Further Relaxations of the SDP Approach to Sensor Network Localization

Zizhuo Wang, Song Zheng*, Stephen Boyd†, and Yinyu Ye‡

September 27, 2006; Revised May 2, 2007

Abstract

Recently, a semi-definite programming (SDP) relaxation approach has been proposed to solve the sensor network localization problem. Although it achieves high accuracy in estimating sensor's locations, the speed of the SDP approach is not satisfactory for practical applications. In this paper we propose methods to further relax the SDP relaxation; more precisely, to relax the single semi-definite matrix cone into a set of small-size semidefinite matrix cones, which we call the smaller SDP (SSDP) approach. We present two such relaxations; and they are, although weaker than the original SDP relaxation, retaining the key theoretical property and tested to be both efficient and accurate in computation. The speed of the SSDP is even faster than that of other further weaker approaches. The SSDP approach may also pave a way to efficiently solve general SDP relaxations without sacrificing their solution quality.

Keywords: Sensor network localization, semi-definite programming, relaxation

1 Introduction

There has been an increase in the use of ad hoc wireless sensor networks for monitoring environmental information (temperature, sound levels, light etc) across an entire physical space, where the sensor network localization problem has received considerable attentions recently. Typical networks of this type consist of a large number of densely deployed sensor nodes which must gather local data and communicate with other nodes. The sensor data from these nodes are relevant only if we know what location they refer to. Therefore knowledge of the node positions becomes imperative. The use of a GPS system could be a very expensive or impossible solution to this requirement. This problem is also related to other practical distance geometry problems.

The mathematical model of the problem can be described as follows. There are n distinct sensor points in R^d whose locations are to be determined, and other m fixed points (called the anchor points) whose locations are known as $a_1, a_2, ...a_m$. The Euclidean distance d_{ij} between the ith and jth sensor points is known if $(i, j) \in N_x$, and the distance $\bar{d}_{ik}$ between the ith sensor

*Department of Mathematical Sciences, Tsinghua University, Beijing, China

†Department of Electrical Engineering, Stanford University, Stanford, CA 94305. E–mail: boyd@stanford.edu

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

and kth anchor points is known if $(i,k) \in N_a$. Usually, $N_x = \{(i,j) : \|x_i - x_j\| = d_{ij} \le rd\}$ and $N_a = \{(i,k) : \|x_i - a_k\| = \bar{d}_{ik} \le rd\}$, where rd is a fixed parameter called radio range. And the sensor network localization problem is to find vector $x_i \in R^d$ for all $i = 1, 2, ..., n$ such that

$$\begin{array}{ll} \|x_i - x_j\|^2 = d_{ij}^2 & \forall\ (i,j) \in N_x \\ \|x_i - a_k\|^2 = \bar{d}_{ik}^2 & \forall\ (i,k) \in N_a. \end{array}$$

Unfortunately, this problem is hard to solve in general even for $d = 1$; see, e.g., [5, 21].

For simplicity, we restrict $d = 2$ in this paper. Many relaxations have been developed to tackle this and other related problems; see, e.g., [1, 2, 3, 6, 7, 8, 10, 11, 12, 14, 16, 17, 20, 26, 29, 30, 31, 32, 34, 35, 37, 38]. Among them, work of [1, 2, 3, 6, 7, 17, 30, 38] used an Euclidean distance matrix-based approach, where no anchor was needed or used to compute the rest unknown portion of the distance matrix [19]; [26, 31] developed a global optimization approach; [20, 37] constructed a second-order cone relaxation; [32, 29] adapted the sum of squares (SOS) approach; [14] modeled a problem similar to the dual of the distance completion problem; and [8, 33] considered bounds on the solution rank of an SDP problem. Recently, a semi-definite programming (SDP) relaxation, see, e.g., [10, 11, 12, 16, 34, 35], which explicitly used anchors' positions as the first-order information, was applied to solving a class of sensor network localization problems. Their relaxation model can be represented by a standard SDP model

$$\begin{array}{lll} \text{(SDP)} & \text{minimize} & \mathbf{0} \cdot Z \\ & \text{subject to} & Z_{(1,2),(1,2)} = I_2, \\ & & (\mathbf{0}; e_i - e_j)(\mathbf{0}; e_i - e_j)^T \bullet Z = d_{ij}^2,\ \forall\ (i,j) \in N_x \\ & & (-a_k; e_i)(-a_k; e_i)^T \bullet Z = \bar{d}_{ik}^2, \quad \forall\ (i,k) \in N_a \\ & & Z \succeq 0. \end{array} \tag{1}$$

Here I_2 is the 2-dimensional identity matrix, $\mathbf{0}$ is a vector or matrix of all zeros, and e_i is the vector of all zeros except an 1 at the i–th position. If a solution $Z = \begin{pmatrix} I_2 & X \\ X^T & Y \end{pmatrix}$ to (1) is of rank 2, or equivalently, $Y = X^T X$, then $X = [x_1, ..., x_n] \in R^{2\times n}$ is a solution to the sensor network localization problem. Note that the SDP variable matrix has two parts: the first-order part X (positions) and the second-order part of Y (position inner products), which two parts will give valuable information on the estimation and confidence measure of the final localization solution.

However, with the size of the SDP problem increases, the dimension of the matrix cone increases simultaneously and the amount of unknown variables increases quadratically, no matter how sparse N_x and N_a might be. It is also known that the arithmetic operation complexity of the SDP is at least $O(n^3)$ to obtain an approximate solution. This complexity bound prevents solving large-size problems. Therefore, it would be very beneficial to further relax the full SDP problem by exploiting the sparsity of N_x and N_a at the relaxation modeling level.

Through out of the paper, R^d denotes the d-dimensional Euclidean space, S^n denotes the space of $n \times n$ symmetric matrices, ^{T} denotes transpose and $r(A)$ denotes the rank of A. For $A \in S^n$, A_{ij} denotes the (i,j) entry of A, and $A_{(i_1,...,i_k),(i_1,...,i_k)}$ denotes the principal submatrix from the rows and columns indexed by $i_1, ..., i_k$. And for $A, B \in S^n$, $A \succeq B$ means that $A - B$ is positive semidefinite.

2 Further Relaxations of the SDP model

We present two such relaxations. The first is a node-based relaxation:

$$
\begin{array}{lll}
\text{(NSDP)} & \text{minimize} & \mathbf{0} \cdot Z \\
& \text{subject to} & Z_{(1,2),(1,2)} = I_2, \\
& & (\mathbf{0}; e_i - e_j)(\mathbf{0}; e_i - e_j)^T \bullet Z = d_{ij}^2, \ \forall\ (i,j) \in N_x \\
& & (-a_k; e_i)(-a_k; e_i)^T \bullet Z = \bar{d}_{ik}^2, \qquad \forall\ (i,k) \in N_a \\
& & Z^i = Z_{(1,2,i,N_i),(1,2,i,N_i)} \succeq 0, \ \forall i,
\end{array}
\tag{2}
$$

where the sensor-i-connected point set

$$N_i = \{j : \ (i,j) \in N_x\}$$

Here, the single $(2+n)$-dimensional matrix cone is replaced by n smaller $3+|N_i|$-dimensional matrix cones, each of which is a principal submatrix of Z. We should mention that a similar idea was proposed in [22] for solving general SDP problems.

The second relaxation is an edge-based relaxation:

$$
\begin{array}{lll}
\text{(ESDP)} & \text{minimize} & \mathbf{0} \cdot Z \\
& \text{subject to} & Z_{(1,2),(1,2)} = I_2, \\
& & (\mathbf{0}; e_i - e_j)(\mathbf{0}; e_i - e_j)^T \bullet Z = d_{ij}^2, \ \forall\ (i,j) \in N_x \\
& & (-a_k; e_i)(-a_k; e_i)^T \bullet Z = \bar{d}_{ik}^2, \qquad \forall\ (i,k) \in N_a \\
& & Z_{(1,2,i,j),(1,2,i,j)} \succeq 0, \ \forall (i,j) \in N_x.
\end{array}
\tag{3}
$$

Here, the single $(2+n)$-dimensional matrix cone is replaced by $|N_x|$ smaller 4-dimensional matrix cones, each of which is a principal submatrix of Z too. If a solution $Z = \begin{pmatrix} I_2 & X \\ X^T & Y \end{pmatrix}$ to (3) satisfies that $r(Z_{(1,2,i,j),(1,2,i,j)})$ equals to 2 for all $(i,j) \in N_x$, then $X = [x_1, ..., x_n]$ is a localization for the localization problem. An edge-based decomposition was also used for the Sum-of-Square (SOS) approach to localization in [32].

In practice, the distances may have certain random measurement errors, so that the ESDP model can be adjusted to a suitable objective optimization problem. For example, if there is a random Laplacian noise added to each d_{ij}^2 and $\bar{d}_{ik}^2$, then we have

$$
\begin{array}{lll}
\text{(ESDPOP)} & \text{minimize} & \sum_{(i,j)\in N_x} |(0, e_i - e_j)(0, e_i - e_j)^T \cdot Z - d_{ij}^2| \\
& & + \sum_{(i,k)\in N_a} |(-a_k, e_i)(-a_k, e_i)^T \cdot Z - \bar{d}_{ik}^2| \\
& \text{subject to} & Z_{(1,2),(1,2)} = I_2, \\
& & Z_{(1,2,i,j),(1,2,i,j)} \succeq 0, \ \forall (i,j) \in N_x;
\end{array}
$$

and it can be written as an SDP minimization problem:

$$
\begin{array}{lll}
\text{(ESDPOP)} & \text{minimize} & \sum_{(i,j)\in N_x}(u_{ij}+v_{ij})+\sum_{(i,k)\in N_a}(u_{ik}+v_{ik}) \\
& \text{subject to} & Z_{(1,2),(1,2)}=I_2, \\
& & (\mathbf{0};e_i-e_j)(\mathbf{0};e_i-e_j)^T\bullet Z-u_{ij}+v_{ij}=d_{ij}^2, \quad \forall\ (i,j)\in N_x \\
& & (-a_k;e_i)(-a_k;e_i)^T\bullet Z-u_{ik}+v_{ik}=\bar{d}_{ik}^2, \quad \forall\ (i,k)\in N_a \\
& & Z_{(1,2,i,j),(1,2,i,j)}\succeq 0,\ u_{ij},v_{ij}\ge 0,\ \forall(i,j)\in N_x, \\
& & u_{ik},v_{ik}\ge 0,\ \forall(i,k)\in N_a.
\end{array}
\tag{4}
$$

Similarly, (NSDP) can be reformulated as

$$
\begin{array}{lll}
\text{(NSDPOP)} & \text{minimize} & \sum_{(i,j)\in N_x}(u_{ij}+v_{ij})+\sum_{(i,k)\in N_a}(u_{ik}+v_{ik}) \\
& \text{subject to} & Z_{(1,2),(1,2)}=I_2, \\
& & (\mathbf{0};e_i-e_j)(\mathbf{0};e_i-e_j)^T\bullet Z-u_{ij}+v_{ij}=d_{ij}^2, \quad \forall\ (i,j)\in N_x \\
& & (-a_k;e_i)(-a_k;e_i)^T\bullet Z-u_{ik}+v_{ik}=\bar{d}_{ik}^2, \quad \forall\ (i,k)\in N_a \\
& & Z^i=Z_{(1,2,i,N_i),(1,2,i,N_i)}\succeq 0,\ \forall i, \\
& & u_{ij},v_{ij}\ge 0,\ \forall(i,j)\in N_x,\ u_{ik},v_{ik}\ge 0,\ \forall(i,k)\in N_a.
\end{array}
\tag{5}
$$

For simplicity, we would focus on the feasibility models of (1), (2) and (3) in most rest discussions of the paper. Obviously, (2) is a relaxation of (1) and (3) is a relaxation of (2). The following proposition will formalize these relations.

Proposition 1. *If $Z^*_{SDP}=\begin{pmatrix} I_2 & X \\ X^T & Y \end{pmatrix}$ is a solution to (1), then Z^*_{SDP}, after removing the unspecified variables, is a solution to relaxation (2); if $Z^*_{NSDP}=\begin{pmatrix} I_2 & X \\ X^T & Y \end{pmatrix}$ is a solution to (2), then Z^*_{NSDP}, after removing the unspecified variables, is a solution to relaxation (3). Hence*

$$F^{SDP}\subset F^{NSDP}\subset F^{ESDP},$$

where $F^{\cdot}$ represents the solution set of the corresponding SDP relaxation.

We notice that (1) has $(n+2)^2$ variables and $|N_x|+|N_a|$ equality constraints, (2) has at most $4+2n+\sum_i|N_i|^2$ variables, and $|N_x|+|N_a|$ equality constraints, and (3) has $4+3n+|N_x|$ variables, and also $|N_x|+|N_a|$ equality constraints. Usually, $4+3n+|N_x|$ is much smaller than $(n+2)^2$, so that (3) has a much less number of variables than (1), hence the NSDP or ESDP relaxation has the potential to be solved much faster than (1), and our computational results will confirm this fact.

But how good is the NSDP or ESDP relaxation? How do these relaxations perform? In the rest of the paper, we will prove that, although they are weaker than the SDP relaxation, the NSDP and ESDP relaxation share some of the same desired theoretical properties possessed by the full SDP relaxation, including the trace criterion for accuracy. We develop a sufficient condition when NSDP coincides with SDP. We also show that the ESDP relaxation is stronger than the Second-Order Cone Programming (SOCP) relaxation. Furthermore, we will present computational results and compare our method with the full SDP, Sum of Squares (SOS), SOCP relaxation, and domain-decomposition methods; and demonstrate that our method is among the fastest ones and its localization quality is comparable or superior to that of other methods.

3 Theoretical Analyses of NSDP

We have the following basic assumption: G is the undirected graph of a sensor network that consists of all the sensors and anchors with edge sets N_x and N_a, then we require that G be connected and contain at least three anchors. Before we state our results, we recall three important concepts: d-uniquely localizable graph, chordal graph, and partial positive semi-definite matrix. The definition of the d-uniquely localizable graph is given by [34]:

Definition 1. *A sensor localization problem is d-uniquely localizable if there is a unique localization $\bar{X} \in \mathcal{R}^{d\times n}$ and there is no $x_i \in \mathcal{R}^h,\ i = 1, \ldots, n$, where $h > d$, such that:*

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

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 simply setting $x_i = (\bar{x}_i; \mathbf{0})$, where anchor points are augmented to $(a_k; \mathbf{0}) \in \mathcal{R}^h$).

d-unique localizability has been proved to be the necessary and sufficient condition for the SDP relaxation to compute a solution in R^d; see [34]. For the case of $d = 2$, if a graph is 2-uniquely localizable, then the SDP relaxation (1) produces a unique solution Z with rank 2, and $X = [x_1, ..., x_n] \in R^{2\times n}$ of Z is the unique localization of the localization problem in R^2.

Definition 2. *An undirected graph is a chordal graph if every cycle of length greater than three has a chord; see, e.g., [13].*

The chordal graph has been used for solving sparse SDP problems or reducing the number of high-order variables in SOS relaxations; see, e.g., [22, 18, 29].

The concept of partial positive semi-definite matrix can be found, e.g., in [24, 27, 28].

Definition 3. *A square matrix is called to be partial symmetric if it is symmetric to the extent of its specified entries, i.e., if the (i, j) entry of the matrix is specified, then so is the (j, i) entry and the two are equal. A partial semi-definite matrix is a partial symmetric matrix and every fully specified principal submatrix is positive semi-definite.*

The following results was proved in [24, 27]

Lemma 1. *Every partial positive semi-definite matrix with undirected graph G has positive semi-definite completion if and only if G is chordal.*

Although the NSDP model is weaker than the SDP relaxation in general, the following theorem implies that they are the same under the chordal condition.

Theorem 1. *Suppose the undirected graph of sensor nodes with edge set N_x is chordal, then*

$$F^{SDP} = F^{NSDP}.$$

Proof. We only need to prove that any solution to (2) can be completed to a solution of (1). Let $Z = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$ be a solution to (2). Then, all entries of Z are specified except those Y_{ij} such that $(i,j) \notin N_x$. The conic constraints of (2) indicate that every fully specified principal submatrix of Z is positive semi-definite, since it is a principal submatrix of Z^i in (2). Thus, Z is a partial semi-definite matrix.

We are also given that the undirected graph induced by Y in Z is chordal. We now prove that the undirected graph induced by Z is also chordal. Notice that the graph of Z has total $n+2$ nodes and every specified entry represents an edge. Let nodes $D1$ and $D2$ represent the first two rows (columns) of Z, respectively. Then each of the two nodes has edges to all other nodes in the graph. Now consider any cycle in the graph of Z. If the cycle constrains D1 or D2 or both, then it must have a chord since each of D1 and D2 connect to every other node; if the cycle contains neither D1 nor D2, then it still contains a chord since the graph of Y is chordal. Therefore, Z has a positive semi-definite completion, say $\bar{Z}$, from Lemma 1, and $\bar{Z}$ must be a solution to (1), since (2) and (1) share the same constraints involving only the specified entries. □

Under the condition of 2-unique localizability, we further have

Corollary 1. *If the sensor network is* 2*-uniquely localizable and the undirected graph of sensor nodes with edge set N_x is chordal, then the solution of (2) is a unique localization for the sensor network.*

4 Theoretical Analyses of ESDP

We now focus on our second relaxation, the ESDP relaxation of (3).

4.1 Relation between ESDP and SDP

In the SDP relaxation model, suppose that $Z_{SDP} = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$ is a solution to (1). It is shown that the individual traces or the diagonal entries of $Y - X^T X$ can indicate the accuracy of the corresponding sensor's location in [10, 34]. These individual traces were also given a confidence interpretation. We will show that the ESDP model retains the same desired property. The following theorem will show that if $Z_{ESDP} = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$ is a solution to (3), then the individual traces of $Y - X^T X$ also serve as the indicators of the accuracy of the corresponding sensor's location.

First, we introduce a lemma involving the rank of SDP solutions.

Lemma 2. *Consider the following semi-definite programming problem*

$$\begin{array}{ll} \min & \sum_i C_i \cdot X_i \\ s.t. & \sum_i A_{ij} \cdot X_i = b_j, \ \forall j \\ & X_i \succeq 0, \ \forall i. \end{array} \tag{6}$$

Then applying the path-following interior-point method will produce a max-rank (relative interior) solution for each X_i, i.e., if X^1 and X^2 are two different optimal solutions satisfying

$$r(X^1_{\bar{i}}) < r(X^2_{\bar{i}}) \text{ for at least one } \bar{i}.$$

Then solving (6) by applying path-following interior-point method will not yield solution X^1.

Proof. Problem (6) can be reformulated into

$$\begin{array}{ll} \min & \overline{C} \cdot X \\ \text{s.t.} & \overline{A}_j \cdot X = b_j, \ \forall j \\ & X = \begin{pmatrix} X_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & X_n \end{pmatrix} \succeq 0, \end{array}$$

where $\overline{C} = \text{diag}(C_i)_{i=1}^n$ and $\overline{A}_j = \text{diag}(A_{ij})_{i=1}^n$. This can be also written as

$$\begin{array}{ll} \min & \overline{C} \cdot X \\ \text{s.t.} & \overline{A_j} \cdot X = b_j, \ \forall j \\ & E_{ij} \cdot X = 0 \ \forall (i,j) \notin D \\ & X \succeq 0, \end{array}$$

where D denotes those positions that do not belong to any diagonal block of X.

Thus, the path-following algorithm will return a max-rank solution to the problem; see, e.g., [23, 25]. In other words, if X^* is a solution calculated by the path-following method, then $\sum_{i=1}^n r(X^*_i)$ is maximal among all solutions. Hence for every i, $r(X^*_i)$ must be maximal among all solutions to (6). This is because when we let

$$X(\alpha) = \alpha X^1 + (1-\alpha) X^2$$

for $0 < \alpha < 1$,

$$r(X(\alpha)_i) \geq \max\{r(X^1_i), r(X^2_i)\}, \ \forall i$$

and

$$r(X(\alpha)_{\bar{i}}) \geq \max\{r(X^1_{\bar{i}}), r(X^2_{\bar{i}})\} \geq r(X^2_{\bar{i}}) > r(X^1_{\bar{i}}).$$

Thus, X^1 cannot be a max-rank solution, and Lemma 2 follows. $\square$

Applying this lemma, we can get the following result which provides a justification for using the individual traces to evaluate the accuracy of computed sensor locations.

Theorem 2. *Let* $Z = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$ *be a max-rank solution of (3). If the diagonal entry or individual trace*

$$(Y - X^T X)_{\bar{i}\bar{i}} = 0 \tag{7}$$

then the $\bar{i}$*th column of* X*,* $x_{\bar{i}}$*, must be the true location of the* $\bar{i}$*th sensor, and* $x_{\bar{i}}$ *is invariant over all solutions* Z *for (3).*

Proof. Our proof is by contradiction. Without losing generality, we assume $(Y - X^T X)_{jj} > 0$ for all $j \neq \bar{i}$. For simplicity, we also denote the principal submatrix $Z_{(1,2,i,j),(1,2,i,j)}$ by $Z_{(1,2,i,j)}$.

Note that the constraints in (3) ensured that $Z_{(1,2,\bar{i},j)} \succeq 0$ for all $(\bar{i}, j) \in N_x$. Thus, $(Y - X^T X)_{\bar{i}\bar{i}} = 0$ implies $(Y - X^T X)_{\bar{i}j} = 0$ for all $(\bar{i}, j) \in N_x$, i.e. $Z_{(1,2,\bar{i},j)}$ has rank 3 for all $(\bar{i}, j) \in N_x$. Moreover, from Lemma 2, the max-rank of $Z_{(1,2,\bar{i},j)}$ is at most 3 for all solutions to (3).

Denote by $\bar{Z}$ a true localization for (3), that is, $\bar{Z}_{(1,2,i,j)}$ has rank 2 for all $(i, j) \in N_x$, where

$$\bar{Z}_{(1,2,i,j)} = \begin{pmatrix} I_2 & \bar{x}_i & \bar{x}_j \\ \bar{x}_i^T & \bar{Y}_{ii} & \bar{Y}_{ij} \\ \bar{x}_j^T & \bar{Y}_{ji} & \bar{Y}_{jj} \end{pmatrix} = \begin{pmatrix} I_2 & \bar{x}_i & \bar{x}_j \\ \bar{x}_i^T & \|\bar{x}_i\|^2 & \bar{x}_i^T \bar{x}_j \\ \bar{x}_j^T & \bar{x}_j^T \bar{x}_i & \|\bar{x}_j\|^2 \end{pmatrix}.$$

Suppose $\bar{x}_{\bar{i}} \neq x_{\bar{i}}$. Since the solution set is convex, then

$$Z^\alpha = \alpha \bar{Z} + (1 - \alpha) Z, \quad 0 \leq \alpha \leq 1,$$

is also a solution to (3). Taking α sufficiently small but strictly positive, we will get another solution Z^α which satisfies

$$r(Z^\alpha_{(1,2,i,j)}) \geq r(Z_{(1,2,i,j)}), \ \forall (i, j) \in N_x,$$

and the *strict inequality* holds for $i = \bar{i}$. This is because for $(\bar{i}, j) \in N_x$

$$\begin{aligned}
&Y^\alpha_{(\bar{i},j)} - [x^\alpha_{\bar{i}}, x^\alpha_j]^T [x^\alpha_{\bar{i}}, x^\alpha_j] \\
&= \alpha \bar{Y}_{(\bar{i},j)} + (1 - \alpha) Y_{(\bar{i},j)} - (\alpha [\bar{x}_{\bar{i}}, \bar{x}_j] + (1 - \alpha)[x_{\bar{i}}, x_j])^T (\alpha [\bar{x}_{\bar{i}}, \bar{x}_j] + (1 - \alpha)[x_{\bar{i}}, x_j]) \\
&= (1 - \alpha)(Y_{(\bar{i},j)} - [x_{\bar{i}}, x_j]^T [x_{\bar{i}}, x_j]) + \alpha(1 - \alpha)([x_{\bar{i}}, x_j] - [\bar{x}_{\bar{i}}, \bar{x}_j])^T ([x_{\bar{i}}, x_j] - [\bar{x}_{\bar{i}}, \bar{x}_j])
\end{aligned}$$

Since $(Y - X^T X)_{\bar{i}\bar{i}} = (Y - X^T X)_{\bar{i}j} = (Y - X^T X)_{j\bar{i}} = 0$,

$$Y_{(\bar{i},j)} - [x_{\bar{i}}, x_j]^T [x_{\bar{i}}, x_j] = \begin{pmatrix} 0 & 0 \\ 0 & \gamma \end{pmatrix}$$

for some $\gamma > 0$.

Also we are given that $\bar{x}_{\bar{i}} \neq x_{\bar{i}}$, so that $([x_{\bar{i}}, x_j] - [\bar{x}_{\bar{i}}, \bar{x}_j])^T ([x_{\bar{i}}, x_j] - [\bar{x}_{\bar{i}}, \bar{x}_j])$ is a positive semi-definite matrix whose first element is positive, which implies that

$$\det \left[(1 - \alpha) \begin{pmatrix} 0 & 0 \\ 0 & \gamma \end{pmatrix} + \alpha(1 - \alpha)([x_{\bar{i}}, x_j] - [\bar{x}_{\bar{i}}, \bar{x}_j])^T ([x_{\bar{i}}, x_j] - [\bar{x}_{\bar{i}}, \bar{x}_j]) \right] > 0.$$

That is, $Z^{\alpha}_{(1,2,\bar{i},j)}$ is a solution to (3) with rank 4, which is a contradiction.

Therefore, we proved that $\bar{x}_{\bar{i}}$ must be the true location of the $\bar{i}$th sensor and $\bar{x}_{\bar{i}}$ is invariant over all solutions to (3). □

Theorem 2 is related to Proposition 2 of [37]. Moreover, the desired invariance property of $x_{\bar{i}}$ extends to ESDPOP, which can be also seen from the proof in [37]. In summary, we have

Corollary 2. *Let* $Z = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$ *be a solution to (3) and condition (7) holds for all* i*. Then, the ESDP model (3) produces a unique solution for the sensor network in* R^2*.*

Next we will intensify Proposition 1 by the following theorem.

Theorem 3. *Let* $Z = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$ *be a solution to (3), and let* $\bar{Z} = \begin{pmatrix} I & \bar{X} \\ \bar{X}^T & \bar{Y} \end{pmatrix}$ *be any solution to (1); both are calculated by the path-following method. If condition (7) holds for* i*, then* $(\bar{Y} - \bar{X}^T\bar{X})_{ii} = 0$*.*

Proof. Our proof is again by contradiction. If (7) holds for Z but not for $\bar{Z}$, i.e., $(\bar{Y} - \bar{X}^T\bar{X})_{ii} > 0$. Since, for $0 \le \alpha \le 1$,

$$Z_\alpha = (1-\alpha)Z + \alpha\bar{Z}$$

is always a solution to (3). Taking α sufficiently small, we will get a solution with a higher rank than Z, and this fact contradicts with Lemma 2. □

Theorem 3 says that if the ESDP relaxation can accurately locate a certain sensor, then so can the SDP relaxation. This implies that the ESDP relaxation is weaker than SDP relaxation. We illustrate this fact using an example.

Example 1: Consider the following graph with 3 sensors and 3 anchors. The 3 anchors are located at $(-0.4, 0)$, $(0.4, 0)$, $(0, 0.4)$ and the 3 sensors are located at $(-0.05, 0.3)$, $(-0.08, 0.2)$, $(0.2, 0.3)$, respectively. We set the radio range to be 0.50 (see Figure 1)

In Figure 1 (and throughout this paper), we use diamonds to represent the anchor positions. In Figure 1 we use solid line to connect two points (sensors and/or anchors) when their Euclidean distance is smaller than the radio range, and the length of such a line segment is known to relaxation models.

First, we use SDP relaxation (1) to solve this sensor localization problem; and the result is accurate (see Figure 2(a)). In Figure 2(a) (and throughout this paper), a circle denotes the true location of a sensor (whose position is not known to the SDP model) and a star denotes the location of a sensor computed by the SDP method. Every true sensor location and its computed corresponding position are connected by a solid line. If we use the quantity of Root Mean Square Deviance to measure the deviance of the computed result:

$$RMSD = \left(\frac{1}{n} \sum_{i=1}^{n} \|x_i - \bar{x}_i\|_2^2 \right)^{\frac{1}{2}} \tag{8}$$

where x_i is the position vector of sensor i computed by the algorithm and $\bar{x}_i$ is its true position vector, then the RMSD of the SDP localization is about $1e-7$. The NSDP model (2) returns the exactly same localization of the SDP from Theorem 1, since N_x is a chordal graph.

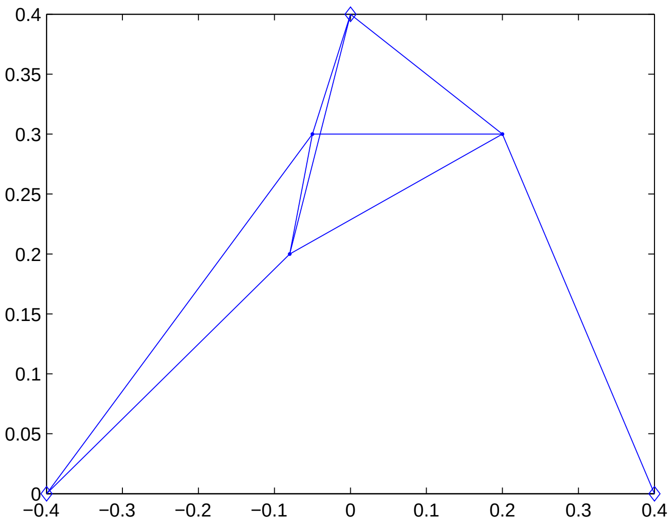
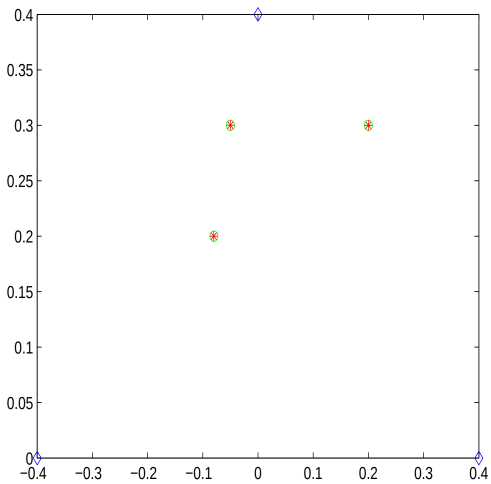
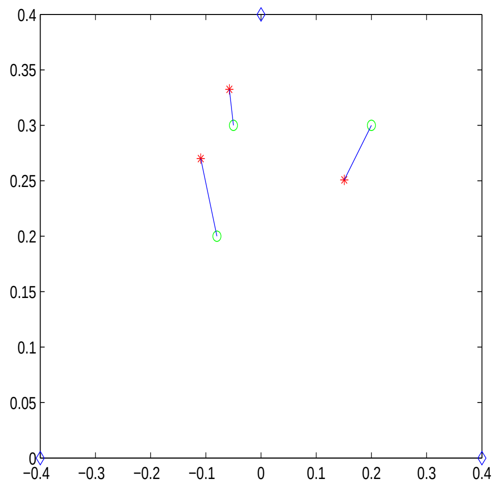


Figure 1: The locations of sensors and anchors and connection edges in Example 1



(a) Graphical localization result of the SDP model



(b) Graphical localization result of the ESDP model

Figure 2: Comparision of graphical localization results generated by the SDP and ESDP on Example 1

Next we use the ESDP model (3) to solve the problem, and this time, the result is inaccurate with RMSD at 0.048 (see Figure 2(b)).

Now we illustrate why this error happened. In SDP model (1), the solution matrix Z^* is required to be positive semi-definite. If we write

$$Z^*_{SDP} = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix},$$

then the matrix $Y - X^T X$ is required to be positive semi-definite. But in model (3),

$$Z^*_{ESDP} = \begin{pmatrix} I & \bar{X} \\ \bar{X}^T & \bar{Y} \end{pmatrix},$$

where we just require that each 2×2 principal submatrix of $\bar{Y} - \bar{X}^T \bar{X}$ be positive semi-definite. This does not imply that the entire matrix is positive semi-definite. In fact, the solution calculated by the ESDP model (3) is

$$Z^*_{ESDP} = \begin{pmatrix} 1 & 0 & -0.07278 & -0.13467 & 0.14884 \\ 0 & 1 & 0.32778 & 0.25467 & 0.24884 \\ -0.07278 & 0.32778 & 0.11072 & 0.09498 & 0.06865 \\ -0.13467 & 0.25467 & 0.09498 & 0.09014 & 0.04540 \\ 0.14884 & 0.24884 & 0.06865 & 0.04540 & 0.08907 \end{pmatrix}.$$

It can be verified that Z^*_{ESDP} satisfies all constraint in (3) as well as in (1), and each 2×2 principal matrix of $\bar{Y} - \bar{X}^T \bar{X}$ is positive semi-definite. But the three eigenvalues of $\bar{Y} - \bar{X}^T \bar{X}$ are $(-0.00048, 0.0048, 0.0091)$, so that the entire matrix of $\bar{Y} - \bar{X}^T \bar{X}$ is indefinite and this is the cause of the difference between the two relaxations.

4.2 Relation between ESDP and SOCP

A Second-Order Cone Programming (SOCP) relaxation for the sensor network localization problem have been proposed; see, e.g., [20, 37].

$$\begin{aligned} (SOCP) \quad \min \quad & \sum_{(i,j)\in N_x} (u_{ij} + v_{ij}) + \sum_{(i,k)\in N_a} (u_{ik} + v_{ik}) \\ \text{s.t.} \quad & x_i - x_j - w_{ij} = 0 \;\; \forall (i,j) \in N_x, x_i - a_k - w_{ik} = 0 \;\; \forall (i,k) \in N_a \\ & y_{ij} - u_{ij} + v_{ij} = d_{ij}^2 \;\; \forall (i,j) \in N_x, y_{ik} - u_{ik} + v_{ik} = \bar{d}_{ik}^2 \;\; \forall (i,k) \in N_a \\ & u_{ij} \geq 0, v_{ij} \geq 0, (y_{ij} + \tfrac{1}{4}, y_{ij} - \tfrac{1}{4}, w_{ij}) \in SOC, \;\; \forall (i,j) \in N_x \\ & u_{ik} \geq 0, v_{ik} \geq 0, (y_{ik} + \tfrac{1}{4}, y_{ik} - \tfrac{1}{4}, w_{ik}) \in SOC, \;\; \forall (i,k) \in N_a \end{aligned} \tag{9}$$

The SOCP relaxation can be also viewed as a further relaxation of SDP relaxation, and it was proved to be faster than the SDP method and to be served as a useful preprocessor of the actual problem. Here in this section, we will show that the ESDP model is stronger than the SOCP relaxation. Our proof refers to Proposition 1 of [37].

Theorem 4. *If* $Z = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$ *is an optimal solution to (4), then*

$$x_i = ith\ column\ of\ X \quad i = 1, 2, ..., n$$

$$y_{ij} = \begin{cases} Y_{ii} + Y_{jj} - 2Y_{ij} & if (i,j) \in N_x \\ \|a_k^2\| - 2a_k^T x_i + Y_{ii} & if (i,k) \in N_a \end{cases}$$

form a feasible solution of (9) with the same objective value.

Proof. Since Z is a feasible solution to (4), we have $Z_{(1,2,i,j),(1,2,i,j)} \succeq 0$ for all $(i,j) \in N_x$. So that for each $(i,j) \in N_x$, we have

$$\begin{pmatrix} Y_{ii} - \|x_i^2\| & Y_{ij} - x_i^T x_j \\ Y_{ij} - x_i^T x_j & Y_{jj} - \|x_j^2\| \end{pmatrix} \succeq 0$$

This implies that $Y_{ii} - \|x_i^2\| \geq 0$, $Y_{jj} - \|x_j^2\| \geq 0$ and $(Y_{ii} - \|x_i^2\|)(Y_{jj} - \|x_j^2\|) \geq (Y_{ij} - x_i^T x_j)^2$

Hence $(Y_{ii} - \|x_i^2\| + Y_{jj} - \|x_j^2\|)^2 \geq 4(Y_{ij} - x_i^T x_j)^2$, i.e.

$$Y_{ii} + Y_{jj} - 2Y_{ij} \geq \|x_i^2\| + \|x_j^2\| - 2x_i^T x_j$$

and the theorem follows. □

Corollary 3. *If* x_i *is invariant over all the solutions of (9), then it is also invariant over all the ESDP solutions. That is, if SOCP relaxation can return the true location for a certain sensor, so can ESDP relaxation.*

The above theorem and corollary indicate that one can always derive the same SOCP relaxation solution from an ESDP relaxation solution, that is, the solution set of the ESDP relaxation is smaller than that of the SOCP relaxation. Thus, the ESDP relaxation is stronger than the SOCP relaxation. And the following example shows that the reverse is not true.

Example 2: Consider the following problem with 3 anchors and 2 sensors. The true location of 3 anchors are $a_1 = (-0.4, 0), a_2 = (0, 0.5)$, and $a_3 = (0.4, 0)$, and the true location of the 2 sensors are $x_1 = (0, -0.3)$ and $x_2 = (0.4, 0.2)$ with radio range 0.7 (see Figure 3).

Since there are only two sensors, the ESDP relaxation is the same with the full SDP relaxation, and it is known that this graph is strongly localizable (see [34]), so we know that the ESDP relaxation will give the *unique* solution Z where X is the accurate positions of the sensors. However, for SOCP relaxation, since the graph is 2-realizable, its optimal value of (9) is 0 so that the optimal solution must satisfy $y_{ij} = d_{ij}^2$ and $y_{ik} = \bar{d}_{ik}^2$. Thus, any $(\bar{x}_1, \bar{x}_2)$ that satisfies

$$\begin{array}{ll} \|\bar{x}_1 - \bar{x}_2\|^2 & \leq 0.4^2 + 0.5^2 = 0.41 \\ \|\bar{x}_1 - a_1\|^2 & \leq 0.3^2 + 0.4^2 = 0.25 \\ \|\bar{x}_1 - a_3\|^2 & \leq 0.3^2 + 0.4^2 = 0.25 \\ \|\bar{x}_2 - a_2\|^2 & \leq 0.4^2 + 0.3^2 = 0.25 \\ \|\bar{x}_2 - a_3\|^2 & \leq 0^2 + 0.2^2 = 0.04 \end{array}$$

must be also optimal to (9).

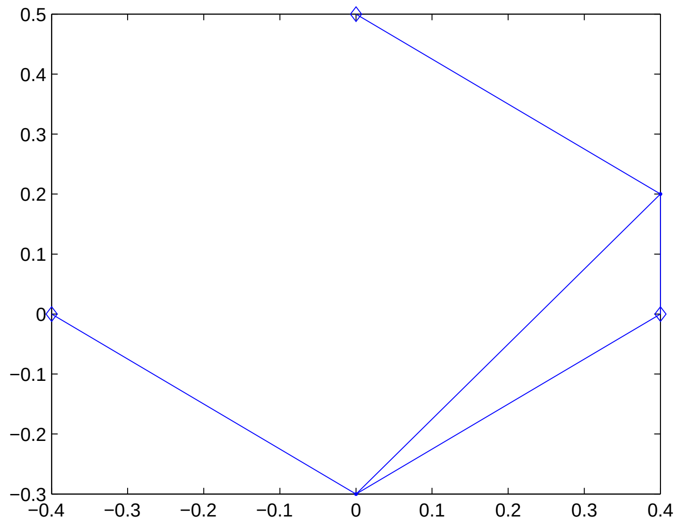


Figure 3: The locations of sensors and anchors and connection edges in Example 2

Now let $\bar{x}_1 = (0,0) \neq x_1$ and $\bar{x}_2 = (0.3, 0.15) \neq x_2$. Then, it is easy to verify that the above inequalities hold, so that $(\bar{x}_1, \bar{x}_2)$ is also an optimal solution to (9). But we know that the interior-point method would always maximize the potential function (see [25, 23])

$$P(x,y) = \sum_{(i,j)\in N_x} \log\left(y_{ij} - \|x_i - x_j\|^2\right) + \sum_{(i,k)\in N_a} \log\left(y_{ik} - \|x_i - a_k\|^2\right)$$

in the optimal solution set; and it is obvious that $P(\bar{x}_1, \bar{x}_2) > P(x_1, x_2)$. Therefore the SOCP relaxation model (9) will not give the true solution x_1 and x_2 and, thereby, the ESDP relaxation is *strictly* stronger than SOCP relaxation for this example.

4.3 The dual problem of ESDP

For a conic programming problem, it is always very important to consider the dual problem. In many cases, the dual problem can present us many important information about the primal problem as well as many useful applications. Here we will present the dual problem of (3) and list some basic properties between the primal and dual.

Consider a general conic programming problem:

$$\begin{array}{ll} \min & C \cdot X \\ \text{s.t.} & A_j \cdot X = b_j, \ \forall j \\ & X_{N_i,N_i} \succeq 0, \ \forall i, \end{array} \tag{10}$$

where $X \in S^n$ and N_i is an index subset of $\{1, 2, ..., n\}$. Then, the dual to the problem is

$$\begin{array}{rl} \max & \sum_j b_j y_j \\ \text{s.t.} & \sum_j y_j A_j + \sum_i S^i = C \\ & \\ & S^i_{N_i,N_i} \succeq 0, \text{ and } S^i_{kj} = 0 \ \forall k \notin N_i \text{ or } j \notin N_i;\ \forall i. \end{array} \tag{11}$$

In other words, S^i is an S^n matrix and its entries are zero outside the principal submatrix of S_{N_i,N_i}.

For the ESDP model (3), the dual problem is

$$\begin{array}{rl} \max & \sum_{(i,j)\in N_x} \omega_{ij} d_{ij}^2 + \sum_{(i,k)\in N_a} \omega_{ik} \bar{d}_{ik}^2 + u_{11} + 2u_{12} + u_{22} \\ \text{s.t.} & \sum_{(i,j)\in N_x} \omega_{ij} (\mathbf{0}; e_i - e_j)^T (\mathbf{0}; e_i - e_j) + \sum_{(i,k)\in N_a} \omega_{ik} (-a_k; e_i)^T (-a_k; e_i) + \\ & \begin{pmatrix} u_{11} + u_{12} & u_{12} & \mathbf{0} \\ u_{12} & u_{22} + u_{12} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} + \sum_{(i,j)\in N_x} S^{(i,j)} = 0 \\ & S^{(i,j)}_{(1,2,i,j),(1,2,i,j)} \succeq 0, \text{ and } S^{(i,j)}_{kl} = 0 \ \forall k \notin \{i,j\} \text{ or } \ l \notin \{i,j\};\ \forall (i,j) \in N_x. \end{array} \tag{12}$$

We have the following complementarity result

Proposition 2. *Let Z be a solution to (3) and $\{S^{(i,j)}\}$ be an optimal solution to the dual, then*

$$S^{(i,j)}_{(1,2,i,j),(1,2,i,j)} \cdot Z_{(1,2,i,j),(1,2,i,j)} = 0, \ \forall (i,j) \in N_x.$$

In particular, if $r\left(S^{(i,j)}_{(1,2,i,j),(1,2,i,j)}\right)$ is 2 for all $(i,j) \in N_x$, then $r\left(Z_{(1,2,i,j),(1,2,i,j)}\right)$ is 2 for all $(i,j) \in N_x$ so that (3) produces a unique localization for the sensor network in R^2.

The beauty of duality is that we can solve the dual problem and simultaneously yield a primal solution from the complementarity proposition. We will demonstrate in the next section that the solution speed of solving the dual is about twice as fast as solving the primal problem, which was originally observed in [9].

5 Computational Results and Comparison to Other Approaches

Now we address the question: Will the improvement in the speed of ESDP relaxation compensate the loss in relaxation quality? In this section, we first present some computational results of the ESDP relaxation model. And then we compare the model with different kinds of approaches, including the SDP approach (1) of [10], the SOCP approach [37], the SOS approach [32], and the domain-decomposition approach of [12, 16].

5.1 Computational results of the ESDP relaxation

In our numerical simulation, we follow [10]. We randomly generate the true positions of all points in a square of 1 by 1, then randomly select a proportion of 2%-10% of the total points to be anchors, then compute all of the edge-length, d_{ij}. We select only those edges whose edge-length is less than a given radio range rd, and add a multiplicative random noise to every selected edge-length

$$d_{ij} = d_{ij}(1 + nf \cdot randn(1)),$$

as the distance input data to the SDP models. Here nf is a specified noisy factor and $randn(1)$ is a standard Gaussian random variable. There may be still many points within the radio range of a sensor or anchor. Thus, in order to maintain the sparsity of the graph, we set a limit on the amount of selected edges connected to each sensor or anchor, i.e., set a upper bound on the degree of each node of the graph.

In our computational experiments we use n to denote the number of points (including both sensors and anchors), m to denote the number of anchors. We limit the node degree to 7 for each sensor or anchor, and the edges are *randomly* chosen for each node. We also implement the steepest-descent local search refinement proposed in [11, 12] for solving noisy problems. All test problems are solved by SeDuMi 1.05 [36] of Matlab7.0 on a DELL D420 laptop with 1.99GB Memory and 1.06GHz CPU.

The first set of test problems has noisy factor $nf = 0.1$ throughout. Table 1 contains a computational comparison of ESDP to the original SDP relaxation [10]. Here three models, the original full SDP model (up to 400 points), the ESDP model, and the dual of the ESDP model itself, are all solved by SeDuMi 1.05. In order to see efficiency of the ESDP model, the solution time (in seconds) in Table 1 includes only the SeDuMi solver time, that is, the data input/preparation time is excluded.

As we can see, while the full SDP solution time increases cubically in size, the SDP solver times of both ESDP and Dual ESDP increase little faster than *linearity.* While this speed up was remarkable, how about the localization quality? Figure 4 shows two graphical results generated by Full SDP and Dual ESDP on solving a smaller problem, where one can barely see much difference. Here diamonds represent the anchor positions, circles represent sensor's true positions, and stars represent the computed sensor positions. (The codes and few test problems have been placed on the public site http://www.stanford.edu/˜yyye/. We welcome the reader to test them and draw their own conclusions.)

Next we compare our approach to the Sum of Squares (SOS) approach, the Second-Order Cone Programming (SOCP) approach, and the domain-decomposition approach. We will use the same examples in their corresponding papers.

5.2 Computational comparison with the SOS method

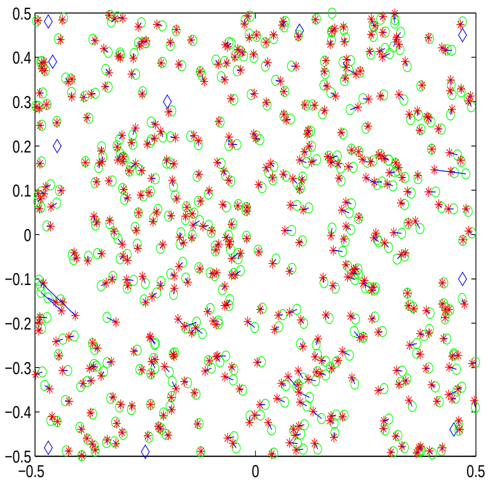
Another possible approach for sensor network localization problem is called the Sum of Squares (SOS) relaxation method. The SOS method is also an SDP relaxation which applies to solving the following problem

$$\min\ f(x) := \textstyle\sum_{(i,j)\in N_x} (\|x_i - x_j\|_2^2 - d_{ij}^2)^2 + \sum_{(i,k)\in N_a} (\|x_i - a_k\|_2^2 - \bar{d}_{ik}^2)^2 \tag{13}$$

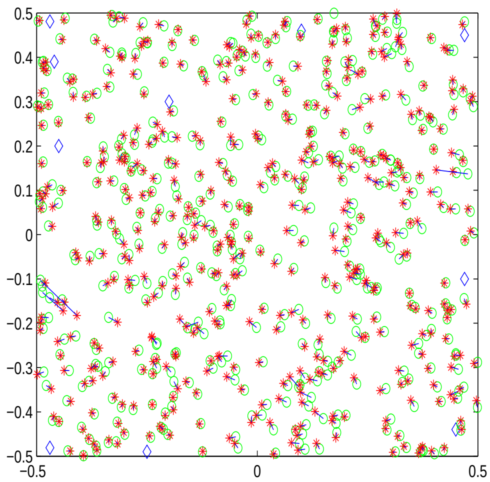
where the objective function is a polynomial.

Noisy Problem #	n	m	rd	Full SDP time	ESDP time	Dual ESDP time
1	50	5	0.35	1.33	1.5	1.22
2	100	5	0.3	4.94	3.22	1.91
3	200	5	0.25	35.21	7.64	4.19
4	400	10	0.2	358.8	18.2	8.98
5	800	20	0.12	*	44.67	18.58
6	1600	40	0.07	*	120.58	43.91
7	3200	80	0.04	*	287.39	104.36
8	5000	100	0.03	*	426.85	192.08
9	6400	160	0.025	*	603.16	250.97

Table 1: Noisy test problems and the SDP solution time comparison



(a) Graphic result of the full SDP model



(b) Graphic result of the dual ESDP model

Figure 4: Comparision of graphical localization results generated by the Full SDP and Dual ESDP on a 10% noisy problem

Recent study [32] has shown that by exploiting the sparsity in SOS relaxation one can get faster computing speed than the SDP relaxation (1) and sometimes as well as higher accuracy. The author demonstrated that this structure can help save the computation time significantly. In [32], the author used the model of 500 sensors, 4 anchors with a radio range of 0.3 and no noises in distance measurements.

The author of [32] reported that it took totally about 1 hour and 25 minutes on a 0.98 GB RAM and 1.46GHz CPU computer to get a result with RMSD=$2.9e-6$. However, with the same parameters, our approach needs only 30 seconds to get the result with RMSD=$1e-6$. Thus, the ESDP approach is much faster than SOS approach in this case and the solution quality is comparable to that of the SOS method; see Figure 5.

5.3 Computational comparison with the SOCP method

The SOCP model performs best with a large fraction of anchors and a low noise. Thus, we test (primal) ESDP on the same set of problems reported in [37] where $m = 0.1n$ (10% of points are anchors) and $nf \leq 0.01$ (less than 1% noise), and the results are shown in Table 2. There are two methods proposed in [37]; one directly uses Matlab SeDuMi to solve the SOCP relaxation model and the other uses a smoothing coordinate gradient descent (SCGD) method coded in FORTRAN 77. The SCGD method is highly parallelizable, similar to distributed methods of [12, 16].

From Table 2, we see that the ESDP approach is much faster than the SOCP approach when both using Matlab SeDuMi, and it is slower than the tailored and FORTRAN coded SCGD method. On the other hand, the localization quality (see RMSD in Table 3) of ESDP is much better than that reported in [37] for both SeDuMi of SOCP and SCGD of SOCP. Figure 6 shows the graphical result of Test Problem 2 (900 sensors, 100 anchors, $nf = 0.001$ and $rd = 0.06$), where the localization of ESDP is quite accurate compared with the graphical result on the same problem reported in [37].

In Table 2, "ESDP time" denotes the *total* solution running time, including Matlab data preparation and SeDuMi input set-up time. Comparing Tables 1 and 2, one can see that, for ESDP, the Matlab data input and SeDuMi set-up time is considerable. This is because Matlab is notoriously slow on matrix loops and data inputs. This problem should go away when the algorithm is coded in C or FORTRAN 77.

Test Problem #	n	nf	rd	ESDP time	SeDuMi of SOCP	SCGD of SOCP
1	1000	0	0.06	59.60sec	3.6min	0.2min
2	1000	0.001	0.06	57.55sec	3.2min	0.4min
3	1000	0.01	0.06	53.60sec	3.9min	1.6min
4	4000	0	0.035	653.7sec	202.5min	1.6min
5	4000	0.001	0.035	668.3sec	193.8min	5.1min
6	4000	0.01	0.035	615.9sec	196.3min	6.2min

Table 2: ESDP times are taken on DELL D420 (1.99GB and 1.06GHz), and SOCP times are reported from [37] on a HP DL360 (1G Memory and 3GHz)

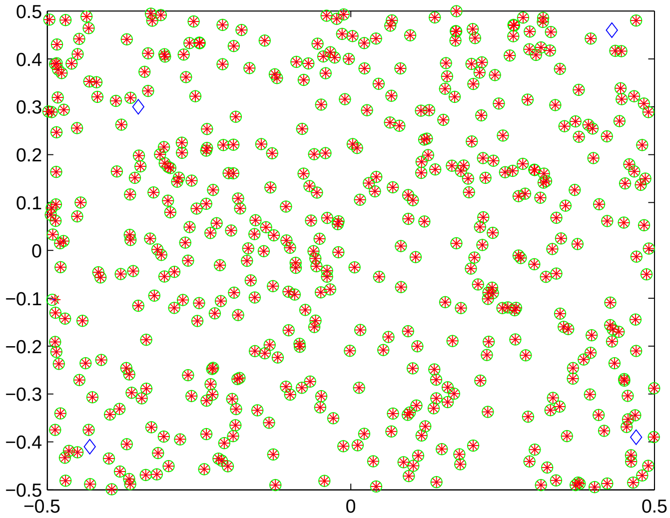


Figure 5: Graphical localization result of the ESDP model on the problem of Nie [32], 500 sensors, 4 anchors, $rd = 0.3$, $nf = 0$, RMSD=$1e - 6$

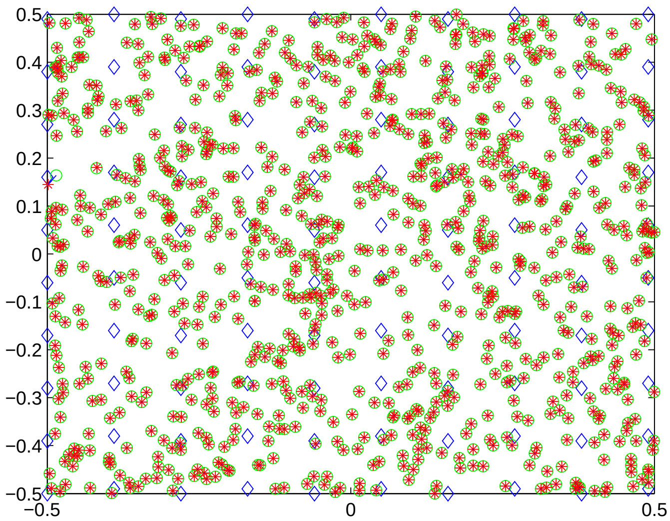


Figure 6: Graphical localization result of the ESDP model on Test Problem 2 in Table 2

Test Problem #	n	nf	rd	SeDuMi SDP dim	CPUtime	obj	RMSD
1	1000	0	0.06	20321×29195	59.60	3e-3	2e-3
2	1000	0.001	0.06	20321×29195	57.55	5e-4	3e-3
3	1000	0.01	0.06	20321×29195	53.66	4e-2	2e-2
4	4000	0	0.035	93727×133285	653.7	3e-3	1e-3
5	4000	0.001	0.035	93727×133285	668.3	7e-3	8e-4
6	4000	0.01	0.035	93727×133285	615.9	2e-2	3e-2

Table 3: Input parameters for the test problems, the corresponding ESDP dimensions, and ESDP computational results

Table 3 contains more detailed statistical results on this test, where "SeDuMi SDP dim" represents problem dimensions solved by SeDuMi, "CPUtime" denotes the total ESDP solution time in seconds (including Matlab data preparation and SeDuMi input set-up time), "obj" denotes the SDP objective value, and RMSD is the localization quality defined in (8).

5.4 Computational comparison with the decomposition method

There are other earlier approaches to speed up the SDP solution time. The domain-decomposition method of [12] and SpaceLoc of [16], are both based on breaking the localization problem into many geographically partitioned and smaller-sized localization problems, since each smaller SDP problem can be solved much faster and more accurate. Thus, they work quite well when many anchors are uniformly distributed in the region so that one is able to partition the network into many smaller domains; and, as a result, each of them contains enough anchors and forms its own individual localization problem. However, when the quantity of anchors is small or most of them are located on the boundary, such as problems in Table 1, these approaches would fail at the beginning, or they are reduced to solving nearly full-size SDP problems, simply because these problems are not computationally decomposable or distributable.

In contrast, our new approach does not depend on the quantity and location of anchors, since it is designed to improve the efficiency of solving one clustered individual SDP problem. In fact, any improvement on solving an individual SDP problem would complement the domain decomposition approaches, since it would be possible to handle much larger-sized subproblems.

6 Future Directions

From the computational results, we can see that the SSDP approaches indeed have a great potential to save the computation time in solving sensor network localization problems and the efficiency of the model is considerable. At the same time, they retain some of most important theoretical features of the original SDP relaxation and achieve high localization quality.

There are many directions for future research. First, although our ESDP relaxation performs very well, we still lack some powerful theorems to illustrate why the model works so well. This is a major issue that needed to be answered. Second, since, in our ESDP model, the decision matrix has its special structure, applying a tailored interior point method (like SCGD for SOCP approach) may save more computational times. We also see that the NSDP relaxation has its own merit, both in theory and in practice. Therefore, further research about the NSDP model is also worth perusing. In fact, we have experimented the NSDP model for solving the Max-Cut problem and would discuss its behavior and performance in another report. Finally, we plan to investigate the applicability of the SSDP relaxation idea for solving general SDP problems.

References

[1] A. Y. Alfakih. Graph Rigidity via Euclidean Distance Matrices. *Linear Algebra and Its Applications* 310:149–165, 2000.

[2] A. Y. Alfakih. On Rigidity and Realizability of Weighted Graphs. *Linear Algebra and Its Applications* 325:57–70, 2001.

[3] A. Y. Alfakih, A. Khandani, H. Wolkowicz. Solving Euclidean Distance Matrix Completion Problems Via Semidefinite Programming. *Comput. Opt. and Appl.* 12:13–30, 1999.

[4] Farid Alizadeh. Interior Point Methods in Semidefinite Programming with Applications to Combinatorial Optimization. *SIAM J. Opt.* 5:13–51, 1995.

[5] James Aspnes, David Goldenberg, Yang Richard Yang. On the Computational Complexity of Sensor Network Localization. *ALGOSENSORS 2004*, in *LNCS* 3121:32–44, 2004.

[6] M. Bădoiu. Approximation Algorithm for Embedding Metrics into a Two–Dimensional Space. *Proc. 14th SODA* 434-443, 2003.

[7] M. Bădoiu, E. D. Demaine, M. T. Hajiaghayi, P. Indyk. Low–Dimensional Embedding with Extra Information. *Proc. 20th SoCG* 320–329, 2004.

[8] A. Barvinok. Problems of Distance Geometry and Convex Properties of Quadratic Maps. *Disc. Comp. Geom.* 13:189–202, 1995.

[9] 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.

[10] P. Biswas, Y. Ye. Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization. *Proc. 3rd IPSN* 46–54, 2004.

[11] P. Biswas, T.-C. Liang, K.-C. Toh, and Y. Ye. Semidefinite Programming Approaches for Sensor Network Localization with Noisy Distance Measurements. Tech. Rep., Dept of Management Science and Engineering, Stanford University, to appear in *IEEE Transactions on Automation Science and Engineering*.

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

[13] J. Blair and B. Peyton. An introduction to chordal graphs and clique trees. In J. George, J. Gilbert, and J. Liu, editors, *Graph Theory and Sparse Matrix Computations*, pages 1-30, Springer Verlag, 1993.

[14] S. Boyd, P. Diaconis and L. Xiao. Fastest mixing Markov chain on a graph. *SIAM Review*, vol. 46, no. 4, pp. 667-689, 2004.

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

[16] M. Carter and H. H. Jin and M. A. Saunders and Y. Ye. Spaseloc: An Adaptable Subproblem Algorithm for Scalable Wireless Network Localization. Working Paper, submitted to *SIAM J. on Optimization*, January, 2005.

[17] J. Costa, N. Patwari, and A. O. Hero III. Distributed multidimensional scaling with adaptive weighting for node localization in sensor networks. To appear in *ACM Trans. on Sensor Networks*.

[18] J. Dahl, V. Roychowdhury, and L. Vandenberghe. Maximum Likelihood Estimation of Gaussian Graphical Models. Working Paper, Electrical Engineering Department, UCLA, in HOPT'06, Delt, 2006.

[19] Y. Ding, N. Krislock, J. Qian, and H. Wolkowicz Sensor Network Localization, Euclidean Distance Matrix. Completions, and Graph Realization. Research Report CORR 2006-23, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada.

[20] L. Doherty, L. E. Ghaoui, S. J. Pister. Convex Position Estimation in Wireless Sensor Networks. *Proc. 20th INFOCOM* Volume 3:1655–1663, 2001.

[21] T. Eren, D. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Moore, B. D. O. Anderson, P. N. Belhumeur. Rigidity, Computation, and Randomization in Network Localization. *Proc. 23rd INFOCOM*, 2004.

[22] M. Fukuda, M. Kojima, K. Murota, K. Nakata. Exploiting Sparsity in Semidefinite Programming via Matrix Completion I: General Framework. *SIAM J. Opt.* 11: 647-674, 2000.

[23] D. Goldfarb, K. Scheinberg. Interior Point Trajectories in Semidefinite Programming. *SIAM J. Opt.* 8(4):871–886, 1998.

[24] B. Grone, C. R. Johnson, E. M. Sa, and H. Wolkowicz. Positive Definite Completions of Partial Hermitian Matrices. *Linear Algebra and its Applications* 58 (1984) 109-124.

[25] O. Güler and Y. Ye. Convergence behavior of interior point algorithms. *Math. Programming* 60:215–228, 1993.

[26] B. Hendrickson. The Molecule Problem: Exploiting Structure in Global Optimization. *SIAM J. Opt.* 5(4):835–857, 1995.

[27] L. Hogben. Graph theoretic methods for matrix completion problems. *Linear Algebra and its Applications* 328 (2001) 161-202.

[28] C. R. Johnson and R. L. Smith. The Positive definite completion problem relative to a subspace. *Linear Algebra and its Applications* 307 (2000) 1-14

[29] M. Kojima, S. Kim and H. Waki. Sparsity in Sums of Squares Polynomials. *Mathematical Programming* 103(1):45-62, 2006.

[30] M. Laurent. Matrix completion problems. *The Encyclopedia of Optimization*, vol. 3, pp. 221–229, 2001.

[31] Jorge Moré and Zhijun Wu. Global continuation for distance geometry problems. *SIAM Journal on Optimization.*, 7:814–836, 1997.

[32] J. Nie. Sum of Squares Method for Sensor Network Localization. Working Paper, submitted to *SIAM Journal of Optimization*, August 2006.

[33] Gábor Pataki. On the Rank of Extreme Matrices in Semidefinite Programs and the Multiplicity of Optimal Eigenvalues. Mathematics of Operations Research 23(2):339–358, 1998.

[34] A. M.-C. So and Y. Ye. Theory of semidefinite programming relaxation for sensor network localization. In SODA'5 and to appear in *Mathematical Programming.*

[35] A. M.-C. So and Y. Ye. A semidefinite programming approach to tensegrity theory and realizability of graphs. Working Paper, Dept of Management Science and Engineering, Stanford University, July 2005, appeared in in SODA'6.

[36] 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.

[37] P. Tseng. Second-order cone programming relaxation of sensor network localization. Working Paper, to appear in *SIAM Journal of Optimization*, August 2005.

[38] 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*, ACM Press, 2004, p. 106.