A discrete search algorithm for finding the structure of protein backbones and side chains

Silas Sallaume

Faculdades Thathi, Araçatuba - SP, Brazil
E:mail: silassallaume@yahoo.com.br

Simone de Lima Martins

Departamento de Ciência da Computação, Universidade Federal Fluminense, Niterói - RJ, Brazil
E:mail: simone@ic.uff.br

Luiz Satoru Ochi

Departamento de Ciência da Computação, Universidade Federal Fluminense, Niterói - RJ, Brazil
E:mail: satoru@ic.uff.br

Warley Gramacho da Silva

Campus Universitário de Palmas, Universidade Federal do Tocantins, Palmas - TO, Brazil
E:mail: wgramacho@uft.edu.br

Carlile Lavor

Departamento de Matemática Aplicada, Universidade Estadual de Campinas, Campinas - SP, Brazil
E:mail: clavor@ime.unicamp.br

Leo Liberti

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

Abstract: An important problem in computational physical chemistry is the determination of the three-dimensional structure of proteins. Some information about protein structure can be obtained by using Nuclear Magnetic Resonance (NMR) techniques, but they provide only a sparse set of distances between atoms in a protein. The Molecular Distance Geometry Problem (MDGP) consists in determining the three-dimensional structure of a molecule using a set of known distances between some atoms. Generally, the MDGP is expressed as a continuous

Copyright © 2009 Inderscience Enterprises Ltd.

optimization problem. Recently, a Branch and Prune (BP) algorithm was proposed to calculate the backbone of a protein, based on a discrete formulation for the MDGP. We present an extension of the BP algorithm that can calculate not only the protein backbone, but the whole three-dimensional structure of proteins. Since this new algorithm preserves the combinatorial approach of the BP algorithm, it can potentially find all solutions of the problem (generally, the methods based on the continuous approach obtain just one solution). The proposed algorithm was successfully tested to find all solutions of a commonly used test set of proteins from the Protein Data Bank (PDB).

Keywords: Algorithms; Discretizable molecular distance geometry problem; Protein structure; Computational physical chemistry

Biographical notes: Silas Sallaume received his Master degree in Computer Science in 2009 from the Fluminense Federal University, Rio de Janeiro. Currently, he is an Associate Professor in the Department of Computer Science, Faculdades Thathi-COC, São Paulo. His research interests include combinatorial optimization and operations research.

Simone de Lima Martins received her PhD in Computer Science in 1999 from the Pontifícia Universidade Católica, Rio de Janeiro. Currently, she is an Associate Professor in the Department of Computer Science, Fluminense Federal University, Rio de Janeiro. Her research interests include combinatorial optimization and parallel processing.

Luiz Satoru Ochi received his PhD in System Engineering in 1988 from the Federal University of Rio de Janeiro, Brazil. Currently, he is a Full Professor in the Department of Computer Science, Fluminense Federal University, Rio de Janeiro. His research interests include combinatorial optimization, graphs, heuristics, operations research.

Warley Gramacho da Silva is an Assistant Professor at Federal University of Tocantins. He received his M.Sc. degree in Computer Science from Fluminense Federal University, Brazil, in 2008. He is currently pursuing his D.Sc. degree in Computer Science at Federal University of Rio de Janeiro, Brazil. His research interests include optimization, algorithms and parallel computing.

Carlile Lavor is a mathematician and received his Ph.D. degree in Computer Science at Federal University of Rio de Janeiro, in 2001. In 2006, he obtained his Habilitation in Combinatorics at State University of Campinas, where he is currently an Associate Professor. He was a Visiting Professor at Politecnico di Milano, Universidad Politecnica de Madrid, École Polytechnique de Paris and Institut Pasteur. His main research interests are molecular geometry optimization, bioinformatics, and quantum information.

Leo Liberti is currently an Associate Professor at LIX, École Polytechnique. He obtained his Ph.D. degree in Global Optimization at Imperial College, London, in 2004, and worked as postdoctoral fellow at Politecnico di Milano, Italy, during 2004-2005. His main research interests are reformulations in mathematical programming, global and combinatorial optimization with applications to complex industrial systems and bioinformatics.

1 Introduction

The function of a protein is determined by its chemical and three-dimensional structures (Creighton, 1993). Some information about the protein structure can be obtained by using Nuclear Magnetic Resonance (NMR) techniques, which are able to give a measure of the distance between pairs of atoms that are not greater than 6Å (Schlick, 2002). The problem of finding the atomic positions of a molecule, when only a given subset of atomic distances is known, is called the Molecular Distance Geometry Problem (MDGP). In practice, the MDGP is solved by continuous optimization methods (Lavor, 2007) which are usually capable to obtain just one of all possible solutions for the problem (for a survey on MDGP methods, see Lavor et al. (2009); Liberti et al. (2011)).

In 2006, Lavor et al. (2006) described an MDGP subclass called Discretizable Molecular Distance Geometry Problem (DMDGP), including protein backbone instances whose 3D structure could be computed by a discrete search algorithm called Branch and Prune (BP) algorithm (Liberti et al., 2008). Other algorithms exhibiting some similarities with BP algorithm can be found in Carvalho et al. (2008); Wu et al. (2008).

We present an extension of the BP algorithm that can calculate not only the protein backbone, but the whole three-dimensional structure of proteins. Since this new algorithm preserves the combinatorial approach of the BP algorithm, it can potentially find all solutions of the problem. This is important because we have a list of all mathematical solutions and we can select one or some of them that satisfy additional physical-chemical restrictions depending of the particular protein under study. The proposed algorithm was able to efficiently find all solutions of the problems associated to some of the most common test proteins in the MDGP literature. All optimal configurations of 11 MDGP instances with roughly 200 to 2000 atoms were solved in just 50s of CPU time.

The remaining of the paper is organized as follows. Section 2 explains how the whole protein structure is determined, first defining an ordering on atoms of the side chain that satisfies the DMDGP assumptions (Section 2.1) and then presenting the algorithm that calculates the whole protein structure (Section 2.2). Computational results are presented in Section 3 and Section 4 concludes the paper.

2 Calculating the whole protein structure

Formally, the MDGP can be defined as the problem of finding Cartesian coordinates $x_1, ..., x_n \in \mathbb{R}^3$ of atoms of a molecule such that

$$\| x_i - x_j \| = d_{i,j}, \qquad (i,j) \in S,$$

where S is the set of pairs of atoms (i,j) whose Euclidean distances $d_{i,j}$ are known. If all distances are known, the problem can be solved in linear time (Dong and Wu, 2002). In general, however, the problem is NP-hard (Saxe, 1979).

In Lavor et al. (2006) and Liberti et al. (2008), it was shown that under the following assumptions, the MDGP can be formulated as a combinatorial problem, called DMDGP:

1. all distances $d_{i-3,i}, d_{i-2,i}, d_{i-1,i}$ must be known for $i \in \{4, \ldots, n\}$,

2. the angles defined by each triplet of consecutive atoms cannot be equal to $k\pi$ (for $k \in \mathbb{Z}$),

for a given ordering on the atoms of the molecule.

The geometrical intuition behind the combinatorial formulation is that the i-th atom lies on the intersection of three spheres centered at atoms $i-3$, $i-2$, $i-1$ having respective radii $d_{i-3,i}, d_{i-2,i}$ and $d_{i-1,i}$. By assumptions above, the intersection of the three spheres defines at most two points labeled i and i' in Figure 1. This allows us to express the position of the i-th atom in terms of the preceding three, leading to the definition of a binary tree of possible atomic positions, where solutions to the DMDGP can be searched. Pruning tests are employed in order to discover infeasible atomic positions. As soon as an atomic position is found to be infeasible, then the corresponding branch is pruned and the search is backtracked. The pruning phase usually reduces the tree within manageable sizes, so that an exhaustive search on the remaining branches is not computationally expensive. As shown in Liberti et al. (2008), the instances of DMDGP have a finite number of solutions with probability 1. The BP algorithm proposed in Lavor et al. (2006) and Liberti et al. (2008) can only find the 3D structure of protein backbones. In fact, a protein is composed of a backbone and many side chains, which are always connected to one of the atoms of the backbone and appear systematically on each three atoms of the backbone related to the amino acids that define the protein (Creighton, 1993). When two amino acids bind during protein synthesis, some of the atoms of their common parts are lost, while the carbon atom C of the first amino acid binds to the nitrogen N of the second one. Therefore, the protein backbone is formed by the sequence of atoms $N - C_\alpha - C$. The general structure of an amino acid is shown in Figure 2, where all atoms are shown and the circle marked by R represents the atoms of the side chain.

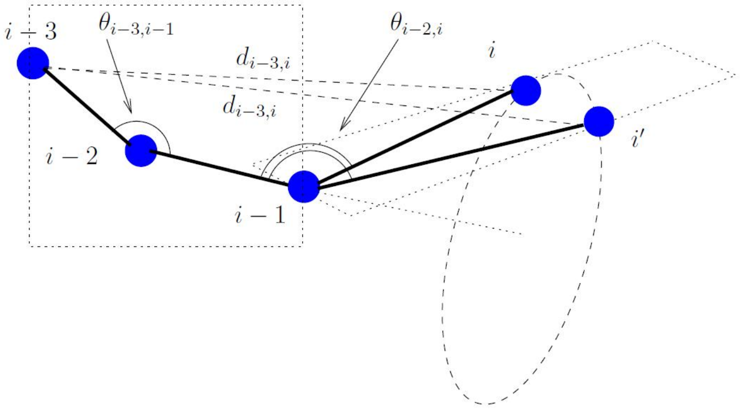

Figure 1 Combinatorial formulation of the MDGP

Figure 3 illustrates a protein backbone with three side chains SC1, SC2 and SC3 (there are 20 different side chains found in all proteins, varying from 1 to 18 atoms (Schlick, 2002)).

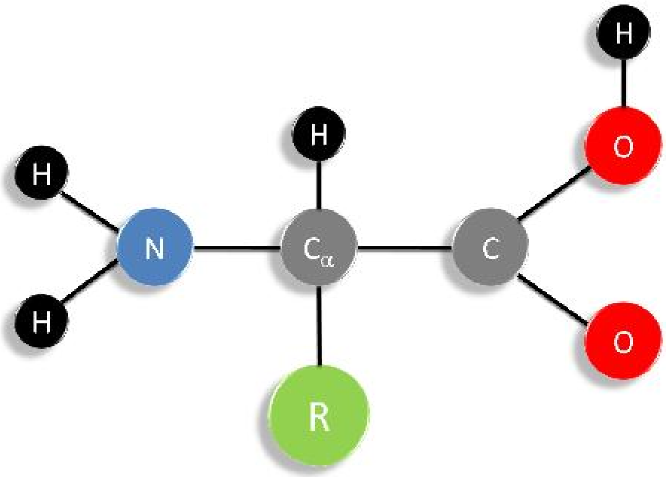

Figure 2 General structure of an amino acid

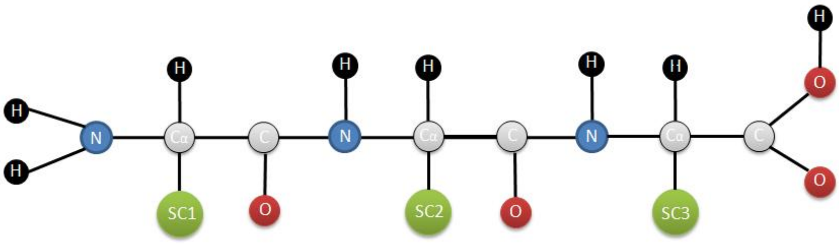

Figure 3 The protein backbone and the side chains SC1, SC2 and SC3

When separated from their backbone, side chains are valid DMDGP instances. Thus, to determine the whole protein structure, several instances of the DMDGP should be solved. The difficulty lies in how to solve these instances in an efficiently and integrated manner.

The algorithm that we developed integrates the calculation of the positions of the atoms belonging to the protein backbone and also the positions of the atoms belonging to the side chains. To consider the atoms of the side chains as instances of the DMDGP, it was necessary to define an ordering on its atoms satisfying the above assumptions 1 and 2.

2.1 Defining an ordering on atoms of the side chains

Finding an atomic ordering on the side chains that satisfies the DMDGP assumptions involves solving the *Discretization Vertex Order Problem* (DVOP), defined as follows (Lavor et al., 2010):

Definition 1 *Given a simple undirected graph $G = (V, E)$ and a positive integer K, establish whether there is an order $<$ on V such that:*

1. *$\{v \in V \mid \rho(v) \leq K\}$ is a K-clique in G,*
2. *for each $v \in V$ with $\rho(v) > K$, we have $|\delta(v) \cap \gamma(v)| \geq K$,*

where $\gamma(v) = \{u \in V \mid u < v\}$ is the set of predecessors of v, $\rho(v) = |\gamma(v)| + 1$ is the rank of the v-th element, and $\delta(v) = \{u \in V \mid \{u, v\} \in E\}$ is the star around v, for all $v \in V$.

The propositions below were proved in Lavor et al. (2010).

Proposition 1 *DVOP is an NP-complete problem.*

Proposition 2 *DVOP with fixed K is polynomially solvable.*

An immediate corollary follows from the above results:

Corollary 1 *The problem of defining an ordering on atoms of the side chains of a protein in order to satisfy the assumptions of the DMDGP is a DVOP with $K = 3$.*

Algorithm 1 presents a polynomial algorithm for ordering the atoms of the side chains. It takes as input the side-chain graph G and returns the order rank ρ if the atoms ordering can be determined, or NULL otherwise.

2.2 The algorithm

The BP algorithm developed in Liberti et al. (2008) for finding the 3D structure of protein backbones is very simple and follows the structure of the problem structure closely. At each step the ith atom can be placed in two possible positions x_i and x'_i. So the search is branched in two branches, one considering position x_i and the other considering position x'_i. The feasibility of each solution is verified by checking if the distances between the calculated position for the ith atom to the previous positions already calculated for the previous atoms of the backbone are equal to the known distances. If the position is feasible, the search of the tree continues in the same way, otherwise the branch is pruned. If neither position is feasible, both branches are pruned and the search is backtracked.

The side chains appear on each three atoms of the protein backbone and can be considered as instances of the DMDGP. So we have two approaches to determine the whole protein structure. The first one is to find the positions of all backbone atoms and of all side chain atoms separately and then integrate all structures based on the known distances. The second one is to determine the positions of the backbone atoms and the side chain atoms in an integrated way. We developed an algorithm based on the second approach.

Algorithm 1 Order_Atoms_Side_Chains (G)

```
for ($\{u_1, u_2, u_3\} \subseteq V$) do
  $\rho(u_1) \leftarrow 1$, $\rho(u_2) \leftarrow 2$, $\rho(u_3) \leftarrow 3$;
  $U \leftarrow V \setminus \{u_1, u_2, u_3\}$;
  while ($U \neq \emptyset$) do
    $w \leftarrow \text{argmax}\{|\delta(v) \cap \gamma(v)| \mid v \in U\}$;
    if ($|\delta(w) \cap \gamma(w)| < 3$) then
      break;
    else
      $\rho(w) \leftarrow |V| - |U| + 1$;
      $U \leftarrow U \setminus \{w\}$;
    end if
  end while
  if ($U = \emptyset$) then
    {instance is YES}
    return $\rho$;
  end if
end for
{instance is NO}
return NULL.
```

Figure 4 illustrates the idea of the algorithm. The numbered vertices correspond to the possible positions of the protein backbone atoms and the subtrees at each third atom represent the possible side chains. The algorithm starts by fixing the position of the first three atoms of the protein backbone formed by the sequence $(H, N, C_\alpha, C, N, C_\alpha, C, ..., N, C_\alpha, C)$. At atom 3, there is a side chain. So, a DMDGP is solved considering the chain formed by the first three atoms of the backbone and the atoms of the side chain, considering an ordering on its atoms previously determined by the Algorithm 1 described in the previous section. The known distances between the backbone atoms and the side chain atoms are used to eliminate some positions which are infeasible. Then, DMDGPs are solved to find positions for the atoms 4, 5 and 6 of the backbone and some of them are eliminated according to the known distances. At the backbone atom 6, another DMDGP is solved to find atom positions for the side chain connected to the backbone atom 6, considering the chain formed by backbone atoms 4, 5 and 6 and the side chain atoms. This procedure follows in the same way until the positions for all atoms of the protein backbone and for all side chain atoms are determined.

Let n be the total number of backbone atoms, F be the set of all known distances and SC be the set of side chains ordered by Algorithm 1.

Let T be a graph representation of the search tree, which is initialized with the first three backbone atoms and v be a node with rank $i - 1$ in T.

The detailed steps are shown in Algorithm 2. In line 2, the two possible placements of the ith backbone atom x_i and x'_i are calculated. If the distances of the position x_i to the positions already determined for some of the backbone atoms and for the atoms of at least one of the side chains are not compatible with the known distances, the tree is pruned in line 11. Otherwise, if the ith atom is a multiple of 3, in line 5, the procedure to determine the side chain atom positions is executed.

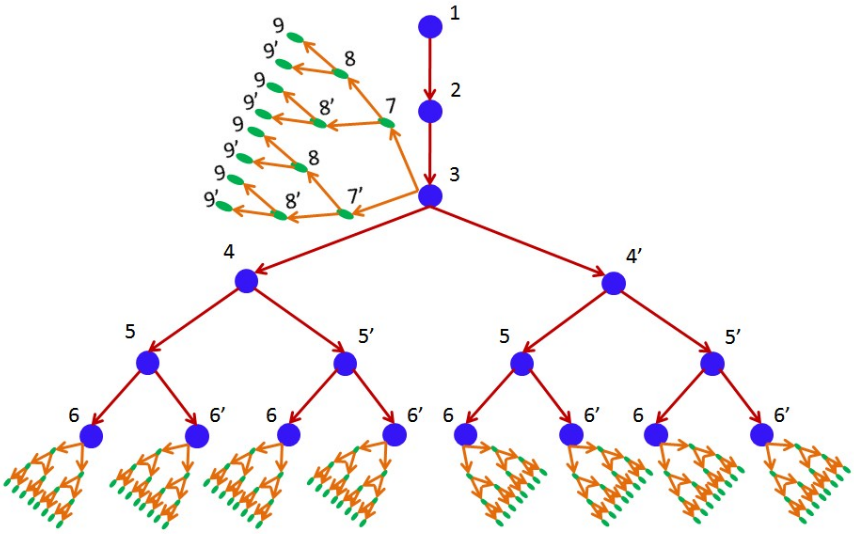

Figure 4 Search tree of Algorithm 2

In line 6, the feasibility of the calculated side chain positions is checked against the known distances, and the unfeasible side chains are pruned. Then a left subnode is created in line 8 related to position x_i and the whole procedure is called again in line 9. From lines 13 to 22 the same process is executed for position x'_i. When the depth search is completed, in line 24, all discovered side chains are checked with each other and with the backbone to verify if the positions of the side chains obey the known distances. If they do not match the known distances, they are discarded.

3 Computational results

The real instances generated for the MDGP were extracted from the structures contained in the Protein Data Bank (PDB) (Berman et al., 2000) and were selected from the set of proteins used by Biswas et al. (2008), Carvalho et al. (2008) and Wu et al. (2007) that use different approaches to solve the same problem. The instances were generated by calculating the distances among all the atoms of a protein and discarding the distances that are above a cutoff value, which is set to a value detectable by NMR techniques.

The code was written in C++ programming language by using the Standard Template Library and compiled by the Visual C++ 2005. All experiments were carried out on an Intel Core 2, 1.6 GHz and 2GB RAM, running Windows XP with Service Pack 2.

Different metrics to measure the quality of solutions and different cutoff values are used in the generation of instances for the MDGP. All these factors hinder a fair comparison between the methods. Thus, for testing the proposed algorithm, we used some instances commonly found in the literature (representing different

Algorithm 2 Whole_Protein_Algorithm(T, v, i)

```
if ($i \leq n - 1$) then
  ith_Positions ← Compute_Possible_Positions_ith_Atom;
  if Feasible_Position_Left(ith_Positions, F) then
    if ($i$ mod $3 = 0$) then
      Obtain_Feasible_SideChains(SC);
      Check_Feasibility_SideChains(F);
    end if
    create a left node $z$;
    Whole_Protein_Algorithm ($T, z, i + 1$);
  else
    Prune_Left(T);
  end if
  if Feasible_Position_Right(ith_Positions, F) then
    if ($i$ mod $3 = 0$) then
      Obtain_Feasible_SideChains(SC);
      Check_Feasibility_SideChains(F);
    end if
    create a right node $z$;
    Whole_Protein_Algorithm ($T, z, i + 1$);
  else
    Prune_Right(T);
  end if
else
  if Solution_is_Feasible(T) then
    Store_Feasible_Backbone_SideChains(T, SC, F);
  end if
end if
```

sizes of proteins) and the Largest Distance Error (LDE) as a measure of solution accuracy (Liberti et al., 2011), defined by

$$LDE = \frac{1}{|S|} \sum_{(i,j) \in S} \frac{| \|x_i - x_j\| - d_{ij} |}{d_{ij}},$$

where S is the set of pairs of atoms (i,j) whose Euclidean distances d_{ij} are known and x_i, x_j are the Cartesian coordinates of atoms (i,j), respectively. The cutoff value for generating the instances was fixed in 6Å, which is the maximum value allowed to simulate data obtained through NMR (Schlick, 2002).

Table 1 presents the obtained results. The column #Atoms indicates the number of atoms of the protein, the column #Sol shows the amount of found solutions, the column PDB indicates which of the found solutions has the greatest degree of similarity with the protein obtained from the PDB (using the RMSD method (Wu et al., 2008)), the column LDE shows the LDE value obtained for the best found solution, and the column Time-W shows the computational time, in seconds, took by the method for finding all solutions.

The last column Time-B shows the computational time, in seconds, took by the algorithm used to find the solutions only for the backbone. We can observe that, sometimes, the procedure to find the solutions for the backbone demanded more computational effort. The algorithm to find the whole protein structure considers also the side chains of a protein, so there may be more distances available to prune the tree of the BP algorithm, which, in general, may reduce the total computational cost of the algorithm.

Protein	*#Atoms*	*#Sol*	*PDB*	LDE	*Time-W*	*Time-B*
1brv	261	2	2	1.95e-17	0.5780	0.00
1ptq	402	8	2	3.91e-15	1.1400	0.05
1aqr	524	2	2	5.25e-17	1.9210	1.03
1hoe	558	4	2	6.93-e17	1.3590	0.01
1lfb	641	4	2	5.87e-17	1.5310	0.02
1ahl	684	2	1	3.97e-17	4.0000	57.62
1pht	811	8	2	6.41e-17	4.0930	0.11
1brz	859	2	1	5.29e-17	6.1710	28.83
1poa	914	32	9	7.50e-17	5.4210	0.03
1acz	1613	8	2	4.71e-17	17.078	2484.24
1rgs	2015	4	1	1.58e-16	5.3590	110.42

Table 1 Solutions obtained for 3D structure of whole proteins

The computational results show the very good performance of the algorithm and the high quality of the solution associated to the greatest degree of similarity with the protein obtained from the PDB. In fact, all found solutions presented LDE values very small, indicating that all solutions are global minimizers of the LDE function. As it was said, one of the main advantages of the combinatorial approach is the possibility to obtain all the solutions of the problem.

4 Conclusions

We presented a method to calculate the three-dimensional structure of a protein, using a set of known distances between some atoms of the protein. It was based on the BP algorithm, which can calculate just the protein backbone. Now, it is possible to calculate the whole protein structure, including the side chains.

Since this new algorithm preserves the combinatorial structure of the BP algorithm, it was able to find all solutions of the selected problems (generally, the methods based on the continuous approach obtain just one solution). Furthermore, the computational times for executing the algorithm were quite small and the quality of the generated solutions was very high.

Acknowledgments: The authors would like to thank the Brazilian agencies FAPESP, FAPERJ, and CNPq for their financial support.

References

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., and Bourne, P.E., The Protein Data Bank, Nucleic Acids Research 28 (2000), 235-242.

Biswas, P., Toh, K.C., Ye, Y., A distributed SDP approach for large-scale noisy anchor-free graph realization with applications to molecular conformation, SIAM Journal on Scientific Computing 30 (2008), 1251-1277.

Carvalho, R.S., Lavor, C., and Protti, F., Extending the geometric buildup algorithm for the molecular distance geometry problem, Information Processing Letters 108 (2008), 234-237.

Creighton, T., Proteins: Structures and Molecular Properties (2nd edition), W.H. Freeman, New York, 1993.

Dong, Q., and Wu, Z., A linear-time algorithm for solving the molecular distance geometry problem with exact interatomic distances, Journal of Global Optimization 22 (2002), 365-375.

Lavor, C., Liberti, L., and Maculan, N. The discretizable molecular distance geometry problem, 2006, arXiv:q-bio/0608012v1.

Lavor, C., Analytic evaluation of the gradient and Hessian of molecular potential energy functions, Physica D 227 (2007), 135-141.

Lavor, C., Liberti, L., and Maculan, N., Molecular distance geometry problem, Encyclopedia of Optimization (2nd edition), Springer, New York, 2305-2311, 2009.

Lavor, C., Lee, J., John, Audrey Lee-St., Liberti, L., Mucherino, A., and Sviridenko, M., Discretization orders for distance geometry problems, 2010, Optimization Letters, accepted for publication.

Liberti, L., Lavor, C., and Maculan, N., A branch-and-prune algorithm for the molecular distance geometry problem, International Transactions in Operational Research 15 (2008), 1-17.

Liberti, L., Lavor, C., and Maculan, N., Molecular distance geometry methods: from continuous to discrete, International Transactions in Operational Research 18 (2011), 33-51.

Saxe, J.B., Embeddability of weighted graphs in k-space is strongly NP-hard, Proceedings of 17th Allerton Conference in Communications, Control and Computing, Monticello, IL, 480–489, 1979.

Schlick, T., Molecular Modeling and Simulation: An Interdisciplinary Guide, Springer, New York, 2002.

Wu, D., and Wu, Z., An updated geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data, Journal of Global Optimization 37 (2007), 661-673.

Wu, D., Wu, Z., Yuan, Y., Rigid versus unique determination of protein structures with geometric buildup, Optimization Letters 2 (2008), 319-331.