



**HAL**  
open science

## Barvinok's naive algorithm in Distance Geometry

Leo Liberti, Ky Vu

► **To cite this version:**

Leo Liberti, Ky Vu. Barvinok's naive algorithm in Distance Geometry. *Operations Research Letters*, 2018, 46 (5), pp.476-481. 10.1016/j.orl.2018.06.006 . hal-02105083

**HAL Id: hal-02105083**

**<https://hal.science/hal-02105083>**

Submitted on 20 Apr 2019

**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.

# Barvinok’s naive algorithm in Distance Geometry<sup>☆</sup>

Leo Liberti

CNRS LIX, Ecole Polytechnique, 91128 Palaiseau, France

Ky Vu

Chinese University of Hong Kong, P.R. China

---

## Abstract

In 1997, A. Barvinok gave a probabilistic algorithm to derive a near-feasible solution of a quadratically (equation) constrained problem from its semidefinite relaxation. We generalize this algorithm to handle matrix variables instead of vectors, and to two-sided inequalities instead of equations. We derive a heuristic for the distance geometry problem, and showcase its computational performance on a set of instances related to protein conformation.

*Keywords:* distance geometry, concentration of measure, protein structure

---

## 1. Introduction

We consider the following

Distance Geometry Problem (DGP). Given an integer  $K > 0$  and a simple graph  $G = (V, E, d)$  where  $d : E \rightarrow \mathbb{R}_+$ , decide whether there is a realization  $x : V \rightarrow \mathbb{R}^K$  such that

$$\forall \{i, j\} \in E \quad \|x_i - x_j\|_2^2 = d_{ij}^2. \quad (1)$$

Let  $n = |V|$  and  $m = |E|$ . The DGP is well known in the literature [1, 2], as it serves as a model for several applications (e.g. to sensor networks [3], molecular conformation [4, 5] and more [6]). A natural extension of the DGP is the interval version (denoted by  $i$ DGP), where  $d : E \rightarrow \mathbb{I}\mathbb{R}_+$  associates intervals (instead of scalars) to edges. This variant is the one often used in applications, since intervals naturally model data uncertainty and noise [7].

We adapt the *naive algorithm* proposed by A. Barvinok in [8, §5] to the DGP setting. Barvinok’s main insight is that the solutions of a Semidefinite Programming (SDP) relaxation are “not too far” from the feasible set of the Quadratically Constrained Problem (QCP) which gives rise to the SDP relaxation: it suffices to factor the SDP solution and multiply it by a random vector having components sampled from a normal distribution. The very natural idea we propose is to use the naive algorithm first, then use this approximate solution as a starting point for a local Nonlinear Programming (NLP) solver deployed on the

original QCP, hoping it will converge to a realization satisfying the feasibility of Eq. (1).

Barvinok’s naive algorithm is a randomized algorithm based on the concentration of measure phenomenon. It applies to SDP relaxations of systems of quadratic equations, i.e. pure feasibility, equation-only QCPs having vector solutions, or, equivalently,  $n \times 1$  matrix solutions. This algorithm cannot be natively applied to the DGP, since a realization  $x$  is naturally represented by an  $n \times K$  matrix  $x = (x_{ik} \mid i \leq n, k \leq K)$  the  $i$ -th row of which is the position vector of vertex  $i \in V$  in  $\mathbb{R}^K$ . This matrix may in general have rank greater than one. In [8], Barvinok gives a proof sketch which only applies to  $n \times 1$  matrices. Of course, an  $n \times K$  matrix can also be represented as an  $n' \times 1$  matrix where  $n' = nK$ , but this would entail a QCP (and hence also an SDP relaxation) with  $nK \times nK$  data matrices, which is practically prohibitive to solve. Moreover, the naive algorithm cannot be applied to the  $i$ DGP since it is constrained by quadratic inequalities rather than equations.

We make two contributions in this paper, one theoretical and the other computational. (i) We propose a generalization of Barvinok’s result [8] in two directions: the  $n \times K$  case, and the case of QCP inequalities of the form  $d^L \leq x^T Q x \leq d^U$  (where  $x$  is  $n \times K$ ). (ii) We establish the practical usefulness of Barvinok’s result on a set of medium and large-scale DGP instances extracted from the Protein Data Bank (PDB) [9]. We also remark that our proof is detailed, and fills many gaps in Barvinok’s “proof sketch” found in [8]. The rest of this paper is organized as follows. In Sect. 2 we present Barvinok’s naive algorithm. In Sect. 3 we state and prove our generalization of Barvinok’s concentration of measure result. In Sect. 5 we discuss our computational results.

---

<sup>☆</sup>We are grateful to A. Barvinok for helpful suggestions and discussions, as well as to an anonymous referee for helping us improve the paper. The first author (LL) has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement n. 764759 (ITN “MINOA”).

*Email addresses:* [liberti@lix.polytechnique.fr](mailto:liberti@lix.polytechnique.fr) (Leo Liberti), [vukhacky@gmail.com](mailto:vukhacky@gmail.com) (Ky Vu)

## 2. The naive algorithm

Consider a system of quadratic equations:

$$\forall i \leq m \quad x^\top Q^i x = a_i. \quad (2)$$

Barvinok's naive algorithm works as follows:

1. solve the SDP relaxation  $[\forall i \leq m \text{trace}(Q^i X) = a_i \wedge X \geq 0]$  of Eq. (2), get a solution  $X^*$ ;
2. factor  $X^* = TT^\top$ ;
3. sample each component of a vector  $y \in \mathbb{R}^n$  from the standard Gaussian distribution  $N(0, 1)$ ;
4. let  $x' = Ty$ .

A concentration of measure argument shows that, with probability at least 0.9,

$$\forall i \leq m \quad \text{dist}(x', A_i) \leq c \sqrt{\|X^*\|_2 \ln(n)},$$

where  $A_i = \{x \in \mathbb{R}^n \mid Q^i x = a_i\}$ ,  $\text{dist}(\cdot, \cdot)$  is the Euclidean distance from a point to a set,  $\|X^*\|_2$  is the largest eigenvalue of  $X^*$ , and  $c$  is a constant which only depends (linearly) on  $\log_n m$ .

As mentioned in the introduction, applying this algorithm to the DGP requires the representation of  $n \times K$  realization matrices using vectors in  $\mathbb{R}^{nK}$ , which, in turn, requires each  $Q^i$  to be  $nK \times nK$ , thereby increasing the size of matrices involved in the DGP solution. This provides our first motivation for extending the naive algorithm to work with realizations  $x$  represented naturally as  $n \times K$  matrices. The second motivation is given by the desire of solving the *i*DGP rather than the (exact) DGP. *i*DGP solutions are realizations satisfying  $L_{ij}^2 \leq \|x_i - x_j\|_2^2 \leq U_{ij}^2$  for each edge  $\{i, j\} \in E$ , where  $[L_{ij}, U_{ij}]$  is an interval weight on  $\{i, j\}$ .

## 3. Extension of the naive algorithm to the $n \times K$ case

Our proposed generalization of the naive algorithm, so that it can find  $n \times K$  realization matrices, is very simple: we replace Step 3 with:

3. sample each component of an  $n \times K$  matrix  $y$  from the standard Gaussian distribution  $N(0, 1/K)$  (where  $\frac{1}{K}$  is the variance).

We shall see that this has the desired effect. We call the algorithm obtained by this replacement *extended naive algorithm*.

### 3.1. Concentration of measure and the isoperimetric inequality

In [8], Barvinok gives the following definition:

**Definition 1.** A measure space  $(\mathcal{X}, \mu)$  has the measure concentration property if there exist positive constants  $c$  and  $C$  (called concentration constants) such that the following holds: for any Lipschitz function  $f : \mathcal{X} \rightarrow \mathbb{R}$  with  $|f(x) - f(y)| < M \text{dist}(x, y)$  for some  $M > 0$  and all  $x, y \in \mathcal{X}$ , and for any  $\varepsilon > 0$ :

$$\mu\{x \in \mathcal{X} \mid |f(x) - \mathbb{E}(f)| > \varepsilon\} < c e^{-C\varepsilon^2/M^2} \quad (3)$$

where  $\mathbb{E}(f)$  is the expectation of  $f$  defined as  $\mathbb{E}(f) = \int_{\mathcal{X}} f(x) d\mu$ .

A notable example of measure concentration is given by the Gaussian distribution in  $\mathbb{R}^n$ . Let  $\mu$  be the probability measure on  $\mathbb{R}^n$  with Gaussian density defined by  $\phi(x) = (2\pi)^{-n/2} e^{-\|x\|^2/2}$ . It is a well known fact that there is measure concentration with  $c = 2$  and  $C = 2/\pi^2$ .

The well-known isoperimetric inequality states that, if  $\mu$  has the measure concentration property, then for any closed set  $A$  that is large enough, then even a small neighborhood  $A(\varepsilon)$  of  $A$  (defined as  $A(\varepsilon) = \{x \in \mathcal{X} \mid \text{dist}(x, A) \leq \varepsilon\}$  for any  $\varepsilon \geq 0$ ) can contain almost the whole measure of the space  $\mathcal{X}$ . This result is stated as [8, Lemma 1.1.2]; the proof is said to “go along the lines of Sect. 7.6 in [10]”. Here we will present a direct proof for it, using elementary arguments.

**Proposition 2.** Assume  $(\mathcal{X}, \mu)$  has measure concentration with constants  $c, C$ , and let  $A \subseteq \mathcal{X}$  be a closed set. For any given  $p \in (0, 1)$ , if  $\varepsilon \geq \frac{\sqrt{\log c - \log \mu(A)} + \sqrt{\log c - \log p}}{\frac{1}{2} \sqrt{C}}$  then we have  $\mu(A(\varepsilon)) > 1 - p$ .

*Proof.* Let us define the function  $f : \mathcal{X} \rightarrow \mathbb{R}_+$  by  $f(x) = \text{dist}(x, A) = \min_{y \in A} \text{dist}(x, y)$ . It is well-known that  $f$  is a 1-Lipschitz function. Then the measure concentration property (3) of  $(\mathcal{X}, \mu)$  implies that, for the given  $\varepsilon > 0$ :

$$\mu\{x \in \mathcal{X} \mid |f(x) - \mathbb{E}(f)| > \varepsilon/2\} < c e^{-C\varepsilon^2/4}. \quad (4)$$

Note that, by the choice of  $\varepsilon$  in the hypothesis, we have  $c e^{-C\varepsilon^2/4} < \max\{\mu(A), p\}$ , since it is equivalent to  $\varepsilon > \max\left(\frac{\sqrt{\log c - \log \mu(A)}}{\frac{1}{2} \sqrt{C}}, \frac{\sqrt{\log c - \log p}}{\frac{1}{2} \sqrt{C}}\right)$ . Thus, (4) implies that

$$\mu\{x \in \mathcal{X} \mid |f(x) - \mathbb{E}(f)| \leq \varepsilon/2\} > 1 - \mu(A). \quad (5)$$

On the other hand

$$\mu\{x \in \mathcal{X} \mid f(x) = 0\} = \mu\{x \in \mathcal{X} \mid \text{dist}(x, A) = 0\} = \mu(A). \quad (6)$$

Therefore, by (5) and (6), there must exist at least one  $x \in A$  such that  $|f(x) - \mathbb{E}(f)| \leq \frac{\varepsilon}{2}$ . But in this case,  $f(x) = 0$ , and thus we have  $\mathbb{E}(f) \leq \frac{\varepsilon}{2}$ . Now we have  $\mu(A_\varepsilon^c) = \mu\{x \in \mathcal{X} \mid f(x) > \varepsilon\} \leq \mu\{x \in \mathcal{X} \mid |f(x) - \mathbb{E}(f)| > \frac{\varepsilon}{2}\} < c e^{-\frac{C\varepsilon^2}{4}} < p$ , where  $A_\varepsilon^c$  is the complement of the set  $A_\varepsilon$ . This concludes the proof.  $\square$

### 3.2. Analysis of the naive algorithm

In this section we prove a generalization of Barvinok's analysis of his naive algorithm which applies to  $n \times K$  matrices and systems of quadratic inequalities such as:

$$\forall i \leq m \quad L_i \leq x^\top Q^i x \leq U_i. \quad (7)$$

Consider the SDP relaxation of Eq. (7):

$$\left. \begin{array}{l} \forall i \leq m \quad L_i \leq \text{trace}(Q^i X) \leq U_i \\ X \geq 0. \end{array} \right\} \quad (8)$$

For technical reasons we assume that the number  $m$  of constraints of Eq. (2) is bounded by a polynomial of  $nK$ , i.e.  $m \leq (nK)^s$  for some fixed number  $s > 0$ . This is a natural

assumption in many cases, e.g. we have  $s = 2$  in the DGP. With this assumption, then  $\log m$  is proportional to  $\log(nK)$ .

We remark that the quadratic form in Eq. (7) can be written as  $\forall i \leq m \quad Q^i(x) = x^\top Q^i x = \sum_{1 \leq j, \ell \leq n} q_{j\ell}^i x_j \cdot x_\ell$ , where  $q_{j\ell}^i$  are real coefficients and  $z \cdot y$  stands for the inner product of the vectors  $z$  and  $y$ . By the naive algorithm, we let  $x' = Ty$ , where  $T$  is an  $n \times n$  factor of the SDP solution and  $y$  is an  $n \times K$  random matrix sampled from the Gaussian distribution  $N(0, \frac{1}{K})$ . Its  $j$ -th row vector is defined by  $x'_j = T_j y$  for  $1 \leq j \leq n$ , where  $T_j$  is the  $j$ -th row of  $T$ . Our objective is to prove that  $x'$  is an approximate solution for the quadratic system (7). In other words, let  $\mathcal{A}$  be the set of all feasible solution for (7); we shall then prove that  $\text{dist}(x', \mathcal{A})$  is small with high probability. The following lemma states that the expectation of quadratic functions  $Q^i(x')$  satisfies the system (7).

**Proposition 3.** *Let  $\mathcal{X}$  be the set of all  $n \times K$  real matrices. Given above notations, we have*

$$\int_{\mathcal{X}} Q^i(Ty) \mu(dy) = \text{trace}(Q^i X^*) \in [L_i, U_i].$$

*Proof.* For each  $1 \leq i \leq m$  we have

$$\begin{aligned} \int_{\mathcal{X}} Q^i(Ty) \mu(dy) &= \int_{\mathcal{X}} \sum_{1 \leq j, \ell \leq n} q_{j\ell}^i (T_j y) (T_\ell y)^\top \mu(dy) \\ &= \sum_{1 \leq j, \ell \leq n} q_{j\ell}^i \int_{\mathcal{X}} T_j (y y^\top) T_\ell^\top \mu(dy). \end{aligned}$$

Since entries in the matrix  $y$  are sampled independently, for each  $j, \ell \leq n$  and  $h, k \leq K$ , we have:

$$\int_{\mathcal{X}} y_{jk} y_{\ell h} \mu(dy) = \begin{cases} \frac{1}{K} & \text{if } j = \ell \text{ and } k = h, \\ 0 & \text{otherwise.} \end{cases}$$

For each  $j, \ell \leq n$ , moreover, we note that  $T_j (y y^\top) T_\ell^\top$  is a linear combination of all terms of the form  $y_{jk} y_{\ell h}$  (for  $h, k \leq K$ ). Therefore, in the integral below, we can simply cancel all the quantities  $y_{ik} y_{jh}$  out whenever  $k \neq h$  or  $i \neq j$ . This yields:

$$\int_{\mathcal{X}} T_j (y y^\top) T_\ell^\top \mu(dy) = \int_{\mathcal{X}} T_j \begin{pmatrix} \|y\|^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \|y\|^2 \end{pmatrix} T_\ell^\top \mu(dy) \quad (9)$$

Since, for all  $p \leq n$ ,  $\|y_p\|^2 = \sum_{k=1}^K y_{pk}^2$ , Eq. (9) can then be rewritten as

$$\begin{aligned} \sum_{p=1}^n \int_{\mathcal{X}} T_{jp} T_{\ell p} \|y_p\|^2 \mu(dy) &= \sum_{p=1}^n \sum_{k=1}^K \int_{\mathcal{X}} T_{jp} T_{\ell p} y_{pk}^2 \mu(dy) \\ \sum_{p=1}^n \sum_{k=1}^K T_{jp} T_{\ell p} \underbrace{\int_{\mathcal{X}} y_{pk}^2 \mu(dy)}_{=1/K} &= \frac{1}{K} \sum_{p=1}^n \sum_{k=1}^K T_{jp} T_{\ell p} = T_j T_\ell^\top = X_{j\ell}^*. \end{aligned}$$

Therefore, we have

$$\int_{\mathcal{X}} Q^i(Ty) \mu(dy) = \sum_{j, \ell \leq n} q_{j\ell}^i X_{j\ell}^* = \text{trace}(Q^i X^*) \in [L_i, U_i] \quad (10)$$

due to the feasibility of  $X^*$  for the system (8).  $\square$

Note that, in Eq. (10),  $\text{trace}(Q^i X^*)$  is also the trace of the quadratic form  $Q^i(Ty)$  with respect to the variable  $y$ : indeed, we have  $Q^i(Ty) = (Ty)^\top Q^i(Ty) = y^\top (T^\top Q^i T) y$ , whence  $\text{trace}(Q^i(Ty)) = \text{trace}(T^\top Q^i T) = \text{trace}(Q^i T T^\top) = \text{trace}(Q^i X^*)$ .

In order to proceed, we will need to use the following result, which was proved in Barvinok's paper (a Fact in page 51). Here we extend the result to the  $n \times K$  dimension case, and provide a proof for completeness.

**Lemma 4.** *Let  $Q(y) = y^\top Q y$  be a quadratic form on  $y \in \mathbb{R}^{n \times K}$ ,  $A^- = \{y \in \mathbb{R}^{n \times K} \mid Q(y) \leq \text{trace}(Q)\}$ , and  $A^+ = \{y \in \mathbb{R}^{n \times K} \mid Q(y) \geq \text{trace}(Q)\}$ . Then for all sufficiently large  $n$  we have  $\mu(A^-) > \frac{1}{3nK}$  and  $\mu(A^+) > \frac{1}{3nK}$ .*

*Proof.* Denote  $nK$  by  $N$ . Without loss of generality, we can assume that  $\text{trace}(Q) \geq 0$ . Consider a coordinate system  $(x_1, \dots, x_N)$  such that  $Q$  is diagonal, i.e.  $Q(x) = c_1 x_1^2 + \dots + c_N x_N^2$ . Now consider a norm-preserving cyclic transform  $\sigma : (x_1, x_2, \dots, x_N) \rightarrow (x_2, x_3, \dots, x_N, x_1)$ . With the convention that  $x_k = x_{k-N}$  if  $k > N$ , then we have for all vectors  $x \in \mathbb{R}^N$ :

$$\sum_{k=1}^N Q(\sigma^k(x)) = \sum_{k=1}^N \sum_{i=1}^N c_i x_{i+k}^2 = \sum_{i=1}^N c_i \sum_{k=1}^N x_{i+k}^2 = \text{trace}(Q) \|x\|^2.$$

Let  $B_N = \{x \in \mathbb{R}^N \mid \|x\| \leq \sqrt{N}\}$ . Then, for any  $x \in B_N$ , we have  $\sum_{k=1}^N Q(\sigma^k(x)) \leq N \text{trace}(Q)$ . It follows that there must be at least one index  $k \geq 0$  such that  $Q(\sigma^k(x)) \leq \text{trace}(Q)$ , which means that  $\sigma^k(x) \in A^-$ . The ball  $B_N$  admits a partition into equivalent classes based on  $\sigma$  (i.e. each class of the form  $\{x, \sigma(x), \dots, \sigma^{N-1}(x)\}$ ). Therefore  $\mu(A^-) \geq \frac{\mu(B_N)}{N}$ . Similarly, we can also prove that  $\mu(A^+) \geq \frac{1 - \mu(B_N)}{N}$ . It is known that  $\lim_{N \rightarrow \infty} \mu(B_N) = \frac{1}{2}$ : indeed, this limit is equal to

$$\lim_{N \rightarrow \infty} \int_{\|x\|^2 \leq N} \frac{1}{(2\pi)^{N/2}} e^{-\frac{\|x\|^2}{2}} dx = \frac{1}{2} = \lim_{N \rightarrow \infty} \text{Prob}(X_1 + \dots + X_N \leq N),$$

where  $X_1, \dots, X_N \sim \chi^2(1)$  are i.i.d. random variables. Denote by  $\mu$  and  $\sigma$  the expectation and variance of  $X_1$  (note that  $\mu = 1$  since it is equal to the expectation of the standard normal distribution). The above limit can now be written as  $\lim_{N \rightarrow \infty} \text{Prob}\left(\frac{X_1 + \dots + X_N - \mu}{\sigma \sqrt{N}} \leq 0\right) = \Phi(0) = \frac{1}{2}$ , where  $\Phi(x)$  is the standard normal cdf evaluated at  $x$  (due to the Central Limit Theorem). Now, since  $\lim_{N \rightarrow \infty} \mu(B_N) = \frac{1}{2}$ , there is  $N$  large enough such that we  $\frac{1}{3} \leq \mu(B_N) \leq \frac{2}{3}$ . Combining with the earlier inequalities  $\mu(A^-) \geq \frac{\mu(B_N)}{N}$  and  $\mu(A^+) \geq \frac{1 - \mu(B_N)}{N}$ , the required inequalities follow.  $\square$

We now state and prove the extension of Barvinok's concentration of measure result to the  $n \times K$  case.

**Theorem 5.** *Let  $X^*$  be a solution of the SDP relaxation (8) and  $x'$  be the output of the extended naive algorithm. For each  $i \leq m$  we introduce the sets  $\mathcal{D}_i = \{x \in \mathbb{R}^{n \times K} \mid Q^i(x) = \text{trace}(Q^i X^*)\}$ . Then for large enough  $n$ , with probability at least 0.9,  $x'$  satisfies  $\text{dist}(x', \mathcal{D}_i) \leq C_0 \sqrt{\|X^*\|_2 \ln(nK)}$ , where  $C_0$  is a constant.*

*Proof.* We define the following sets for each  $i \leq m$ :

$$\begin{aligned} A_i^- &= \{y \in \mathbb{R}^{n \times K} \mid Q^i(Ty) \leq \text{trace}(Q^i X^*)\} \\ A_i^+ &= \{y \in \mathbb{R}^{n \times K} \mid Q^i(Ty) \geq \text{trace}(Q^i X^*)\} \\ A_i &= \{y \in \mathbb{R}^{n \times K} \mid Q^i(Ty) = \text{trace}(Q^i X^*)\}. \end{aligned}$$

From Lemma 4, we have  $\mu(A_i^-) \geq \frac{1}{3nK}$  and  $\mu(A_i^+) \geq \frac{1}{3nK}$  for all sufficiently large  $n$ . Now we can apply Proposition 2 for  $p = \frac{1}{20m}$  to obtain  $\mu(A_i^-(\varepsilon)) \geq 1 - \frac{1}{20m}$  whenever  $\varepsilon \geq \frac{1}{\sqrt{C}} \left( \sqrt{\ln c - \ln \mu(A_i^-)} + \sqrt{\ln c - \ln \frac{1}{20m}} \right)$  (the measure concentration constants can be taken to be  $c = 2$  and  $C = 2/\pi^2$ ). Since

$$\begin{aligned} \sqrt{\ln c - \ln \mu(A_i^-)} &\leq \sqrt{\ln c - \ln \left( \frac{1}{3nK} \right)} = \sqrt{\ln c + \ln(3nK)} = O(\sqrt{\ln(nK)}) \\ \sqrt{\ln c - \ln \frac{1}{20m}} &= \sqrt{\ln 20c + \ln(m)} \leq \sqrt{\ln 20c + \ln(nK)^s} = O(\sqrt{\ln(nK)}), \end{aligned}$$

if we choose some  $\varepsilon = O(\sqrt{\ln(nK)})$  with a large enough constant, then we have  $\mu(A_i^-(\varepsilon)) \geq 1 - \frac{1}{20m}$  and similarly  $\mu(A_i^+(\varepsilon)) \geq 1 - \frac{1}{20m}$  for  $i \leq m$ . By the union bound we have  $\mu(A_i^+(\varepsilon) \cap A_i^-(\varepsilon)) \geq 1 - \frac{1}{10m}$ .

We now claim that  $A_i^+(\varepsilon) \cap A_i^-(\varepsilon) = A_i(\varepsilon)$  for all  $i \leq m$ . Indeed, since  $A_i \subset A_i^+$  and  $A_i \subset A_i^-$ , we have  $A_i(\varepsilon) \subseteq A_i^+(\varepsilon) \cap A_i^-(\varepsilon)$ . For the other inclusion, take any  $\hat{y} \in A_i^+(\varepsilon) \cap A_i^-(\varepsilon)$ . It means that there exists  $y_i^+ \in A_i^+$  and  $y_i^- \in A_i^-$  such that  $\|\hat{y} - y_i^+\| \leq \varepsilon$  and  $\|\hat{y} - y_i^-\| \leq \varepsilon$ . By definition it then follows that

$$Q^i(Ty_i^-) \leq \text{trace}(Q^i X^*) \text{ and } Q^i(Ty_i^+) \geq \text{trace}(Q^i X^*).$$

Since  $Q^i(Ty)$  is a continuous function of  $y$ , on the interval defined by two endpoints  $y_i^+$  and  $y_i^-$ , there must exist some  $y'$  such that  $Q^i(Ty') = \text{trace}(Q^i X^*)$ , i.e.  $y' \in A_i$ . The claim follows since

$$\|\hat{y} - y'\| \leq \max\{\|\hat{y} - y_i^+\|, \|\hat{y} - y_i^-\|\} \leq \varepsilon.$$

From the claim above,  $\mu(A_i(\varepsilon)) \geq 1 - \frac{1}{10m}$  for  $i \leq m$ . Applying the union bound for all  $m$  sets  $A_i(\varepsilon)$  with  $i \leq m$ , we have  $\mu\left(\bigcap_{i=1}^m A_i(\varepsilon)\right) \geq 1 - m \frac{1}{10m} = 0.9$ . Hence, our randomly generated point  $y$  is in  $A_i(\varepsilon)$  for each  $i \leq m$  with probability at least 90%. Therefore

$$x' = Ty \in T(A_i(\varepsilon)) \subseteq \mathcal{D}_i + \varepsilon \|T\| \subseteq \mathcal{D}_i(\varepsilon \|T\|) \subseteq \mathcal{D}_i(\varepsilon \sqrt{\|X^*\|_2}),$$

which proves the theorem.  $\square$

We remark that in Thm. 5 the constant  $C_0$  depends on  $s$  (assumed fixed).

### 3.3. The DGP case: removing the dependence on $K$

Note that the results obtained above are for the general quadratic case. For the DGP case, the problem has a specific block structure which allows us to improve upon them. We will show in this section that the dependence on  $K$  in Lemma 4 and Theorem 5 can be removed.

The equality constraint in Eq. (1) corresponding to  $\{i, j\} \in E$  can be explicitly written as  $\sum_k x_{ik}^2 + \sum_k x_{jk}^2 - 2 \sum_k x_{ik} x_{jk} = d_{ij}^2$ . This is a quadratic form on the  $n \times K$  matrix variable  $x$ . If we

write it in matrix form  $x^T Q_{ij} x = d_{ij}^2$ , the matrix  $Q_{ij}$  has the regular structure

$$Q_{ij} = \begin{pmatrix} Q_{ij}^n & 0 & 0 & \dots & 0 \\ 0 & Q_{ij}^n & 0 & \dots & 0 \\ 0 & 0 & Q_{ij}^n & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & Q_{ij}^n \end{pmatrix}$$

with  $K$  diagonal blocks, where each for each  $i, j$   $Q_{ij}^n$  is an  $n \times n$  matrix with zeros everywhere except for the entries  $(i, i), (i, j), (j, i), (j, j)$  set to  $\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ . Now we can rewrite Lemma 4 for this specific block matrix, in which the dependence on  $K$  is removed:

**Lemma 6.** *Let  $Q(y) = y^T Q y$  be a quadratic form on  $y \in \mathbb{R}^{n \times K}$ , where  $Q$  consists of  $K$  diagonal identical symmetric blocks  $Q_n \in \mathbb{R}^{n \times n}$ , namely  $Q = \text{diag}(Q_n, \dots, Q_n)$ . Let  $A^- = \{y \in \mathbb{R}^{n \times K} \mid Q(y) \leq \text{trace}(Q)\}$  and  $A^+ = \{y \in \mathbb{R}^{n \times K} \mid Q(y) \geq \text{trace}(Q)\}$ . Then for all sufficiently large  $n$  we have  $\mu(A^-) > \frac{1}{3n}$  and  $\mu(A^+) > \frac{1}{3n}$ .*

*Proof.* Without loss of generality, we can assume that  $\text{trace}(Q) \geq 0$ . By changing of the coordinate system if necessary, we can assume that each block  $Q^n$  is diagonal, i.e.  $Q^n = \text{diag}(\lambda_1, \dots, \lambda_n)$ . Now consider an orthogonal transform  $\sigma$  that sends  $(x_1, x_2, \dots, x_{nK})$  to

$$(x_2, \dots, x_n, x_1; x_{n+2}, \dots, x_{2n}, x_{n+1}; x_{2n+2}, \dots, x_{3n}, x_{2n+1}; \dots; x_{nK}, x_{nK-n+1}).$$

Then  $\sigma$  generates a cyclic group of order  $n$ . Averaging over  $n$  shifts, we get  $\sum_{k=1}^n Q(\sigma^k(x)) = \frac{1}{K} \text{trace}(Q) \|x\|^2$ . Let  $N = nK$  and let  $B_N = \{x \in \mathbb{R}^N \mid \|x\| \leq \sqrt{N}\}$ . Then, for any  $x \in B_N$ , we have  $\sum_{k=1}^n Q(\sigma^k(x)) \leq \frac{N}{K} \text{trace}(Q) = n \text{trace}(Q)$ . It follows that there must be at least one index  $k \geq 0$  such that  $Q(\sigma^k(x)) \leq \text{trace}(Q)$ , which means that  $\sigma^k(x) \in A^-$ . The ball  $B_N$  admits a partition into equivalent classes based on  $\sigma$  (i.e. each class of the form  $\{x, \sigma(x), \dots, \sigma^{n-1}(x)\}$ ). Therefore  $\mu(A^-) \geq \frac{\mu(B_N)}{n}$ . Similarly, we can also prove that  $\mu(A^+) \geq \frac{1 - \mu(B_N)}{n}$ . From Lemma 4, we have proved that  $\lim_{N \rightarrow \infty} \mu(B_N) = \frac{1}{2}$ . Therefore, when  $N = nK$  is sufficiently large, then we have  $\mu\{y \in \mathbb{R}^{n \times K} \mid Q(y) \leq \text{trace}(Q)\} > \frac{1}{3n}$  and  $\mu\{y \in \mathbb{R}^{n \times K} \mid Q(y) \geq \text{trace}(Q)\} > \frac{1}{3n}$  as claimed.  $\square$

Now we can obtain an improvement for the result in Theorem 5. We will state it without proof here because the proof is exactly the same except for the application of Lemma 6 instead of Lemma 4.

**Theorem 7.** *Let  $X^*$  be a solution of the SDP relaxation (8) for the DGP (1) and  $x'$  be the output of the extended naive algorithm. For each  $i \leq m$  we introduce the set  $\mathcal{D}_i = \{x \in \mathbb{R}^{n \times K} \mid Q^i(x) = \text{trace}(Q^i X^*)\}$ . Then for large enough  $n$ , with probability at least 0.9,  $x'$  satisfies  $\text{dist}(x', \mathcal{D}_i) \leq C_0 \sqrt{\|X^*\|_2 \ln(n)}$ , where  $C_0$  is a constant.*

Note that the logarithmic term under the square root in the RHS was improved from  $nK$  to  $n$ .

#### 4. A different approach to the iDGP case

In this section, we argue that with appropriate assumptions on the solution  $X^*$  and the interval bounds  $[L_i, U_i]$ , we can obtain a different bound, which depends on  $\ln m$  instead of  $\ln nK$ , which might be. We will use the following result, which is proved in [11].

**Proposition 8.** *Let  $Q : \mathbb{R}^N \rightarrow \mathbb{R}$  be a positive semidefinite quadratic form, i.e.  $Q(x) = x^T Q x$  for a positive semidefinite matrix  $Q$ . Let  $\|Q\| = \sqrt{\mu_1^2 + \dots + \mu_N^2}$  be the Schatten 2-norm of  $Q$ , in which  $\mu_1, \dots, \mu_N$  are the eigenvalues of  $Q$ . Then*

1. For any  $\tau \geq 0$  we have

$$\mu(\{x \in \mathbb{R}^N \mid Q(x) < \text{trace}(Q) - \tau\}) \leq \exp\left(-\frac{\tau^2}{4\|Q\|^2}\right). \quad (11)$$

2. For any  $\tau \geq 0$  such that  $\tau\mu_i \leq \|Q\|^2$  for all  $i \leq N$ , we have

$$\mu(\{x \in \mathbb{R}^N \mid Q(x) > \text{trace}(Q) + \tau\}) \leq \exp\left(-\frac{\tau^2}{8\|Q\|^2}\right). \quad (12)$$

Now we apply this proposition to the naive algorithm. Since  $X^*$  is a feasible solution of the SDP relaxation of the iDGP, then by (8) we have  $L_i \leq \text{trace}(Q^i X^*) \leq U_i$  for all  $i \leq m$ . We define some parameters  $\lambda_i$  such that  $\lambda_i \leq \min\left\{\frac{U_i - \text{trace}(Q^i X^*)}{\text{trace}(Q^i X^*)}, \frac{\text{trace}(Q^i X^*) - L_i}{\text{trace}(Q^i X^*)}\right\}$  for  $i \leq m$ . By this definition, we have  $L_i \leq \text{trace}(Q^i X^*) - \lambda_i \text{trace}(Q^i X^*)$ . Therefore, by applying (11) for  $\tau = \lambda_i \text{trace}(Q^i X^*)$ , we have

$$\begin{aligned} \mu(\{y \in \mathbb{R}^{n \times K} \mid Q^i(Ty) < L_i\}) &\leq \mu(\{y \in \mathbb{R}^{n \times K} \mid Q^i(Ty) \\ &< \text{trace}(Q^i X^*) - \lambda_i \text{trace}(Q^i X^*)\}) \\ &\leq \exp\left(-\frac{\lambda_i^2 \text{trace}(Q^i X^*)^2}{4\|Q^i X^*\|^2}\right), \end{aligned}$$

which will be smaller than or equal to  $\frac{1}{3}$  if we have  $\lambda_i \geq \frac{\sqrt{4 \ln 3}}{\text{trace}(Q^i X^*)} \frac{\|Q^i X^*\|}{\text{trace}(Q^i X^*)}$ .

Next, we do the same with the inequality in (12). By the definition of  $\lambda_i$ , we have  $U_i \geq \text{trace}(Q^i X^*) + \lambda_i \text{trace}(Q^i X^*)$ , therefore by applying (12) for  $\tau = \lambda_i \text{trace}(Q^i X^*)$ , we have

$$\begin{aligned} \mu(\{y \in \mathbb{R}^{n \times K} \mid Q^i(Ty) > U_i\}) &\leq \mu(\{y \in \mathbb{R}^{n \times K} \mid Q^i(Ty) \\ &> \text{trace}(Q^i X^*) + \lambda_i \text{trace}(Q^i X^*)\}) \\ &\leq \exp\left(-\frac{\lambda_i^2 \text{trace}(Q^i X^*)^2}{8\|Q^i X^*\|^2}\right), \end{aligned}$$

which will be smaller than or equal to  $\frac{1}{\sqrt{3}}$  if we have  $\lambda_i \geq \frac{\sqrt{4 \ln 3}}{\text{trace}(Q^i X^*)} \frac{\|Q^i X^*\|}{\text{trace}(Q^i X^*)}$  (similarly to the previous case). Note that,

to ensure the assumption in (12) is satisfied, we have to further assume that  $\lambda_i \text{trace}(Q^i X^*) \sigma_{\max}(Q^i X^*) \leq \|Q^i X^*\|^2$  as a sufficient condition, where  $\sigma_{\max}(A)$  stands for the maximum singular value of  $A$ . In conclusion, the above conditions are equivalent to

$$2\sqrt{3} \leq \min\left\{\frac{U_i - \text{trace}(Q^i X^*)}{\|Q^i X^*\|}, \frac{\text{trace}(Q^i X^*) - L_i}{\|Q^i X^*\|}, \frac{\|Q^i X^*\|^2}{\text{trace}(Q^i X^*) \sigma_{\max}(Q^i X^*)}\right\}. \quad (13)$$

If this condition holds, then by using the union bound we have

$$\mu(\{y \in \mathbb{R}^{n \times K} \mid L_i \leq Q^i(Ty) \leq U_i\}) \geq 1 - \frac{1}{3} - \frac{1}{\sqrt{3}} = \frac{2 - \sqrt{3}}{3}. \quad (14)$$

**Proposition 9.** *Let  $x'$  be the output of the extended naive algorithm and assume that Eq. (13) holds. For each  $1 \leq i \leq m$ , denote  $\mathcal{D}_i = \{x \in \mathbb{R}^{n \times K} \mid L_i \leq Q^i(x) \leq U_i\}$ . Then with the probability at least 90%,  $x'$  satisfies the  $\text{dist}(x', \mathcal{D}_i) \leq C_0 \sqrt{\ln m \|X^*\|_2}$ , where  $C_0$  is a positive universal constant.*

*Proof.* For each  $1 \leq i \leq m$ , we define a closed set  $A_i = \{y \in \mathbb{R}^{n \times K} \mid L_i \leq Q^i(Ty) \leq U_i\}$ . From the estimation in (14), we have  $\mu(A_i) \geq \frac{2 - \sqrt{3}}{3}$  for all  $i \leq m$ . Then for any  $\varepsilon$  and  $0 < p < 1$  such that

$$\begin{aligned} &\frac{\sqrt{\log c - \log \mu(A_i)} + \sqrt{\log c - \log p}}{\frac{1}{2} \sqrt{C}} \\ &\leq \frac{\sqrt{\log c - \log \left(\frac{2 - \sqrt{3}}{3}\right)} + \sqrt{\log c - \log p}}{\frac{1}{2} \sqrt{C}} = O\left(\sqrt{\log\left(\frac{1}{p}\right)}\right) \leq \varepsilon, \end{aligned}$$

we have  $\mu(A_i(\varepsilon)) > 1 - p$  (by Proposition 2). Take  $p = \frac{1}{10m}$  and apply the union bound, then we have  $\mu\left(\bigcap_{i=1}^m A_i(\varepsilon)\right) > 1 - mp = 0.9$ . Hence, our randomly generated point  $y$  is in  $A_i(\varepsilon)$  for each  $i \leq m$  with probability at least 90%; thus  $x' = Ty \in T(A_i(\varepsilon)) \subseteq \mathcal{D}_i + \varepsilon\|T\| \subseteq \mathcal{D}_i(\varepsilon\|T\|) \subseteq \mathcal{D}_i(\varepsilon\sqrt{\|X^*\|_2})$ . Note that we can choose  $\varepsilon = O\left(\sqrt{\log\left(\frac{1}{p}\right)}\right) = O\left(\sqrt{\log m}\right)$ . Therefore  $\text{dist}(x', \mathcal{D}_i) \leq O\left(\sqrt{\ln m \|X^*\|}\right)$ , which proves the theorem.  $\square$

We remark that in Prop. 9 the constant  $C_0$  does not depend on  $s$  as in Thm. 5.

#### 5. Computational results

Testing probabilistic algorithms based on the phenomenon of concentration of measure involves the difficulty of having to determine the minimum instance size starting from which the algorithm will start producing useful outputs. Most proofs involve unknown constants (e.g.  $c, C$  in Eq. (3)). Valid values for these constants can be determined (see e.g. [12]), but such values often make empirical applications difficult or impossible. This is why computational tests are sometimes employed in the attempt to determine practically useful values [13].

The objective of our computational testbed is to ascertain whether Barvinok's naive algorithm is beneficial or not. We therefore considered two heuristics, the only difference between them being the activation/deactivation of this algorithm as a sub-step. We call `bvk` the version of this algorithm *with* Barvinok's naive algorithm, and `sdpnlp` the version without. Both heuristics have the following structure:

1. compute the solution  $X^*$  of the SDP relaxation of the DGP or iDGP instance;
2. map the  $n \times n$  matrix  $X^*$  into an  $n \times K$  matrix  $x'$ ;
3. let  $x^*$  be the  $n \times K$  realization obtained by applying a local NLP solver to the starting point  $x'$ .

The `bvk` algorithm is obtained by replacing Step 2 with Barvinok's naive algorithm, while `sdpnlp` replaces Step 2 with Principal Component Analysis (PCA):

- factor  $X^*$  as  $V^T \Lambda V$ , where  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$  are the eigenvalues, and  $V$  is the matrix of (column) eigenvectors;

- since  $X^* \geq 0$ ,  $\lambda_i \geq 0$  for all  $i$ , which means that  $\sqrt{\Lambda}$  is real;
- let  $\Lambda' = \text{diag}(\lambda_1, \dots, \lambda_K, 0, \dots, 0)$ , and let  $x' = V^T \sqrt{\Lambda'}$ .

It is quite usual to compare DGP algorithms on solution quality and CPU time [2, 14]. The two most established error measures are the *mean distance error* (MDE), defined as  $\frac{1}{|E|} \sum_{(i,j) \in E} \left| \|x_i - x_j\|_2 - d_{ij} \right|$ , and the *largest distance error* (LDE), defined as  $\max_{(i,j) \in E} \left| \|x_i - x_j\|_2 - d_{ij} \right|$ . The equivalent measures for the *iDGP* replace  $\|x_i - x_j\|_2 - d_{ij}$  in the above formulæ with  $\max(0, L_{ij} - \|x_i - x_j\|_2) + \max(0, \|x_i - x_j\|_2 - U_{ij})$  [1, 14]. Both errors only take into account the discrepancy of the computed solution with the input data. While a low MDE can still yield a significantly different realization (if only locally), a low LDE usually gives a higher confidence of the reconstructed realization being close to the input data.

As a local NLP solver, we chose the “root” option out of the solvers in the `scipy` library [15]. Limited to the DGP, we found this solver to be superior (both CPU-time and quality-wise) to the other solvers in the library, except on really tiny instances where it appears to fail often. In all cases, the total CPU time is dominated by the SDP solution and by the local NLP solution; this is natural as optimization solvers are large-scale software packages implementing complicated algorithms, while both Barvinok’s naive algorithm and PCA are quite simple.

Note that Step 2 is stochastic in `bvk` and deterministic in `sdpnlp`. It therefore makes sense to consider the best outcome (w.r.t. LDE) out of a certain number of iterations of Barvinok’s algorithm, while PCA need only be run once. While this might appear to penalize the former as regards the CPU time, note that its worst-case complexity is dominated by the matrix product (resulting in  $O(Kn^2)$ ), while PCA requires a matrix factorization, which is  $O(n^3)$  in general. We empirically set the maximum number of iterations of Barvinok’s algorithm to 500.

The computational results, obtained using a Python 2.7 code on an Intel i5 CPU with 8GB RAM running Windows 10, are presented for the DGP in Table 1, and for the *iDGP* in Table 2. For the DGP, it is quite obvious from Table 1 that `bvk` yields

instance	MDE		LDE		CPU	
	bvk	sdpnlp	bvk	sdpnlp	bvk	sdpnlp
names	<b>0.00</b>	0.11	<b>0.07</b>	1.00	39.33	<b>22.44</b>
pept	<b>0.00</b>	0.10	<b>0.03</b>	1.81	83.91	<b>56.65</b>
C0020pdb	<b>0.00</b>	0.12	<b>0.01</b>	2.72	76.73	<b>49.39</b>
1guu-1	0.03	<b>0.00</b>	0.26	<b>0.08</b>	370.73	<b>322.66</b>
1guu-4000	<b>0.03</b>	0.12	<b>0.73</b>	1.15	415.66	<b>397.87</b>
1guu	0.02	<b>0.01</b>	<b>0.29</b>	0.33	305.54	<b>267.52</b>
res_5000	<b>0.00</b>	0.15	<b>0.00</b>	2.24	84.19	<b>53.36</b>
res_2000	<b>0.00</b>	0.07	<b>0.00</b>	1.46	85.26	<b>53.39</b>
res_0	0.00	0.00	<b>0.00</b>	0.01	93.08	<b>62.64</b>
res_3000	<b>0.00</b>	0.01	<b>0.00</b>	1.08	88.51	<b>53.43</b>
res_1000	<b>0.00</b>	0.10	<b>0.00</b>	3.05	87.88	<b>52.98</b>
res_2kxa	<b>0.00</b>	0.15	<b>0.00</b>	2.92	764.34	<b>713.35</b>
C0030pk1	<b>0.00</b>	0.11	<b>0.07</b>	2.19	1178.73	<b>1024.86</b>

Table 1: Computational comparison of Barvinok’s algorithm and PCA on DGP instances.

better solutions than `sdpnlp`. It is easy to see that the latter is faster than the former; it is, however, also easy to notice that `bvk` seems to “catch up” for instances of increasing size.

This is consistent with the fact that we run a fixed number of iterations of Barvinok’s naive algorithm (which is  $O(Kn^2)$ ) and only one iteration of PCA (which is  $O(n^3)$ ). The outcome of

instance	MDE		LDE		CPU	
	bvk	sdpnlp	bvk	sdpnlp	bvk	sdpnlp
names	0.04	<b>0.00</b>	2.11	<b>0.07</b>	53.91	<b>36.86</b>
pept	0.01	<b>0.00</b>	0.46	<b>0.40</b>	133.28	<b>99.60</b>
C0020pdb	0.02	<b>0.00</b>	1.64	<b>0.42</b>	112.38	<b>79.22</b>
1guu-1	0.02	<b>0.01</b>	1.09	<b>0.58</b>	500.64	<b>440.50</b>
1guu-4000	0.03	<b>0.02</b>	1.49	1.49	522.19	<b>461.53</b>
1guu	memory overflow					
res_5000	0.01	<b>0.00</b>	0.69	<b>0.08</b>	30764.21	<b>30465.16</b>
res_2000	0.01	<b>0.00</b>	1.78	<b>0.10</b>	33017.88	<b>32713.91</b>
res_0	0.00	0.00	0.11	0.11	22897.14	<b>22619.79</b>
res_3000	0.00	0.00	<b>0.05</b>	0.08	26095.91	<b>25846.81</b>
res_1000	0.00	0.00	<b>0.05</b>	0.07	27790.87	<b>27542.96</b>
res_2kxa	memory overflow					
C0030pk1	memory overflow					

Table 2: Computational comparison of Barvinok’s algorithm and PCA on *iDGP* instances.

the experiments on the *iDGP* presented in Table 2 are not as clear. We notice that three instances cause a memory overflow error on both heuristics, which is a consequence of the NLP solver. Again, the `sdpnlp` heuristic is superior to `bvk` as concerns CPU time; and the ratio between the two values again becomes smaller as the size increases. In terms of solution quality, `sdpnlp` is superior to `bvk` for all but the three largest instances, and slightly inferior for the two largest. While the discrepancies in LDE are too small to derive any meaningful conclusion, we remark that concentration of measure phenomena always “kick in” from some instance size onwards, but it is difficult to establish the threshold size. The larger instances we tested all yielded memory overflows, due to both SDP and NLP solvers.

An anonymous referee suggested a direction for future work which we find very interesting. Our idea for deploying a local descent method from the starting point  $x'$  found by Barvinok’s naive algorithm was given by the geometric intuition that, if the distance from  $x'$  to the feasible set of the given QCP is bounded with high probability, then a local descent from  $x'$  might be all that is needed to find a feasible point of the QCP. The referee asks the following question: are there natural conditions under which a *gradient descent* from  $x'$  is guaranteed (perhaps with some probability) to perform nicely?

While researchers in mathematics and theoretical computer science (TCS) produce many algorithms with interesting theoretical properties, their adaptation to applications and consequent testing is scattered at best. Some TCS researchers have been heard, during talks at conferences, making various statements to the effect that their algorithms are not for practical use, and that their value is wholly theoretical. We beg to differ: whether these algorithms have a practical value or not is simply *not known* (for lack of testing). While we realize that no amount of testing can ever prove anything, an indication of usefulness might remarkably increase the impact of a “wholly theoretical algorithm” (witness e.g. the Goemans-Williamson randomized approximation algorithm for `MAX CUT` [16]). We hope this paper will contribute to increase the impact of the wonderfully elegant, simple and yet effective “naive algorithm” that A. Barvinok invented twenty years ago.

## References

- [1] L. Liberti, C. Lavor, A. Mucherino, N. Maculan, Molecular distance geometry methods: from continuous to discrete, *International Transactions in Operational Research* 18 (2010) 33–51.
- [2] L. Liberti, C. Lavor, N. Maculan, A. Mucherino, Euclidean distance geometry and applications, *SIAM Review* 56 (1) (2014) 3–69.
- [3] N. Krislock, H. Wolkowicz, Explicit sensor network localization using semidefinite representations and facial reductions, *SIAM Journal on Optimization* 20 (2010) 2679–2708.
- [4] M. Cucuringu, A. Singer, D. Cowburn, Eigenvector synchronization, graph rigidity and the molecule problem, *Information and Inference: a journal of the IMA* 1 (2012) 21–67.
- [5] A. Cassioli, B. Bordeaux, G. Bouvier, A. Mucherino, R. Alves, L. Liberti, M. Nilges, C. Lavor, T. Malliavin, An algorithm to enumerate all possible protein conformations verifying a set of distance constraints, *BMC Bioinformatics* 16 (2015) 23–38.
- [6] S. Billinge, P. Duxbury, D. Gonçalves, C. Lavor, A. Mucherino, Assigned and unassigned distance geometry: Applications to biological molecules and nanostructures, *4OR* 14 (2016) 337–376.
- [7] B. Berger, J. Kleinberg, T. Leighton, Reconstructing a three-dimensional model with arbitrary errors, *Journal of the ACM* 46 (2) (1999) 212–235.
- [8] A. Barvinok, Measure concentration in optimization, *Mathematical Programming* 79 (1997) 33–53.
- [9] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, P. Bourne, The protein data bank, *Nucleic Acid Research* 28 (2000) 235–242.
- [10] V. Milman, G. Schechtman, Asymptotic theory of finite dimensional normed spaces, no. 1200 in *Lecture Notes in Mathematics*, Springer, Berlin, 1986.
- [11] A. Barvinok, *A Course in Convexity*, no. 54 in *Graduate Studies in Mathematics*, American Mathematical Society, Providence, 2002.
- [12] A. Barvinok, *Math 710: Measure Concentration*, class notes (2005).
- [13] S. Venkatasubramanian, Q. Wang, The Johnson-Lindenstrauss transform: An empirical study, in: *Algorithm Engineering and Experiments*, Vol. 13 of *ALENEX*, SIAM, Providence, 2011, pp. 164–173.
- [14] C. D’Ambrosio, V. K. Ky, C. Lavor, L. Liberti, N. Maculan, New error measures and methods for realizing protein graphs from distance data, *Discrete and Computational Geometry* 57 (2) (2017) 371–418.
- [15] E. Jones, T. Oliphant, P. Peterson, *SciPy: Open source scientific tools for Python*, [Online; accessed 2016-03-01] (2001).  
URL <http://www.scipy.org/>
- [16] M. Goemans, D. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *Journal of the ACM* 42 (6) (1995) 1115–1145.