An Analysis on the Degrees of Freedom of Binary Representations for Solutions to Discretizable Distance Geometry Problems

Antonio Mucherino

An Analysis on the Degrees of Freedom of Binary Representations for Solutions to Discretizable Distance Geometry Problems

Antonio Mucherino

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

Abstract. The degrees of freedom in special binary representations for instances of the Discretizable Distance Geometry Problem (DDGP) are studied in this note article. The focus is on DDGP instances where the underlying graphs, together with their associated vertex orders, are able to satisfy the so-called consecutivity assumption. This additional assumption, in fact, makes it possible to group together subsets of consecutive binary variables, which turn out to strongly depend on each other, so that they can actually be replaced by a smaller subset of binary variables. As a consequence, new binary representations, with reduced degrees of freedom w.r.t the trivial binary representations for DDGP instances, can be introduced and potentially be exploited in DDGP solution methods.

1 Introduction

Given a positive integer K, the Discretizable Distance Geometry Problem (DDGP) [5] asks whether a weighted undirected graph $G = (V, E, d)$ satisfying the following assumptions:

(a) $G[\{1, 2, \ldots, K\}]$ is a clique;
(b) $\forall v \in \{K+1, \ldots, |V|\}$, there exist $u_1, u_2, \ldots, u_K \in V$ such that
  (b.1) $u_1 < v$, $u_2 < v$, $\ldots$, $u_K < v$;
  (b.2) $\{\{u_1, v\}, \{u_2, v\}, \ldots, \{u_K, v\}\} \subset E$;
  (b.3) $\mathcal{V}_S(u_1, u_2, \ldots, u_K) > 0$ (if $K > 1$),

admits a realization $x : v \in V \longrightarrow x_v \in \mathbb{R}^K$ such that, for all $\{u, v\} \in E$, the distance constraints $||x(u) - x(v)|| = d(u, v)$ are satisfied (the symbol $||\cdot||$ indicates the Euclidean norm). The vertices $u_1, u_2, \ldots, u_K$, for every selected vertex $v > K$, are referred to as *reference vertices*, and the corresponding distances $d(u_1, v)$, $d(u_2, v)$, $\ldots$, $d(u_K, v)$ are generally referred to as *reference distances*. Notice that $G[\cdot]$ is the subgraph induced by a subset of vertices of V, and $\mathcal{V}_S(\cdot)$ is the volume of the simplex generated by a valid realization of the vertices $u_1, u_2, \ldots, u_K$. Notice that a total *vertex ordering* on the vertices of the graph is implicitly given.

By exploiting the discretization assumptions **(a)** and **(b)**, it is possible to construct a search tree containing all solutions to the DDGP. This work is focused on DDGPs where the distance information can be considered as *exact*, which is: "extremely precise", and consequence of this fact is that the search tree is binary. The DDGP is an NP-hard

subclass [1] of the graph embeddability problem, which was proved to belong to the NP-hard class by Saxe in [8].

Over the last years, several variants of the DDGP have been proposed in the scientific literature [2]. The variant which was introduced with the acronym DMDGP[1] is based on an additional assumption on G, where it is required that the reference vertices are the ones that immediately precede the current vertex v in the given vertex ordering. In some previous publications, this additional assumption is named *the consecutivity assumption*. When this assumption is satisfied, it is possible to define a sequence of $(K+1)$-cliques of G which admit an overlap of K vertices [4]. Every clique groups together every vertex $v > K$ with its reference vertices $u_1, u_2, \ldots, u_K$, so that the feasibility of each of them can be verified independently, with an overall computational complexity which is polynomial.

The focus of this contribution is on binary representations for solutions to DDGPs where the consecutivity assumption is satisfied. Actually, it is rather trivial to verify that all instances of the DDGP admit a binary representation, because the paths over the branches of the binary trees that are employed for the representation of the DDGP search domain can be simply identified by vectors of binary variables [6]. In addition to this, however, the consecutivity assumption is exploited in this work to group together subsets of consecutive vertices which are not independent, so that the actual number of degrees of freedom in the binary representations can be reduced. To this aim, a theoretical result previously presented in [3] is exploited. Although the current work may seem a rather light contribution to the research on the DDGP, the use, in any solution methods, of the proposed binary representations can potentially give important benefits for the solution of DDGP instances.

After a short review on solution methods for the DDGP (see Section 2), binary representations for DDGP solutions will be studied in Section 3. Finally, Section 4 will conclude this note article with some future works.

2 Current DDGP solution methods

The majority of proposed methods for the DDGP are based on a standard branch-and-prune (BP) algorithmic framework [2]. As mentioned in the Introduction, the discretization assumptions make it possible to represent solutions to DDGP instances as a discrete (and finite) domain having the structure of a tree. The idea therefore, in all BP frameworks, is to explore such a tree in a depth-first fashion by exploiting the distance information which is ensured by the discretization assumptions; any additional distance information is subsequently used for pruning infeasible tree branches: the distances used to perform this "pruning" action are generally named *pruning distances*. When a leaf node of the search tree is reached, one solution to the instance is given by the path over the tree edges leading from the tree root node to the current leaf node [7].

Several methods based on this algorithmic framework have been proposed over the last years [2]. In none of these methods, however, the pruning distances are exploited to perform a reduction on the degrees of freedom for the binary representation associated

[1] The additional "*M*" stands for *Molecular* and it reminds that this class of problems seemed, when introduced, to be particularly suitable to structural biology problems.

to the solutions. When the BP framework is implemented, the depth-first search is supposed to perform its exploration layer by layer, without using possible local information about subgroups of vertices that may have already been explored in previous steps. The reduction on the degrees of freedom is actually possible when this "local information" is exploited (see next section).

3 A binary representation for DDGP solutions

The search domain for a DDGP instance with exact distances is a binary tree [5]. One solution to the problem can be represented as a path from the root to one of its leaf nodes, where layer by layer, the path has only two possible "ways to go": either it may take the left-handed branch, or the right-handed one. A vector of binary variables b_i can therefore be employed to represent one possible solution to a given DDGP instance [7]:

b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	$\ldots$	$\ldots$	b_n

where b_i corresponds to the vertex v_i, and where the indices i reflect the vertex order associated to the graph G. An immediate observation is that, since the initial clique is generally fixed in solution methods (see assumption **(a)** in the Introduction), the first K binary variables in this binary representation are actually not necessary. Moreover, all DDGP instances admit search trees where a symmetry is present at layer $K+1$: if the instance is feasible, both binary options 0 and 1 are feasible for the variable b_{K+1}, so that it can actually be neglected (once solutions have been found with $b_{K+1} = 0$, all others can be obtained by changing its value to 1).

The following representation is therefore more convenient (the example is given for $K = 3$):

b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	$\ldots$	$\ldots$	b_n

There are $|V|-K-1$ degrees of freedom in this representation, corresponding to $2^{|V|-K-1}$ potential solutions to be explored by DDGP solution methods. This initial result is not new and is valid for any kind of DDGP instance.

When the DDGP instance satisfies the consecutivity assumption, every subsequence of consecutive $K+1$ vertices corresponds to a $(K+1)$-clique of G, where the first K vertices play the role of reference when constructing the possible positions for the vertex which is the last, in the given vertex ordering, in the clique (see Introduction). The feasibility of all these cliques can be verified in advance: given a binary representation, the only infeasibility that can be detected is related to possible violations on the "pruning distances".

Suppose now that there exists such a pruning distance between the vertex v_6 and v_{12} of the example above (which is, between the binary variables b_6 and b_{12}). Consequence of the consecutivity assumption is the possibility to state that the sub-instance $G[\{v_6,\ldots,v_{12}\}]$ is itself a DDGP instance satisfying the consecutivity assumption. Moreover, this sub-instance has a special property: its first and last vertex (in the vertex ordering inherited from G) are connected by a pruning distances. The symmetry results in [3] indicate therefore that there exist only two (symmetric) solutions to the sub-instance $G[\{v_6,\ldots,v_{12}\}]$. Thus, once this sub-instance is solved independently and its

two solutions are encoded by a binary representation, say $c_9(0) = (\hat{b}_9, \hat{b}_{10}, \hat{b}_{11}, \hat{b}_{12})$ and $c_9(1) = \neg c_9(0) = (\neg\hat{b}_9, \neg\hat{b}_{10}, \neg\hat{b}_{11}, \neg\hat{b}_{12})$, the new binary representation for the original instance is:

b_5	b_6	b_7	b_8	c_9	b_{13}	b_{14}	...	...	b_n

which replaces 4 binary variables with only one. In fact, the binary variables b_6, b_7 and b_8 are not replaced (recall that $K = 3$ in the example), because they represent the given set of values for the initial clique of the sub-instance.

In general, this kind of manipulation on the binary representations makes it possible to perform a reduction on the degree of freedom by combining subsets of consecutive vertices that depend on each other. For every degree of freedom that is removed from the representation, there is a consequent reduction of one order of magnitude on the number of total DDGP solutions that can be represented. If $m > K+1$ is the number of vertices forming the sub-instance, the degrees of freedom become equal to $|V| - m$ after the first modification, so that 2^{m-K-1} undesired solutions are "erased" from the search tree. Moreover, this operation can be performed for all pruning distances in the DDGP instance, starting from the ones defining smaller sub-instances, up to the larger ones. Even if it remains exponential, the reduction of the corresponding computational complexity can become important, so that any solution method for the DDGP can find a great benefit in using these new binary representations.

It is important to remark that this complexity reduction is possible because every sub-instance delimited by a pruning distance is solved in advance. Naturally, the solution of all these sub-instances introduces an extra computational cost. However, when the overall complexity is computed, one needs to *sum up* all these complexities, for each identified sub-instance, as well as for the final representation of the original instance. This is different from the computation of the complexity associated to the trivial representation consisting of only b_i's variables, where all these single complexities would need to be *multiplied* by each other to obtain the total complexity.

4 Conclusions and perspectives

This short contribution shows that the properties of DDGP instances satisfying the consecutivity assumption can be exploited to reduce the degrees of freedom in suitable binary representations of its solutions, and in turn decrease the size of the search space that is explored by solution methods. This short article is presented in a simple style by making reference to a running example: future works will be aimed at a formalization of the content of this article, as well as at the possible extension of (at least a part of) these results to DDGPs instances for which the consecutivity assumption is not satisfied. An analysis on DDGP instances where not all its distances can be considered as "exact" will also be investigated.

Acknowledgments

This work is partially supported by the international project MULTIBIOSTRUCT funded by the ANR French funding agency (ANR-19-CE45-0019).

References

1. C. Lavor, L. Liberti, N. Maculan, A. Mucherino, *The Discretizable Molecular Distance Geometry Problem*, Computational Optimization and Applications **52**, 115–146, 2012.
2. L. Liberti, C. Lavor, N. Maculan, A. Mucherino, *Euclidean Distance Geometry and Applications*, SIAM Review **56**(1), 3–69, 2014.
3. L. Liberti, B. Masson, J. Lee, C. Lavor, A. Mucherino, *On the Number of Realizations of Certain Henneberg Graphs arising in Protein Conformation*, Discrete Applied Mathematics **165**, 213–232, 2014.
4. A. Mucherino, *A Pseudo de Bruijn Graph Representation for Discretization Orders for Distance Geometry*, Lecture Notes in Computer Science **9043**, Lecture Notes in Bioinformatics series, F. Ortuño and I. Rojas (Eds.), Proceedings of the 3^{rd} International Work-Conference on Bioinformatics and Biomedical Engineering (IWBBIO15), Granada, Spain, 514–523, 2015.
5. A. Mucherino, C. Lavor, L. Liberti, *The Discretizable Distance Geometry Problem*, Optimization Letters **6**(8), 1671–1686, 2012.
6. A. Mucherino, C. Lavor, L. Liberti, E-G. Talbi, *A Parallel Version of the Branch & Prune Algorithm for the Molecular Distance Geometry Problem*, IEEE Conference Proceedings, ACS/IEEE International Conference on Computer Systems and Applications (AICCSA10), Hammamet, Tunisia, 1–6, 2010.
7. A. Mucherino, L. Liberti, C. Lavor, *MD-jeep: an Implementation of a Branch & Prune Algorithm for Distance Geometry Problems*, Lectures Notes in Computer Science **6327**, K. Fukuda et al. (Eds.), Proceedings of the 3^{rd} International Congress on Mathematical Software (ICMS10), Kobe, Japan, 186—197, 2010.
8. 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.