



# The discrete side of Distance Geometry: a focus on the 1-dimensional case

Simon Hengeveld, Antonio Mucherino

## ► To cite this version:

Simon Hengeveld, Antonio Mucherino. The discrete side of Distance Geometry: a focus on the 1-dimensional case. Proceedings of Discrete Math Days, Jul 2022, Santander, Spain. hal-03688779

**HAL Id: hal-03688779**

**<https://inria.hal.science/hal-03688779v1>**

Submitted on 5 Jun 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# The discrete side of Distance Geometry: a focus on the 1-dimensional case\*

Simon B. Hengeveld<sup>†</sup> and A. Mucherino<sup>‡</sup>

IRISA, University of Rennes 1, Rennes, France

## Abstract

We review some recent results in the context of Distance Geometry in dimension 1, and give some new research directions for problem instances where the distance information is imprecise and represented by real-valued intervals.

## 1 Distance Geometry in dimension 1

If a sketch comedy were to introduce the problem we consider, it would probably show a few improvised robbers meeting with the gang boss in an attempt to get ready for their *mission*. You would then hear the boss shout “Synchronize your watches”, and hence listen the robbers give diverging answers for the current time...until the last robber, after a little hesitation, would finally admit that his watch has since long been broken<sup>1</sup>...

Try it yourself with some friends and you will see that the situation above is not so uncommon (exception made for the broken watch), for in our everyday life it is not necessary to be extremely precise with time. The problem of synchronizing clocks with high precision is rather a problem that computer machines need more often, and more importantly, deal with. Over the Internet, for example, several messages are sent and received each second by the machines connected to the Net, and, as pointed out by Lamport in the seminal paper [3], the *logical nature* of time is of primary importance when designing or analyzing distributed systems. Since then, several research lines have developed in this context [1, 11, 12, 13].

Distance Geometry (DG) is a more general problem that is able to model the clock synchronization problem when its dimensionality is set to 1 [7]. We are given a simple weighted undirected graph  $G = (V, E, d)$  whose edges represent the available distance information between vertex pairs, and the main problem in DG is to find out what are the positions we can assign to the vertices of the graph  $G$  so that the available distances are satisfied. These positions can, in general, be given in any dimension  $K > 0$ . In the context of clock synchronization, the vertices  $v \in V$  are computer machines, the presence of an edge  $\{u, v\} \in E$  between two machines indicates that a time offset is available, and the weight function  $d$  provides the numerical value for such an offset. And as mentioned above,  $K$  is set to 1. Saxe proved in 1979, in the context of graph embeddability, that the problem is NP-complete [9] when  $K = 1$ .

In this short contribution, we will review some of our most recent results in the context of DG in dimension 1, and we will give some new directions for extending these results to the case where the distances are represented by real-valued intervals (instead of singletons).

---

\*This research is supported by the ANR-PRCI MULTIBIOSTRUCT project (ANR-19-CE45-0019).

<sup>†</sup>Email: [simon.hengeveld@irisa.fr](mailto:simon.hengeveld@irisa.fr)

<sup>‡</sup>Email: [antonio.mucherino@irisa.fr](mailto:antonio.mucherino@irisa.fr)

<sup>1</sup>One of us remembers having seen this comedy sketch in a movie, probably an Italian comedy of the 80s; however, it is likely that the same comedy sketch was repoposed in other places.

## 2 Paradoxical Distance Geometry

The *paradoxical* subclass of DG instances in dimension 1 has recently been introduced in [2]. Instances belonging to this special class have the particularity of being on the one hand “easy” to solve because most vertices in  $G$  depend on only one other vertex, and the total number of expected solutions may be known in advance [5]; on the other hand, however, these instances admit a search space that grows exponentially with the cardinality of  $V$ , and its complete exploration is generally necessary in order to identify the solutions. Thus, these are rather “hard” instances.

Paradoxical instances of the DG are represented in dimension 1 by cycle graphs. Once the vertex to play the role of *first* vertex  $v_1$  is selected, and one of its two neighbours is selected to play the role of the *second* vertex, say  $v_2$ , then a total vertex order on  $V$  can be derived. The first vertex can be simply placed in the origin of the real line  $\mathbb{R}$ . Subsequently, the positions for every vertex  $v_k$ , with  $k > 1$ , can be obtained by exploiting the distance information related to its predecessor  $v_{k-1}$ .

If we suppose that all available distances are *exact*, which is, extremely precise, then, for every known position  $x_{k-1}$  for  $v_{k-1}$ , we can compute two new possible positions,  $x_{k-1} - d_{k-1,k}$  and  $x_{k-1} + d_{k-1,k}$ , for the vertex  $v_k$ . We remark that this procedure builds up a binary tree whose paths, from the root to any of its leaf nodes, represent possible solutions to our problem [4]. Each position  $x_k$  is computed by performing partial sums of the distance values, each taken with the sign associated to the tree branches of the considered path. The feasible solutions are those for which the only “extra distance”, i.e. the only one that is not employed for the computation of the positions and is associated to the edge  $\{1, n\}$ , is satisfied by the positions finally assigned to  $v_n$ . It is only when the last layer of the tree is reached, indeed, that the feasible solutions can be selected from the search space.

As pointed out in [2], the introduction of a fictive vertex  $v_{n+1}$  in  $G$ , artificially connected to  $v_n$  by the edge  $\{n, n+1\}$  holding the same weight as  $\{1, n\}$ , together with the suppression of the original edge  $\{1, n\}$ , allows us to have a uniform distribution of the distances over the vertices of the graph. The immediate consequence of this transformation consists in the possibility of *blindly* applying the same rule over the vertices of the graph, in the derived vertex order, for the computation of all vertex positions. After the graph transformation, the selection of the feasible solutions from the search space can be performed by verifying whether the positions for  $v_1$  and  $v_{n+1}$  coincide (this is a valid solution), or not (the solution is not valid). An important consequence of this transformation is the matrix-by-vector reformulation that we will describe in Section 3.

In order to establish the high importance of this special class of instances, we remark that every DG instance can be seen as the union of paradoxical instances [8]. A solution method capable to recognize, extract and solve with priority the paradoxical parts of DG instances could in fact benefit of better convergence properties: once the paradoxical sub-instances are solved, they can be replaced by their relatively tiny solution sets, aiding in this way at reducing the degrees of freedom for the solutions of the original instance.

## 3 Matrix-by-vector reformulation

The procedure above for the solution of paradoxical DG instances is based on the idea to build up a binary tree containing vertex positions and to select the feasible solutions by verifying that  $x_1 = x_{n+1}$ . We introduce the matrix<sup>2</sup>

$$M_{ij} = \begin{cases} -1 & \text{if } (i-1)/2^{j-1} \bmod 2 = 0, \\ 1 & \text{otherwise,} \end{cases}$$

and the vector  $y_j = d_{j,j+1}$ , which contains the distance information related to our paradoxical instance. Notice that the index  $i$  varies from 1 to  $2^n$ , while the index  $j$  varies from 1 to  $n$ . It can be easily

---

<sup>2</sup>We can point out here again the paradoxical side of our instances: the multiplication of a matrix by a vector is nothing difficult to perform; however, we remark that the number of rows of our matrices grow exponentially with the number of vertices in  $G$ , so that very large matrices may be defined for relatively small instances.

verified [2] that the vector

$$r = My, \quad (1)$$

resulting from the multiplication of the matrix  $M$  by the vector  $y$ , contains the positions  $x_{n+1}$  for all possible solutions. The feasible solutions to our paradoxical instances are therefore the ones for which  $r_i = 0$  (because  $x_1$  is implicitly set to 0). Moreover, for a given matrix row  $i$ , the value of each  $x_k^i$  (which is the position for the vertex  $v_k$  in the solution encoded by the  $i^{\text{th}}$  row of the matrix) can be obtained by performing the partial sums  $\sum_{j=1}^k M_{ij}y_j$ . We point out that this matrix-by-vector reformulation is particularly adapted to the optical processor presented in [2], and that similar approaches have also been proposed for other well-known NP-complete problems [10].

#### 4 Generalization to interval distances

An approach to deal with DG instances in dimension 1 where distances are represented by real-valued intervals was previously proposed in [7]. The main idea is to replace standard arithmetic with interval arithmetic for the calculation of possible subsets of interval values for the vertices of the graph  $G$ . Basically, only the addition and subtraction operations need to be borrowed from interval arithmetic. We notice that interval arithmetic has already been successfully employed in the context of global optimization [6], and that, in our context, it is particularly useful for modeling distance uncertainty.

Let us suppose that the graph  $G = (V, E, d)$  represents a paradoxical DG instance where the distance information is provided through real-valued intervals, which is, the weight function  $d$  actually provides a lower bound  $\ell$  and an upper bound  $u$  on each distance value. Let  $z$  be a vector consisting of  $n$  intervals, where  $n = |V|$ . Once the degenerate interval  $\{0\}$  is associated to the first element  $z_1$  of the vector, the subsequent intervals  $z_k$  can be obtained via the formulæ:

$$\begin{aligned} [\underline{z}_k, \bar{z}_k]' &= [\underline{z}_{k-1}, \bar{z}_{k-1}] - [\ell_{k-1,k}, u_{k-1,k}] = [\underline{z}_{k-1} - u_{k-1,k}, \bar{z}_{k-1} - \ell_{k-1,k}], \\ [\underline{z}_k, \bar{z}_k]'' &= [\underline{z}_{k-1}, \bar{z}_{k-1}] + [\ell_{k-1,k}, u_{k-1,k}] = [\underline{z}_{k-1} + \ell_{k-1,k}, \bar{z}_{k-1} + u_{k-1,k}], \end{aligned}$$

where  $\underline{z}$  and  $\bar{z}$  are the two vectors of real numbers containing, respectively, the lower and the upper bounds of the intervals in  $z$ . When the fictive vertex  $v_{n+1}$  is employed, the verification of feasibility for the obtained solutions reduces to verifying that  $[\underline{z}_{n+1}, \bar{z}_{n+1}] \cap [\underline{z}_1, \bar{z}_1] \neq \emptyset$ , which is equivalent to saying that the last interval  $z_{n+1}$  needs to contain the origin 0 (when  $z_1$  is set up as indicated above).

But how to compute all intervals for the vector  $z$ ? We claim in this short contribution that the use of interval distances in  $y$ , coupled with the use of interval arithmetic in equ. (1), naturally extends the computations for the interval case. However, when classical arithmetic is replaced by interval arithmetic, the range of the computed intervals tends to increase more and more for larger indices  $k$ . As a consequence, a larger and larger growth uncertainty is likely to be associated to the possible values of the vertex positions as our instances grow in size.

The matrix-by-vector formulation, however, when employed for the interval case, allows us to rewrite in a straightforward manner the *back-propagation* procedure previously introduced in [7]. This procedure aims to reduce the range of the obtained intervals  $z_k$  by back-propagating any additional distance information that may be available for the computation of intervals  $z_h$  with  $h > k$ . When considering paradoxical DG instances, this situation only occurs when positions for the fictive vertex  $v_{n+1}$  are computed, which are supposed to be compatible with the position assigned to  $v_1$  (because of the original edge  $\{1, n\}$ ).

Let  $S$  be the subset of indices for which the corresponding row in  $M$  represents a feasible solution:

$$S = \{i \in \{1, \dots, n\} : z_1 \cap [\underline{r}_i, \bar{r}_i] \neq \emptyset\}.$$

We claim that the actual position intervals  $z_k^i$ , for every feasible solution with index  $i \in S$  and every

vertex  $v_k$  of our graph  $G$ , are given by the following interval arithmetic formula:

$$[\underline{z}_k^i, \bar{z}_k^i] = \left( \sum_{j=1}^k M_{ij}[\underline{y}_j, \bar{y}_j] \right) \cap \left( \sum_{j=n}^k M_{ij}[\underline{y}_j, \bar{y}_j] \right).$$

The level of uncertainty on these intervals can therefore be reduced for all vertex positions having a very small, or a very large (close to  $n$ ), rank in the considered vertex order. From the vector  $z$ , a solution  $x$  formed by singletons can be trivially extracted. Future works will be aimed at extending the present work to DG paradoxical instances in higher dimensions.

## Acknowledgments

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

## References

- [1] N.M. Freris, S.R. Graham, P.R. Kumar, *Fundamental Limits on Synchronizing Clocks Over Networks*, IEEE Transactions on Automatic Control **56**(6), 1352–1364, 2010.
- [2] S.B. Hengeveld, N. Rubiano da Silva, D.S. Gonçalves, P.H. Souto Ribeiro, A. Mucherino, *An Optical Processor for Matrix-by-Vector Multiplication: an Application to the Distance Geometry Problem in 1D*, Journal of Optics **24**(1), 015701, 2022.
- [3] L. Lamport, *Time, Clocks and the Ordering of Events in a Distributed System*, Communications of the ACM **21**, 558–564, 1978.
- [4] 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.
- [5] 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.
- [6] F. Messine, *Extensions of Affine Arithmetic: Application to Unconstrained Global Optimization*, Journal of Universal Computer Science **8**(11), 992–1015, 2002.
- [7] A. Mucherino, *On the Exact Solution of the Distance Geometry with Interval Distances in Dimension 1*. In: “Recent Advances in Computational Optimization”, S. Fidanova (Ed.), Studies in Computational Intelligence **717**, 123–134, 2018.
- [8] A. Mucherino, *An Analysis on the Degrees of Freedom of Binary Representations for Solutions to Discretizable Distance Geometry Problems*. In “Recent Advances in Computational Optimization”, S. Fidanova (Ed.), Studies in Computational Intelligence **986**, 251–255, 2022.
- [9] J. Saxe, *Embeddability of Weighted Graphs in  $k$ -Space is Strongly NP-hard*, Proceedings of 17<sup>th</sup> Allerton Conference in Communications, Control and Computing, 480–489, 1979.
- [10] N.T. Shaked, S. Messika, S. Dolev, J- Rosen, *Optical Solution for Bounded NP-complete Problems*, Applied Optics **46**(5), 711–724, 2007.
- [11] A. Singer, *Angular Synchronization by Eigenvectors and Semidefinite Programming*, Applied and Computational Harmonic Analysis **30**(1), 20–36, 2011.
- [12] P. Verissimo, M. Raynal, *Time in Distributed System Models and Algorithms*. In: “Advances in Distributed Systems, Advanced Distributed Computing: From Algorithms to Systems”, S.K. Shrivastava, S. Krakowiak (Eds.), Springer, 1–32, 1999.
- [13] Y-C. Wu, Q. Chaudhari, E. Serpedin, *Clock Synchronization of Wireless Sensor Networks*, IEEE Signal Processing Magazine **28**(1), 124–138, 2011.