New methods for the Distance Geometry Problem

Gustavo Dias and Leo Liberti [1,2]

CNRS LIX, Ecole Polytechnique, 91128 Palaiseau, France

Abstract

Given an integer K and a simple edge-weighted undirected graph $G = (V, E)$, the Distance Geometry Problem questions the existence of a vertex realization function $V \to \mathbb{R}^K$ such that each vertex pair adjacent to an edge is placed at a distance which is equal to the edge weight. This problem has many applications to science and engineering, and many methods have been proposed to solve it. We propose some new formulation-based methods.

Keywords: DGP, Semidefinite Programming, Diagonally dominant matrices.

1 Introduction

The problem studied in this paper is the

> Distance Geometry Problem (DGP). Given an integer $K \geq 1$ and a simple, edge-weighted, undirected graph $G = (V, E, d)$, where $d : E \to \mathbb{R}_+$, verify the existence of a vertex *realization* function $x : V \to \mathbb{R}^K$ such that:

$$\forall \{i, j\} \in E \quad \|x_i - x_j\| = d_{ij}. \tag{1}$$

[1] Gustavo Dias is financially supported by a CNPq PhD thesis award.

[2] Email: {dias,liberti}@lix.polytechnique.fr

A recent survey on the DGP with the Euclidean norm is given in [2]. The DGP is **NP**-hard, by reduction from Partition. Three well-known applications are to clock synchronization ($K = 1$), sensor network localization ($K = 2$), and protein conformation ($K = 3$). A related problem, the Distance Matrix Completion Problem (DMCP), asks whether a partially defined matrix can be completed to a distance matrix. The difference is that while K is part of the input in the DGP, it is part of the output in the DMCP, in that a realization into *any* Euclidean space which allows the computation of the missing distances provides a certificate. It is remarkable that, by virtue of this seemingly minor difference, it is not known whether the Euclidean DMCP (EDMCP) is in **P** or **NP**-hard. It is currently thought to be "between the two classes".

In this short paper we sketch several new formulation-based methods for solving the DGP.

2 MILP formulations for 1- and ∞-norms

To the best of our knowledge, no method for solving DGPs with the 1- and ∞-norm currently exists.[3] Yet, since both norms can be linearized exactly, it is not difficult to derive Mixed-Integer Linear Programming (MILP) formulations for either. We first re-write Eq. (1) as follows:

$$\min_x \sum_{\{i,j\}\in E} |\, \|x_i - x_j\|_\ell - d_{ij}\,|, \tag{2}$$

for $\ell \in \{1, \infty\}$. Then, for $\ell = 1$ we write:

$$\min_x \sum_{\{i,j\}\in E} \left| \sum_{k\le K} |x_{ik} - x_{jk}| - d_{ij} \right|,$$

and equivalently for $\ell = \infty$. For $\ell = 1$, we apply some standard absolute value reformulations to obtain a MILP. The case $\ell = \infty$ is slightly more involved, but still easy to model. These formulations can be solved using any off-the-shelf MILP solver.

[3] We shall gladly take corrections to this statement!

3 SDP formulations for the 2-norm

Many Semidefinite Programming (SDP) formulations for the 2-norm case are well known from the sensor network localization literature (see [2]). Note that a realization x can be represented in matrix form by an $n \times K$ matrix where $n = |V|$, and where each of the n rows is a vector $x_i \in \mathbb{R}^K$ which places vertex $i \in V$. The Euclidean DGP (EDGP) can be modelled as follows:

$$\forall \{i,j\} \in E \quad \|x_i - x_j\|_2^2 = x_i \cdot x_i + x_j \cdot x_j - 2x_i \cdot x_j = d_{ij}^2. \tag{3}$$

Since the EDGP involves sums $x_i \cdot x_j$ of quadratic terms for various $i, j \in V$, we can linearize these sums by replacing them with variables X_{ij} organized in an $n \times n$ matrix, i.e. $X = xx^\top$. This provides an easy reformulation of Eq. (3):

$$\begin{aligned} \forall \{i,j\} \in E \quad X_{ii} + X_{jj} - 2X_{ij} &= d_{ij}^2 \\ X &= xx^\top. \end{aligned}$$

The rank constraint $X = xx^\top$ can be readily relaxed to $X \succeq xx^\top$, which in turn can be written as the Schur complement $\begin{pmatrix} I_K & x^\top \\ x & X \end{pmatrix} \succeq 0$, yielding a well-known pure feasibility SDP formulation. Usually, in an attempt to reduce the rank of the solution X, many papers propose the objective function $\min \mathsf{trace}(X)$. Some empirical experience suggests that this particular objective is suitable for instances from the sensor network localization application, since the so-called "anchor nodes" are usually evenly scattered among the sensors, and play a regularization role. For protein conformation instances, on the other hand, trace minimization yields poor results. A better formulation turns out to be:

$$\begin{aligned} \min_X \sum_{\{i,j\} \in E} (X_{ii} + X_{jj} - 2X_{ij}) & \\ \forall \{i,j\} \in E \quad X_{ii} + X_{jj} - 2X_{ij} &\ge d_{ij}^2 \\ X - xx^\top &\succeq 0. \end{aligned}$$

For the EDMCP, where the rank is of no importance, we only require that X should be the Gram matrix of a realization x (of any rank). Since Gram matrices are exactly positive semidefinite (PSD) matrices, the formulation is simplified to

$$\begin{aligned} \forall \{i,j\} \in E \quad X_{ii} + X_{jj} - 2X_{ij} &= d_{ij}^2 \\ X &\succeq 0. \end{aligned}$$

4 Diagonally dominant approximation

One serious drawback of SDP is that current solving technology is limited to instances of fairly low sizes. A. Ahmadi recently remarked [1] that diagonal dominance provides a useful tool for inner approximating the PSD cone. A matrix (Y_{ij}) is *diagonally dominant* (DD) if

$$\forall i \le n \quad Y_{ii} \ge \sum_{j \ne i} |Y_{ij}|. \tag{4}$$

It follows from Gershgorin's theorem that diagonally dominant matrices are PSD (the converse does not hold, hence the inner approximation). The crucial observation is that Eq. (4) is easy to linearize as follows:

$$\begin{aligned} \forall i \le n \quad & \sum_{j \ne i} T_{ij} \le Y_{ii} \\ \forall i, j \le n \quad & -T_{ij} \le Y_{ij} \le T_{ij}. \end{aligned}$$

This yields a new LP formulation related to the EDGP:

$$\left.\begin{array}{lr} \min\limits_{X,Y,T} \sum\limits_{\{i,j\} \in E} (X_{ii} + X_{jj} - 2X_{ij}) & \\ \forall \{i,j\} \in E & X_{ii} + X_{jj} - 2X_{ij} \ge d_{ij}^2 \\ & \begin{pmatrix} I_K & x^\top \\ x & X \end{pmatrix} = Y \\ \forall i \le n + K & \sum\limits_{\substack{j \le n+K \\ j \ne i}} T_{ij} \le Y_{ii} \\ & -T \le Y \le T. \end{array}\right\} \tag{5}$$

Let $\mathcal{D}(U) = \{U^\top M U \mid M \text{ is DD}\}$. The approximation Eq. (5) can be iteratively improved by requiring that $Y \in \mathcal{D}(U)$ with $U^0 = I$ and $U^h = \sqrt{\bar{Y}^{h-1}}$ for all $h > 0$, where $\bar{Y}^{h-1}$ is the solution of Eq. (5) at the previous iteration $h - 1$.

References

[1] Ahmadi, A. and G. Hall, *Sum of squares basis pursuit with linear and second order cone programming*, Technical Report 1510.01597v1, arXiv (2015).

[2] Liberti, L., C. Lavor, N. Maculan and A. Mucherino, *Euclidean distance geometry and applications*, SIAM Review **56** (2014), pp. 3–69.