Proceedings of the 2013 Federated Conference on
Computer Science and Information Systems pp. 341–346

Energy-based Pruning Devices for the BP Algorithm applied to Distance Geometry

Douglas Gonçalves*, Antonio Mucherino*, Carlile Lavor†

*IRISA, University of Rennes 1, Rennes, France.
{douglas.goncalves,antonio.mucherino}@irisa.fr

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

***Abstract*—The Molecular Distance Geometry Problem (MDGP) is the one of finding an embedding of a molecular graph in the three dimensional space, where graph vertices represent atoms and edges represent known distances between some pairs of atoms. The MDGP is a constraint satisfaction problem and it is generally cast as a continuous global optimization problem. Moreover, under some assumptions, this optimization problem can be discretized and so that it becomes combinatorial, and it can be solved by a Branch & Prune (BP) algorithm. The solution set found by BP, however, can be very large for some instances, while only the most energetically stable conformations are of interest. In this work, we propose and integrate the BP algorithm with two new energy-based pruning devices. Computational experiments show that the newly added pruning devices are able to improve the performance of the BP algorithm, as well as the quality (in terms of energy) of the conformations in the solution set.**

I. Introduction

THE Molecular Distance Geometry Problem (MDGP) is the one of finding the possible three-dimensional conformations of a molecule from the information about the relative distances between some pairs of its atoms [1], [2]. Let $G = (V, E, d)$ be a weighted undirected graph representing an instance of the MDGP. The vertex set describes the atoms forming the molecule, and there is an edge joining two vertices if and only if the distance between the two corresponding atoms is known. The MDGP can be seen therefore as the problem of finding a function $x : V \longrightarrow \Re^3$ such that

$$\forall (u, v) \in E, \qquad ||x(u) - x(v)|| = d_{uv},$$

where $||\cdot||$ represents the computed Euclidean distance between the coordinates $x(u)$ and $x(v)$, whereas d_{uv} is the weight of the edge (u, v). The MDGP is NP-hard [3].

By its nature, the MDGP is a constraint satisfaction problem that is generally formulated as a global optimization problem in a continuous space [4]. When some particular assumptions are satisfied, moreover, the search space of the optimization problem can be discretized, so that it becomes combinatorial. We refer to a subclass of MDGPs that can be discretized as the Discretizable MDGP (DMDGP) [5], [6], [7]. Instances of the DMDGP can be solved by applying an ad-hoc Branch & Prune (BP) algorithm [8].

The basic idea behind BP is as follows. Suppose that possible positions have already been computed for all the atoms of a given molecule that have rank smaller than i (we suppose that a total order relationship for the vertices of G exists). Because of the discretization assumptions, there are up to two possible positions for the current atom i, that can be obtained by intersecting three spheres centered in the already placed atoms $i-3$, $i-2$ and $i-1$, and having radii $d_{i-3,i}$, $d_{i-2,i}$ and $d_{i-1,i}$ respectively. In this way, a binary tree can be defined, which is the conformational search space of the discretized problem, where branches duplicate in number when passing on higher level layers. By using some additional information about the distances (that are not considered in the tree construction), the feasibility of the atomic positions can be verified, so that branches of the tree can be pruned in case they contain infeasible positions. Moreover, additional pruning devices can be conceived for improving the performance of the BP algorithm [9], [10].

In this paper, we propose two new pruning devices to be included in the BP algorithm in order to improve its performance. For the first time, we consider pruning devices that are based on the chemical nature of MDGP instances; in other words, in these new pruning devices, we exploit the chemical structure of the molecule, and not only the distance information. We will present two new pruning devices. The former basically considers the van der Waals (vdW) radii [11] and forbids any configuration where non-bonded atoms are too close to each other, by verifying the relative distances between spheres centered in the atoms and having as radii the corresponding vdW radii. The latter pruning device is instead based on the well-known Lennard Jones (LJ) potential [12], which is related to the internal energy of the molecule.

The rest of the paper is organized as follows. In Section II, we will briefly describe the BP algorithm and we will focus our attention on the symmetry properties of BP trees, that will be exploited later in the paper. Section III will introduce the two new pruning devices, while computational experiments will be presented in Section IV. Conclusions and future works will be discussed in Section V.

978-1-4673-4471-5/$25.00

Algorithm 1 The BP algorithm.

```
1: BP(v, n, d)
2: compute x'_v;
3: if (x'_v is feasible) then
4:    if (v = n) then
5:       let nsols = nsols + 1;
6:    else
7:       BP(v + 1, n, d);
8:    end if
9: end if
10: compute x''_v;
11: if (x''_v is feasible) then
12:    if (v = n) then
13:       let nsols = nsols + 1;
14:    else
15:       BP(v + 1, n, d);
16:    end if
17: end if
```

II. The BP algorithm

The BP algorithm [8] is an exact algorithm for the solution of DMDGPs. It explores recursively the discrete search domain (a binary tree) of the DMDGP and it prunes infeasible branches of such a tree as soon as they are discovered. Alg. 1 gives a sketch of the algorithm. In the algorithm call, $v \in V$ is the current vertex for which we are looking for a position, n is the cardinality of V, and d represents the weights associated to the edges. During each call, the two possible positions for the vertex v, x'_v and x''_v, are computed, and their feasibility is verified. If, for example, x'_v is feasible, then this position could be part of a solution, and therefore the branch of the binary tree rooted at x'_v needs to be explored. In this case, the algorithm invokes itself for exploring the branch rooted at x'_v. Instead, if for example x''_v is not feasible, then the current branch does not contain any solution. It is therefore pruned: the algorithm does not invoke itself in this case.

The conditions in the two **if** control structures (see lines 3 and 11 of Alg. 1) can be verified by employing the so-called *pruning devices*. The easiest to conceive and to implement (and probably the most efficient) is the Direct Distance Feasibility (DDF) pruning device. DDF simply verifies whether additional distances (that are not employed in the computation of x'_v and x''_v) are satisfied by the obtained candidate positions. In the following, we will refer to such additional distances as *pruning distances*. The BP algorithm, together with the DDF pruning device, was shown to be very efficient for the solution of protein-like instances [5].

Fig. 1 shows a BP tree for a small instance and the set of solutions obtained when only the DDF pruning device is employed. In the following, we will say that this is the DDF solution set, in order to make a distinction between this solution set and the ones that we will obtain while employing the new pruning devices. In the tree representation, a solution is given by a path from the tree root to one of its leaf nodes.

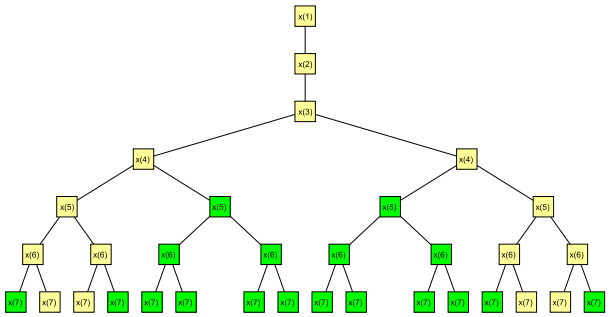

Fig. 1. The BP tree for a 7-atom instance, and the DDF solution set (in yellow).

As it is graphically shown in the picture, DDF solution sets are symmetric, i.e. one solution can be obtained from another by replacing a branch with its symmetric one [13], [14].

On the other hand, BP trees are highly symmetric: when no additional distances are available for applying the DDF pruning device, then every tree branch has a symmetric branch. A pair of symmetric branches shares the following property: given two vertices v and w, in the two branches, even if their coordinates are different, their relative distance is the same. Conversely, if we consider a vertex u having rank smaller than the common root of the two symmetric branches, its distance from any vertex v on the two branches is branch-dependent.

Let us consider a pair of symmetric branches of the BP tree that are rooted at the vertex v. If there is a vertex with rank $u < v$ and another vertex $w > v$ (which belongs to the two branches) such that the pruning distance d_{uw} is available, then, with probability 1, at most one of the two coordinates for w (one on the first branch, another on the second one) can satisfy this distance [15]. As a consequence, this kind of pruning distances are able to break the symmetries in BP trees, so that only a subset of symmetries is left in the DDF solution set. An easy test for verifying the presence of a symmetry in the solution set (before its explicit computation) is to check the existence of this kind of pruning distances. As reported in [13], there is a symmetry on the layer v of the solution set if and only if the vertex v belongs to the set

$$B = \{v \in V : \nexists (u, w) \text{ s.t. } u + 3 < v \leq w\}.$$

In this work, we will exploit these symmetry properties for generating instances having a predefined minimum number of solutions, and we will also discuss a possible improvement for the LJ pruning device which exploits the symmetries in BP trees.

III. Energy-based pruning devices

In this section, we propose two new pruning devices to be included in the BP algorithm. The first one is based on vdW radii (see Section III-A), while the second one is based on LJ energy (see Section III-B).

A. vdW radii pruning device

In this work, we consider the very common representation for an atom through the coordinates of its center. An atom, however, fills a certain portion of space: its nucleus, consisting of neutrons and protons, has a predetermined volume, while electrons orbit around this nucleus, at a given distance from it. Therefore, an atom can be seen as a sphere centered in its nucleus (which corresponds to the center of the atom) and having radius equal to the distance between the nucleus and the orbiting electrons. This distance can be estimated for each kind of atoms, and it is generally referred to as *atomic radius*. It is common to say that these electrons form a sort of *cloud*.

When two atoms are chemically bonded, their clouds of electrons tend to overlap. If they are not bonded, however, repulsion forces do not allow them to be too close to each other. The half of the distance (between atoms of the same kind) for which the attraction and repulsion forces are in equilibrium is called *van der Waals* (vdW) radius [11]. The vdW pruning device is therefore based on this simple idea: when two atoms are not bonded, their relative distance should be greater than the sum of the two corresponding vdW radii. This verification can be applied to all pairs of atoms for which no pruning distance is available.

Notice that, differently from the DDF pruning device, a precise distance is not available but rather only a lower bound for this distance. When the relative distance between two vertices u and v, with $u < v$ just positioned somewhere, goes below the predefined threshold, then the candidate position for v can be pruned. In the practice, we consider a relaxed condition, by setting the threshold to the 80% of the sum of the vdW radii. In the experiments presented in Section IV, we will consider instances containing Carbons (C) and Nitrogens (N) only. The vdW radius for C is set to 1.875; the vdW radius for N is set to 1.688. These values were extracted from the default parameters of the force field described in [16].

B. LJ pruning device

This pruning device is based on the overall internal energy of a molecule. As it is well-known, an accurate description of all interactions among the atoms in a molecule can be very complex, so that the overall energy can be only approximated by taking into consideration the most important interactions.

The vdW interactions between pairs of non-bonded atoms [17] are usually modeled by the Lennard Jones energy. In this case, we consider both repulsion and attraction forces, and for modeling the overall energy, we use the sum of pairwise LJ potentials 12–6 [12] :

$$E_{LJ} = \sum_{uv} 4\varepsilon_{uv} \left[\left(\frac{\sigma_{uv}}{d_{uv}} \right)^{12} - \left(\frac{\sigma_{uv}}{d_{uv}} \right)^{6} \right], \qquad (1)$$

where ε_{uv} and σ_{uv} are two parameters that can be defined by the relationships between the pairs of atoms u and v. The parameter σ_{uv} is the distance where the pair potential is zero, whereas ε_{uv} is the well depth. The minimum value for the LJ pair potential is $-\varepsilon_{uv}$ achieved in $r_{uv} = 2^{1/6}\,\sigma_{uv}$ (which corresponds to the sum of vdW radii).

During the execution of the algorithm, every time a leaf node is reached (on the layer n), a complete conformation is found, and its energy E_n can be computed. Let us suppose that $\hat{E}_n$ is the lowest energy found so far. The basic idea behind the LJ pruning device is to verify in advance whether new branches of the tree can actually contain conformations with an energy that can be potentially smaller than $\hat{E}_n$. This can be done by computing a lower bound on the energy concerning all the conformations belonging to a common branch.

Depending on the range in which the inter-atomic distances can vary, however, we can compute an accurate lower bound for the actual value. In case the BP algorithm is currently positioned on the layer v, then we have a partial energy value $E_{n(\leq v)}$ (computed by using the available coordinates) and a lower bound $L_{(>v)}$ on the energy $E_{n(>v)}$ (approximated by summing the minimum values given by the Lennard Jones terms for which no distance is available yet). Therefore, if $E_{n(\leq v)} + L_{(>v)} > \hat{E}_n$, there is no hope to identify a conformation with an energy smaller than $\hat{E}_n$ while exploring the current branch of the tree. This branch can be therefore pruned.

Notice that the LJ pruning device considers implicitly the vdW pruning device (see previous section). If a distance between a pair of atomic coordinates (for the atom v and a previous one) is small enough for the atomic position x_v to be pruned by the vdW pruning device, then the corresponding LJ potential is large, so that the LJ pruning device declares the atomic position infeasible as well. Therefore, when the vdW and LJ pruning devices work together, it is appropriate to apply LJ only after vdW.

IV. Computational experiments

This section presents some computational experiments where the BP algorithm is integrated with the new proposed pruning devices. All codes were written in C programming language and all the experiments were carried out on an Intel Core 2 Duo @ 2.4 GHz with 2GB RAM, running Mac OS X. The codes have been compiled by the GNU C compiler v.4.0.1 with the `-O3` flag.

In this paper, we suppose that all considered instances consist of a list of precise distances between some pairs of atoms of the molecule. This is an unrealistic assumption [18], because this information can be obtained through experiments of Nuclear Magnetic Resonance (NMR), where lower and upper bounds on the distances are actually provided. This assumption, however, allowed us to begin the investigation of new ideas that are potentially able to help in the solution of real instances of the problem (see Section V).

The instances considered in this paper are artificially generated by using the following procedure. Protein conformations are downloaded from the Protein Data Bank (PDB) [19] and the backbone atoms N–C_α–C of such proteins are extracted from such conformations. Distances are then computed between each possible pair of atoms, and the distances that are greater than 5Å are rejected (this is done because NMR data only consists of short range distances). As previously remarked

TABLE I
SOME COMPUTATIONAL EXPERIMENTS WITH BP AND THE NEW ENERGY-BASED PRUNING DEVICES.

Instance			only DDF			DDF + vdW			DDF + LJ			DDF + vdW + LJ		
name	n	m	BP calls	#L	time	BP calls	#L	time	BP calls	#L	time	BP calls	#L	time
1mbn-0	459	3200	1951	8	**0.03**	1951	8	0.05	1951	3	**0.03**	1951	3	0.05
1mbn-2	459	3169	11327	512	0.35	4071	96	0.16	9229	5	0.27	4071	4	**0.15**
1rgs-0	792	4936	10883	8	0.43	9059	8	0.51	9127	5	**0.35**	9059	5	0.51
1rgs-1	792	4857	165317	128	6.82	108033	96	6.27	123025	13	**4.90**	108033	11	6.29
1bpm-1	1443	9056	73147	128	6.88	14775	20	**1.89**	36599	7	3.26	14775	4	**1.89**
1bpm-2	1443	9027	1150409	2048	111	90143	108	**11.9**	272474	9	24.9	90143	6	**11.9**
1n4w-1	1610	10860	51521	32	3.39	21785	6	**1.52**	32037	7	1.79	21785	3	1.53
1n4w-2	1610	10675	182433	512	18.75	26855	24	**2.42**	52962	17	4.24	26855	4	2.43
1mqq-1	2032	12820	54175	128	8.62	17347	32	4.64	22067	4	**3.22**	17347	4	4.65
1mqq-2	2032	12807	812927	2048	144	161265	344	49.80	256059	6	**42.25**	161625	4	49.84
1rwh-1	2265	13908	38261	32	5.39	9727	2	**1.43**	17101	4	2.06	9227	2	**1.43**
1rwh-2	2265	13868	1152495	1024	169	35649	8	**5.43**	77613	9	9.32	35649	4	5.45
2e7z-1	2907	27509	342675	64	38.63	229483	8	30.91	256046	7	**25.44**	229483	3	30.89
2e7z-2	2907	27157	2041939	2048	391	271435	16	**42.61**	470990	21	68.11	271435	5	42.63
1epw-0	3861	35028	11975	2	3.12	11299	2	5.67	11489	2	**2.88**	11299	2	5.72
1epw-1	3861	34707	123561	32	43.27	22457	4	12.72	38580	4	**11.16**	22457	2	12.74
1epw-2	3861	34052	2815081	4096	1282	48297	32	**36.79**	192497	23	81.09	48297	3	36.81

in [5], the obtained set of distances forms a DMDGP instance in the majority of the cases (this is always the case for the proteins considered in this paper).

Successively, for all generated instances, a certain number of pruning distances is discarded. Let $\hat{V} \subset V$, containing K randomly selected vertices. For each $v \in \hat{V}$, all the pruning distances d_{uw} such that $u + 3 < v < w$ are removed from the instance. In this way, a symmetry in the DDF solution set is generated, and this makes the total number of solutions increase (see Section II). An instance generated by using this procedure has at least 2^K solutions [15], [13].

Table I shows some computational experiments. The name of each instance is composed by its label on the PDB, plus a number, representing the cardinality K of $\hat{V}$. For every instance, we also provide the number of atoms ($n = |V|$), and the total number of available distances ($m = |E|$).

The experiments are performed for different setups of the BP algorithm. We consider the four following setups:

- only the DDF pruning device is exploited;
- DDF is integrated with the vdW pruning device;
- DDF is integrated with the LJ pruning device;
- the three pruning devices are considered together.

For every setup and for every instance, we monitor the total number of BP calls necessary for enumerating the whole solution set, the number #L of times that BP reaches a leaf node of the search tree (#L corresponds to the cardinality of the solution set when only DDF is employed), and finally the CPU time, in seconds. CPU times in bold are used to mark the fastest executions.

The quality of the obtained solutions can be evaluated in two ways. In order to verify whether all available distances are satisfied, we use the Largest Distance Error (LDE) function:

$$LDE(x) = \sum_{(u,v)\in E} \frac{1}{m} \frac{|\,||x(u) - x(v)|| - d_{uv}\,|}{d_{uv}}.$$

In this work, moreover, it is of interest to verify the LJ energy of the obtained solutions, by employing Equation (1). In the presented experiments, the LJ energy is computed even when the LJ pruning device is disabled. The partial energy $E_{(>k)}$ is computed at each recursive call of BP, because we noticed empirically that this is more efficient than computing the LJ energy for the whole DDF solution set after the execution of the algorithm. For lack of space in the table, LDE and LJ values are omitted for all experiments, but some examples will be given in the text.

The experiments show the effectiveness of the two new pruning devices. Even if very simple, the vdW pruning device is able to reduce the total number of BP calls, as well as the computational time, because it can identify infeasible branches that were not selected by DDF. A similar observation can be done for the LJ pruning device. It is interesting to remark that, when the two new pruning devices are considered together with DDF, the performance of BP is similar to the case in which only the vdW pruning device is considered. As a consequence, the repulsive term in LJ, which dominates the potential for small distances (also considered in vdW), plays the most important role during the pruning process. However, notice that #L decreases when vdW and LJ work together: fewer leaf nodes are reached during the execution of BP.

Fig. 2 shows two solutions for the instance 1mbn-2. The leftmost conformation has minimum LDE value 1.79e-10 (energy -131.21), but it does not correspond to the solution having the smallest energy value. In fact, as it can be seen from the figure, one helix in this conformation slightly diverges from the rest of the molecule, so that some energetic terms increase in value. The rightmost conformation is the one with minimum LJ potential energy: -136.12, while the LDE value is 5.36e-08.

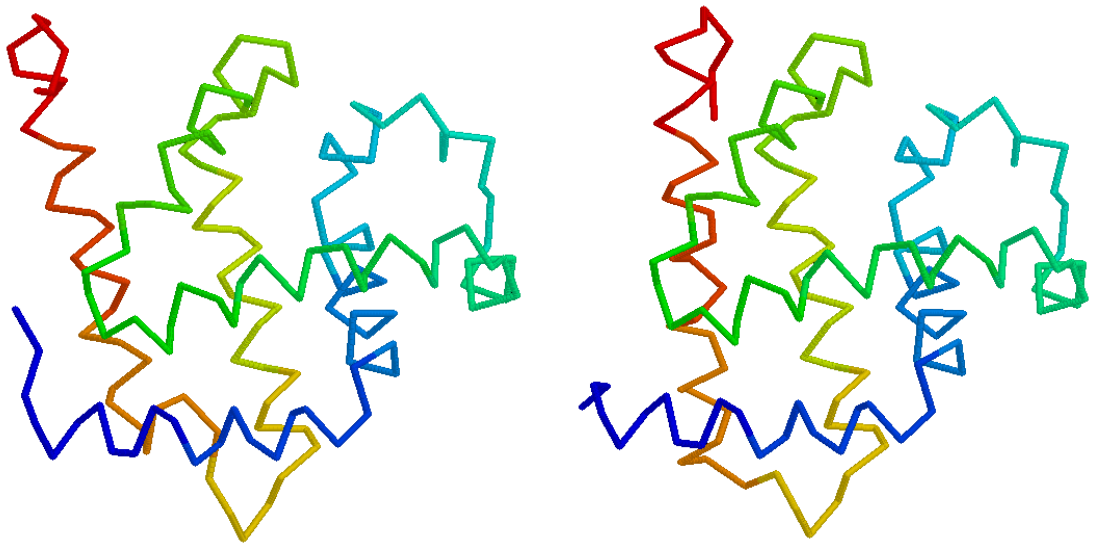

Fig. 2. Two solutions for the instance `1mbn-2`. The leftmost conformation has minimum LDE value, while the rightmost conformation has minimum LJ energy.

V. Conclusions and future works

We introduced two new pruning devices that are based on internal energy in molecules. We showed, through computational experiments, that they are able to improve the pruning capabilities of BP algorithm. The energetically most stable conformations, that previously could be selected from the entire DDF solution set, can be actually obtained by employing these two new pruning devices. The solution set is therefore reduced to energetically stable conformations only, while the performance of BP improves.

This work represents the first step for the integration of energy-based pruning devices inside BP. Next step will consist in testing such new pruning devices on instances of the problem containing interval data. As mentioned above, this kind of instances is more realistic, because, in biological applications, experimental data are generally imprecise. While the adaptation of these pruning devices to interval data is rather trivial, their impact on the performance of the algorithm is expected to be much more pronounced.

When working with interval data, the number of branches in BP trees increases because, instead of 2 possible positions for the current atom, we have $2D$ possible positions, where D corresponds to the number of samples in the discretized interval distance [20]. Thus, it is also expected an increase in the computational cost for the execution of these new pruning devices. For this reason, it will be worth implementing suitable strategies for making the new pruning devices more efficient. In the LJ pruning device, for example, the quality of the lower bound $L_{(>v)}$ can be improved by taking into consideration the symmetry properties of BP trees. Distances between pairs of atoms belonging to two symmetric branches coincide, and therefore their LJ energy is the same. Once one of the two branches has been explored, and its total energy has been computed, it is known in advance that the second branch has the same energy. This property can therefore help in finding better approximations of the lower bound $L_{(>v)}$. Unfortunately, when using this strategy in BP with exact distances, the trade-off between increased cost and performance improvement is not relevant.

VI. Acknowledgments

We wish to thank Thérèse Malliavin for the fruitful comments on this paper. We are also thankful to Brittany Region (France), for funding a 1-year postdoc for DG, and to UNICAMP, CNPq, FAPESP and INRIA, for partial financial support.

References

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

[2] A. Mucherino, C. Lavor, L. Liberti, and N. M. (Eds.), *Distance Geometry: Theory, Methods and Applications*. 410 pages, Springer, 2013.

[3] J. B. Saxe, "Embeddability of weighted graphs in k-space is strongly np-hard," in *Proceedings of* 17^{th} *Allerton Conference in Communications, Control and Computing*, Monticello, IL, 1979, pp. 480–489.

[4] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino, "Euclidean distance geometry and applications," *to appear in SIAM Review*, 2013.

[5] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino, "The discretizable molecular distance geometry problem," *Computational Optimization and Applications*, vol. 52, pp. 115–146, 2012.

[6] L. Liberti, C. Lavor, A. Mucherino, and N. Maculan, "Molecular distance geometry methods: from continuous to discrete," *International Transactions in Operational Research*, vol. 18, pp. 33–51, 2010.

[7] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino, "Recent advances on the discretizable molecular distance geometry problem," *European Journal of Operational Research*, vol. 219, pp. 698—706, 2012.

[8] L. Liberti, C. Lavor, and N. Maculan, "A branch-and-prune algorithm for the molecular distance geometry problem," *International Transactions in Operational Research*, vol. 15, pp. 1–17, 2008.

[9] C. Lavor, L. Liberti, A. Mucherino, and N. Maculan, "On a discretizable subclass of instances of the molecular distance geometry problem," in *ACM Conference Proceedings, 24th Annual ACM Symposium on Applied Computing*, Hawaii, USA, 2009, pp. 804–805.

[10] A. Mucherino, C. Lavor, T. Malliavin, L. Liberti, M. Nilges, and N. Maculan, "Influence of pruning devices on the solution of molecular distance geometry problems," in *Proceedings of the* 10^{th} *International Symposium on Experimental Algorithms (SEA11)*, ser. Lecture Notes in Computer Science, P. M. Pardalos and S. Rebennack, Eds., Crete, Greece, 2011, vol. 6630, pp. 206–217.

[11] A. Bondi, "van der waals volumes and radii," *Journal of Physical Chemistry*, vol. 68, no. 3, pp. 441–451, 1964.

[12] J. E. L. Jones, "Cohesion," in *Proceedings of the Physical Society*, vol. 43, 1931, pp. 461–482.

[13] A. Mucherino, C. Lavor, and L. Liberti, "A symmetry-driven bp algorithm for the discretizable molecular distance geometry problem," in *IEEE Conference Proceedings, Computational Structural Bioinformatics Workshop (CSBW11), International Conference on Bioinformatics & Biomedicine (BIBM11)*, Atlanta, GA, USA, 2011, pp. 390–395.

[14] ——, "Exploiting symmetry properties of the discretizable molecular distance geometry problem," *Journal of Bioinformatics and Computational Biology*, vol. 10, no. 1242009, pp. 1–15, 2012.

[15] L. Liberti, B. Masson, J. Lee, C. Lavor, and A. Mucherino, "On the number of solutions of the discretizable molecular distance geometry problem," in *Proceedings of the* 5^{th} *Annual International Conference on Combinatorial Optimization and Applications (COCOA11)*, ser. Lecture Notes in Computer Science, vol. 6831, 2011, pp. 322–342.

[16] J. P. Linge and M. Nilges, "Influence of non-bonded parameters on the quality of nmr structures: A new force field for nmr structure calculation," *Journal of Biomolecular NMR*, vol. 13, no. 1, pp. 51–59, 1999.

[17] H. C. Hamaker, "The london–van der waals attraction between spherical particles," *Physica*, vol. 4, no. 10, pp. 1058–1072, 1937.

[18] T. E. Malliavin, A. Mucherino, and M. Nilges, "Distance geometry in structural biology: New perspectives," in *Distance Geometry: Theory, Methods and Applications*, L. L. N. M. A. Mucherino, C. Lavor, Ed. Springer, 2013, pp. 329–350.

[19] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne, "The protein data bank," *Nucleic Acids Research*, vol. 28, pp. 235–242, 2000.

[20] C. Lavor, L. Liberti, and A. Mucherino, "The interval branch-and-prune algorithm for the discretizable molecular distance geometry problem with inexact distances," *Journal of Global Optimization*, vol. 56(3), pp. 855–871, 2013.