ON THE DISCRETIZATION OF DISTANCE GEOMETRY PROBLEMS

Antonio Mucherino, Carlile Lavor, Leo Liberti, Nelson Maculan

Abstract: *Distance geometry consists of finding an embedding of a weighted undirected graph in $\mathbb{R}^n$. Since some years, we are working on suitable discretizations for this problem. Because of the discretization, the search domain is reduced from a continuous to a discrete set which has the structure of a tree. Based on this combinatorial structure, we developed an efficient branch-and-prune (BP) algorithm for the solution of distance geometry problems. In this paper, we focus on two important aspects of the discretization: the identification of suitable vertex discretizing orderings and the analysis of the symmetries that can be found in BP trees.*

Keywords: *distance geometry, discretization, combinatorial optimization, discretizing orderings, symmetries.*

ACM Classification Keywords*: G.2.1 Combinatorics - Combinatorial algorithms; B.5.2 Design Aids - Optimization; J.3 Life and Medical Sciences - Biology and genetics;*

MSC*: 05C85, 90C27, 51K99.*

Introduction

The Distance Geometry Problem (DGP) is the problem of finding an embedding of a weighted undirected graph G in $\mathbb{R}^n$. Let $G = (V, E, d)$ be a weighted undirected graph representing an instance of the DGP, where each vertex $v \in V$ corresponds to a point x_v in $\mathbb{R}^n$, and there is an edge between two vertices if and only if their relative distance is known (the weight associated to the edge). That is, the DGP is the problem of finding a function

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

such that

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

where $x_u = x(u)$ and $x_v = x(v)$. In its basic form, the DGP is a constraint satisfaction problem, because a set of coordinates x_v must be found that satisfies the constraints (1). In the problem definition, the symbol $|| \cdot ||$ represents the computed distance between x_u and x_v, whereas $d(u, v)$ refers to the known distances. The DGP is NP-hard Saxe [1979].

The DGP has various interesting applications. An example is given by the problem of identifying sensors in telecommunication networks Biswas et al. [2006]; Krislock [2010]. The distances between pairs of sensors can be estimated by the time needed for a two-way communication, and such distances can be exploited for identifying the position in space of each sensor. In this application, some sensors are fixed (they are called *anchors* in this domain) and the dimension of the space is usually $n = 2$.

A very interesting application arises in the field of biology. Experiences of Nuclear Magnetic Resonance (NMR) are able to estimate distances between some pairs of atoms of a molecule. Such distances, therefore, can be used for defining a DGP, whose solutions correspond to the set of conformations for the molecules that satisfy all given distances. In this context, the DGP is referred to as Molecular DGP (MDGP) Crippen et al. [1988]; Havel [1995]. This problem is particularly interesting for molecules such as proteins, because the conformation of the protein can provide clues about its function. MDGPs are DGPs in three-dimensional spaces ($n = 3$).

The basic approach to the DGP is to reformulate it as a continuous global optimization problem, where a penalty function is employed in order to measure the satisfaction of the constraints based on the known distances. Many methods and algorithms have been proposed for the solution of this optimization problem, and most of them are

based on a search in a continuous space, and/or on heuristic approaches to optimization. Recent surveys on this topic can be found in Liberti et al. [2010] and Lavor et al. [2012c].

We are working on suitable discretizations of the DGP, which are possible when some assumptions are satisfied. We give in the following the definition of two classes of DGPs that can be discretized.

The Discretizable MDGP (DMDGP) Lavor et al. [2012b]

Supposing there is a *total order relation* on the vertices of V, the DMDGP consists of all the instances of the MDGP satisfying the two assumptions:

A1 E contains all cliques on quadruplets of consecutive vertices;

A2 the triangular inequalities

$$\forall v \in \{1, \ldots, |V| - 2\} \quad d(v, v+2) < d(v, v+1) + d(v+1, v+2)$$

must hold strictly.

The Discretizable DGP (DDGP) Mucherino et al. [2012]

Supposing there is a *partial order relation* on the vertices of V, the DDGP consists of all the instances of the DGP satisfying the two assumptions:

B1 there exists a subset V_0 of V such that

- $|V_0| = 4$;
- the order relation on V_0 is total;
- V_0 is a clique;
- $\forall v_0 \in V_0 \quad \forall v \in V \setminus V_0, \quad v_0 < v$.

B2 $\forall v \in V \setminus V_0, \exists\, u^1, u^2, u^3 \in V$ such that:

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

Note that we removed the word *Molecular* in the definition of the DDGP, because this is a more general problem and not necessarily related to protein molecules. We proved that DMDGP $\subset$ DDGP.

The idea behind the discretization is that the intersection among three spheres in the three-dimensional space can produce, with probability 1, at most two points in the hypothesis in which their centers are not aligned. Consider four vertices u^1, u^2, u^3 and v. If the coordinates for u^1, u^2 and u^3 are known, as well as the distances $d(u^1, v)$, $d(u^2, v)$ and $d(u^3, v)$, then the three spheres can be defined, and their intersection provides the two possible positions in the space for the last vertex v. In the hypothesis of the DMDGP, the four vertices u^1, u^2, u^3 and v are consecutive in the ordering associated to V. In the DDGP, this condition is relaxed, so that the three vertices u^1, u^2, u^3 only have to precede v. In both cases, if the vertices u^1, u^2 and u^3 are already placed somewhere, there are only two possible positions for v.

This suggests a recursive search on a binary tree containing the potential coordinates for the vertices of G. The Branch & Prune (BP) algorithm Liberti et al. [2008] is an exact algorithm which is based on a search on this tree. The binary tree of possible solutions (to which we will also refer as "BP tree") is explored starting from its top, where

the first vertex is positioned, and by placing one vertex per time. At each step, two possible positions for the current vertex v are computed, and two new branches are added to the tree. As a consequence, the size of the binary tree can get very large quite quickly, but the presence of additional distances (not employed in the construction of the tree) can help in verifying the feasibility of the computed positions. To this aim, we consider pruning devices: as soon as a position is found to be infeasible, the corresponding branch can be pruned and the search can be backtracked. We noticed that protein-like instances contain a sufficient number of distances that can be used in the BP pruning phase, so that the complete set of solutions can be found very efficiently Liberti et al. [2011a]. The strong point of the BP algorithm is given by the possibility to enumerate all solutions to a given D(M)DGP.

Solutions to the DMDGP or to the DDGP can be represented as a complete path on the BP tree, from the root (the node associated to the first vertex) to one of the leaf nodes (corresponding to the last vertex). Layer by layer, this path shows all branches associated to the solution: in particular, it shows the selected branches on each layer of the tree. Since only two branches (related to the two positions of the sphere intersection) can be chosen at each layer based on the preceding branches that were already selected, this information can be coded as a binary variable. A solution can therefore be also represented as a binary vector of length $|V|$. If we associate an ordering to the branches having the same root on the previous layer, we can say that each binary variable indicates if the "first" or the "second" branch is selected, or if the "left" or the "right" branch is selected (see Figure 3 for a graphical representation).

In both DMDGP and DDGP, the ordering in which the vertices of G are considered is of fundamental importance. As widely discussed in Mucherino et al. [2012], an instance may or may not satisfy the assumptions for the discretization depending on this ordering. Supposing we have an instance with a predefined ordering for its vertices for which the discretization is not possible, a suitable rearrangement of these vertices may transform the instance in a discretizable instance. Since DMDGP $\subset$ DDGP, this task is, in general, easier for the DDGP.

The problem of sorting the vertices of a graph in order to satisfy certain conditions can be seen as a combinatorial optimization problem. We developed an efficient greedy algorithm for the solution of this problem Lavor et al. [2012a] (for the DDGP), but in the easier case in which all distances are supposed to be exact. Especially in the field of biology, distances are not precisely known, but rather approximations of such distances are available. Since the discretization is still possible if only one of the three reference distances (for both DMDGP and DDGP) is represented by an interval (which gives the uncertainty on the value of the distance), discretization orderings must take this condition into consideration.

Another interesting feature of DMDGPs and DDGPs is given by the definition of BP trees that are symmetric. This implies the possibility to reduce the search on a subset of branches of the tree. If one or more solutions are identified in this subset of branches, other solutions can be reconstructed by applying symmetry rules. As a consequence, the study of the symmetries present in BP trees is essential for improving the algorithm. This is particularly important for instances containing uncertain data, where such an uncertainty makes the computational cost higher.

In this paper, we focus our attention on these two important aspects of the DGP discretization: the identification of suitable vertex discretizing orderings, and the analysis of the symmetries that can be found in BP trees. We will devote the next two session to these two topics. Conclusions will be drawn in the last session.

Discretization orderings

Let $G = (V, E, d)$ be a weighted undirected graph representing an instance of the DGP. As mentioned in the Introduction, the ordering given to the vertices in V plays a very important role in the discretization. There exist indeed orderings for which neither the assumptions for the DMDGP nor the ones for the DDGP can be satisfied. Moreover, there can be orderings for which G represents an instance of the DDGP which is not in the DMDGP class. If the assumptions for the DMDGP are satisfied (**A1** and **A2**), then the assumptions for the DDGP are also satisfied (**B1** and **B2**) Mucherino et al. [2012].

We noticed the importance of the ordering associated to the vertex set V since the beginning of our work. Later, we studied the problem of finding suitable vertex orderings by solving a combinatorial optimization problem and we found an efficient solution for the DDGP Lavor et al. [2012a]. In this work, however, we supposed that all available

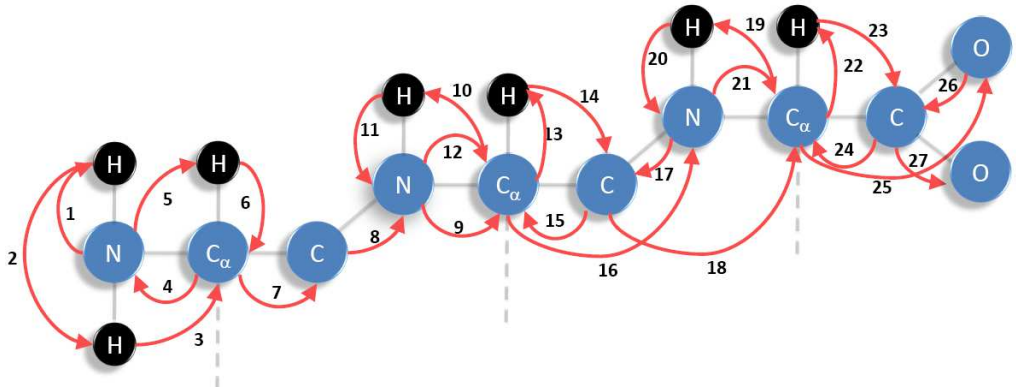

Figure 1: The hand-crafted artificial ordering r_{PB}.

distances were exact, while real-life instances usually contain a percentage of distances whose values are subject to uncertainty. In the field of biology, distances obtained by NMR are commonly represented by a lower bound and an upper bound on the actual value for the distance.

As widely explained in Lavor et al. [2012e], the discretization is still possible in presence of uncertain distances, but, in each sphere intersection, only one reference distance can be imprecise, i.e. at least two of them must be exact. In this case, indeed, the problem of intersecting three spheres (related to 3 exact distances) is transformed in the problem intersecting two spheres with one spherical shell (the uncertainty gives a certain thickness to one of the three spheres). Once two spheres are intersected, the intersection between the obtained circle and the spherical shell produces two disjoint curves in the three-dimensional space. At this point, a predetermined number of sample points can be chosen on the two curves. Notice that, in this case, BP trees are not binary anymore, and that a solution to the problem cannot be represented by a binary vector, but rather by a vector of integer numbers (there are several left branches, and several right branches).

Suitable vertex orderings do not only have to satisfy the assumptions **A1** and **A2** (for the DMDGP) or the assumptions **B1** and **B2** (for the DDGP), but, in addition, for each quadruplet of vertices u^1, u^2, u^3, v, only one of the distances $d(u^1, v)$, $d(u^2, v)$, $d(u^3, v)$ can be represented by an interval. We are currently working for finding a solution to this problem.

Meanwhile, we are also working for identifying suitable orderings for particular classes of molecules. If the chemical composition of a molecule is known, indeed, the distances that are needed for the discretization can be obtained by analyzing its chemical structure. Proteins, for example, are chains of amino acids, and the chemical structure of each amino acid is known. Exact distances can be obtained by observing the atomic bonds, while imprecise distances can be found by analyzing the degrees of freedom of the structure. In this situation, NMR data are not necessary for performing the discretization (they can be rather employed in the pruning phase of BP), and therefore the ordering can be identified independently by the available information.

We carefully hand-crafted a special ordering for the protein backbones Lavor et al. [2012e] (see Figure 1). We found a particular ordering for the generic amino acid which composes a protein, and hence the ordering r_{PB} related to the whole protein backbone as the composition of the several orderings for its amino acids (without side chains):

$$r_{PB} = \{r^1_{PB}, r^2_{PB}, \ldots, r^i_{PB}, \ldots, r^p_{PB}\}.$$

The ordering r^i_{PB} corresponds to the i^{th} amino acid of the protein (superscripts indicate the amino acid to which each atom belongs):

$$r^i_{PB} = \{N^i, C^{i-1}, C^i_\alpha, H^i, N^i, C^i_\alpha, H^i_\alpha, C^i, C^i_\alpha\}.$$

In order to artificially add exact distances, atoms are considered more than once: the distance between two copies of the same atom is 0. It is important to remark that this trick for allowing the discretization does not increase the complexity of the problem, because the second copy of an atom can only be positioned as its first copy. In other

words, there is no branching on the BP tree in correspondence with duplicated atoms. For example, in the ordering for the generic amino acid r^i_{PB}, the nitrogen N^i is considered twice, the carbon C^i_α is considered 3 times, and the carbon C^{i-1} belonging to the previous amino acid is repeated among the atoms of this amino acid. In total, there are four copies of atoms that already appeared earlier in the sequence. Hydrogen atoms are not duplicated.

The first and the last amino acids of the chain have a slightly different structure, and therefore we designed particular orderings for such amino acids. At the beginning of the sequence, we consider the following ordering for the first amino acid:

$$r^1_{PB} = \{N^1, H^1, H^0, C^1_\alpha, N^1, H^1_\alpha, C^1_\alpha, C^1\}.$$

One of the hydrogens bound to N^1 (in general, in r^i_{PB}, there is only one hydrogen) is indicated by the symbol H^0. Between r^1_{PB} and the generic amino acid ordering r^i_{PB}, we consider the following ordering, which makes a sort of *bridge* between the beginning of the sequence and the generic ordering:

$$r^2_{PB} = \{N^2, C^2_\alpha, H^2, N^2, C^2_\alpha, H^2_\alpha, C^2, C^2_\alpha\}.$$

Finally, for the last amino acid of the sequence, we have the following ordering:

$$r^p_{PB} = \{N^p, C^{p-1}, C^p_\alpha, H^p, N^p, C^p_\alpha, H_\alpha, C^p, C^p_\alpha, O^p, C^p, O^{p+1}\}.$$

Note that this is the only case in which oxygen atoms appear. The two oxygens O^p and O^{p+1} which are present in the last amino acid r^p_{PB} correspond to the two oxygens of the C-terminal carboxyl group COO^- of the protein.

The special ordering r_{PB} is constructed in order to satisfy the assumptions for the DMDGP. In particular, for each atom $v \in V$, the three edges $(v-3, v)$, $(v-2, v)$ and $(v-1, v)$ are always contained in E. The corresponding distances are obtained from known bond lengths and bond angles, that only depend on the kind of bonded atoms. The two edges $(v-2, v)$ and $(v-1, v)$ are always associated to exact distances, whereas only the edge $(v-3, v)$ may be associated to an interval distance. In particular, there are three different possibilities. If $d(v-3, v) = 0$, then v represents a duplicated atom, and therefore the only feasible coordinates for v are the same of its previous copy. If $d(v-3, v)$ is an exact distance, the standard discretization process can be applied (intersection among three spheres), and two possible positions for v can be computed. Finally, if $d(v-3, v)$ is represented by an interval, we discretize the interval and take D sample distances from it. For each sample distance, we perform the sphere intersection: $2 \times D$ possible atomic positions can be computed for v.

In order to discretize instances related to entire protein conformations, it is necessary to identify an ordering for all side chains for the 20 amino acids that can be involved in the protein synthesis. This is not trivial, because side chains have more complex structures with respect to the part which is common to each amino acid. Figure 2 shows a discretization ordering for the *glycine*, the smallest amino acid that can be found in proteins, whose side chain is composed by a hydrogen atom only:

$$r_{GLY} = \{C^i_\alpha, N^i, H^i_\alpha, C^i_\alpha, H^i_\beta, C^i, H^i_\alpha, C^i_\alpha, C^i\}.$$

Here, we denote with H^i_β the hydrogen forming the glycine side chain. The ordering r_{GLY} can be plugged in the ordering r_{PB}. If we consider the first amino acid, for example, the last two vertices C^1_α and C^1 need to be replaced by r_{GLY} in order to consider this side chain. In spite of the simplicity of glycine, this ordering is rather complex and it considers various atoms of the protein backbone, which are duplicated in order to satisfy the needed assumptions. Future works will be devoted to the development of suitable hand-crafted orderings for all other 19 amino acids. This will allow us to solve real NMR instances by using our discrete approach to DGPs.

Symmetries

If a D(M)DGP instance admits solutions, then there is an even number of solutions Lavor et al. [2012b]. This is a theoretical result proved for the DMDGP which is immediately extensible to the DDGP. Our computational experiments confirmed this result. However, the sets of solutions found by the BP algorithm always satisfied a stronger property: the cardinality of the set of solutions is always a power of 2.

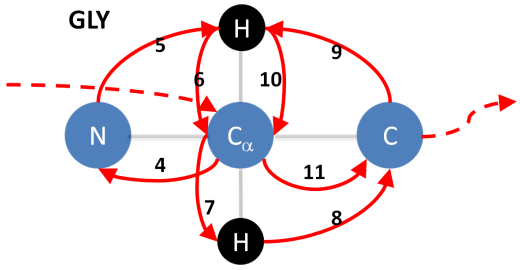

Figure 2: The artificial ordering r_{GLY} for the glycine.

This theoretical result remained unproved for a long time. At a certain point, we found indeed a counterexample, i.e. an instance, artificially generated in a particular way, for which the total number of solutions was not a power of 2. But then we were able to prove that the Lebesgue measure of the subset of instances for which this property is not satisfied is 0 Liberti et al. [2011b]. As a consequence, we can say that, in practice, real-life instances should always have a power of 2 of solutions. This result has been formally proved for the DMDGP; we are currently working for extending this result to the DDGP.

The "power of 2" property is due the presence of various symmetries in BP binary trees Lavor et al. [2012b]. First of all, there is a symmetry at layer 4 of all BP trees, which makes even the total number of solutions. We usually refer to this symmetry as the *first symmetry*. At layer 4, there are no distances for pruning, and the two branches rooted at node 3 are perfectly symmetric. In other words, any solution found on the first branch is related to another solution on the second one, which can be obtained by inverting, at each layer, left with right branches, and vice versa.

In the DMDGP, as for the first symmetry, each partial reflection symmetry appears every time there are no pruning distances concerning some layer v. In such a case, the number of feasible branches on layer v is duplicated with respect to the one of the previous layer $v-1$, and pairs of branches rooted at the same node x_{v-1} are perfectly symmetric. Figure 3 shows a BP tree containing 3 symmetries.

As mentioned in the Introduction, a solution to a DMDGP can be represented in different ways, such as a path on the tree and a list of binary choices 0–1 (we suppose here that all distances are exact). Since solutions sharing symmetric branches of the tree have symmetric local binary representations, we can derive a very easy strategy for generating all solutions to a DMDGP from one found solution and the information on the symmetries in the

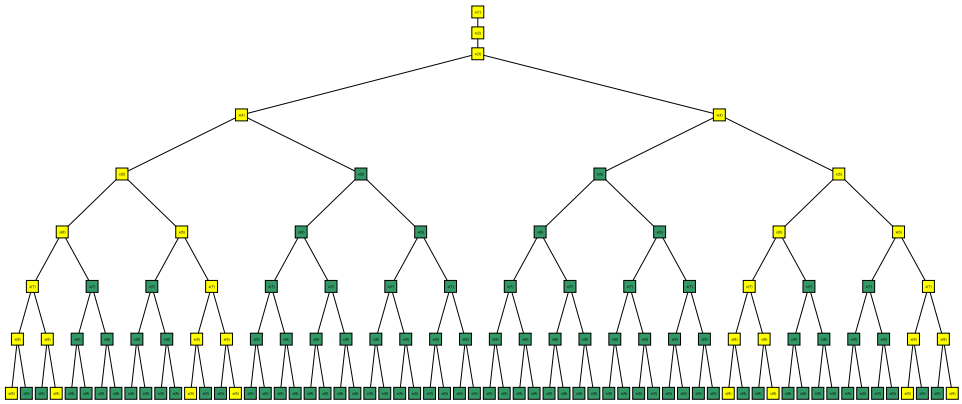

Figure 3: All symmetries of an instance with 9 vertices and $B = \{4, 6, 8\}$. Feasible branches are marked in light yellow.

corresponding tree Mucherino et al. [2011]. Let us consider for example the solution in Fig. 3 corresponding to the second leaf node (from left to right). The binary vector corresponding to this solution is

$$\mathbf{s}_2 = (0, 0, 0, 0, 0, 0, 0, 1, 1),$$

where we suppose that 0 represents the choice *left*, and 1 represents *right* (the first three zeros are associated to the first three fixed vertices of the graph). Since there is a symmetry at layer 6, another solution to the problem can be easily computed by repeating all choices from the root node until the layer 5, and by inverting all other choices. On the binary vector, repeating means *copying*, and inverting means *flipping*. So, another solution to the problem is

$$\mathbf{s}_3 = (0, 0, 0, 0, 0, 1, 1, 0, 0).$$

This solution corresponds to the third feasible leaf node in Fig. 3.

This property can be exploited for speeding up the solution to DMDGPs. The procedure we mentioned above can indeed be used for reconstructing any solution to the problem. Thus, once one solution to the problem is known, all the others can be obtained by exploiting information on the symmetries of BP trees. The set

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

contains all layers v of the tree where there is a symmetry Mucherino et al. [2011]. As a consequence, $|B|$ is the number of symmetries that are present in the tree. Naturally, since the first symmetry is present in all BP trees, $|B| \geq 1$. The total number of solutions is, with probability 1, equal to $2^{|B|}$.

If the current layer is related to the vertex $v \in B$, for each x_{v-1} on the previous layer, both the newly generated positions for x_v are feasible. If $v \notin B$, instead, only one of the two positions can be part of a branch leading to a solution. The other position is either infeasible or it defines a branch that will be pruned later on at a further layer v, in correspondence with a pruning distance whose graph edge $\{u, w\}$ is such that $u + 3 < v \leq w$. Therefore, we can exploit such information for performing the selection of the branches that actually define a solution to the problem. When $v \notin B$ (only one position is feasible), it is not known a priori which of the two branches (left/right) is the correct one. This is the reason why at least one solution must be computed before having the possibility of exploiting the symmetries for computing all the others.

Computational experiments presented in Mucherino et al. [2011] showed that the BP algorithm can be enhanced by exploiting this a priori knowledge on the symmetries, specially when instances having many solutions are considered. Once one solution is obtained by BP, all the others can be quickly obtained by using this information. Even if protein-like instances usually contain a few symmetries Liberti et al. [2011a], the extension of the BP algorithm to interval data actually needs to be integrated with symmetry-based strategies in order to improve its efficiency. Research in this direction will be performed in the future.

Conclusions

The discretization of DGPs allows us to solve these problems by employing the BP algorithm, which is based on a search on the BP trees. This algorithm is the first exact algorithm which is potentially able to solve DGPs containing interval data. Differently from other proposed algorithms for this problem, BP can enumerate all the solutions for a given instance, allowing to obtain multiple solutions and to leave any other out with certainty.

These works on the DGP also open the doors for suitable discretizations of other important problems, specially in the field of biology. The discretizing ordering for proteins that we detailed in this paper, for example, is completely independent on NMR data, and therefore it could also be applied in the context of *protein folding* and *protein docking*. The exploitation of the symmetries in BP trees can be essential for managing these more complex problems. These are the main directions that we will take for our future research.

Acknowledgements

This work is partially supported by the ANR project "Bip:Bip". The authors would also like to thank the Brazilian research agencies FAPESP and CNPq, and the French research agency CNRS and École Polytechnique, for financial support.

Bibliography

P. Biswas, T.-C. Liang, T.-C. Wang, Y. Ye, *Semidefinite Programming Based Algorithms for Sensor Network Localization*, ACM Transactions on Sensor Networks **2**, 188–220, 2006.

G.M. Crippen and T.F. Havel, *Distance Geometry and Molecular Conformation*, John Wiley & Sons, New York, 1988.

T.F. Havel, *Distance Geometry*, D.M. Grant and R.K. Harris (Eds.), Encyclopedia of Nuclear Magnetic Resonance, Wiley, New York, 1701-1710, 1995.

N. Krislock, *Semidefinite Facial Reduction for Low-Rank Euclidean Distance Matrix Completion*, PhD thesis, University of Waterloo, 2010.

C. Lavor, J. Lee, A. Lee-St.John, L. Liberti, A. Mucherino, M. Sviridenko, *Discretization Orders for Distance Geometry Problems*, Optimization Letters **6**, 783–796, 2012.

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

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.

C. Lavor, L. Liberti, A. Mucherino, *On the Solution of Molecular Distance Geometry Problems with Interval Data*, IEEE Conference Proceedings, International Workshop on Computational Proteomics, International Conference on Bioinformatics & Biomedicine (BIBM10), Hong Kong, 77–82, 2010.

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, 2012.

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.

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.

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.

L. Liberti, B. Masson, J. Lee, C. Lavor, A. Mucherino, *On the Number of Solutions of the Discretizable Molecular Distance Geometry Problem*, Lecture Notes in Computer Science **6831**, Proceedings of the 5^{th} Annual International Conference on Combinatorial Optimization and Applications (COCOA11), Zhangjiajie, China, 322–342, 2011.

A. Mucherino, C. Lavor, L. Liberti, *A Symmetry-Driven BP Algorithm for the Discretizable Molecular Distance Geometry Problem*, IEEE Conference Proceedings, Computational Structural Bioinformatics Workshop (CSBW11), International Conference on Bioinformatics & Biomedicine (BIBM11), Atlanta, GA, USA, 390–395, 2011.

A. Mucherino, C. Lavor, L. Liberti, *The Discretizable Distance Geometry Problem*, to appear in Optimization Letters, 2012.

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

Authors' Information

Antonio Mucherino *IRISA, University of Rennes 1, Rennes, France.*
Email: antonio.mucherino@irisa.fr

Carlile Lavor *IMECC-UNICAMP, Campinas-SP, Brazil. Email: clavor@ime.unicamp.br*

Leo Liberti *LIX, École Poleytechnique, Palaiseau, France. Email: liberti@lix.polytechnique.fr*

Nelson Maculan *COPPE, Federal University of Rio de Janeiro, Rio de Janeiro-RJ, Brazil.*
Email: maculan@cos.ufrj.br