Semidefinite Programming Approaches for Sensor Network Localization with Noisy Distance Measurements

Pratik Biswas, Tzu-Chen Liang, Kim-Chuan Toh, Ta-Chung Wang and Yinyu Ye

***Abstract*— A sensor network localization problem is to determine the positions of the sensor nodes in a network given incomplete and inaccurate pairwise distance measurements. Such distance data may be acquired by a sensor node by communicating with its neighbors. We describe a general semidefinite programming (SDP) based approach for solving the graph realization problem, of which the sensor network localization problems is a special case. We investigate the performance of this method on problems with noisy distance data. Error bounds are derived from the SDP formulation. The sources of estimation error in the SDP formulation are identified.**

The SDP solution usually has a rank higher than the underlying physical space, which when projected onto the lower dimensional space generally results in high estimation error. We describe two improvements to ameliorate such a difficulty. First, we propose a regularization term in the objective function that can help to reduce the rank of the SDP solution. Second, we use the points estimated from the SDP solution as the initial iterate for a gradient-descent method to further refine the estimated points. A lower bound obtained from the optimal SDP objective value can be used to check the solution quality. Experimental results are presented to validate our methods and show that they outperform existing SDP methods.

***Index Terms*— Position Estimation, Graph Realization, Semidefinite Programming, Dimensionality Reduction**

I. Introduction

A typical sensor network consists of a large number of sensors which are densely deployed in a geographical area. Sensors collect the local environmental information such as temperature or humidity and can communicate to each other. The advance of Micro-electro-mechanical system (MEMS) and wireless communication technology have made the sensor network a low cost and highly efficient method for environmental observations. A successful example of habitat monitoring system is the Great Duck Island (GDI) [1] system. Other applications like battlefield surveillance, moving object tracking and asset location are also attracting large research efforts.

Author names are in alphabetical order. This work is partially supported by the Institute for Mathematical Sciences of the National University of Singapore.

Department of Electrical Engineering, Stanford University. E-mail: pbiswas@stanford.edu.

Department of Aeronautics & Astronautics, Stanford University.

Department of Mathematics, National University of Singapore, Singapore 117543. E-mail: mattohkc@nus.edu.sg.

Department of Aeronautics & Astronautics, Stanford University.

Department of Management Science and Engineering and, by courtesy, Electrical Engineering, Stanford University, Stanford, CA 94305. E-mail: yinyu-ye@stanford.edu.

The information collected through a sensor network can be interpreted and relayed far more effectively if we know where it is coming from and where it needs to be sent. Therefore, it is often very useful to know the positions of the sensor nodes in a network. Usually sensor networks consist of a large number of sensors and it is too expensive to use Global Positioning System (GPS)[2] to locate their positions. Therefore positioning methods using neighboring distance measurements are developed. The ad hoc sensor network localization problem is: assuming the accurate positions of some nodes (called anchors) are known, how to use them and partial pair-wise distance measurements to locate the positions of all sensor nodes in the network, see [3], [4], [5], [6], [7], [8], [9], [10], [11]. The difficulty of this problem arises from 2 factors. First, the distance measurements always have some noise or uncertainty. Since the effect of the measurement uncertainty usually depends on the geometrical relationship between sensors and is not known a priori, an unbiased model is not easy to build. Second, even if the distance measurements are perfectly accurate, a sufficient condition for this sensor network to be localizable cannot be identified easily, see [12], [13].

In practice, increasing the number of anchors and the radio range of communication usually make the localization result more satisfactory, but this also implies incurring more cost. Hence reliable algorithms to solve the localization problem, as well as the knowledge of how to deploy a sensor network so that the sensors can be located accurately, are both important research issues. Various techniques have been developed to overcome the measurement uncertainties. Most of these methods are based on minimizing some global error functions (called the objective function), which can be different when the model of uncertainty changes. Depending on the kind of optimization model being formulated, the characteristic and computation complexity varies. For example, the maximum likelihood estimation for sensor network localization problem is a non-convex optimization problem. Existing algorithms have limited success, even for small problem sizes, see [14]. On the other hand, if we relax the distance constraints, the problem can be formulated as a second order cone program (SOCP) and cheap and scalable algorithms can be applied [5], [15]. However, the solution is acceptable only when the anchors are densely deployed on the boundary of the sensor network. Another approach is to allow a large radio range for every sensor so that they can communicate with many other sensors or anchors. But this needs a higher consumption of battery power for each sensor.

Most existing approaches apply various degrees of relaxation to the original problem. One recent relaxation is the SDP relaxation developed in [16], [17], [18], [19], [20], [21], [22], which is closely related to SDP relaxations for other distance geometry problems, see [23], [24], [25], [26]. It was subsequently proved that when there is no noise in the measured distances, the localization problem can be solved in polynomial time under a uniquely localizable assumption [16]. However, if the sensor network is not uniquely localizable, there must exist a higher (rank) dimensional localization that minimizes the objective function, and SDP relaxation always produces this maximal (rank) dimensional solution. A classical way to obtain a 2-dimensional solution is to project the high dimensional solution onto the 2-dimensional space, but this generally does not produce satisfactory results. Moreover, in a real world ad hoc wireless sensor network, the distance measurements inevitably have noises. Using this noisy distance information, we can only formulate the localization problem as an optimization problem that minimizes the difference between the measured distances and that of the estimated locations. The max-rank property of the SDP solution [16] makes the relaxation solution almost always lies in a higher (rank) dimensional space, and no robust and reliable method is available to round the high dimensional SDP solution.

In this paper, we formulate the sensor network localization problem in terms of a general graph realization problem, relax it to an SDP, and derive results on the upper and lower bounds on the SDP objective function. The SDP objective function gives us an idea of the accuracy of the technique and also provides a "proof of quality" against which we can measure any estimation.

We have developed 2 new techniques to obtain better estimation accuracy. One approach is to add a regularization term to the objective function to force the SDP solution to lie close to a low dimensional subspace of $\mathcal{R}^n$. The addition of a regularization term penalizes folding between the points and maximizes the spacing between them. Previously, a similar idea has been used for dimensionality reduction in machine learning [27].

The other approach is to improve the SDP estimated solution using a gradient-descent method. One should note that gradient methods generally do not deliver a global optimal solution when the problem is non-convex. However, our numerical results show that the SDP estimated solution generally give a good starting point for the gradient-descent method to converge to the global optimal solution. It is demonstrated that the improved solution is very close to the optimal one by comparing the minimal objective value to the SDP lower bound.

Section II describes the problem setup and a variety of formulations using the SDP relaxation model. In Section III, we derive upper and lower bounds for the SDP objective function. The estimation accuracy of SDP based algorithms for noisy distance data is also discussed. Section IV introduces the use of regularization terms in the SDP objective function in order to obtain lower dimensional solutions. Section V deals with the use of post processing gradient methods using the SDP estimated solution as a starting point. Finally, experimental results that show that these methods significantly improve the performance of SDP based localization algorithms are presented in Section VI.

II. The Semidefinite Programming Model

We start by introducing the notation used in this paper. For two symmetric matrices A and B, $A \succeq B$ means $A - B \succeq 0$, i.e. $A - B$ is a positive semidefinite matrix. The standard trace inner product between A, B is denoted by $\langle A, B\rangle$. We use I_d and $\mathbf{0}$ to denote the $d \times d$ identity matrix and the vector of all zeros, whose dimensions will be clear in the context. The 2-norm of a vector x is denoted as $\|x\|$. We use the notation $[A\,;\,B]$ to denote the matrix obtained by appending B to the last row of A.

Consider a sensor network in $\mathcal{R}^2$ with m anchors and n sensors. An anchor is a node whose location a_k in $\mathcal{R}^2$, $k = 1, \ldots, m$, is known, and a sensor is a node whose location x_i in $\mathcal{R}^2$ is yet to be determined, $i = 1, \ldots, n$. For a pair of sensors x_i and x_j, their Euclidean distance is denoted as d_{ij}. Similarly, for a pair of sensor x_i and anchor a_k their Euclidean distance is denoted as d_{ik}. In general, not all pairs of sensor/sensor and sensor/anchor distances are known, so the known pair-wise distances of sensor/sensor and sensor/anchor are denoted as $(i,j) \in \mathcal{N}$ and $(i,k) \in \mathcal{M}$, respectively.

Mathematically, the localization problem in $\mathcal{R}^2$ can be stated as follows: given m anchor locations $a_k, k = 1, \ldots, m$ and some distance measurements $d_{ij}, (i,j) \in \mathcal{N}, d_{ik}, (i,k) \in \mathcal{M}$, find the locations of n sensors $x_i, i = 1, \ldots, n$. One should note that this problem is in fact a special case of general distance geometry and graph realization problems in $\mathcal{R}^d$ where $d = 2$. The algorithms we propose are not restricted to $\mathcal{R}^2$, so for generality, we will deal with the graph realization problem and illustrate its applicability to the sensor network localization problem as an example.

Let $\mathcal{V} := \{1, \ldots, n\}$ and $\mathcal{A} := \{a_1, \ldots, a_m\}$. The realization problem in $\mathcal{R}^d$ for the graph $(\mathcal{V}, \mathcal{A}; D)$ is to determine the coordinates of the unknown points $x_1, \ldots, x_n$ from the partial distance data $D = \{d_{ij} : (i,j) \in \mathcal{N} \cup \mathcal{M}\}$.

The realization problem can be formulated as the following error minimization problem:

$$\min_{x_1,\ldots,x_n \in \mathcal{R}^d} \left\{ \begin{array}{l} \sum_{(i,j)\in\mathcal{N}} \gamma_{ij} \left| \|x_i - x_j\|^2 - d_{ij}^2 \right| \\ + \sum_{(i,j)\in\mathcal{M}} \gamma_{ij} \left| \|x_i - a_j\|^2 - d_{ij}^2 \right| \end{array} \right\}, \qquad (1)$$

where $\gamma_{ij} > 0$ are given weights. By adding weights to the distance constraints, we are essentially giving extra emphasis to those distance measurements for which we have higher confidence and vice versa. This would be particularly useful in cases where the distance measures are noisy but there is some prior information about which of those are more reliable. In such cases, we can assign higher weights to the distance constraints corresponding to the more reliable measures. Similar ideas have been explored for other localization algorithms in [28]. However, this idea has not been explored in much detail for our algorithm. The use of intelligent weighting schemes is being investigated and will be reported elsewhere.

Let $X = [x_1 \ x_2 \ \dots \ x_n] \in \mathcal{R}^{d\times n}$ be the position matrix that needs to be determined. It is readily shown that

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

$$\|x_i - a_j\|^2 = a_{ij}^T [X \quad I_d]^T [X \quad I_d] a_{ik}$$

where $e_{ij} = e_i - e_j$, and $a_{ij} = [e_i; -a_j]$. Here e_i is the ith unit vector in $\mathcal{R}^n$, and a_{ij} is the vector obtained by appending $-a_j$ to e_i.

For convenience, we let $g_{ij} = a_{ij}$ for $(i,j) \in \mathcal{M}$ and $g_{ij} = [e_{ij}; \mathbf{0}_m]$ for $(i,j) \in \mathcal{N}$. We further let $\mathcal{E} = \mathcal{N} \cup \mathcal{M}$. Let $Y = X^T X$. Then the problem (1) can be rewritten as:

$$\min \left\{ \sum_{(i,j)\in\mathcal{E}} \gamma_{ij} \left| g_{ij}^T [Y, X^T; X, I_d] g_{ij} - d_{ij}^2 \right| \, : \, Y = X^T X \right\} \tag{2}$$

We should emphasize that when there are no anchors, it is important to eliminate the translational degree of freedom by fixing the centroid of the configuration at the origin. This is equivalent to adding the constraint $e^T (X^T X) e = \|Xe\|^2 = 0$ to (2).

Unfortunately, the problem (2) is not a convex optimization problem. Our method is to relax the problem (2) to a semidefinite program by relaxing the constraint $Y = X^T X$ to $Y \succeq X^T X$. The last matrix inequality is equivalent to (Boyd et al. [25]) the condition that $[Y \ X^T; \ X \ I_d] \succeq 0$. Let $\mathcal{K} = \{Z \, : \, Z = [Y \quad X^T; X \quad I_d] \succeq 0\}$. The SDP relaxation of (2) can be written as the following SDP problem:

$$v^* := \min_{Z\in\mathcal{K}} \left\{ g(Z; D) := \sum_{(i,j)\in\mathcal{E}} \gamma_{ij} \left| g_{ij}^T Z g_{ij} - d_{ij}^2 \right| \right\}. \tag{3}$$

Let $\mathcal{I} = \{(i,j) \, : \, n+1 \le i \le j \le n+d\}$ and $E_{ij} = (e_i e_j^T + e_j e_i^T)/2$ for $(i,j) \in \mathcal{I}$. Also let δ_{ij} be the Kronecker delta. The dual of (3) is as follows:

$$\max \left\{ \begin{array}{l} \sum_{(i,j)\in\mathcal{E}} \lambda_{ij} d_{ij}^2 + \sum_{(i,j)\in\mathcal{I}} \lambda_{ij}\delta_{ij} \ : \\ \sum_{(i,j)\in\mathcal{E}} \lambda_{ij} g_{ij} g_{ij}^T + \sum_{(i,j)\in\mathcal{I}} \lambda_{ij} E_{ij} \ \preceq \ 0, \\ |\lambda_{ij}| \ \le \ \gamma_{ij} \ \forall\, (i,j) \in \mathcal{N} \end{array} \right\}. \tag{4}$$

When the given distance data is exact, the optimal objective value in problem (1) is zero. The question is whether there is enough distance information to determine the position of all the unknown points. This problem has been discussed in [16] in detail, and a condition of unique localizability or realizability is given. We repeat it here,

Definition 1: Problem (1) is uniquely localizable if it has a unique optimal solution $\bar{X}$ in $\mathcal{R}^{d\times n}$, with objective value 0, and there is no x_j in $\mathcal{R}^l, j = 1, ..., n$, where $l > d$ (excluding the case appending all zeros to $\bar{X}$), such that

$$\begin{aligned} \|x_i - x_j\|^2 &= d_{ij}^2, \ \forall\, (i,j) \in \mathcal{N} \\ \|x_i - (a_j; \mathbf{0})\|^2 &= d_{ij}^2, \ \forall\, (i,j) \in \mathcal{M}. \end{aligned}$$

The latter condition in the definition says that the problem cannot be localized in a higher dimensional space where anchor points are augmented to $(a_k; \mathbf{0}) \in \mathcal{R}^l, j = 1, ..., m$. The result in [16] stated that if the problem is uniquely localizable, then the relaxation problem (3) solves (1) exactly. Without solving (3) , it is difficult to tell whether a problem is uniquely localizable. But once solved, one can check whether $Y = X^T X$ in the SDP solution and immediately know if the problem is uniquely localizable.

For the localization problem with measurement noises, the story can be quite different. In general, the optimal objective value in (1) is not zero. Therefore, the optimization problem aims to minimize the difference between the measured distances and those derived from the estimated positions.

III. SDP WITH NOISY DISTANCE DATA

In this section, we analyze the effect of noisy distance measures on the SDP objective function value. The SDP objective gives us an insight into the actual estimation error that can be introduced and its dependence on the noise.

We assume that the distances d_{ij} and d_{ik} are perturbed by random noises ϵ_{ij} and ϵ_{ik}:

$$\begin{aligned} d_{ij} &= \hat{d}_{ij}|1+\epsilon_{ij}|, \quad (i,j) \in \mathcal{N} \\ d_{ik} &= \hat{d}_{ik}|1+\epsilon_{ik}|, \quad (i,k) \in \mathcal{M}, \end{aligned}$$

where $\hat{d}_{ij}$ is the true distance between points i and j, and $\hat{d}_{ik}$ is the true distance between point i and anchor k. We further assume that $E(\epsilon_{ij}) = 0$, $E(\epsilon_{ik}) = 0$. We take the absolute value because the noise maybe higher than 1, and so we would get a negative value for distance, which is not meaningful.

Let $\hat{v}$ and $\widehat{Z}$ be the optimal objective value and minimizer of (3) for the problem $(\mathcal{V}, \mathcal{A}; \widehat{D})$, respectively. Let $\hat{\lambda}$ be the maximizer of (4) for the problem $(\mathcal{V}, \mathcal{A}; \widehat{D})$.

Assumption 1. The problem $(\mathcal{V}, \mathcal{A}; \widehat{D})$ is localizable, i.e., $\hat{v} = 0$ and $g_{ij}^T \widehat{Z} g_{ij} = \hat{d}_{ij}^2$ for all $(i,j) \in \mathcal{E}$.

A. Sum of Absolute Errors

Proposition 1: Consider the problem $(\mathcal{V}, \mathcal{A}; D)$ and assume that Assumption 1 holds. Let v_* be the optimal objective value of (3) for this problem. We have the following results.

(a)

$$E[v_*] \ \le \ \sum_{(i,j)\in\mathcal{E}} \gamma_{ij} \hat{d}_{ij}^2 E \left| 2\epsilon_{ij} + \epsilon_{ij}^2 \right| \tag{5}$$

(b)

$$E[v_*] \ge \max \left\{ \begin{array}{l} \displaystyle\sum_{(i,j)\in\mathcal{E}} \lambda_{ij} \hat{d}_{ij}^2 (1 + E[\epsilon_{ij}^2]) + \sum_{(i,j)\in\mathcal{I}} \lambda_{ij}\delta_{ij} \, : \\ \lambda \text{ is feasible for (4)} \end{array} \right\}$$

(c) If $\epsilon_{ij} \sim N(0, \tau^2)$ for $(i,j) \in \mathcal{E}$, then

$$E[v_*] \ \le \ (1.6\tau + \tau^2) \sum_{(i,j)\in\mathcal{E}} \gamma_{ij} \hat{d}_{ij}^2. \tag{6}$$

Proof. (a) Since $\widehat{Z} \in \mathcal{K}$, it is clear that

$$v_* \ \le \ g(\widehat{Z}; D) \ \le \ \sum_{(i,j)\in\mathcal{E}} \gamma_{ij} \left| \hat{d}_{ij}^2 (2\epsilon_{ij} + \epsilon_{ij}^2) \right|,$$

and (5) follows readily.

(b) For a fixed λ that is feasible for (4), we have by weak duality that

$$v_* \;\ge\; \sum_{(i,j)\in\mathcal{E}} \lambda_{ij} d_{ij}^2 + \sum_{(i,j)\in\mathcal{I}} \lambda_{ij}\delta_{ij}.$$

By taking expectation on both sides, we get

$$\begin{aligned} E[v_*] \;&\ge\; \sum_{(i,j)\in\mathcal{E}} \lambda_{ij} E[d_{ij}^2] + \sum_{(i,j)\in\mathcal{I}} \lambda_{ij}\delta_{ij} \\ &=\; \sum_{(i,j)\in\mathcal{E}} \lambda_{ij} \hat{d}_{ij}^2 (1 + E[\epsilon_{ij}^2]) + \sum_{(i,j)\in\mathcal{I}} \lambda_{ij}\delta_{ij}. \end{aligned}$$

Since the above inequality holds for any λ that is feasible for (4), the required inequality follows.

(c) The inequality in (6) follows from (5) by noting that $E\left|2\epsilon_{ij} + \epsilon_{ij}^2\right| \le \frac{4}{\sqrt{2\pi}}\tau + \tau^2 \le 1.6\tau + \tau^2$. ■

Proposition 1 (c) indicates that the expected optimal SDP objective value $E[v_*]$ can potentially grow as fast as the standard deviation τ of the noises ϵ_{ij}. The example in subsection III-C and Figure 1(a) shows empirically that $E[v_*]$ indeed can grow as fast as the upper bound predicted in Proposition 1 (c).

B. Sum of Squares Errors

We can also formulate the localization problem for $(\mathcal{V}, \mathcal{A}; D)$ as the following error minimization problem:

$$\min_{X\in\mathcal{R}^{d\times n}} \left\{ \begin{array}{c} \sum_{(i,j)\in\mathcal{N}} \gamma_{ij}^2 \left(\|x_i - x_j\|^2 - d_{ij}^2\right)^2 \\ + \sum_{(i,j)\in\mathcal{M}} \gamma_{ij}^2 \left(\|x_i - a_j\|^2 - d_{ij}^2\right)^2 \end{array} \right\}. \tag{7}$$

The advantage of the model in (7) is that the objective function is smooth, and standard local optimization methods can be applied directly. But unfortunately, its SDP relaxation is slightly more expensive to solve compared to (3).

We consider the following SDP relaxation of (7):

$$v_* := \min_{Z\in\mathcal{K}} \left\{ h(Z;D) := \sqrt{\sum_{(i,j)\in\mathcal{E}} \gamma_{ij}^2 (g_{ij}^T Z g_{ij} - d_{ij}^2)^2} \right\}. \tag{8}$$

The dual of (8) is given by

$$\max \left\{ \begin{array}{l} \sum_{(i,j)\in\mathcal{E}} \lambda_{ij} d_{ij}^2 + \sum_{(i,j)\in\mathcal{I}} \lambda_{ij}\delta_{ij} \; : \\ \sum_{(i,j)\in\mathcal{E}} \lambda_{ij} g_{ij} g_{ij}^T + \sum_{(i,j)\in\mathcal{I}} \lambda_{ij} E_{ij} \preceq 0 \\ \sum_{(i,j)\in\mathcal{E}} \lambda_{ij}^2/\gamma_{ij}^2 \le 1 \end{array} \right\}. \tag{9}$$

Let $B_{ij} = g_{ij} g_{ij}^T$ for $(i,j)\in\mathcal{E}$. It is not difficult to show that

$$\begin{aligned} h(Z;D)^2 \;&=\; \langle Z - \widehat{Z},\, \mathcal{Q}(Z - \widehat{Z}) - 2W\rangle \\ &\quad + \sum_{(i,j)\in\mathcal{E}} t_{ij}^2 \gamma_{ij}^2 \hat{d}_{ij}^4 + R(Z;D), \end{aligned}$$

where

$$\mathcal{Q} = \sum_{(i,j)\in\mathcal{E}} \gamma_{ij}^2 B_{ij}\otimes B_{ij}, \quad W = \sum_{(i,j)\in\mathcal{E}} s_{ij}\gamma_{ij}^2 \hat{d}_{ij}^2 B_{ij} \succeq 0$$

$$\begin{aligned} R(Z;D) \;&=\; -2\left\langle \sum_{(i,j)\in\mathcal{E}} (\xi_{ij} - s_{ij})\gamma_{ij}^2 \hat{d}_{ij}^2 B_{ij},\, Z - \widehat{Z} \right\rangle \\ &\quad + \sum_{(i,j)\in\mathcal{E}} \gamma_{ij}^2 \hat{d}_{ij}^4 (\xi_{ij}^2 - t_{ij}^2). \end{aligned}$$

Here $\xi_{ij} = 2\epsilon_{ij} + \epsilon_{ij}^2$, $s_{ij} = E[\xi_{ij}] = E[\epsilon_{ij}^2]$, and $t_{ij}^2 = E[\xi_{ij}^2] = E[4\epsilon_{ij}^2 + 4\epsilon_{ij}^3 + \epsilon_{ij}^4]$. Note that $E[R(Z;D)] = 0$ and $\langle Z - \widehat{Z},\, \mathcal{Q}(Z - \widehat{Z})\rangle = \sum_{(i,j)\in\mathcal{E}} \gamma_{ij}^2 (g_{ij}^T Z g_{ij} - \hat{d}_{ij}^2)^2$.

Proposition 2: Assume that $\gamma_{ij} = 1$ for all $(i,j)\in\mathcal{E}$ for simplicity. Consider the problem $(\mathcal{V}, \mathcal{A}; D)$ and assume that Assumption 1 holds. Then we have the following results.

(a)

$$\begin{aligned} &(E[v_*])^2 - \sum_{(i,j)\in\mathcal{E}} t_{ij}^2 \hat{d}_{ij}^4 \\ &\le \min_{Z\in\mathcal{K}} \left\{ \langle Z - \widehat{Z},\, \mathcal{Q}(Z - \widehat{Z}) - 2W\rangle \right\} \qquad (10) \\ &\le \min_{Y\succeq 0} \left\{ \begin{array}{l} \sum_{(i,j)\in\mathcal{N}} (e_{ij}^T Y e_{ij})^2 - 2 s_{ij} \hat{d}_{ij}^2 (e_{ij}^T Y e_{ij}) \\ + \sum_{i=1}^n |\mathcal{M}_i| Y_{ii}^2 - 2\beta_i Y_{ii} \end{array} \right\} \qquad (11) \\ &\le \min_{y\ge 0} \left\{ \begin{array}{l} \sum_{(i,j)\in\mathcal{N}} (y_i + y_j)^2 - 2 s_{ij} \hat{d}_{ij}^2 (y_i + y_j) \\ + \sum_{i=1}^n |\mathcal{M}_i| y_i^2 - 2\beta_i y_i \end{array} \right\} \qquad (12) \\ &\le 0 \end{aligned}$$

where $\mathcal{M}_i = \{j : (i,j)\in\mathcal{M}\}$ and $\beta_i = \sum_{j\in\mathcal{M}_i} s_{ij}\hat{d}_{ij}^2$.

(b)

$$E[v_*] \ge \max \left\{ \begin{array}{l} \sum_{(i,j)\in\mathcal{E}} \lambda_{ij} \hat{d}_{ij}^2 (1 + s_{ij}) + \sum_{(i,j)\in\mathcal{I}} \lambda_{ij}\delta_{ij} \; : \\ \lambda \text{ is feasible for (9)} \end{array} \right\}$$

(c) If $\epsilon_{ij} \sim N(0, \tau^2)$ for $(i,j)\in\mathcal{E}$, then

$$E[v_*] \;\le\; \tau\sqrt{4 + 3\tau^2} \sqrt{\sum_{(i,j)\in\mathcal{E}} \hat{d}_{ij}^4}. \tag{13}$$

Proof. (a) For any fixed $Z\in\mathcal{K}$, we have $v_*^2 \;\le\; h(Z;D)^2$. By taking expectation and using Jensen's inequality , we get

$$\begin{aligned} &(E[v_*])^2 - \sum_{(i,j)\in\mathcal{E}} t_{ij}^2 \hat{d}_{ij}^4 \le E[v_*^2] - \sum_{(i,j)\in\mathcal{E}} t_{ij}^2 \hat{d}_{ij}^4 \\ &\le E[h(Z;D)^2] - \sum_{(i,j)\in\mathcal{E}} t_{ij}^2 \hat{d}_{ij}^4 \\ &= \langle Z - \widehat{Z},\, \mathcal{Q}(Z - \widehat{Z}) - 2W\rangle \qquad (14) \end{aligned}$$

Since the above inequality holds for any $Z\in\mathcal{K}$, thus the inequality (10) hold.

Let the optimal objective value of the minimization problem in (10) be z_*. For a given $Y\succeq 0$, let $Y_0 = [Y, 0;\; 0, 0]$. Since $\widehat{Z} + Y_0 \in\mathcal{K}$, we have

$$\begin{aligned} z_* \;&\le\; \langle Y_0,\, \mathcal{Q}(Y_0) - 2W\rangle \\ &=\; \sum_{(i,j)\in\mathcal{N}} (e_{ij}^T Y e_{ij})^2 - 2 s_{ij} \hat{d}_{ij}^2 (e_{ij}^T Y e_{ij}) \\ &\quad + \sum_{(i,j)\in\mathcal{M}} Y_{ii}^2 - 2 s_{ij} \hat{d}_{ij}^2 Y_{ii}. \end{aligned}$$

Note that the last sum above is equal to $\sum_{i=1}^n |\mathcal{M}_i| Y_{ii}^2 - 2\beta_i Y_{ii}$. Since the above inequality holds for any $Y \succeq 0$, the inequality (11) follows readily. The inequality (12) follows readily from (11) by relaxing the constraint $Y \succeq 0$ to $Y = \text{diag}(y)$ with $y \geq 0$.

(b) The proof is similar to that in Proposition 1.

(c) The required result follows readily from (a). ■

Corollary 1: If $\mathcal{N} = \emptyset$, then $Y = \text{diag}((\beta_i/|\mathcal{M}_i|)_{i=1}^n)$ is a minimizer of (11), where $\mathcal{M}_i$ and β_i are defined as in Proposition 2.

Proof. In this special case, the objective function in (11) is given by $\sum_{i=1}^n |\mathcal{M}_i| Y_{ii}^2 - 2\beta_i Y_{ii}$, which is the sum of independent quadratic functions in Y_{ii}. Thus the optimum can easily be shown to be $\beta_i/|\mathcal{M}_i|$ for each i. ■

The above corollary indicates that when $\mathcal{N} = \emptyset$, the quantity $E[\|x_i - \hat{x}_i\|^2] = Y_{ii}$ is the estimation error variance of an unknown point given its distances only to a set of known points. The error variance turns out to be the average of the error variances of all the distances the unknown point has with the known points.

Proposition 2 (c) indicates that the expected optimal SDP objective value $E[v_*]$ can potentially grow as fast as the standard deviation τ of the noises ϵ_{ij}. Indeed the example in subsection III-C and Figure 1(a) shows empirically that $E[v_*]$ can grow proportionately to the upper bound in Proposition 2 (c).

C. Simulations

We consider randomly generated test problems. Throughout this section, the weights γ_{ij} in (1) are set to 1.

First, sixty points $\{\hat{x}_i : i = 1, \ldots, 60\}$ are randomly generated in the unit square $[-0.5, 0.5] \times [-0.5, 0.5]$ via the MATLAB command: `rand('state',0); x = rand(2,1) - 0.5`. Then the edge set $\mathcal{N}$ is generated by considering only pairs of points that have distances less than $R = 0.3$, i.e.,

$$\mathcal{N} = \{(i,j) : \|\hat{x}_i - \hat{x}_j\| \leq 0.3, \ 1 \leq i < j \leq 60\}.$$

If there are anchors, the edge set $\mathcal{M}$ is similarly defined by

$$\mathcal{M} = \{(i,k) : \|\hat{x}_i - a_k\| \leq 0.3, \ 1 \leq i \leq 60, \ 1 \leq k \leq m\}.$$

For the example in Figure 1, 4 anchors are placed at the positions $(\pm 0.45, \pm 0.45)$.

We assume that the distances d_{ij} and d_{ik} are perturbed by random noises ϵ_{ij} and ϵ_{ik} as follows:

$$\begin{aligned} d_{ij} &= \hat{d}_{ij}|1 + \tau\epsilon_{ij}|, \quad (i,j) \in \mathcal{N} \\ d_{ik} &= \hat{d}_{ik}|1 + \tau\epsilon_{ik}|, \quad (i,k) \in \mathcal{M}, \end{aligned}$$

where $\hat{d}_{ij}$ is the true distance between point i and j, and $\hat{d}_{ik}$ is the true distance between point i and anchor k. We assume that ϵ_{ij}, ϵ_{ik} are independent standard Normal random variables. In future examples, we will express the noise in terms of percentage, that is, a value of 0.1 for τ corresponds to 10% noise.

Also for later purpose, we will use the Root Mean Square Distance (RMSD) to measure the accuracy of the estimated positions $\{x_i : i = 1, \ldots, n\}$:

$$RMSD = \frac{1}{\sqrt{n}} \left(\sum_{i=1}^n \|\hat{x}_i - x_i\|^2 \right)^{1/2}. \qquad (15)$$

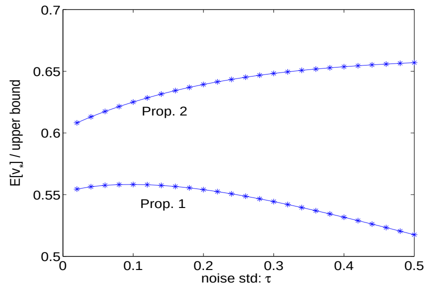
For each given τ, we generated 50 samples of $\{d_{ij} : (i,j) \in \mathcal{E}\}$. We solve the corresponding SDP (3) for these 50 samples to estimate the the expected optimal SDP objective value $E[v_*]$. In Figure 1(a), the ratio between the estimated $E[v_*]$ and the upper bounds given in Proposition 1 and 2 are plotted against the standard deviation τ of the noises. Figure 1(b) plots the ratios between the estimated $E[v_*]$ and the corresponding lower bounds for the 2 different models. As can be seen from the graphs, the upper bound is reasonably tight and is a good indicator of the expected objective value. The lower bound however is very loose and does not reveal too much information. The construction of tighter lower bounds are being investigated.

Figure 1(c) plots the ratios between the average RMSD error and $\sqrt{\tau}$ obtained from the SDP relaxation of (1) and (7). Observe that the average RMSD error roughly grows like $\tau^{1.5}$ when τ is small and $\sqrt{\tau}$ when τ is large. Therefore the RMSD grows at a higher rate when τ is large, because the distance measures become so erroneous that the structure of the graphs is altered significantly and they become nonuniquely localizable.

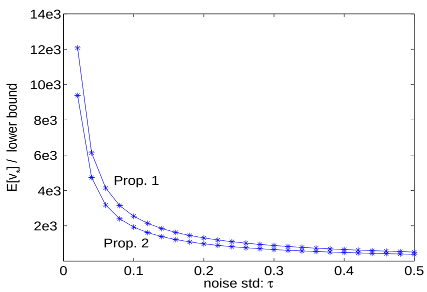
D. The High Rank Property of SDP Relaxations

When the distance measurements have errors, the distance constraints usually contradict each other and so there is no localization in $\mathcal{R}^d$. In other words, $Y \neq X^T X$. However, since the SDP approach relaxes the constraint $Y = X^T X$ into $Y \succeq X^T X$, it is still possible to locate the sensor in a higher dimensional space (or choose a Y with a higher rank) and to make the objective function value zero. The optimal solution in a higher dimensional space always results in a smaller objective function value than the one constrained in $\mathcal{R}^d$. Furthermore, the 'max-rank' property [29] implies that solutions obtained through interior point methods for solving SDPs converge to the maximum rank solutions. Hence, because of the relaxation of the rank requirement, the solution is "lifted" to a higher dimensional space. For example, imagine a rigid structure consisting of set of points in a plane (with the points having specified distances from each other). If we perturb some of the specified distances, the configuration may need to be readjusted by setting some of the points outside the plane.

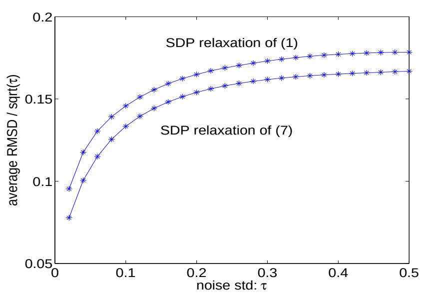
A main research topic is how to round the higher-dimensional (higher rank) SDP solution into a lower-dimensional (rank-2) solution. One way is to ignore the augmented dimensions and use the projection X^* as a suboptimal solution, which is the case in [18]. However, the projection typically leads to points getting 'crowded' together as shown in Figure 2(a). (Imagine the projection of the top vertex of a pyramid onto its base.) This is because a large contribution to



(a) $E[v_*]$ versus the upper bounds in Propositions 1 and 2.

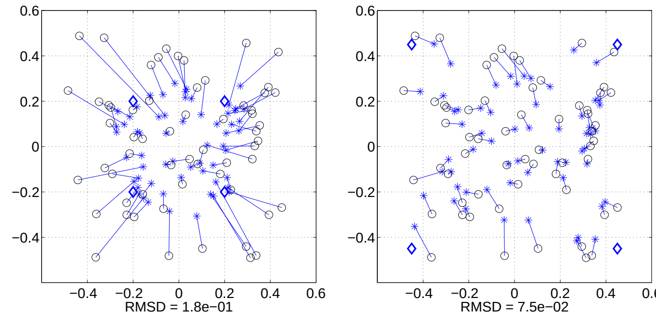


(b) $E[v_*]$ versus the lower bounds in Propositions 1 and 2.



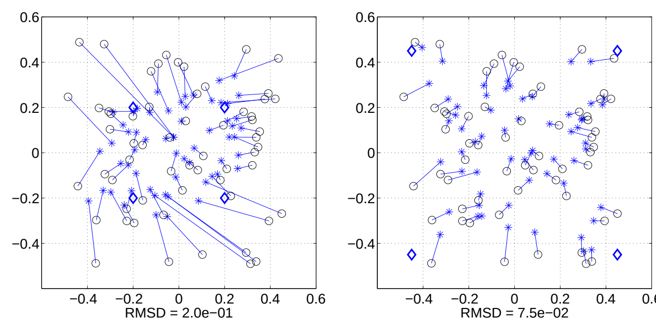
(c) average RMSD versus $\sqrt{\tau}$

Fig. 1. Ratios between $E[v_*]$ and the upper and lower bounds in Propositions 1 and 2; and the ratios between average RMSD and $\sqrt{\tau}$ obtained from the SDP relaxation of (1) and (7).



(a) 4 inner anchors: $(\pm 0.2, \pm 0.2)$ (b) 4 outer anchors: $(\pm 0.45, \pm 0.45)$

Fig. 2. 60 sensors, 4 anchors, 20 % noise (following the standard Normal distribution), Radio range=0.3. The positions are estimated from the SDP problem in (3).



(a) 4 inner anchors: $(\pm 0.2, \pm 0.2)$ (b) 4 outer anchors: $(\pm 0.45, \pm 0.45)$

Fig. 3. 60 sensors, 4 anchors, 20 % noise (following the uniform distribution in $[-\sqrt{3}, \sqrt{3}]$), Radio range=0.3. The positions are estimated from the SDP problem in (3).

the distance between 2 points could come from the dimensions we choose to ignore.

This problem can be reduced to some extent by placing anchors at the perimeter like in Figure 2(b). Since these anchors are constrained to be in a plane and are stretched out, the other points cannot fold together into higher dimensions and still maintain the distance constraints with these anchors. An intelligent anchor placement design that ensures lower dimensional embedding is part of our future research.

Example 1: An example of the effect of anchor placement is demonstrated in Figure 2 where for the same network of points and different anchor positions, the estimations are drastically different. The (blue) diamonds refer to the positions of the anchors; (black) circles to the original locations of the unknown sensors; and (blue) asterisks to their estimated positions from the SDP solution of (3). The discrepancies between the original and the estimated points are indicated by solid lines. (The same setup and notations will be used in all examples of this article).

Example 2: Here we consider the same network of points in Example 1, but the true distances are perturbed by random noises ϵ_{ij} and ϵ_{ik} following the uniform distribution in $[-\sqrt{3}, \sqrt{3}]$. The corresponding results are displayed in Figure 3. Observe that the plots in this figure are qualitatively similar to those in Figure 2.

The use of bounding away constraints (that correspond to points that must be far away from each other) can also reduce the problem of crowding. However, there may be too many such constraints for all pairs of points thus increasing the computational complexity of the technique. Warm start approaches have been discussed in [19] where bounding away constraints are added to a second SDP if they are being violated in the initial SDP (without any bounding constraints). However, it would be ideal if the SDP needs only to be solved once.

This motivates the use of regularization methods. In [27], regularization terms have been incorporated to the SDP arising from kernel learning in nonlinear dimensionality reduction. The purpose is to penalize folding and try to find a stretched map of the set of points while maintaining local distance relations. This idea is formally connected to the tensegrity theory for graph realization, that is, certain stretched graphs can always be realized in a lower dimensional space than those graphs without stretching; see [30]. In the next section, we describe a similar technique to encourage lower dimensional solutions. We will also describe the use of a gradient descent method to improve the SDP estimated solution. These methods turn out to be extremely effective and efficient.

IV. Regularization

It has been observed by many researchers that when the distance data is noisy, the points estimated from (1) tend to crowd together towards the center of the configuration. Here we propose a strategy to ameliorate this difficulty.

Let e be the n-dimensional vector of all ones, $\hat{e} = e/\sqrt{n+m}$, and $\hat{a} = \sum_{k=1}^{m} a_k/\sqrt{n+m}$. Let $\boldsymbol{a} = [\hat{e}; \hat{a}]$ and $A = [a_1, \ldots, a_m]$. Our strategy is to subtract the following regularization term from the objective function of (3) to prevent the estimated points from crowding together:

$$\frac{\lambda}{n+m}\Big(\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|x_i - x_j\|^2 + \sum_{i=1}^{n}\sum_{k=1}^{m}\|x_i - a_k\|^2\Big)$$

$$= \lambda\Big(\sum_{i=1}^{n}\|x_i\|^2 - \|X\hat{e} + \hat{a}\|^2\Big) + \lambda\Big(\frac{n}{n+m}\|A\|_F^2 + \|\hat{a}\|^2\Big)$$

$$= \lambda\langle I - \boldsymbol{a}\boldsymbol{a}^T,\, Z\rangle + \lambda\Big(\frac{n}{n+m}\|A\|_F^2 + \|\hat{a}\|^2 - d\Big),$$

where λ is a positive regularization parameter that is to be chosen.

Ignoring the constant term, the regularized form of (3) is given by

$$\min_{Z \in \mathcal{K}} \left\{ g(Z; D) \; - \lambda\langle I - \boldsymbol{a}\boldsymbol{a}^T,\, Z\rangle \right\}. \tag{16}$$

A. Choice of regularization parameter

An important issue in (16) is the optimum choice of the parameter λ. If it is too low, the regularization term will not affect the SDP solution significantly enough. However, if it is chosen to be too high, the regularization term will overwhelm the error term and thus causing the SDP to either be infeasible or to yield a solution where the points are stretched out too far apart.

The optimum choice of the parameter λ is dependent upon the size and geometry of the network and the distance information available. Currently, it is not known how to determine such an optimum value prior to solving the SDP for a particular network. We can start with $\lambda = 1$ and if the SDP is infeasible or the objective function has a high negative value below a predefined threshold, we scale λ by a factor less than 1 and solve a new SDP. The factor could be chosen to balance the magnitudes of the error term and the regularization term in (16). While this method works satisfactorily, it suffers the disadvantage of possibly having to solve several SDPs.

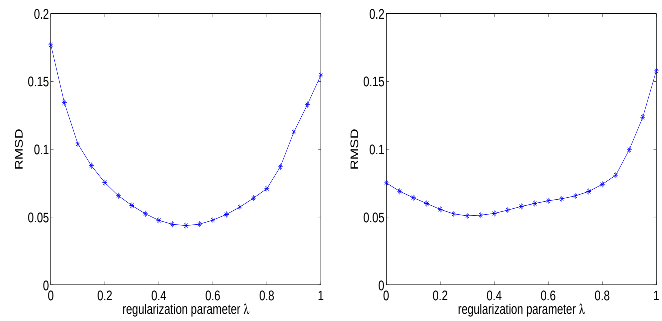
A heuristic choice of λ that we found to be reasonably effective is the following. Suppose Z^* is obtained from the non-regularized SDP (3). Let

$$\lambda^* \;=\; \frac{g(Z^*; D)}{\langle I - \boldsymbol{a}\boldsymbol{a}^T,\, Z^*\rangle}. \tag{17}$$

Our numerical experience show that the actual choice of λ should not be larger than λ^*, and $\lambda = \lambda^*$ typically works reasonably well. The heuristic choice of λ just described requires the solution of two SDPs.

It was observed from simulations that within a certain range of network sizes and radio ranges, a particular range of λ tends to be ideal. By setting the value of λ to be in this range prior to localization, it is very likely that an accurate solution can be obtained from the SDP (16) in the first attempt. However this behavior needs to be investigated in more detail in order to obtain a suitable choice of λ that guarantees a good estimate.

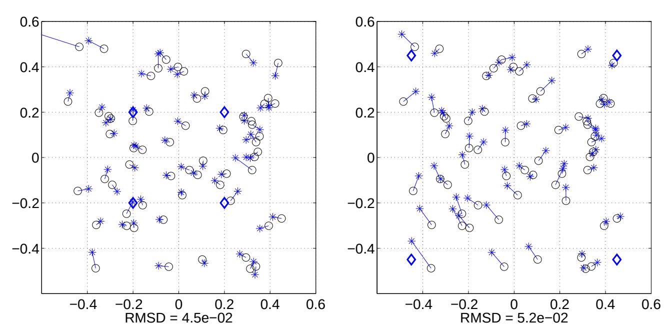
Figure 4 shows the RMSD error versus the regularization parameter λ for the problem in Figure 2. Observe that for a reasonably wide range of λ, the RMSD error are smaller than the RMSD error obtained from the non-regularized SDP in (3), which corresponds to the special case $\lambda = 0$.



(a) 4 inner anchors (b) 4 outer anchors

Fig. 4. RMSD error of the estimated positions obtained from SDP with regularization parameter λ for the examples in Figure 2.

Example 3: In the following example, we show the result of adding a regularization term for the cases discussed in Figure 2(a). The problem of crowding at the center as was seen before is eliminated as seen in Figure 5(a) for anchors placed in the interior. In Figure 5(b), we see that by choosing an appropriate value of λ, the regularization can be used to improve the solution even for the case where the anchors are in the exterior.



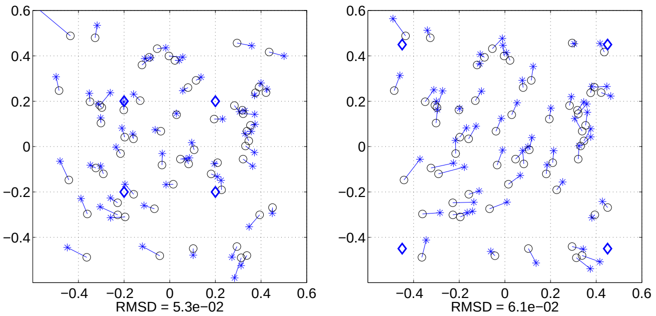
(a) 4 inner anchors (b) 4 outer anchors

Fig. 5. Same as Figure 2, but for SDP with regularization. The regularization parameter λ is chosen to be value in (17), which equals to 0.551 and 0.365, respectively.

Example 4: Here we show the result of adding a regularization term for the cases discussed in Figure 3, where the noises ϵ_{ij} follow an uniform distribution in $[-\sqrt{3}, \sqrt{3}]$.

B. Ideas for improvement

The idea of regularization can be linked to tensegrity theory and realizability of graphs in lower dimensions, see [30].



(a) 4 inner anchors (b) 4 outer anchors

Fig. 6. Same as Figure 3, but for SDP with regularization. The regularization parameter λ is chosen to be the value in (17), which equals to 0.532 and 0.363, respectively.

The notion of stress is used to explain this. By maximizing the distance between some vertices in a graph, the graph gets stretched out and there is a non-zero stress induced on the edges in the graph. For the configuration to remain in equilibrium, the total stress on a vertex must sum to zero. In order for the overall stress to cancel out completely, the graph must be in a low dimensional space. In this paper, we have not explored a strategy of selecting any particular edges to stretch. Techniques to select such edges can improve the robustness of the regularization technique and are a part of our future research.

V. Refinement by a Gradient Descent Method

The positions estimated from the SDP relaxation (3) or (16) can further be refined by applying a local optimization method to (1). However, as the objective function in (1) is non-smooth, it is more convenient to apply a local optimization method to model below:

$$\min_{X\in\mathcal{R}^{d\times n}} \left\{ \begin{array}{rcl} f(X) & := & \displaystyle\sum_{(i,j)\in\mathcal{N}} \gamma_{ij}^2 \left(\|x_i - x_j\| - d_{ij}\right)^2 \\ & & + \displaystyle\sum_{(i,j)\in\mathcal{M}} \gamma_{ij}^2 \left(\|x_i - a_j\| - d_{ij}\right)^2 \end{array} \right\}. \quad (18)$$

The reason that we use the above model instead of the one in (7) is purely based on our empirical experience that the former tends to perform better than the latter. The method we suggest to improve the SDP estimated solution is to move every sensor location along the negative gradient direction of the function $f(X)$ in (18) to reduce the error function value. Now, we will explain the gradient method in more detail.

Recall that $\mathcal{N}_j = \{i : (i,j) \in \mathcal{N}\}$ and $\mathcal{M}_j = \{k : (j,k) \in \mathcal{M}\}$. By using the fact that $\nabla_x \|x - b\| = (x-b)/\|x-b\|$ if $x \neq b$, it is easy to show that for the objective function $f(X)$ in (18), the gradient $\nabla_j f$ with respect to sensor x_j is given by:

$$\nabla_j f(X) \;=\; 2\sum_{i\in\mathcal{N}_j} \gamma_{ij}^2 \Big(1 - \frac{d_{ij}}{\|x_j - x_i\|}\Big)(x_j - x_i)$$
$$+\, 2\sum_{i\in\mathcal{M}_j} \gamma_{ij}^2 \Big(1 - \frac{d_{ij}}{\|x_j - a_i\|}\Big)(x_j - a_i)$$

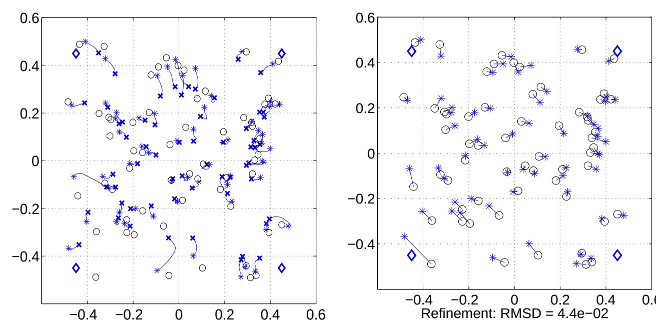
Notice that $\nabla_j f$ only involves sensors and anchors that are connected to x_j. Thus $\nabla_j f$ can be computed in a distributed fashion.

Let

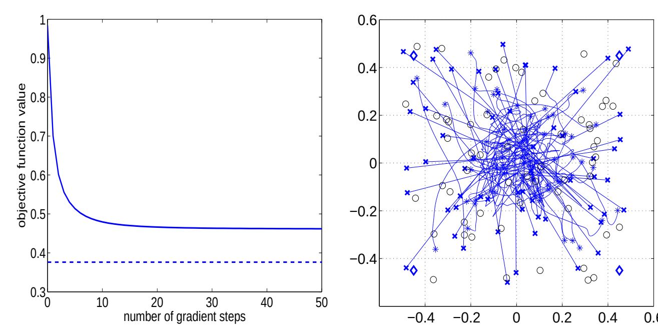
$$X(\alpha) \;=\; [x_1 - \alpha\nabla_1 f(X),\, \ldots,\, x_n - \alpha\nabla_n f(X)]. \quad (19)$$

By choosing the step size $\alpha \in (0,1]$ appropriately, the function value $f(X(\alpha))$ can be reduced. In our method, the step size is chosen by a simple back-tracking procedure: starting with $\alpha = 1$, if $f(X(\alpha)) < f(X)$, set the next iterate to be $X(\alpha)$ and stop; else, reduce α by half, and recurse.

Example 5: Consider the example presented in Figure 7. These are the results of the gradient method applied to the network of 60 sensors and 4 anchor nodes (shown as blue diamonds), a radio range of 0.30 and 20% noise, as discussed in Figure 2(b). The SDP estimated positions shown in Figure 2(b) are used as the initial solution. The SDP estimated positions are indicated by the blue crosses in 7(a). Figure 7(a) also shows the update trajectories in 50 iterations. The blue asterisks indicate the final positions of the sensors; the trajectories are indicated by blue lines. It can be observed clearly that most sensors are moving toward their actual locations marked by the black circles. The final positions after 50 gradient steps are plotted in Figure 7(b). which is a more accurate localization.



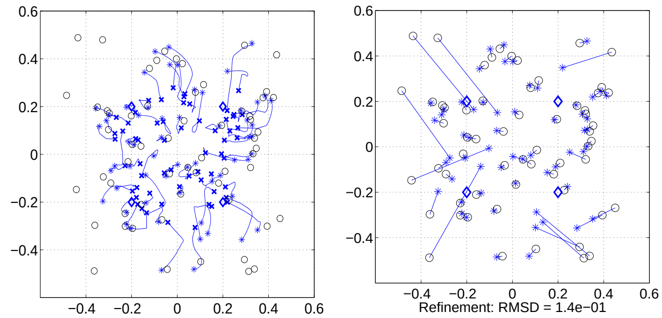
(a) Gradient search trajectories (b) Gradient result, 50 iterations



(c) The sum of error squares vs. number of gradient search steps (d) Result of gradient search with random initial starting point

Fig. 7. Refinement through gradient-descent method, for the example in Figure 2(b)

To demonstrate that the refined localization is indeed better, we can compute the objective function at every iteration to see if its value is reduced. In Figure 7(c) the objective function values vs number of gradient steps are plotted. One can see that in the first 15 steps the objective function value



(a) Gradient search trajectories (b) Gradient result, 50 iterations

Fig. 8. Refinement through gradient method, for case in Figure 2(a)

drops rapidly, and then the curve levels off. This demonstrates that the gradient search method does improve the overall localization result. A natural question is how good the new localization is. To answer this question we need a lower bound for the objective function value. One trivial lower bound of the objective function value is 0; but a tighter lower bound is given by the optimal objective value of the SDP relaxation of (18). In this case, the SDP objective value is about 0.376, plotted in Figure 7(c) in a dashed line, and the gradient search method produces an objective function value of about 0.462. Thus an error gap of 0.086 of suboptimality is obtained, which is about than 23% of the lower bound provided by the optimal SDP objective value.

The gradient descent method is a local optimization method that generally does not deliver the global optimal solution of a non-convex problem, unless a good starting iterate is available. The sensor network localization problem based on (18) is a non-convex optimization problem. Hence a pure gradient-descent method would not work. To see this, another experiment is performed. We use random starting points as the initial sensor locations and update them by the same rule (19). The updated trajectories are shown in Figure 7(d). Most of these sensors do not converge to their actual positions. Comparing it to the result in Figure 7(b), we can see that the use of the SDP solution as the initial localization makes all the difference. The effect of the initial position is also illustrated through the results of applying the gradient method to the case where the anchors are in the interior (as seen in Figure 2(a)). The results of the gradient method are shown in Figure 8. Clearly, due to bad initialization, some points are unable to converge close to their actual locations.

Some finer points need to be mentioned. First, since the regularized SDP is already quite accurate, it is a better starting point for the refinement and the gradient method offers some minor improvement. For the non regularized case, the initial estimate is highly erroneous to begin with and the gradient method can offer significant improvement but still may have high error. A combination of the regularized and gradient method yields very accurate estimates even in the presence of high noise. Secondly, because the distance measurements have noises, convergence to the true location should not be expected. Hence, although some of the trajectories do not converge to their corresponding true locations, this does not mean that the gradient method is not working. Actually, from the objective function value, we know that the updates work. Finally, we can also use an exact line-search method to choose the step size α to guarantee a drop in the objective value.

VI. Experimental Results

For the purpose of simulations, the scenario described in subsection III-C is used. A network of 60 uniform randomly distributed unknown points is generated in the square area $[-0.5, 0.5] \times [-0.5, 0.5]$. m anchors are also generated in the same manner. The distances between the nodes is calculated. If the distance between 2 nodes was less than the specified radio range R, the distance is included in the edge set for solving the SDP after adding a random error to it in the following manner.

$$d_{ij} = \hat{d}_{ij} \cdot |1 + N(0,1) * nf|,$$

where $\hat{d}_{ij}$ is the actual distance between the 2 nodes, nf (noise factor) is a given number between $[0, 1]$ that is used to control the amount of noise variance and $N(0, 1)$ is a standard normal random variable. The radio range R and noise factor nf are varied and the effects on the RMSD error as defined in (15) are observed. For each of the graphs shown below, experiments are performed on 30 independent configurations and the average is taken. Our program is implemented with MATLAB and it uses either SDPT3 [31], [32], SEDUMI [33] or DSDP [34] as the SDP solver.

A. *Effect of Varying Radio Range and Noise Factor*

Figure 9 shows the variation of average estimation error (normalized by the radio range R) when the noise in the distance measurements increases and with different radio ranges. 6 anchors are placed randomly in the network area.

The results for the SDP model (1) and the results from [17] where detailed results of different SDP models without regularization are discussed will serve as comparisons against which the effectiveness of the regularization technique can be gauged.

As can be seen from the results, for random anchor placement, the use of regularization provides about $30\%R$ improvement even in very high noise and low radio range. This is a significant improvement over the previous methods as far as estimation accuracy for random anchor placement is concerned. This frees the algorithm from the constraint that anchors must be placed at the perimeter for good estimation. For reasonably large radio ranges, the estimation error stays under control even in highly noisy scenarios. Even for 30% noise, the estimation error can be kept below $20\%R$ for $R \geq 0.3$. It can also be observed that the gradient method offers some minor improvement over the regularized SDP. In fact, it is because the regularized SDP provides an excellent starting point that the gradient method error is also lowered significantly as compared to results mentioned in [17].

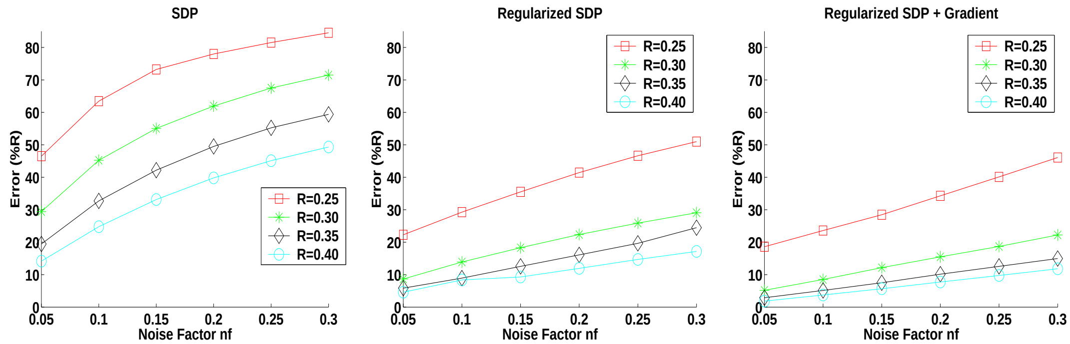


Fig. 9. Variation of estimation error (RMSD) with measurement noise, 6 anchors randomly placed.

B. Effect of Number of Anchors

Figure 10 shows the variation of estimation error by increasing the radio range while varying the number of randomly placed anchors from 4 to 8, and the nf was fixed at 0.2. For the same networks, Figure 11 shows the variation of estimation error by increasing the measurement noise (nf) while varying the number of anchors from 4 to 8, where the radio range is fixed at 0.3. We remind the reader that the number of anchors is denoted by m.

From these results, it becomes clear that beyond a certain number of anchors, the regularized SDP provides nearly the same accuracy irrespective of the number of anchors as opposed to the SDP model (3) which is more sensitive to the number of anchors. This is encouraging since the regularized SDP can therefore be used to provide better estimation accuracy with fewer anchors.

VII. Conclusion

In this article, we have presented robust techniques to solve the sensor network localization problem using semidefinite programming. The results show significant improvement in the performance of the SDP based algorithms in highly noisy scenarios by adding regularization terms followed by refinement using a gradient-descent method. Our future work concentrates on making the algorithms more efficient and robust. The links with tensegrity theory need to examined more closely in order to develop more effective regularization terms through intelligent edge selection. By choosing optimal weights within the objective function with regard to both the error term and the regularization term, the accuracy of the estimates provided by a single SDP solution can potentially be improved significantly.

As the size of the networks grows, solving one large SDP may become intractable. For this purpose, a distributed algorithm based on SDP has been described in [19], [20]. The ideas described in these paper are also being incorporated into a distributed formulation where the entire network is divided into multiple clusters which are solved in parallel and the solutions from the clusters are subsequently stitched together.

We have also demonstrated the use of SDP for localization using combinations of distance and angle constraints or pure angle constraints [21], [35] . This is particularly useful in multi-modal sensor networks where distance information may be collected by a variety of sensing mechanisms, such as cameras, ultrasound and RF. The extension of these techniques to provide robust noise handling for angle based localization is a part of our future research.

The general graph realization problem for which an algorithm was described in this paper can be extended to solve other distance geometry problems such as molecule structure determination [20] and ball packing as well.

References

[1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, Atlanta, GA, Sept. 2002. [Online]. Available: citeseer.ist.psu.edu/mainwaring02wireless.html

[2] E. Kaplan, *Understanding GPS: Principles and Applications*. Artech House, 1996.

[3] J. Beutel, "Geolocation in a picoradio environment," Master's thesis, ETH Zurich/UC Berkeley, 1999.

[4] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low cost outdoor localization for very small devices." Computer science department, University of Southern California, Tech. Rep., April 2000.

[5] L. Doherty, L. E. Ghaoui, and K. S. J. Pister, "Convex position estimation in wireless sensor networks." in *Proceedings of IEEE Infocom*, Anchorage, Alaska, April 2001, pp. 1655 –1663.

[6] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer*, vol. 34, no. 8, pp. 57–66, 2001.

[7] A. Howard, M. Matarić, and G. Sukhatme, "Relaxation on a mesh: a formalism for generalized localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Wailea, Hawaii, Oct 2001, pp. 1055–1060.

[8] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *GLOBECOM (1)*, 2001, pp. 2926–2931.

[9] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from connectivity in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, pp. 961–974, 2004.

[10] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Mobile Computing and Networking*, 2001, pp. 166–179. [Online]. Available: citeseer.ist.psu.edu/savvides01dynamic.html

[11] C. Savarese, J. M. Rabaey, and K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *Proceedings of the General Track: 2002 USENIX Annual Technical Conference*. USENIX Association, 2002, pp. 317–327.

[12] T. Eren, D. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur, "Rigidity, computation, and randomization in network localization," in *Proceedings of IEEE Infocom*, 2004, pp. 1655 –1663.

[13] B. Jackson and T. Jordan, "Connected rigidity matroids and unique realizations of graphs," 2003. [Online]. Available: citeseer.ifi.unizh.ch/jackson03connected.html

[14] J. Moré and Z. Wu, "Global continuation for distance geometry problems." *SIAM Journal on Optimization.*, vol. 7, pp. 814–836, 1997.

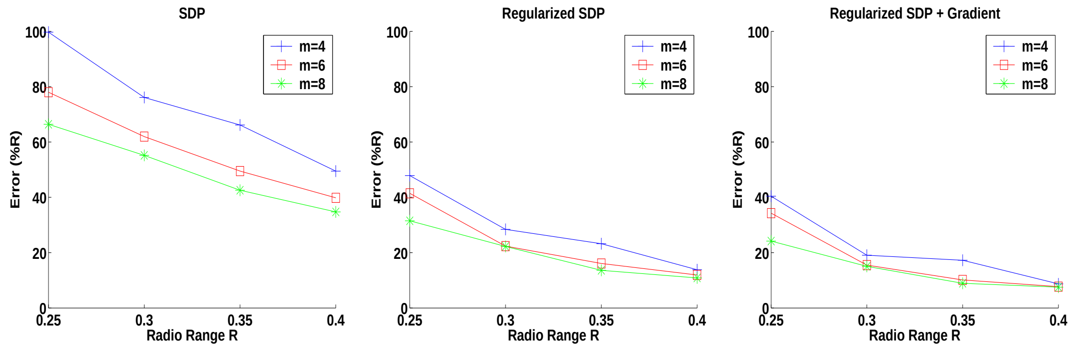


Fig. 10. Variation of estimation error (RMSD) with number of anchors (randomly placed) and varying radio range, $nf = 0.2$.

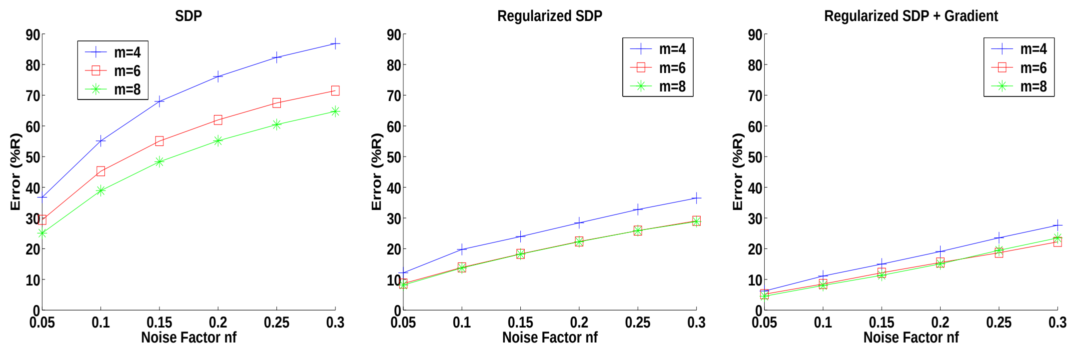


Fig. 11. Variation of estimation error (RMSD) with number of anchors (randomly placed) and varying measurement noise, $R = 0.3$.

[15] P. Tseng, "Second-order cone programming relaxation of sensor network localization, *submitted to SIAM Journal of Optimization*," August 2005.

[16] A. M.-C. So and Y. Ye, "Theory of semidefinite programming relaxation for sensor network localization," Dept of Management Science and Engineering, Stanford University, *in SODA'5 and to appear in Mathematical Programming*, Tech. Rep., April 2004.

[17] P. Biswas, T.-C. Liang, T.-C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," Dept of Management Science and Engineering, Stanford University, *submitted to ACM Transactions on Sensor Networks*, Tech. Rep., October 2005.

[18] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *Proceedings of the third international symposium on Information processing in sensor networks*. ACM Press, 2004, pp. 46–54.

[19] ——, "A distributed method for solving semidefinite programs arising from ad hoc wireless sensor network localization." Dept of Management Science and Engineering, Stanford University, *to appear in Multiscale Optimization Methods and Applications*, Tech. Rep., October 2003.

[20] P. Biswas, T.-C. Liang, K.-C. Toh, and Y. Ye, "An SDP based approach for anchor-free 3d graph realization," Dept of Management Science and Engineering, Stanford University, *submitted to SIAM Journal on Scientific Computing*, Tech. Rep., March 2005.

[21] P. Biswas, H. Aghajan, and Y. Ye, "Integration of angle of arrival information for multimodal sensor network localization using semidefinite programming," Wireless Sensor Networks Lab, Stanford University,, Tech. Rep., May 2005.

[22] T.-C. Liang, T.-C. Wang, and Y. Ye, "A gradient search method to round the semidefinite programming relaxation solution for ad hoc wireless sensor network localization," Dept of Management Science and Engineering, Stanford University, Tech. Rep., August 2004.

[23] A. Y. Alfakih, A. Khandani, and H. Wolkowicz, "Solving euclidean distance matrix completion problems via semidefinite programming," *Comput. Optim. Appl.*, vol. 12, no. 1-3, pp. 13–30, 1999.

[24] A. I. Barvinok, "Problems of distance geometry and convex properties of quadratic maps." *Discrete & Computational Geometry*, vol. 13, pp. 189–202, 1995.

[25] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. SIAM., 1994.

[26] M. Laurent, "Matrix completion problems," *The Encyclopedia of Optimization*, vol. 3, pp. 221–229, 2001. [Online]. Available: citeseer.ist.psu.edu/laurent01matrix.html

[27] K. Q. Weinberger, F. Sha, and L. K. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *ICML '04: Proceedings of the twenty-first international conference on Machine learning*. New York, NY, USA: ACM Press, 2004, p. 106.

[28] J. Costa, N. Patwari, and A. O. HeroIII, "Distributed multidimensional scaling with adaptive weighting for node localization in sensor networks," *to appear in ACM Trans. on Sensor Networks*, 2005.

[29] O. Güler and Y. Ye, "Convergence behavior of interior point algorithms," *Mathematical Programming*, vol. 60, pp. 215–228, 1993.

[30] A. M.-C. So and Y. Ye, "A semidefinite programming approach to tensegrity theory and realizability of graphs," Dept of Management Science and Engineering, Stanford University, *to appear in SODA'6*, Tech. Rep., July 2005.

[31] K. C. Toh, M. J. Todd, and R. Tutuncu, "SDPT3 — a Matlab software package for semidefinite programming," *Optimization Methods and Software*, vol. 11, pp. 545–581, 1999. [Online]. Available: citeseer.ist.psu.edu/article/toh98sdpt.html

[32] R. Tütüncü, K. Toh, and M. Todd, "Solving semidefinite-quadratic-linear programs using SDPT3," *Mathematical Programming, Series B*, vol. 95, pp. 189–217, 2003.

[33] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11 & 12, pp. 625–633, 1999.

[34] S. J. Benson, Y. Ye, and X. Zhang, "Solving large-scale sparse semidefinite programs for combinatorial optimization," *SIAM Journal on Optimization*, vol. 10(2), pp. 443–461, 2000. [Online]. Available: citeseer.ist.psu.edu/benson98solving.html

[35] P. Biswas, H. Aghajan, and Y. Ye, "Semidefinite programming algorithms for sensor network localization using angle of arrival information," in *to appear, in Proceedings of Thirty-Ninth Annual Asilomar Conference on Signals, Systems, and Computers*, October 2005.