Math. Program. Ser. B (2007) 109:367–384
DOI 10.1007/s10107-006-0040-1

FULL LENGTH PAPER

Theory of semidefinite programming for Sensor Network Localization

Anthony Man-Cho So · Yinyu Ye

Received: 22 November 2004 / Accepted: 1 June 2005 /
Published online: 19 September 2006

Abstract We analyze the semidefinite programming (SDP) based model and method for the position estimation problem in sensor network localization and other Euclidean distance geometry applications. We use SDP duality and interior-point algorithm theories to prove that the SDP localizes any network or graph that has unique sensor positions to fit given distance measures. Therefore, we show, for the first time, that these networks can be localized in polynomial time. We also give a simple and efficient criterion for checking whether a given instance of the localization problem has a unique realization in $\mathcal{R}^2$ using graph rigidity theory. Finally, we introduce a notion called strong localizability and show that the SDP model will identify all strongly localizable sub-networks in the input network.

Keywords Euclidean distance geometry · Graph realization · Sensor network localization · Semidefinite programming · Rigidity theory

A preliminary version of this paper has appeared in the *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2005.

A. Man-Cho So
Department of Computer Science,
Stanford University, Stanford, CA 94305, USA
e-mail: manchoso@cs.stanford.edu

Y. Ye (✉)
Department of Management Science and Engineering and, by courtesy,
Electrical Engineering, Stanford University,
Stanford, CA 94305, USA
e-mail: yinyu-ye@stanford.edu

Mathematics Subject Classification (2000) 51K05 · 52C25 · 68Q25 · 90C22 · 90C35

1 Introduction

One of the most studied problems in distance geometry is the *Graph Realization* problem, in which one is given a graph $G = (V, E)$ and a set of non-negative weights $\{d_{ij} : (i,j) \in E\}$ on its edges, and the goal is to compute a realization of G in the Euclidean space $\mathcal{R}^d$ for a given dimension d, i.e. to place the vertices of G in $\mathcal{R}^d$ such that the Euclidean distance between every pair of adjacent vertices v_i, v_j equals to the prescribed weight d_{ij}. This problem and its variants arise from applications in various areas, such as molecular conformation, dimensionality reduction, Euclidean ball packing, and more recently, wireless sensor network localization [3,12,14,21,27,30]. In the sensor networks setting, the vertices of G correspond to sensors, the edges of G correspond to communication links, and the weights correspond to distances. Furthermore, the vertices are partitioned into two sets – one is the *anchors*, whose exact positions are known (via GPS, for example); and the other is the *sensors*, whose positions are unknown. The goal is to determine the positions of all the sensors. We shall refer to this problem as the *Sensor Network Localization* problem. Note that we can view the Sensor Network Localization problem as a variant of the Graph Realization problem in which a subset of the vertices are constrained to be in certain positions.

In many sensor networks applications, sensors collect data that are location dependent. Thus, another related question is whether the given instance has a unique realization in the required dimension (say, in $\mathcal{R}^2$). Indeed, most of the previous works on the Sensor Network Localization problem fall into two categories – one deals with computing a realization of a given instance [12,14, 15,21,26–28,30], and the other deals with determining whether a given instance has a unique realization in $\mathcal{R}^d$ using graph rigidity [15,18]. It is interesting to note that from an algorithmic viewpoint, the two problems above have very different characteristics. Under certain non-degeneracy assumptions, the question of whether a given instance has a unique realization on the plane can be decided efficiently [22], while the problem of computing a realization on the plane is NP-complete in general, even if the given instance has a unique realization on the plane [7]. Thus, it is not surprising that all the aforementioned heuristics for computing a realization of a given instance do not guarantee to find it in the required dimension. On another front, there have been attempts to characterize families of graphs that admit polynomial time algorithms for computing a realization in the required dimension. For instance, Eren et al. [15] have shown that the family of *trilateration graphs* has such property. (A graph is a trilateration graph in dimension d if there exists an ordering of the vertices $1, \ldots, d+1, d+2, \ldots, n$ such that (i) the first $d+1$ vertices form a complete graph, and (ii) each vertex $j > d+1$ has at least $d+1$ edges to vertices earlier in the sequence.) However, the question of whether there exist larger families of graphs with such property is open.

1.1 Our contribution

In this paper, we resolve this question by showing that the family of *uniquely localizable* graphs also enjoys such a property. Informally, a graph is uniquely localizable in dimension d if (i) it has a unique realization in $\mathcal{R}^d$, and (ii) it does not have any realization whose affine span is $\mathcal{R}^h$, where $h > d$. Specifically, we present an SDP model that guarantees to find the unique realization in polynomial time when the input graph is uniquely localizable. The proof employs SDP duality theory and properties of interior-point algorithms for SDP. To the best of our knowledge, this is the first time such a theoretical guarantee is proven for a general localization algorithm. Moreover, our results are interesting in view of the hardness result of Aspnes et al. [7], as they identify a large family of efficiently realizable graphs. Next, using the theory of graph rigidity, we give a simple and efficient criterion for checking whether a given instance has a unique realization on the plane. We remark that this result has been independently proven by Eren et al. [15]. However, our approach is different in that we use techniques from kinematics and these techniques may be of independent interest. Lastly, we introduce the concept of strong localizability. Informally, a graph is strongly localizable if it is uniquely localizable and remains so under slight perturbations. We show that the SDP model will identify all the strongly localizable subgraphs in the input graph.

We should mention here that the Sensor Network Localization problem (or its variants) has been studied in various contexts before. However, these earlier works have quite different emphases from ours. For instance, Schoenberg [29] and Young and Householder [34] have studied the problem in the context of Euclidean distance matrix characterizations. They have considered the case where there are no anchors, but *all* pairwise distances among the sensors are known. They have shown that the given pairwise distances arise from points in an d-dimensional (but not $(d-1)$-dimensional) Euclidean space if and only if a certain matrix is positive semidefinite and has rank d. Such a characterization forms the basis for the classical approach to multidimensional scaling (see, e.g., [17,31]), where various algorithms are developed for constructing a configuration of points in $\mathcal{R}^d$ (where d is part of the input) such that the induced distance matrix matches or approximates the given (complete) distance matrix. Later, Trosset [32,33] has extended classical multidimensional scaling to include the case where the given distance matrix is incomplete, i.e. some of the pairwise distances may be missing. He has shown that a realization in the required dimension exists if and only if the global optimum of a certain optimization problem is zero, and has provided a numerical procedure for finding such a realization. However, it is not clear under what conditions would Trosset's algorithm terminate with a desired realization in polynomial time. On another front, Barvinok [10] has studied the Sensor Network Localization problem in the context of quadratic maps and used SDP theory to analyze the possible dimensions of the realization. In addition, Alfakih et al. [3–5] have related this problem to the *Euclidean Distance Matrix Completion* problem and obtained an SDP formulation for the former. Moreover, Alfakih has obtained a characterization of rigid graphs in [1] using Euclidean distance matrices and has studied some of the computational aspects of such characterization in [2] using SDP. However, these papers mostly address the question of realizability of the input graph, and the analyses of their SDP models only guarantee that they will find a realization whose dimension lies within a certain range. Thus, these models are not quite suitable for our application. In contrast, our analysis takes advantage of the presence of anchors and gives a condition which guarantees that our SDP model will find a realization in the required dimension. We remark that SDP has also been used to compute and analyze distance geometry problems where the realization is allowed to have a certain amount of distortion in the distances [11,24]. Again, these methods can only guarantee to find a realization that lies in a range of dimensions. Thus, it would be interesting to extend our method to compute low-distortion realizations in a given dimension. For some related work in this direction, see, e.g., [8].

1.2 Outline of the paper

The rest of the paper is organized as follows. In Sect. 2, we give a formal definition of the Sensor Network Localization problem and introduce the notations that will be used in the paper. In Sect. 3, we provide a formulation of the problem as an SDP. We remark that the SDP model used here is developed earlier in a companion paper [12]. In that paper the authors have reported the model's superb experimental performance, and the current work is an attempt to provide theoretical justifications for using that model. Specifically, we analyze the SDP and discuss its characteristics in Sect. 4. In Sect. 5 we discuss our results in the context of rigidity theory. In Sect. 6 we introduce the notion of strong localizability and show how the SDP model can identify strongly localizable subgraphs in the input graph. In Sect. 7 we compare the different notions introduced in this paper and demonstrate their differences via examples. In particular, we show that rigidity in $\mathcal{R}^2$, unique localizability, and strong localizability are all distinct concepts. Lastly, we summarize our results in Sect. 8 and discuss some possible future directions.

2 Preliminaries

We begin with some notations. The trace of a matrix A is denoted by $\mathrm{Trace}(A)$. We use I and $\mathbf{0}$ to denote the identity matrix and the matrix of all zeros, respectively, whose dimensions will be clear from the context. The inner product of two matrices P and Q is denoted by $P \bullet Q = \mathrm{Trace}(P^{\mathrm{T}}Q)$. The 2-norm of a vector x, denoted by $\|x\|$, is given by $\sqrt{x \bullet x}$. A positive semidefinite matrix X is denoted by $X \succeq \mathbf{0}$.

In this paper we study the *Sensor Network Localization* problem, which is defined as follows. We are given m *anchor* points $a_1, \ldots, a_m \in \mathcal{R}^d$ whose locations are known, and n *sensor* points $x_1, \ldots, x_n \in \mathcal{R}^d$ whose locations we wish to determine. Furthermore, we are given the Euclidean distance values $\bar{d}_{kj}$ between a_k

and x_j for some k, j, and d_{ij} between x_i and x_j for some $i < j$. Specifically, let $N_a = \{(k,j) : \bar{d}_{kj} \text{ is specified}\}$ and $N_x = \{(i,j) : i < j,\ d_{ij} \text{ is specified}\}$. The Sensor Network Localization problem is then to find a realization of $x_1, \ldots, x_n \in \mathcal{R}^d$ such that:

$$\begin{aligned} \|a_k - x_j\|^2 &= \bar{d}_{kj}^2 \quad \forall (k,j) \in N_a \\ \|x_i - x_j\|^2 &= d_{ij}^2 \quad \forall (i,j) \in N_x \end{aligned} \tag{1}$$

We would like to develop fast algorithms to answer questions like: Does the network have a realization of x_j's? Is the realization unique? As we shall see in subsequent sections, these questions can be answered efficiently.

3 Semidefinite programming method

In general, problem (1) is a non-convex optimization problem and difficult to solve. In fact, most previous approaches adopt global optimization techniques such as nonlinear least square methods, or geometric methods such as triangularization. An alternate approach, called the semidefinite programming method, is recently developed in [12] and related earlier work [3,23]. We shall review this approach below.

Let $X = [x_1\ x_2\ \cdots\ x_n]$ be the $d \times n$ matrix that needs to be determined. Then, for all $(i,j) \in N_x$, we have:

$$\|x_i - x_j\|^2 = e_{ij}^{\mathrm{T}} X^{\mathrm{T}} X e_{ij}$$

and for all $(k,j) \in N_a$, we have:

$$\|a_k - x_j\|^2 = (a_k; e_j)^{\mathrm{T}} [I_d; X]^{\mathrm{T}} [I_d; X] (a_k; e_j)$$

Here, $e_{ij} \in \mathcal{R}^n$ is the vector with 1 at the ith position, -1 at the jth position and zero everywhere else; $e_j \in \mathcal{R}^n$ is the vector of all zeros except an -1 at the jth position; $(a_k; e_j) \in \mathcal{R}^{d+n}$ is the vector of a_k on top of e_j; and I_d is the d-dimensional identity matrix. Thus, problem (1) becomes: find a symmetric matrix $Y \in \mathcal{R}^{n \times n}$ and a matrix $X \in \mathcal{R}^{d \times n}$ such that:

$$\begin{aligned} & e_{ij}^{\mathrm{T}} Y e_{ij} = d_{ij}^2 && \forall (i,j) \in N_x \\ & (a_k; e_j)^{\mathrm{T}} \begin{pmatrix} I_d & X \\ X^{\mathrm{T}} & Y \end{pmatrix} (a_k; e_j) = \bar{d}_{kj}^2 && \forall (k,j) \in N_a \\ & Y = X^{\mathrm{T}} X \end{aligned}$$

The SDP method is to relax the constraint $Y = X^{\mathrm{T}} X$ to $Y \succeq X^{\mathrm{T}} X$, where $Y \succeq X^{\mathrm{T}} X$ means that $Y - X^{\mathrm{T}} X \succeq \mathbf{0}$. It is well-known [13] that the condition

$Y \succeq X^{\mathrm{T}}X$ is equivalent to:

$$Z = \begin{pmatrix} I_d & X \\ X^{\mathrm{T}} & Y \end{pmatrix} \succeq \mathbf{0} \tag{2}$$

Thus, we can write the relaxed problem as a standard SDP problem, namely, find a symmetric matrix $Z \in \mathcal{R}^{(d+n)\times(d+n)}$ to:

$$\begin{array}{lll} \text{maximize} & 0 & \\ \text{subject to} & Z_{1:d,1:d} = I_d & \\ & (\mathbf{0}; e_{ij})(\mathbf{0}; e_{ij})^{\mathrm{T}} \bullet Z = d_{ij}^2 & \forall (i,j) \in N_x \\ & (a_k; e_j)(a_k; e_j)^{\mathrm{T}} \bullet Z = \bar{d}_{kj}^2 & \forall (k,j) \in N_a \\ & Z \succeq \mathbf{0} & \end{array} \tag{3}$$

where $Z_{1:d,1:d}$ is the $d \times d$ principal submatrix of Z. Note that this formulation forces any possible feasible solution matrix to have rank at least d.

The dual of the SDP relaxation is given by:

$$\begin{array}{ll} \text{minimize} & I_d \bullet V + \sum_{(i,j)\in N_x} y_{ij} d_{ij}^2 + \sum_{(k,j)\in N_a} w_{kj} \bar{d}_{kj}^2 \\ \text{subject to} & \begin{pmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \sum_{(i,j)\in N_x} y_{ij} (\mathbf{0}; e_{ij})(\mathbf{0}; e_{ij})^{\mathrm{T}} \\ & \quad + \sum_{(k,j)\in N_a} w_{kj} (a_k; e_j)(a_k; e_j)^{\mathrm{T}} \succeq \mathbf{0} \end{array} \tag{4}$$

Note that the dual is always feasible, as $V = \mathbf{0}$, $y_{ij} = 0$ for all $(i,j) \in N_x$ and $w_{kj} = 0$ for all $(k,j) \in N_a$ is a feasible solution.

4 Analysis of the SDP relaxation

We now investigate when will the SDP (3) have an exact relaxation, i.e. when will the solution matrix Z have rank d. Suppose that problem (3) is feasible. This occurs when, for instance, $\bar{d}_{kj}$ and d_{ij} represent exact distance values for the positions $\bar{X} = [\bar{x}_1\ \bar{x}_2\ \cdots\ \bar{x}_n]$. Then, the matrix $\bar{Z} = (I_d; \bar{X})^{\mathrm{T}}(I_d; \bar{X})$ is a feasible solution for (3). Now, since the primal is feasible, the minimal value of the dual must be 0, i.e. there is no duality gap between the primal and dual.

Let U be the $(d+n)$-dimensional dual slack matrix, i.e.:

$$U = \begin{pmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \sum_{(i,j)\in N_x} y_{ij} (\mathbf{0}; e_{ij})(\mathbf{0}; e_{ij})^{\mathrm{T}} + \sum_{(k,j)\in N_a} w_{kj} (a_k; e_j)(a_k; e_j)^{\mathrm{T}}$$

Then, from the duality theorem for SDP (see, e.g., [6]), we have:

Theorem 1 *Let $\bar{Z}$ be a feasible solution for (3) and $\bar{U}$ be an optimal slack matrix of* (4). *Then,*

1. *complementarity condition holds: $\bar{Z} \bullet \bar{U} = 0$ or $\bar{Z}\bar{U} = \mathbf{0}$;*
2. *$Rank(\bar{Z}) + Rank(\bar{U}) \leq d + n$;*
3. *$Rank(\bar{Z}) \geq d$ and $Rank(\bar{U}) \leq n$.*

An immediate result from the theorem is the following:

Corollary 1 *If an optimal dual slack matrix has rank n, then every solution of* (3) *has rank d. That is, problems* (1) *and* (3) *are equivalent and* (1) *can be solved as an SDP in polynomial time.*

Another technical result is the following:

Proposition 1 *If every sensor point is connected, directly or indirectly, to an anchor point in* (1), *then any solution to* (3) *must be bounded, that is, Y_{jj} is bounded for all $j = 1, \ldots, n$.*

Proof If sensor point x_j is connected to an anchor point a_k, then we have:

$$\|x_j\|^2 - 2a_k^{\mathrm{T}} x_j + \|a_k\|^2 \leq Y_{jj} - 2a_k^{\mathrm{T}} x_j + \|a_k\|^2 = \bar{d}_{kj}^2$$

so that from the triangle inequality $\|x_j\|$ in (2) is bounded. Hence, we have:

$$Y_{jj} \leq \bar{d}_{kj}^2 + 2\|a_k\|\,\|x_j\| - \|a_k\|^2$$

Furthermore, if x_i is connected to x_j and Y_{jj} is bounded, we have:

$$Y_{ii} - 2\sqrt{Y_{ii} Y_{jj}} + Y_{jj} \leq Y_{ii} - 2Y_{ij} + Y_{jj} = d_{ij}^2$$

so that from the triangle inequality Y_{ii} must be also bounded. □

In general, a primal (dual) max-rank solution is a solution that has the highest rank among all solutions for primal (3) (dual (4)). It is known [16,19] that various path-following interior-point algorithms compute the max-rank solutions for both the primal and dual in polynomial time. This motivates the following definition.

Definition 1 *Problem* (1) *is uniquely localizable if there is a unique localization $\bar{X} \in \mathcal{R}^{d \times n}$ and there is no $x_j \in \mathcal{R}^h$, $j = 1, \ldots, n$, where $h > d$, such that:*

$$\begin{array}{ll} \|(a_k; \mathbf{0}) - x_j\|^2 = \bar{d}_{kj}^2 & \forall (k, j) \in N_a \\ \|x_i - x_j\|^2 = d_{ij}^2 & \forall (i, j) \in N_x \\ x_j \neq (\bar{x}_j; \mathbf{0}) & \textit{for some } j \in \{1, \ldots, n\} \end{array}$$

The latter says that the problem cannot have a non-trivial localization in some higher dimensional space $\mathcal{R}^h$ (i.e. a localization different from the one obtained by setting $x_j = (\bar{x}_j; \mathbf{0})$ for $j = 1, \ldots, n$), where anchor points are augmented to $(a_k; \mathbf{0}) \in \mathcal{R}^h$, for $k = 1, \ldots, m$.

We now develop the following theorem:

Theorem 2 *Suppose that the network is connected. Then, the following are equivalent:*

1. *Problem* (1) *is uniquely localizable.*
2. *The max-rank solution matrix of* (3) *has rank d.*
3. *The solution matrix of* (3)*, represented by* (2)*, satisfies $Y = X^{\mathrm{T}}X$.*

Proof The equivalence between 2. and 3. is straightforward.

Now, since any rank d solution of (3) is a solution to (1), from 2. to 1. we need to prove that if the max-rank solution matrix of (3) has rank d then it is unique. Suppose not, i.e., (3) has two rank-d feasible solutions:

$$Z_1 = \begin{pmatrix} I_d & X_1 \\ X_1^{\mathrm{T}} & X_1^{\mathrm{T}}X_1 \end{pmatrix} \quad \text{and} \quad Z_2 = \begin{pmatrix} I_d & X_2 \\ X_2^{\mathrm{T}} & X_2^{\mathrm{T}}X_2 \end{pmatrix}$$

Then, the matrix $Z = \alpha Z_1 + \beta Z_2$, where $\alpha + \beta = 1$ and $\alpha, \beta > 0$ is a feasible solution and its rank must be d, since all feasible solution of (3) has rank at least d but the max-rank is assumed to be d. Therefore, we have:

$$Z = \begin{pmatrix} I_d & \alpha X_1 + \beta X_2 \\ \alpha X_1^{\mathrm{T}} + \beta X_2^{\mathrm{T}} & \alpha X_1^{\mathrm{T}}X_1 + \beta X_2^{\mathrm{T}}X_2 \end{pmatrix} = \begin{pmatrix} I_d & B \\ B^{\mathrm{T}} & B^{\mathrm{T}}B \end{pmatrix}$$

where $B = \alpha X_1 + \beta X_2$. It follows that $(X_1 - X_2)^{\mathrm{T}}(X_1 - X_2) = \mathbf{0}$, or $\|X_1 - X_2\| = 0$, i.e. $Z_1 = Z_2$, which is a contradiction.

Next, we prove the direction from 1. to 2., that is, the rank of a max-rank solution of (3) is d. Suppose that there is a feasible solution Z of (3) whose rank is greater than d. Then, we must have $Y \succeq X^{\mathrm{T}}X$ and $Y \neq X^{\mathrm{T}}X$. Thus, we have the decomposition $Y - X^{\mathrm{T}}X = (X')^{\mathrm{T}}X'$, where $X' = [x'_1, \ldots, x'_n] \in \mathcal{R}^{r \times n}$ and r is the rank of $Y - X^{\mathrm{T}}X$. Now, consider the point:

$$\tilde{x}_j = \begin{pmatrix} x_j \\ x'_j \end{pmatrix} \in \mathcal{R}^{d+r} \quad \text{for } j = 1, \ldots, n$$

Then, we have:

$$\|\tilde{x}_j\|^2 = Y_{jj},\ (\tilde{x}_i)^{\mathrm{T}}\tilde{x}_j = Y_{ij} \quad \forall i, j$$

Moreover, since the network is connected, we conclude from Proposition 1 that Y_{ii} and Y_{ij} are bounded for all i,j. Hence, we have:

$$\begin{aligned} \|(a_k;\mathbf{0}) - \tilde{x}_j\|^2 &= \bar{d}_{kj}^2 \quad \forall (k,j) \in N_a \\ \|\tilde{x}_i - \tilde{x}_j\|^2 &= d_{ij}^2 \quad \forall (i,j) \in N_x \end{aligned}$$

In other words, $\tilde{x}_j$ is a localization of problem (1) in $\mathcal{R}^{d+r}$, which is a contradiction. □

Theorem 2 establishes, for the first time, that as long as problem (1) is uniquely localizable, then the realization can be computed in polynomial time by solving the SDP relaxation. Conversely, if the relaxation solution computed by an interior-point algorithm (which generates max-rank feasible solutions) has rank d (and hence $Y = X^{\mathrm{T}}X$), then X is the unique realization of problem (1). Moreover, Theorem 2 implies the existence of a large family of efficiently realizable graphs, even though the recent result of Aspnes et al. [7] shows that the problem of computing a realization of the sensors on the plane is NP-complete in general (this is true even when the instance has a unique solution on the plane).

5 Connections to rigidity theory

In this section we give a simple and efficient criterion for checking whether a graph G with anchors has a unique realization on the plane using rigidity theory. The main idea is to augment G to another graph G' by adding edges between the anchors in G and check whether G' is rigid. We remark that the main theorem of this section, Theorem 3, has been independently proven by Eren et al. in [15]. However, our proof (in particular, Propositions 2, 3 and 4) gives a connection between the graphs G and G' which [15] did not offer.

Before we state our result, we shall briefly review the theory of rigidity. For a more detailed account, see, e.g., [18,20].

The theory of rigidity concerns with the study of frameworks. A *framework* is a pair (G,p), where G is a graph and $p : V \to \mathcal{R}^d$ is an embedding mapping the vertices into an Euclidean space. Equivalently, we can view p as an $|V|d$-dimensional vector assigning coordinates to the vertices. Given a framework (G,p), we can define the edge function $f : \mathcal{R}^{|V|d} \to \mathcal{R}^{|E|}$ by $f(p)_{ij} = \|p_i - p_j\|^2$, where $(i,j) \in E$. A natural question is then whether there exists another non-congruent realization of the framework (G,p), i.e. whether there exists an $q \in \mathcal{R}^{|V|d}$ not congruent to p such that $\|q_i - q_j\|^2 = \|p_i - p_j\|^2$ for all $(i,j) \in E$. By non-congruence we mean that q is not obtained by applying a rigid motion to p. In this section, we shall only consider *generic* embeddings, which means that the vertex coordinates assigned by the embeddings are algebraically independent over the rationals. There are several ways in which a framework in $\mathcal{R}^2$ can have non-congruent realizations. We first consider the following.

Definition 2 *A finite flexing of a framework (G,p) is a family of realizations of G, parametrized by t, such that the location of each vertex is a differentiable function of t, and $\|p_i(t) - p_j(t)\|^2 = c_{ij}$ for all $(i,j) \in E$.*

Now, upon differentiating with respect to t, we have the relation:

$$(p_i - p_j)^{\mathrm{T}}(v_i - v_j) = 0 \quad \forall (i,j) \in E \tag{5}$$

where v_i is the instantaneous velocity of vertex i. An assignment of velocities such that the above relation is satisfied is called an *infinitesimal motion* of the framework. We say that the infinitesimal motion is *trivial* if it is simply a translation or rotation. Thus, it follows that if a framework has a non-trivial infinitesimal motion, then the framework has a non-congruent realization. In this case, we say that the framework is *infinitesimally flexible*. Otherwise, the framework is *infinitesimally rigid*.

Note that the above definition does not restrict the assignment of velocities besides the requirement that it satisfies (5). Thus, the theory cannot be applied directly to our setting, since we require certain vertices be anchored. However, if there is a non-zero assignment that satisfies $v_i = 0$ for all anchored vertex i, then the framework is clearly not uniquely localizable. In addition, such an assignment will necessarily preclude translations and rotations when there are more than one anchors, as such motions do not fix two or more vertices.

The observation in the preceding paragraph gives us a clue on relating unique localizability and rigidity. We say that G has a *fixing infinitesimal motion* if it has an infinitesimal motion that fixes the anchors. Given a graph G, consider the graph G' obtained from G by including the edges connecting the anchors. In other words, if a_i, a_j are anchors, then $(a_i, a_j) \in E(G')$. We then have the following proposition.

Proposition 2 *G has no fixing infinitesimal motion iff G' is infinitesimally rigid.*

Proof Suppose that G has an infinitesimal motion $\mathcal{M}$ that fixes the anchors. Then, $\mathcal{M}$ would also be an infinitesimal motion for G'. For necessity, suppose that G has no infinitesimal motion that fixes the anchors, but that G' is not infinitesimally rigid. Then, G' must have an infinitesimal motion $\mathcal{M} = \{v(a_1), \ldots, v(a_k), v_1, \ldots, v_n\}$ that assigns some non-zero velocity to an anchor. However, the subgraph induced by the anchors is complete, and hence rigid. Thus, $\mathcal{M}$ restricted to this subgraph is an *infinitesimal isometry* [18]. Without loss of generality, we consider the two cases where this isometry is a translation or a rotation. For the translation case, $v(a_i) = v$ for $1 \le i \le k$. Then, the assignment $\mathcal{M}' = \{0, \ldots, 0, v_1 - v, \ldots, v_n - v\}$ is an infinitesimal motion of G' (and hence of G) that fixes the anchors, which is a contradiction. For the rotation case, we may assume, by a change of reference frame if necessary, that the center of rotation is at one of the anchors, say a_1. Thus, we have $v(a_1) = 0$. Let ω be the angular velocity of the rotation, and for sensor x_i, define $\bar{v}_i = \omega \times \|x_i - a_1\|$, for $1 \le i \le n$. In other words, $\bar{v}_i$ is the velocity of sensor x_i if the whole network is to rotate around a_1 at an angular velocity of ω. Note that $\bar{v}_i$ satisfy the following relations:

$$(x_i - a_j)^{\mathrm{T}}(\bar{v}_i - v(a_j)) = 0 \quad 1 \le i \le n, 1 \le j \le k \tag{6}$$

$$(x_i - x_j)^{\mathrm{T}}(\bar{v}_i - \bar{v}_j) = 0 \quad 1 \le i \le j \le n \tag{7}$$

Now, consider the velocity assignment:

$$\mathcal{M}' = \{0, \ldots, 0, v_1 - \bar{v}_1, \ldots, v_n - \bar{v}_n\}$$

We claim that it is an infinitesimal motion of G' that fixes the anchors. To see this, it suffices to check that:

$$\begin{aligned} (x_i - a_j)^{\mathrm{T}}((v_i - \bar{v}_i) - 0) &= 0 \quad \forall (i,j) \in N_a \\ (x_i - x_j)^{\mathrm{T}}((v_i - \bar{v}_i) - (v_j - \bar{v}_j)) &= 0 \quad \forall (i,j) \in N_x \end{aligned}$$

The first equation follows since we have:

$$(x_i - a_j)^{\mathrm{T}}(v_i - \bar{v}_i) = (x_i - a_j)^{\mathrm{T}}((v_i - v(a_j)) + (v(a_j) - \bar{v}_i)) = 0$$

by definition of $\mathcal{M}$ and relation (6). The second equation again follows directly from the definition of $\mathcal{M}$ and the relation (7). This again leads to a contradiction. Therefore, the proof is completed. □

Next, we consider another way in which a graph can have non-congruent realizations. We say that a set of vertices form a *mirror* if they lie on a line, and there are no edges crossing this line. Obviously, by reflecting across this mirror, we would have two non-congruent realizations of the graph. We say that a framework allows a *partial reflection* if such a mirror exists. Then, we have the following proposition.

Proposition 3 *G allows a partial reflection that fixes the anchors iff G' allows a partial reflection.*

Proof It suffices to observe that if there is a partial reflection, then all the anchors will lie on one side of the mirror. □

As indicated in [20], the above two conditions are still not sufficient to guarantee a unique realization of a graph on the plane. To state the third condition, we begin with some definition.

Definition 3 *A framework $\mathcal{F}$ is said to be redundant if the framework $\mathcal{F}'$ obtained from $\mathcal{F}$ by removing an edge is infinitesimally rigid.*

Definition 4 *A framework is said to be redundantly rigid if all its edges are redundant.*

We then have the following proposition:

Proposition 4 *Suppose that G has at least four anchors. Then, G has no fixing infinitesimal motion after the removal of any of its edges iff G' is redundantly rigid.*

Proof Consider an edge $e \in E(G)$. If G has a fixing infinitesimal motion after the removal of e, then G' is not redundantly rigid. Conversely, suppose that G has no fixing infinitesimal motion after the removal of any of its edges. To show that G' is redundantly rigid, it suffices to note that the subgraph induced by the anchors in G' is rigid, even after the removal of any one of its edges. Thus, if G' has an infinitesimal motion after the removal of e, we have a contradiction by a similar argument in Proposition 2. □

A recent result of Jackson and Jordán [22] shows that infinitesimal rigidity, three-connectivity and redundant rigidity are necessary and sufficient conditions for unique realization of a graph in $\mathcal{R}^2$. Thus, from the results of Propositions 2, 3 and 4, we obtain the following theorem:

Theorem 3 *The graph G with anchors is uniquely realizable in $\mathcal{R}^2$ iff the associated graph G' is uniquely realizable in $\mathcal{R}^2$.*

Corollary 2 *In order for the graph G with anchors to be uniquely localizable, it is necessary that the associated graph G' is uniquely realizable in $\mathcal{R}^2$.*

We remark that the unique realizability of G' in $\mathcal{R}^2$ can be checked efficiently, and we refer the interested readers to [20].

Note that the graph G' has $\Omega(m^2)$ edges, where m is the number of anchors. An examination of the proofs above would immediately reveal that all we need is a graph G' such that the subgraph induced by the anchors is uniquely realizable. There exist graphs with only $O(m)$ edges that possess such property. One example is the trilateration graph defined in [15]. In order to improve computational efficiency, we should use one of these graphs instead.

6 Strongly localizable problem

Although unique localizability is a useful notion in determining the solvability of the Sensor Network Localization problem, it is not stable under perturbation. As we shall see in Sect. 7, there exist networks which are uniquely localizable, but may no longer be so after small perturbation of the sensor points. This motivates us to define another notion called strong localizability.

Definition 5 *We say problem (1) is strongly localizable if the dual of its SDP relaxation (4) has an optimal dual slack matrix with rank n.*

Note that if a network is strongly localizable, then it is uniquely localizable from Theorems 1 and 2, since the rank of all feasible solution of the primal is d.

We show how we can construct a rank-n optimal dual slack matrix. First, note that if U is an optimal dual slack matrix of rank n, then it can be written in the form $U = (-X^{\mathrm{T}}; I_n)^{\mathrm{T}} W (-X^{\mathrm{T}}; I_n)$ for some positive definite matrix W of rank n. Now, consider the dual matrix U. It has the form:

$$U = \begin{pmatrix} U_{11} & U_{12} \\ U_{12}^{\mathrm{T}} & U_{22} \end{pmatrix}$$

where U_{22} is an $n \times n$ matrix. Moreover, it can be decomposed as $U_{22} = A + D$, where $A_{ij} = y_{ij}$ if $(i,j) \in N_x$, $A_{ii} = -\sum_{j:(i,j)\in N_x} A_{ij}$; and D is a diagonal matrix where $D_{ii} = -\sum_{(k,i)\in N_a} w_{ki}$. (If there is no $(k,i) \in N_a$, then $D_{ii} = 0$.) Note that if we impose the constraints $y_{ij} \le 0$ and $w_{ki} \le 0$, then both A and D are positive semidefinite. Moreover, we have the following:

Proposition 5 *Suppose that the network is connected. Furthermore, suppose that $y_{ij} < 0$ for all $(i,j) \in N_x$, and that $w_{ki} < 0$ for all $(k,i) \in N_a$, with $N_a \neq \emptyset$. Then, U_{22} is positive definite, i.e. it has rank n.*

Proof Since A and D are positive semidefinite, we have $x^{\mathrm{T}} U_{22} x \ge 0$ for all $x \in \mathcal{R}^n$. We now show that there is no $x \in \mathcal{R}^n \backslash \{\mathbf{0}\}$ such that $x^{\mathrm{T}} A x = x^{\mathrm{T}} D x = 0$. Suppose to the contrary that we have such an x. Then, since D is diagonal, we have $x^{\mathrm{T}} D x = \sum_{i=1}^n D_{ii} x_i^2 = 0$. In particular, for $D_{ii} > 0$, we have $x_i = 0$. Now, note that:

$$x^{\mathrm{T}} A x = -\sum_{(i,j)\in N_x} (x_i - x_j)^2 A_{ij}$$

Thus, $x^{\mathrm{T}} A x = 0$ implies that $x_i = x_j$ for all $(i,j) \in N_x$. Since $N_a \neq \emptyset$, there exists an i such that $D_{ii} > 0$, whence $x_i = 0$. Since the network is connected, it follows that $x = \mathbf{0}$. □

Proposition 5 gives us a recipe for putting U into the desired form. First, we set U_{22} to be a positive definite matrix. Then, we need to set $U_{12} = -\bar{X} U_{22}$, where $\bar{X}$ is the matrix containing the true locations of the sensors. We now investigate when this is possible. Note that the above condition is simply a system of linear equations. Let A_i be the set of sensors connected to anchor i, and let E be the number of sensor–sensor edges. Then, the above system has $E + \sum_i |A_i|$ variables. The number of equations is $E + 3m$, where m is the number of sensors that are connected to some anchors. Hence, a sufficient condition for solvability is that the system of equations are linearly independent, and that $\sum_i |A_i| \ge 3m$. In particular, this shows that the trilateration graphs defined in [15] are strongly localizable.

We now develop the next theorem.

Theorem 4 *If a problem (graph) contains a subproblem (subgraph) that is strongly localizable, then the submatrix solution corresponding to the subproblem in the SDP solution has rank d. That is, the SDP relaxation computes a solution that localizes all possibly localizable unknown points.*

Proof Let the subproblem have n_s unknown points and they are indexed as $1, \ldots, n_s$. Since it is strongly localizable, an optimal dual slack matrix U_s of the SDP relaxation for the subproblem has rank n_s. Then, in the dual problem of the SDP relaxation for the whole problem, we set V and those w_{kj}'s associated with the subproblem to the optimal slack matrix U_s and set all other w_{kj}'s to 0. Then, the slack matrix:

$$U = \begin{pmatrix} U_s & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \succeq \mathbf{0}$$

must be optimal for the dual of the (whole-problem) SDP relaxation, and it is complementary to any primal feasible solution of the (whole-problem) SDP relaxation:

$$Z = \begin{pmatrix} Z_s & * \\ * & * \end{pmatrix} \succeq \mathbf{0} \quad \text{where} \quad Z_s = \begin{pmatrix} I_d & X_s \\ X_s^{\mathrm{T}} & Y_s \end{pmatrix}$$

However, we have $0 = Z \bullet U = Z_s \bullet U_s$ and $U_s, Z_s \succeq \mathbf{0}$. The rank of U_s is n_s implies that the rank of Z_s is exactly d, i.e. $Y_s = (X_s)^{\mathrm{T}} X_s$, so X_s is the unique realization of the subproblem. □

7 A comparison of notions

In this section, we will show that the notions of unique localizability, strong localizability and rigidity in $\mathcal{R}^2$ are all distinct.

7.1 Unique localizability $\not\Rightarrow$ strong localizability

We have already remarked earlier that a strongly localizable graph is necessarily uniquely localizable. However, as we shall see, the converse is not true.

Let G_1 be the network shown in Fig. 1(a). The key feature of G_1 is that the sensor x_2 lies on the line joining anchors a_1 and a_3. It is not hard to check that this network is uniquely localizable. Now, suppose to the contrary that G_1 is strongly localizable. Then, the dual slack matrix U admits the decomposition $U = (-\bar{X}^{\mathrm{T}}, I)^{\mathrm{T}} W (-\bar{X}^{\mathrm{T}}, I)$. It is easy to verify that:

$$U_{12} = (\bar{y}_{21} a_2 + \bar{y}_{31} a_3, \bar{y}_{12} a_1 + \bar{y}_{32} a_3)$$
$$U_{22} = \begin{pmatrix} -(\bar{y}_{21} + \bar{y}_{31}) - y_{12} & y_{12} \\ y_{12} & -(\bar{y}_{12} + \bar{y}_{32}) - y_{12} \end{pmatrix}$$

and the form of U requires that $U_{12} = -\bar{X} U_{22}$. This is equivalent to the following system of equations:

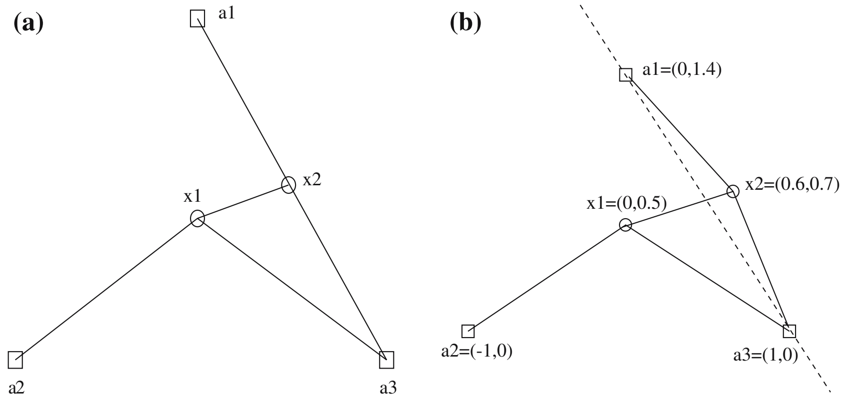

Fig. 1 A comparison of graph notions. **a** A uniquely localizable, but not strongly localizable network **b** A rigid network that is not uniquely localizable

$$(\bar{x}_1 - a_2)\bar{y}_{21} + (\bar{x}_1 - a_3)\bar{y}_{31} = (\bar{x}_1 - \bar{x}_2)y_{12} \tag{8}$$

$$(\bar{x}_2 - a_1)\bar{y}_{12} + (\bar{x}_2 - a_3)\bar{y}_{32} = -(\bar{x}_1 - \bar{x}_2)y_{12} \tag{9}$$

Since $\bar{x}_2$ lies on the affine space spanned by a_1 and a_3, Eq. (9) implies that $y_{12} = 0$. However, Eq. (8) would then imply that $\bar{x}_1$ lies on the affine space spanned by a_2 and a_3, which is a contradiction. Thus, we conclude that G_1 is not strongly localizable.

7.2 Rigid in $\mathcal{R}^2 \not\Rightarrow$ unique localizability

By definition, a uniquely localizable network is rigid in $\mathcal{R}^2$. However, the converse is not true. To see this, let G_2 be the network shown in Fig. 1(b).

Note that G_2 can be viewed as a perturbed version of G_1. It is easy to verfiy that G_2 is rigid. Thus, by Theorem 2, it can fail to be uniquely localizable only if it has a realization in some higher dimension. Indeed, the above network has an three-dimensional realization. The idea for constructing such a realization is as follows. Let us first remove the edge (x_1, x_2). Then, reflect the subgraph induced by a_1, x_2, a_3 across the dotted line. Now, consider two spheres, one centered at a_2 and the other centered at a_3, both having radius $\sqrt{5}/2$. The intersection of these spheres is a circle, and we can move x_1 along this circle until the distance between x_1 and x_2 equals to the prespecified value. Then, we can put the edge (x_1, x_2) back and obtain an three-dimensional realization of the network.

More precisely, for the above realization, the reflected version of x_2 has coordinates $x_2' = \left(\frac{173}{370}, \frac{112}{185}, 0\right)$. Now, let $x_1' = \left(0, \frac{23}{64}, \frac{\sqrt{495}}{64}\right)$. It is straightforward to verify that:

$$\|x_1 - a_2\|^2 = \|x_1' - a_2\|^2 = \frac{5}{4}$$

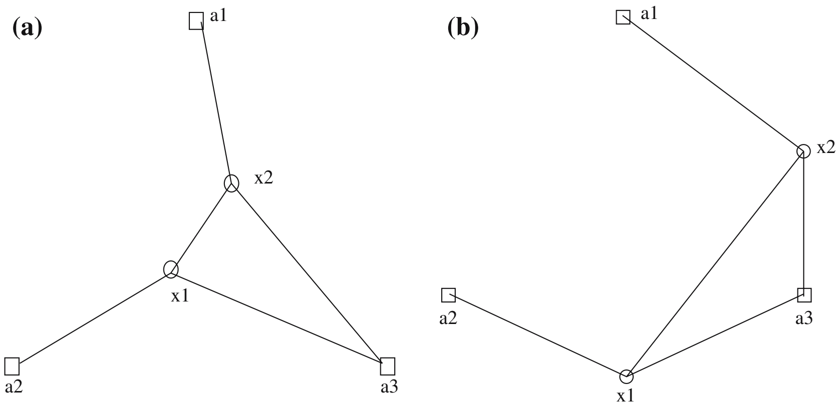

Fig. 2 Strongly localizable networks

$$\|x_1 - a_3\|^2 = \|x_1' - a_3\|^2 = \frac{5}{4}$$
$$\|x_1 - x_2\|^2 = \|x_1' - x_2'\|^2 = \frac{2}{5}$$

Hence, we conclude that G_2 is not uniquely localizable.

It would be nice to have a characterization on those graphs which are rigid in the plane but have higher dimensional realizations. However, finding such a characterization remains a challenging task, as such characterization would necessarily be non–combinatorial, and would depend heavily on the geometry of the network. For instance, the networks shown in Fig. 2, while having the same combinatorial property as the one shown in Fig. 1b, are uniquely localizable (in fact, they are both strongly localizable):

8 Conclusion

In this paper we have studied the Sensor Network Localization problem, which is a variant of the Graph Realization problem. We have shown for the first time that the SDP method yields an algorithm that guarantees to find the solution if the input graph is uniquely localizable. Moreover, we have defined various notions of localizability and demonstrated their relationship with classical rigidity theory. However, this work has still left many interesting open questions unanswered. First, for those instances that are not uniquely localizable, it would be interesting to investigate how many anchors are needed and how should they be placed in order to make the instance uniquely localizable. Secondly, our SDP model assumes that the input data are noise-free. However, sensor measurements are often noisy, and it is important to have a model that can handle noisy data and has good theoretical properties. Thirdly, besides the distance measurements, there may be extra information available to help us determine the desired realization. For instance, we may have angle estimates between a pair of sensors that are within communication range. It would be desirable to develop a model that incorporates and exploits these information. For some results in this direction, see, e.g., [9].

Acknowledgments We would like to thank Jiawei Zhang for his careful reading of the manuscript. We are also grateful to the referees whose detailed comments greatly improve the presentation of the paper.

References

1. Alfakih, A.Y.: Graph rigidity via euclidean distance matrices. Linear Algebra Appl. **310**, 149–165 (2000)
2. Alfakih, A.Y.: On rigidity and realizability of weighted graphs. Linear Algebra Appl. **325**, 57–70 (2001)
3. Alfakih, A.Y., Khandani, A., Wolkowicz, H.: Solving euclidean distance matrix completion problems via semidefinite programming. Comput. Opt. Appl. **12**, 13–30 (1999)
4. Alfakih, A.Y., Wolkowicz, H.: On the embeddability of weighted graphs in euclidean spaces. Research Report CORR 98-12, University of Waterloo, Department of Combinatorics and Optimization (1998)
5. Alfakih, A.Y., Wolkowicz, H.: Euclidean distance matrices and the molecular conformation problem. Research Report CORR 2002-17, University of Waterloo, Department of Combinatorics and Optimization (2002)
6. Alizadeh, F.: Interior point methods in semidefinite programming with applications to combinatorial optimization. SIAM J. Opt. **5**, 13–51 (1995)
7. Aspnes, J., Goldenberg, D., Yang, Y.R.: On the computational complexity of sensor network localization. ALGOSENSORS 2004, in LNCS **3121**, 32–44 (2004)
8. Bădoiu, M.: Approximation algorithm for embedding metrics into a two-dimensional space. Proceedings 14th SODA, pp. 434–443 (2003)
9. Bădoiu, M., Demaine, E.D., Hajiaghayi, M.T., Indyk, P.: Low-dimensional embedding with extra information. Proceedings 20th SoCG, pp. 320–329 (2004)
10. Barvinok, A.: Problems of distance geometry and convex properties of quadratic maps. Disc. Comput. Geom. **13**, 189–202 (1995)
11. Barvinok, A.: A course in convexity. AMS (2002)
12. Biswas, P., Ye, Y.: Semidefinite programming for Ad Hoc wireless sensor network localization. Proceedings 3rd IPSN, pp. 46–54 (2004)
13. Boyd, S., Ghaoui, L.E., Feron, E., Balakrishnan, V.: Linear matrix inequalities in system and control theory. SIAM (1994)
14. Doherty, L., Ghaoui, L.E., Pister, S.J.: Convex position estimation in wireless sensor networks. Proceedings 20th INFOCOM, Vol. 3, pp. 1655–1663 (2001)
15. Eren, T., Goldenberg, D.K., Whiteley, W., Yang, Y.R., Moore, A.S., Anderson, B.D.O., Belhumeur, P.N.: Rigidity, computation, and randomization in network localization. Proceedings 23rd INFOCOM (2004)
16. Goldfarb, D., Scheinberg, K.: Interior point trajectories in semidefinite programming. SIAM J. Opt. **8**(4), 871–886 (1998)
17. Gower, J.C.: Some distance properties of latent root and vector methods in multivariate analysis. Biometrika **53** 325–338 (1966)
18. Graver, J., Servatius, B., Servatius, H.: Combinatorial rigidity. AMS (1993)
19. Güler, O., Ye, Y.: Convergence behavior of interior point algorithms. Math. Prog. **60**, 215–228 (1993)
20. Hendrickson, B.: Conditions for unique graph realizations. SIAM J. Comput. **21**(1), 65–84 (1992)
21. Hendrickson, B.: The molecule problem: exploiting structure in global optimization. SIAM J. Opt. **5**(4), 835–857 (1995)
22. Jackson, B., Jordán, T.: Connected rigidity matroids and unique realizations of graphs. Preprint (2003)
23. Laurent, M.: Matrix completion problems. The Encycl. Optim. **3**, 221–229 (2001)

24. Linial, N., London, E., Rabinovich, Yu.: The geometry of graphs and some of its algorithmic applications. Combinatorica **15**(2), 215–245 (1995)
25. Moré, J., Wu, Z.: Global continuation for distance geometry problems. SIAM J. Opt. **7**, 814–836 (1997)
26. Savarese, C., Rabay, J. Langendoen, K.: Robust positioning algorithms for distributed Ad-Hoc wireless sensor networks. USENIX Annual Technical Conference (2002)
27. Savvides, A., Han, C.-C., Srivastava, M.B.: Dynamic fine-grained localization in Ad-Hoc networks of sensors. Proceedings 7th MOBICOM, pp. 166–179 (2001)
28. Savvides, A., Park, H., Srivastava, M.B.: The bits and flops of the n-hop multilateration primitive for node localization problems. Proceedings 1st WSNA, pp. 112–121 (2002)
29. Schoenberg, I.J.: Remarks to Maurice Fréchet's Article "Sur la Définition Axiomatique d'une Classe d'Espace Distanciés Vectoriellement Applicable sur l'Espace de Hilbert". Ann. Math. **36**(3), 724–732 (1935)
30. Shang, Y., Ruml, W., Zhang, Y., Fromherz, M.P.J.: Localization from mere connectivity. Proceedings 4th MOBIHOC, pp. 201–212 (2003)
31. Torgerson, W.S.: Multidimensional scaling: I. theory and method. Psychometrika **17**, 401–419 (1952)
32. Trosset, M.W.: Distance matrix completion by numerical optimization. Comput. Opt. Appl. **17**, 11–22 (2000)
33. Trosset, M.W.: Extensions of classical multidimensional scaling via variable reduction. Comput. Stat. **17**(2), 147–162 (2002)
34. Young, G., Householder, A.S.: Discussion of a set of points in terms of their mutual distances. Psychometrika **3**, 19–22 (1938)