|r|$ layers. Each layer contains all possible positions for the vertex r_j. The tree is constructed as the search proceeds: for a given branch at layer $j-1$, a certain number of branches are added at layer j, and this number can be equal to 1, 2 or $2 \times D$, depending on the three situations mentioned above.

The possible combinatorial explosion makes the iBP algorithm exponential in the worth case. However, the strong point of this algorithm is given by its pruning phase, where the feasibility of the computed atomic positions is verified, and, in case of infeasibilities, the corresponding branch is immediately pruned and not considered anymore. This allows us to focus the searches on the parts on the tree where there are solutions. Moreover, it was recently proved that instances regarding protein conformations generally bring to the definition of trees having a bounded width [8].

III. An order for the protein backbone

Let $G = (V, E, d)$ be a weighted undirected graph representing an instance of the iDMDGP. Let us divide the edge set E in two parts: E_d, which contains distances necessary for the discretization (Assumption 1), and E_p, which contains all the other available distances. In each step of the iBP algorithm, a certain number of possible positions for the current vertex r_j are computed by using the distances in E_d, and the feasibility of these positions is immediately verified

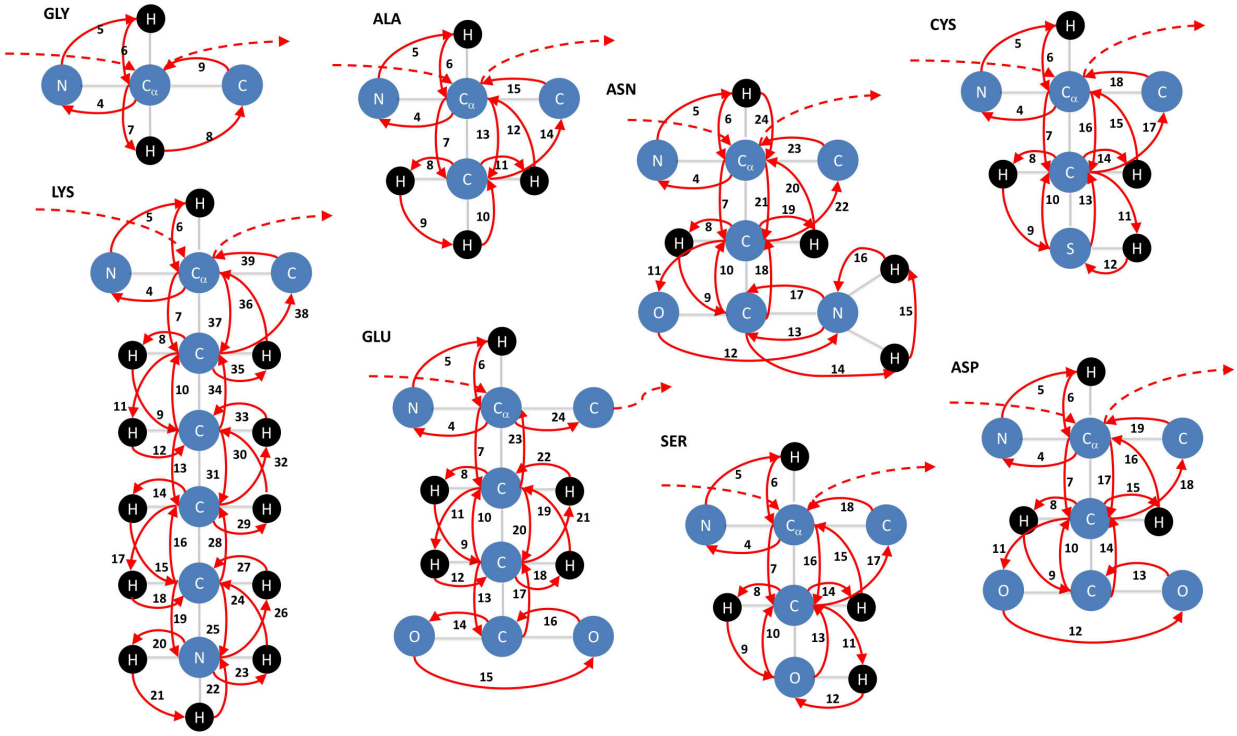

Fig. 2. The hand-crafted orders for 8 side chains.

by exploiting information on additional distances contained in E_p. In this work, we suppose that E_d only contains distances obtained by analyzing the chemical composition of proteins. In practice, all the distances in E_d are derived from known bond lengths and bond angles that are present in protein structures. The distances in E_p, instead, represent the pruning distances, which can be obtained by NMR experiments. One important consequence of the fact that E_d do not contain NMR data is that our search tree (and hence our atomic coordinates) are not computed by exploiting NMR information, which could be affected by errors.

As mentioned in the Introduction, there exist special orderings for V so that Assumptions 1 and 2 of the iDMDGP are always satisfied. In [5], we found one of such orders for the protein backbones. Fig. 1 shows the hand-crafted order for a small protein backbone containing 3 amino acids. For a protein containing p amino acids, the following order can be used for discretizing the backbone of proteins:

$$\begin{aligned} r_{\mathrm{PB}} \quad = \quad & (N^1, H^1, H^0, C_\alpha^1, N^1, H_\alpha^1, C_\alpha^1, C^1, \\ & N^2, C_\alpha^2, H^2, N^2, C_\alpha^2, H_\alpha^2, C^2, C_\alpha^2, \ldots, \\ & N^i, C^{i-1}, C_\alpha^i, H^i, N^i, C_\alpha^i, H_\alpha^i, C^i, C_\alpha^i, \ldots, \\ & N^p, C^{p-1}, C_\alpha^p, H^p, N^p, C_\alpha^p, H_\alpha^p, C^p, C_\alpha^p, \\ & O^p, C^p, O^{p+1}). \end{aligned}$$

Superscripts indicate the amino acid to which each atom belongs: in our notation, H^0 is the second hydrogen bonded to N in the first amino acid, whereas O^{p+1} is the second oxygen of the last amino acid. A particular order has been identified for the first, the second, the generic (labeled with i) and the last amino acid for protein backbones. This order can be used in conjunction with the iBP algorithm described in the previous section: for each vertex r_j of the order, the necessary distances for the discretization are all available and only the distance between r_{j-3} and r_j may be represented by an interval. Up to 3 copies of the same atom can be present in this ordering. We point out that this does not increase the complexity of the problem, because there is no branching on the tree in correspondence with duplicated atoms.

IV. New orders for the side chains

We present in this section new hand-crafted discretization orders for the side chains of 8 of the smallest amino acids that can be involved in the protein synthesis. Such orders can be combined with the backbone order in Figure 1 for discretizing instances concerning entire protein conformations (backbone and side chains). In Section V, our computational experiments will show that the addition of the distance information regarding the side chains improves the pruning capabilities of the iBP algorithm, by allowing a stricter selection of the tree branches where feasible solutions to the iDMDGP can be searched.

Figure 2 contains the discretization orders for 8 side chains. The glycine (GLY) is the smallest amino acid that can be found in proteins, whose side chain is composed by a hydrogen atom

only. A possible discretization order for GLY is the following:

$$r_{\mathrm{GLY}} = \{C_\alpha^i, N^i, H_\alpha^i, C_\alpha^i, H_\beta^i, C^i, C_\alpha^i\},$$

where, in our notation, H_β^i represents the only hydrogen atom which forms the side chain. It can be noted that the inclusion of only one atom needs the duplication of some backbone atoms in order to have the discretization assumptions satisfied.

The other amino acids that we consider in this paper are: the alanine (ALA), the aspargine (ASN), the aspartic acid (ASP), the cysteine (CYS), the glutamic acid (GLU), the lysine (LYS) and the serine (SER). ALA is the second amino acid in order of size. Its side chain is composed by 4 atoms: one carbon atom bonded to three hydrogens. However, in order to have the discretization, the ordering associated to ALA is composed by much more vertices, connected by 11 edges, labeled from 4 to 15. The same observation can be made for all other side chains that we consider. LYS is the longest in Figure 2. The proposed order is represented by 36 edges, labeled from 4 to 39. Since Figure 2 is rather self-explicative, we omit the explicit definition of the orders r_{SC}, with $\mathrm{SC} \in \{\mathrm{ALA}, \mathrm{ASN}, \mathrm{ASP}, \mathrm{CYS}, \mathrm{GLU}, \mathrm{LYS}, \mathrm{SER}\}$, which can be easily derived from the pictures.

V. Computational experiments

We present in this section some computational experiments where we use the side chain orders proposed in Section IV. All codes were written in C programming language and all the experiments were carried out on an Intel Core i7 2.30GHz with 8GB RAM, running Linux. The codes have been compiled by the GNU C compiler v.4.6.1 with the `-O3` flag.

The experiments here presented prove that the information regarding the protein side chains plays an important role in the identification of solutions to iDMDGPs. At this stage of our work, we did not try yet to use real data from NMR (as it was done in [11] for protein backbones), but rather we did study the influence of side chain information on the iBP algorithm, and, in particular, on its pruning phase.

To this aim, we artificially generated a set of instances of the iDMDGP by combining the backbone order in Figure 1 and the 8 side chains orders in Figure 2. The first considered set contains instances formed by 4 amino acids. Given a certain sequence of amino acids, say ALA-GLY-ASN-CYS, we construct two graphs G and G', which represent two instances of the iDMDGP. The graph G represents an instance where the side chains of its amino acids are considered, the graph G' represents the same instance without the side chain information. We will refer to the first class of instances with the symbol $\mathbb{C}_{sidechains}$, and to the second class with $\mathbb{C}_{backbone}$.

Graphs $G \in \mathbb{C}_{sidechains}$ are generated as follows. Recall that the edge set E can be divided in two subsets E_d and E_p (see Section III). Since G contains the information related to the side chains of the amino acids, the vertex set V and the subset of edges E_d are obtained from the orderings (backbone and side chains) presented in Sections III and IV. The subset E_p contains some randomly generated interval distances concerning only pairs of hydrogens in V. In order to simulate NMR data, only distances shorter than 5Å are considered. All distances are represented by an interval of length 1.4Å.

TABLE I
These experiments show the influence of the side chain information on the number of solutions of iDMDGPs.

sequence	$\min(D)$	#Sol for G'	#Sol for G
GLY GLY GLY GLY	4	11424	48
GLY ALA GLY ALA	5	9792	256
ALA ALA ALA ALA	5	11518848	1536
CYS CYS CYS CYS	5	35840	512
CYS GLY GLY CYS	5	3216256	768
GLY ASN GLY GLY	5	320	64
SER SER SER SER	5	35840	512
ALA GLY ASN CYS	6	10240	3200
CYS ASN ALA SER	8	3785264	784
ASN ASN ASN ASN	6	264720	6800
GLY ASP SER GLY	5	165120	128
GLY ASP GLY ALA	4	244128	864
GLY ALA GLU GLY	6	10175840	800
GLY GLY GLY LYS	14	130282650	37856

Graphs $G' \in \mathbb{C}_{backbone}$ are derived from the corresponding graph $G \in \mathbb{C}_{sidechains}$. This is done in order to compare pairs of similar instances, one containing the side chain information, the other without. The vertex set $V' \subset V$ and the subset of edges E'_d are obtained from the special ordering for the backbone atoms in Figure 1. Then, we set $E'_p = \{(u,v) \in E_p : u, v \in V'\}$, so that the pruning distances in G' are exactly the same in G, exception made for the pruning distances related to side chain atoms.

Table I shows some experiments where our instances G and G' are related to sequences that are formed by 4 amino acids. In the table, $\min(D)$ is the smallest D value that guarantees the generation of enough tree branches for obtaining at least one solution to the problem. In general, if $\bar{D} = \min(D)$ for an instance $G \in \mathbb{C}_{sidechains}$, then $\bar{D}$ is also a suitable number of sample distances for the corresponding instance $G' \in \mathbb{C}_{backbone}$, because it contains, by definition, fewer pruning distances. #Sol represents the number of solutions that we obtain for each instance G (containing the side chain information) or G' (containing the backbone information only). All experiments lasted no more than 20 seconds.

The experiments show the importance of side chain information. In all experiments, the number #Sol of solutions is smaller in correspondence with instances that contain side chains. Instances without side chains are hence much more flexible, many more possible conformations are allowed, whereas instances with side chains have a reduced number of possible conformations. The use of side chain information, therefore, improves the pruning capabilities of the iBP algorithm, allowing to focus the search on a smaller number of branches of the tree.

In Table II, we consider another set of instances containing longer sequences of amino acids. In this table, we only consider the instances represented by graphs $G \in \mathbb{C}_{sidechains}$,

TABLE II
SOME EXPERIMENTS ON INSTANCES CORRESPONDING TO LONGER SEQUENCES OF AMINO ACIDS.

n_{aa}	$\|V\|$	$\min(D)$	#Sol for G	CPU time
10	145	7	1	0.36
20	325	8	1	5.42
30	447	8	1	77.37
40	623	10	1	80.63
50	750	8	1	114.22

and, for the minimum $\bar{D} = \min(D)$, we compute only one solution for the instance (#Sol is always 1). For each instance, n_{aa} is the number of amino acids forming the simulated molecule (the sequences of amino acids have been randomly generated, and they only contain the amino acids considered in Section IV), and $|V|$ is the total number of vertices in G (duplicated atoms are counted). We also report the CPU time in seconds.

The experiments in Table II show that side chain information can also be added to discretized iMDGPs related to longer sequences of amino acids. For a sequence containing 50 amino acids, there are 750 vertices in the graph G representing the artificial instance, and the iBP algorithm is still able to find a solution in less than 2 minutes. At this stage of our work, we do not compute yet all solutions for instances in the set, because suitable strategies need to be developed in order to control the combinatorial explosion in iBP trees in the case of long sequences.

VI. CONCLUSIONS

We presented new discretization orders for 8 of the smallest amino acids that can be found in proteins. Together with a discretization order for the protein backbone, these orders allow for discretizing all instances concerning proteins composed by these amino acids. This way, the employment of the iBP algorithm is possible, and all distances estimated through NMR experiments can be exploited for efficiently pruning parts of the iBP search tree. The computational experiments show that the side chain information is able to improve the pruning phase of the iBP algorithm.

Future research will be devoted to the identification of special orders for the other side chains in proteins. Longer side chains, however, imply the following issue: duplicated atoms (i.e. atoms appearing more than once in the order) may be very far from each other in the sequence (in the proposed order for LYS, for example, copies of the C_α atom are quite far but not too much). As a consequence, because of some small error propagation, the coordinates of two copies of the same atom may be different. This phenomenon could bring to incompatibilities between partial coordinates and available distances. For this reason, we are currently studying possible ways to overcome this issue.

Moreover, as it can be noted in the experiments in Section V, the total number of solutions for an instance (even containing side chains) can be large. However, we noticed that some found solutions are very similar to others, so that clusters of solutions may be identified in order to reduce the number of representative solutions that are obtained. Work is currently in progress for improving the iBP algorithm so that it can only output representative solutions.

VII. ACKNOWLEDGMENTS

The authors wish to thank CAPES, that funded a 4-month visit to Rennes for Virginia Costa: this work was mainly performed during such a visit. We are also thankful to the French Embassy in São Paulo, UNICAMP and CNPq.

REFERENCES

[1] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, P. Bourne, *The Protein Data Bank*, Nucleic Acids Research **28**, 235–242, 2000.

[2] G. Crippen and T. Havel, *Distance Geometry and Molecular Conformation*, John Wiley & Sons, New York, 1988.

[3] C. Lavor, L. Liberti, N. Maculan, A. Mucherino, *The Discretizable Molecular Distance Geometry Problem*, Computational Optimization and Applications **52**, 115–146, 2012.

[4] C. Lavor, L. Liberti, N. Maculan, A. Mucherino, *Recent Advances on the Discretizable Molecular Distance Geometry Problem*, European Journal of Operational Research **219**, 698–706, 2012.

[5] C. Lavor, L. Liberti, A. Mucherino, *The interval Branch-and-Prune Algorithm for the Discretizable Molecular Distance Geometry Problem with Inexact Distances*, to appear in Journal of Global Optimization, 2011.

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

[7] L. Liberti, C. Lavor, A. Mucherino, N. Maculan, *Molecular Distance Geometry Methods: from Continuous to Discrete*, International Transactions in Operational Research **18**, 33–51, 2010.

[8] L. Liberti, B. Masson, C. Lavor, A. Mucherino, *Branch-and-Prune Trees with Bounded Width*, Proceedings of the 10^{th} Cologne-Twente Workshop on Graphs and Combinatorial Optimization (CTW11), Rome, Italy, 189–193, 2011.

[9] M. Nilges, A. Gronenborn, A. Brunger, G. Clore, *Determination of Three-Dimensional Structures of Proteins by Simulated Annealing with Interproton Distance Restraints. Application to Crambin, Potato Carboxypeptidase Inhibitor and Barley Serine Proteinase Inhibitor 2*, Protein Engineering **2**, 27–38, 1988.

[10] J. Moré, Z. Wu, *Distance Geometry Optimization for Protein Structures*, Journal of Global Optimization **15**, 219–234, 1999.

[11] A. Mucherino, C. Lavor, T. Malliavin, L. Liberti, M. Nilges, N. Maculan, *Influence of Pruning Devices on the Solution of Molecular Distance Geometry Problems*. In: P.M. Pardalos and S. Rebennack (Eds.), Lecture Notes in Computer Science **6630**, Proceedings of the 10^{th} International Symposium on Experimental Algorithms (SEA11), Crete, Greece, 206–217, 2011.

[12] J. Saxe, *Embeddability of Weighted Graphs in k-Space is Strongly NP-hard*, Proceedings of 17^{th} Allerton Conference in Communications, Control and Computing, 480–489, 1979.