SIAM J. OPTIM.
Vol. 19, No. 2, pp. 655–673

FURTHER RELAXATIONS OF THE SEMIDEFINITE PROGRAMMING APPROACH TO SENSOR NETWORK LOCALIZATION*

ZIZHUO WANG†, SONG ZHENG‡, YINYU YE§, AND STEPHEN BOYD¶

Abstract. Recently, a semidefinite programming (SDP) relaxation approach has been proposed to solve the sensor network localization problem. Although it achieves high accuracy in estimating the sensor 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 semidefinite matrix cone into a set of small-size semidefinite submatrix cones, which we call a sub-SDP (SSDP) approach. We present two such relaxations. Although they are weaker than the original SDP relaxation, they retain the key theoretical property, and numerical experiments show that they are both efficient and accurate. The speed of the SSDP is even faster than that of other approaches based on weaker relaxations. The SSDP approach may also pave a way to efficiently solving general SDP problems without sacrificing the solution quality.

Key words. sensor network localization, semidefinite programming, second-order cone programming, principal submatrix, chordal graph

AMS subject classifications. 90C22, 49M20, 65K05

DOI. 10.1137/060669395

1. Introduction.

There has been an increase in the use of ad hoc wireless sensor networks for monitoring environmental information (e.g., temperature, sound levels, and light) across an entire physical space, where the sensor network localization problem has received considerable attention recently. Typical networks of this type consist of a large number of densely deployed sensor nodes which gather local data and communicate with other nearby nodes. The sensor data from these nodes are relevant only if we know to what location they refer. Therefore knowledge of the node positions becomes imperative. The use of a GPS system could be a very expensive or otherwise impossible approach 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 m other fixed points (called the anchor points), whose locations $a_1, a_2, \ldots, a_m$ are known. 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 and kth anchor points is known if $(i,k) \in N_a$. Usually, $N_x = \{(i,j) : \|x_i - x_j\| = d_{ij} \le r_d\}$ and $N_a = \{(i,k) : \|x_i - a_k\| = \bar{d}_{ik} \le r_d\}$, where r_d is a fixed parameter called the radio range. The sensor network

*Received by the editors September 8, 2006; accepted for publication (in revised form) February 13, 2008; published electronically July 2, 2008. Part of this research was done when the first two authors were students and the third author was a visiting professor at the Department of Mathematical Sciences, Tsinghua University, Beijing, China.

http://www.siam.org/journals/siopt/19-2/66939.html

†Department of Management Science and Engineering, Stanford University, Stanford, CA 94305 (zzwang@stanford.edu).

‡Department of Industrial Engineering and Logistics Management, Hong Kong University of Science and Technology, Hong Kong SAR (zhengs@ust.hk).

§Department of Management Science and Engineering and, by courtesy, Electrical Engineering, Stanford University, Stanford, CA 94305 (yinyu-ye@stanford.edu).

¶Department of Electrical Engineering, Stanford University, Stanford, CA 94305 (boyd@stanford.edu).

localization problem is to find $x_i \in R^d$, $i = 1, 2, \ldots, n$, for which

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

Unfortunately, this problem is hard to solve in general even for $d = 1$; see, e.g., [15, 35].

For simplicity, we restrict ourselves to $d = 2$ in this paper. Many relaxations have been developed to tackle this and other related problems; see, e.g., [1, 4, 3, 5, 23, 6, 7, 28, 29, 33, 25, 16, 21, 12, 18, 20, 22, 26, 2, 31, 30, 19]. Among them, the work of [1, 4, 3, 5, 23, 16, 20, 19] used a Euclidean distance matrix-based approach, where no anchor was needed or used to compute the unknown portions of the distance matrix [36]; [12, 22] developed a global optimization approach; [21, 30] constructed a second-order cone relaxation; [26, 18] adapted the sum-of-squares (SOS) approach; [33] modeled a problem similar to the dual of the distance completion problem; and [6, 27] considered bounds on the solution rank of a semidefinite programming (SDP) problem. Recently, an SDP relaxation (see, e.g., [7, 28, 29, 25, 2, 31]) which explicitly used the 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

$$\begin{array}{ll} \text{minimize} & \mathbf{0} \bullet Z \\ \text{subject to} & Z_{(1,2)} = I, \\ & (\mathbf{0}; e_i - e_j)(\mathbf{0}; e_i - e_j)^T \bullet Z = d_{ij}^2 \quad \forall\ (i,j) \in N_x, \\ & (-a_k; e_i)(-a_k; e_i)^T \bullet Z = \bar{d}_{ik}^2 \quad \forall\ (i,k) \in N_a, \\ & Z \succeq 0. \end{array} \tag{1.1}$$

Here I is the 2-dimensional identity matrix and $Z_{(1,2)}$ is the upper-left 2×2 principal submatrix of Z, $\mathbf{0}$ is a vector or matrix of all zeros, and e_i is the vector of all zeros, except for a one in the ith position. If a solution

$$Z = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$$

to (1.1) is of rank 2, or, equivalently, $Y = X^T X$, then $X = [x_1, \ldots, x_n] \in R^{2\times n}$ 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). Both parts give valuable information about the estimation and confidence measure of the final localization solution.

As the size of the SDP problem increases, the dimension of the matrix cone increases and the number of 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.

Throughout the paper, R^d denotes d-dimensional Euclidean space, S^n denotes the space of $n \times n$ symmetric matrices, and $\mathrm{Rank}(A)$ denotes the rank of A. For $A \in S^n$, A_{ij} denotes the (i,j) entry of A, and $A_{(i_1,\ldots,i_k)}$ denotes the principal submatrix from the rows and columns indexed by $i_1, \ldots, i_k$. For $A, B \in S^n$, $A \succeq B$ means that $A - B$ is positive semidefinite, and $A \bullet B$ denotes the inner product, i.e., $A \bullet B = \mathrm{Tr}(AB)$.

2. Further relaxations of the SDP model. We will give two such relaxations. The first is a node-based relaxation, which we call the NSDP relaxation:

$$\begin{array}{ll} \text{minimize} & \mathbf{0} \bullet Z \\ \text{subject to} & Z_{(1,2)} = I, \\ & (\mathbf{0}; e_i - e_j)(\mathbf{0}; e_i - e_j)^T \bullet Z = d_{ij}^2 \quad \forall\ (i,j) \in N_x, \\ & (-a_k; e_i)(-a_k; e_i)^T \bullet Z = \bar{d}_{ik}^2 \quad \forall\ (i,k) \in N_a, \\ & Z^i = Z_{(1,2,i,N_i)} \succeq 0 \quad \forall i, \end{array} \tag{2.1}$$

where $N_i = \{j : \ (i,j) \in N_x\}$ is the sensor-i-connected point set. 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 [24] for solving general SDP problems.

The second relaxation is an edge-based relaxation, which we call the ESDP relaxation:

$$\begin{array}{ll} \text{minimize} & \mathbf{0} \bullet Z \\ \text{subject to} & Z_{(1,2)} = I, \\ & (\mathbf{0}; e_i - e_j)(\mathbf{0}; e_i - e_j)^T \bullet Z = d_{ij}^2 \quad \forall\ (i,j) \in N_x, \\ & (-a_k; e_i)(-a_k; e_i)^T \bullet Z = \bar{d}_{ik}^2 \quad \forall\ (i,k) \in N_a, \\ & Z_{(1,2,i,j)} \succeq 0 \quad \forall (i,j) \in N_x. \end{array} \tag{2.2}$$

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. If a solution

$$Z = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$$

to (2.2) satisfies Rank$(Z_{(1,2,i,j)} = 2$ for all $(i,j) \in N_x$, then $X = [x_1, \ldots, x_n]$ is a localization for the localization problem. An edge-based decomposition was also used for the SOS approach to localization in [26].

In practice, the distances may be corrupted by random measurement errors. In this case the ESDP model can be adjusted by forming a suitable objective. For example, if there is a random Laplacian noise added to each d_{ij}^2 and $\bar{d}_{ik}^2$, then we solve

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

which can be written as an SDP:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{(i,j)\in N_x} (u_{ij} + v_{ij}) + \sum_{(i,k)\in N_a} (u_{ik} + v_{ik}) \\ \text{subject to} & Z_{(1,2)} = I, \\ & (\mathbf{0}; e_i - e_j)(\mathbf{0}; e_i - e_j)^T \bullet Z - u_{ij} + v_{ij} = d_{ij}^2 \quad \forall\ (i,j) \in N_x, \\ & (-a_k; e_i)(-a_k; e_i)^T \bullet Z - u_{ik} + v_{ik} = \bar{d}_{ik}^2 \quad \forall\ (i,k) \in N_a, \\ & Z_{(1,2,i,j)} \succeq 0, \quad u_{ij}, v_{ij} \geq 0 \quad \forall (i,j) \in N_x, \\ & u_{ik}, v_{ik} \geq 0 \quad \forall (i,k) \in N_a. \end{array} \tag{2.3}$$

Similarly, NSDP can be reformulated as

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

For simplicity, we focus on the feasibility models of (1.1), (2.1), and (2.2) in the rest of this paper.

Obviously, (2.1) is a relaxation of (1.1), and (2.2) is a relaxation of (2.1). The following proposition will formalize these relations.

PROPOSITION 2.1. *If*

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

is a solution to (1.1)*, then* Z_{SDP}^**, after removing the unspecified variables, is a solution to relaxation* (2.1)*; if*

$$Z_{NSDP}^*=\begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$$

is a solution to (2.1)*, then* Z_{NSDP}^**, after removing the unspecified variables, is a solution to relaxation* (2.2)*. Hence*

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

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

We notice that (1.1) has $(n+2)^2$ variables and $|N_x|+|N_a|$ equality constraints, (2.1) has at most $4+2n+\sum_i |N_i|^2$ variables and $|N_x|+|N_a|$ equality constraints, and (2.2) 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 (2.2) has a much smaller number of variables than (1.1); hence, the NSDP or ESDP relaxation has the potential to be solved much faster than (1.1). 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 relaxations 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, SOS, SOCP relaxation, and domain-decomposition methods. One will see that our method is among the fastest methods, and its localization quality is comparable or superior to that of other methods.

3. Theoretical analyses of NSDP. We make the following basic assumption: G, the undirected graph of a sensor network consisting of all sensors and anchors, with edge sets N_x and N_a, is connected and contains at least three anchors. Before we present our results, we recall three basic concepts: the d-uniquely localizable graph, the chordal graph, and the partial positive semidefinite matrix.

The definition of a d-uniquely localizable graph is given by [2].

DEFINITION 3.1. *A sensor localization problem is d-uniquely localizable if there is a unique localization $\bar{X} \in R^{d \times n}$ and there is no $x_i \in R^h,\ i = 1, \ldots, n$, 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 nontrivial localization in some higher-dimensional space 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 $(a_k; \mathbf{0}) \in R^h$).

The condition of a 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 [2]. For the case of $d = 2$, if a graph is 2-uniquely localizable, then the SDP relaxation (1.1) produces a unique solution Z with rank 2, and $X = [x_1, \ldots, x_n] \in R^{2 \times n}$ of Z is the unique localization of a localization problem in R^2.

DEFINITION 3.2. *An undirected graph is a chordal graph if every cycle of length greater than three has a chord; see, e.g.,* [8].

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., [24, 17, 18].

DEFINITION 3.3. *A square matrix, possibly containing some unspecified entries, is called partial symmetric if whenever the (i,j) entry of the matrix is specified, then so is the (j,i) entry, and the two are equal. A partial semidefinite matrix is a partial symmetric matrix for which every fully specified principal submatrix is positive semidefinite.*

The concept of a partial positive semidefinite matrix can be found, e.g., in [9, 13, 14].

The following result was proved in [9, 13].

LEMMA 3.4. *Every partial positive semidefinite matrix with undirected graph G has positive semidefinite 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 equivalent under the chordal condition.

THEOREM 3.5. *Let the undirected graph of sensor nodes with edge set N_x be chordal. Then*

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

Proof. We need only to prove that any solution to (2.1) can be completed to a solution of (1.1). Let

$$Z = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$$

be a solution to (2.1). Then all entries of Z are specified except those Y_{ij} such that $(i,j) \notin N_x$. The conic constraints of (2.1) indicate that every fully specified principal submatrix of Z is positive semidefinite, since it is a principal submatrix of Z^i in (2.1). Thus, Z is a partial semidefinite 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 a total of $n+2$ nodes, and every specified entry represents an edge. Let nodes D_1 and D_2 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 contains D_1 or D_2 or both, then it must have a chord since each of D_1 and D_2 connect to every other node; if the cycle contains neither D_1 nor D_2, then it still contains a chord since the graph of Y is chordal. Therefore, Z has a positive semidefinite completion, say, $\bar{Z}$, from Lemma 3.4, and $\bar{Z}$ must be a solution to (1.1), since (2.1) and (1.1) share the same constraints involving only the specified entries. ∎

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

COROLLARY 3.6. *If a sensor network is* 2*-uniquely localizable and its undirected graph of sensor nodes with edge set* N_x *is chordal, then the solution of* (2.1) *is a unique localization for the sensor network.*

4. Theoretical analyses of ESDP.

We now focus on our second relaxation, the ESDP relaxation of (2.2).

4.1. Relation between ESDP and SDP.

In the SDP relaxation model, let

$$Z_{SDP} = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$$

be a solution to (1.1). Then it is shown that the individual traces or the diagonal entries of $Y - X^T X$ represent confidence measures in the accuracy of the corresponding sensor's location; see [7, 2]. We will show that the ESDP model retains this very desired property. More precisely, if

$$Z_{ESDP} = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$$

is a solution to (2.2), then the individual traces of $Y - X^T X$ also represent confidence measures in the accuracy of the corresponding sensor's location.

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

LEMMA 4.1. *Consider the following SDP:*

$$\begin{array}{ll} \text{minimize} & \sum_i C_i \bullet X_i \\ \text{subject to} & \sum_i A_{ij} \bullet X_i = b_j \quad \forall j, \\ & X_i \succeq 0 \quad \forall i. \end{array} \tag{4.1}$$

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*

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

Then solving (4.1) *by applying the path-following interior-point method will not yield solution* X^1.

Proof. Problem (4.1) can be reformulated into

$$\begin{array}{ll} \text{minimize} & \overline{C} \bullet X \\ \text{subject to} & \overline{A}_j \bullet X = b_j \quad \forall j, \\ & X = \begin{pmatrix} X_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & X_n \end{pmatrix} \succeq 0, \end{array}$$

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

$$\begin{array}{ll} \text{minimize} & \overline{C} \bullet X \\ \text{subject to} & \overline{A_j} \bullet X = b_j \ \forall j, \\ & E_{ij} \bullet X = 0 \ \forall (i,j) \notin D, \\ & X \succeq 0, \end{array}$$

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., [10, 11]. In other words, if X^* is a solution calculated by the path-following method, then $\sum_{i=1}^n \operatorname{Rank}(X_i^*)$ is maximal among all solutions; hence, for every i, $\operatorname{Rank}(X_i^*)$ must be maximal among all solutions to (4.1). Thus, X^1 cannot be a solution generated by the interior-point method. □

By applying this lemma, we have the following result which provides a justification for using the individual traces to measure the accuracy of computed sensor locations.

THEOREM 4.2. *Let*

$$Z = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$$

be a max-rank solution of (2.2). *If the diagonal entry or individual trace*

$$(Y - X^T X)_{\bar{i}\bar{i}} = 0, \tag{4.2}$$

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* (2.2).

Proof. Our proof is by contradiction. Without losing generality, we assume that $(Y - X^T X)_{jj} > 0$ for all $j \neq \bar{i}$.

Note that the constraints in (2.2) 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 that $(Y - X^T X)_{\bar{i}j} = 0$ 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 4.1, the max-rank of $Z_{(1,2,\bar{i},j)}$ is at most 3 for all solutions to (2.2).

Denote by $\bar{Z}$ a true localization for (2.2), that is, $\bar{Z}_{(1,2,i,j)}$ has rank 2 for all $(i, j) \in N_x$, where

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

Suppose that $\bar{x}_{\bar{i}} \neq x_{\bar{i}}$. 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 (2.2). By taking α sufficiently small but strictly positive, we will get another solution Z^α which satisfies

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

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 semidefinite 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 (2.2) 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 (2.2). ∎

Theorem 4.2 is related to Proposition 2 of [30]. Moreover, the desired invariance property of $x_{\bar{i}}$ extends to the case with noises, which can also be seen from the proof in [30]. In summary, we have the following.

COROLLARY 4.3. *Let*

$$Z = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$$

be a solution to (2.2) *and condition* (4.2) *hold for all* i. *Then the ESDP model* (2.2) *produces a unique solution for the sensor network in* R^2.

Next we enhance Proposition 2.1 by the following theorem.

THEOREM 4.4. *Let*

$$Z = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$$

be a solution to (2.2), *and let*

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

be any solution to (1.1); *both are calculated by the path-following method. If condition* (4.2) *holds for* Z, *so it does for* $\bar{Z}$.

Proof. Our proof is again by contradiction. If (4.2) holds for Z but not for $\bar{Z}$, e.g., $(\bar{Y} - \bar{X}^T \bar{X})_{ii} > 0$. Since, for $0 \leq \alpha \leq 1$,

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

is always a solution to (2.2), by taking α sufficiently small, we will get a solution with a higher rank than Z, and this fact contradicts Lemma 4.1. ∎

Theorem 4.4 says that if the ESDP relaxation can accurately locate a certain sensor, so can the SDP relaxation. This implies that the ESDP relaxation is weaker than the SDP relaxation. We illustrate this by using an example.

Example 1. Consider the following graph with 3 sensors and 3 anchors. The 3 anchors are located at $(-0.4, 0)$, $(0.4, 0)$, and $(0, 0.4)$, and the 3 sensors are located at $(-0.05, 0.3)$, $(-0.08, 0.2)$, and $(0.2, 0.3)$, respectively. We set the radio range to be 0.50; see Figure 4.1.

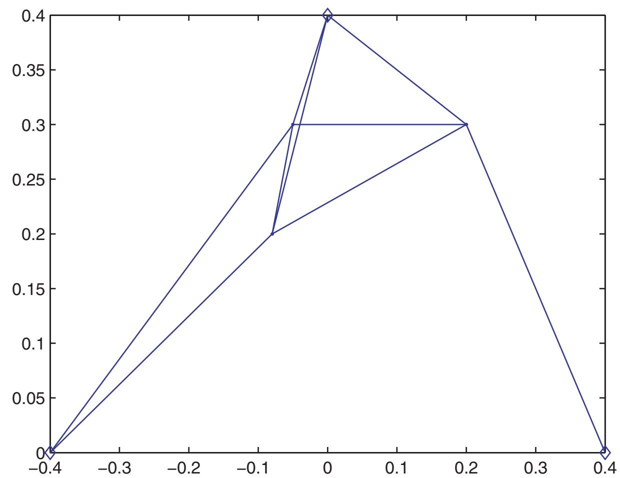


FIG. 4.1. *The locations of sensors, anchors, and connection edges in Example* 1.

In Figure 4.1 (and throughout this paper), we use diamonds to represent the anchor positions. We use a solid line to connect two points (sensors and/or anchors) when their Euclidean distance is smaller than the radio range, so that the length of the line segment is known.

First, we use full SDP relaxation (1.1) to solve this sensor localization problem, where the result is accurate (see Figure 4.2(a)). In Figure 4.2(a) (and throughout this paper), a circle denotes the true location of a sensor (they are not known to the SDP models), and a star denotes the location of a sensor computed by the SDP model. If we use the quantity of the root mean square deviance (RMSD) to measure the deviance of the computed result:

$$RMSD = \left(\frac{1}{n} \sum_{i=1}^{n} \|x_i - \bar{x}_i\|_2^2 \right)^{\frac{1}{2}}, \tag{4.3}$$

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 full SDP localization is about $1e-7$. Note that the NSDP model (2.1) returns the exactly same localization of the full SDP from Theorem 3.5, since N_x is a chordal graph.

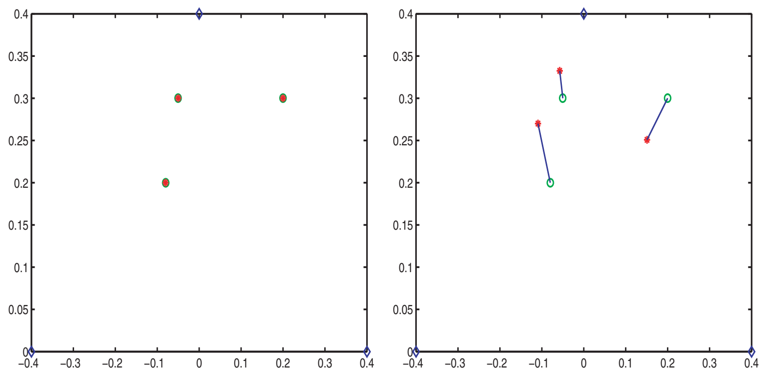
Next we use the ESDP model (2.2) to solve the problem, and this time the result is inaccurate with the RMSD at 0.048; see Figure 4.2(b), where every true sensor location and its computed corresponding position are connected by a solid line.

Now we illustrate why this error happened. In SDP model (1.1), the solution matrix Z^* is required to be positive semidefinite. 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 semidefinite. But in model (2.2),

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



(a) Graphical localization result of the SDP model

(b) Graphical localization result of the ESDP model

FIG. 4.2. *Comparision of graphical localization results generated by the SDP and ESDP in Example* 1.

where we just require that each 2×2 principal submatrix of $\bar{Y} - \bar{X}^T\bar{X}$ be positive semidefinite. This does not imply that the entire matrix is positive semidefinite. In fact, the solution calculated by the ESDP model (2.2) 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 constraints in (2.2) as well as in (1.1), and each 2×2 principal matrix of $\bar{Y} - \bar{X}^T\bar{X}$ is positive semidefinite. 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 SOCP relaxation for the sensor network localization problem has been proposed (see, e.g., [21, 30]):

$$\begin{aligned} &\text{minimize} && \sum_{(i,j)\in N_x} (u_{ij} + v_{ij}) + \sum_{(i,k)\in N_a} (u_{ik} + v_{ik}) \\ &\text{subject to} && x_i - x_j - w_{ij} = 0 \quad \forall (i,j) \in N_x, \quad x_i - a_k - w_{ik} = 0 \quad \forall (i,k) \in N_a, \\ &(4.4) && y_{ij} - u_{ij} + v_{ij} = d_{ij}^2 \ \ \forall (i,j) \in N_x, y_{ik} - u_{ik} + v_{ik} = \bar{d}_{ik}^2 \ \ \forall (i,k) \in N_a, \\ & && u_{ij} \geq 0, v_{ij} \geq 0, (y_{ij} + \tfrac{1}{4}, y_{ij} - \tfrac{1}{4}, w_{ij}) \in SOC \ \ \forall (i,j) \in N_x, \\ & && u_{ik} \geq 0, v_{ik} \geq 0, (y_{ik} + \tfrac{1}{4}, y_{ik} - \tfrac{1}{4}, w_{ik}) \in SOC \ \ \forall (i,k) \in N_a. \end{aligned}$$

The SOCP relaxation can be also viewed as a further relaxation of the SDP relaxation, and it was proved to be faster than the SDP method and to serve as a useful preprocessor of the actual problem. In this section, we will show that the ESDP model is stronger than the SOCP relaxation. Our proof refers to Proposition 1 of [30].

THEOREM 4.5. *If*

$$Z = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$$

is an optimal solution to (2.3), *then the ith column of* X, x_i, $i = 1, \dots, n$, *and*

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

form a feasible solution for (4.4) *with the same objective value.*

Proof. Since Z is a feasible solution to (2.3), we have $Z_{(1,2,i,j),(1,2,i,j)} \succeq 0$ for all $(i,j) \in N_x$. So, 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 4.6. *If* x_i *is invariant over all of the solutions of* (4.4), *then it is also invariant over all of the ESDP solutions. That is, if SOCP relaxation can return the true location for a 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. The following example shows that the reverse is not true.

Example 2. Consider the following problem with 3 anchors and 2 sensors. The true locations of 3 anchors are $a_1 = (-0.4, 0), a_2 = (0, 0.5)$, and $a_3 = (0.4, 0)$, and the true locations of the 2 sensors are $x_1 = (0, -0.3)$ and $x_2 = (0.4, 0.2)$ with radio range 0.7 (see Figure 4.3).

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 [2]), 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 (4.4) 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

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

must be also optimal to (4.4).

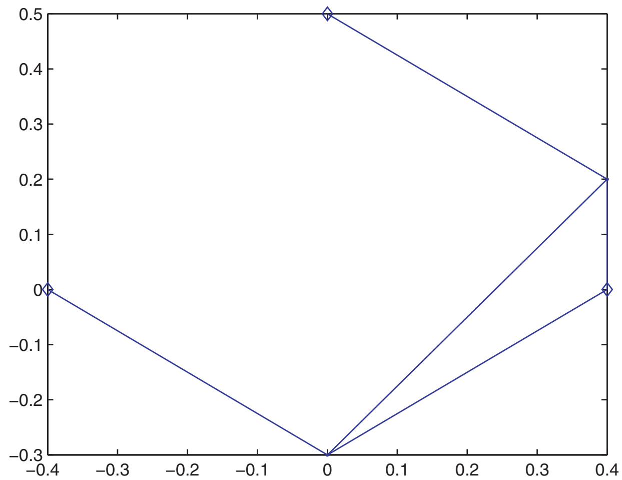


FIG. 4.3. *The locations of sensors, anchors, and connection edges in Example* 2.

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 (4.4). But we know that the interior-point method would always maximize the potential function (see [11, 10])

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

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 (4.4) will not give the true solution x_1 and x_2, and, thereby, the ESDP relaxation is *strictly* stronger than the SOCP relaxation for this example.

4.3. The dual problem of ESDP. For a conic programming problem, it is important to consider its dual problem. In many cases, the dual problem can give much important information about the primal problem as well as many useful applications. Here we will present the dual problem of (2.2) and list some basic properties between the primal and dual.

Consider a general conic programming problem:

$$\begin{array}{ll} \text{minimize} & C \bullet X \\ \text{subject to} & A_j \bullet X = b_j \quad \forall j, \\ & X_{(N_i)} \succeq 0 \quad \forall i, \end{array} \tag{4.5}$$

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

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_j b_j y_j \\ \text{subject to} & \displaystyle\sum_j y_j A_j + \sum_i S^i = C, \\ & S^i_{(N_i)} \succeq 0, \text{ and } S^i_{kj} = 0 \quad \forall k \notin N_i \text{ or } j \notin N_i; \quad \forall i. \end{array} \tag{4.6}$$

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 (2.2), the dual problem is

$$
\begin{aligned}
&\text{maximize} && \sum_{(i,j)\in N_x} \omega_{ij} d_{ij}^2 + \sum_{(i,k)\in N_a} \omega_{ik} \bar{d}_{ik}^2 + u_{11} + 2u_{12} + u_{22} \\
&\text{subject to} && \sum_{(i,j)\in N_x} \omega_{ij} (\mathbf{0}; e_i - e_j)^T (\mathbf{0}; e_i - e_j) + \sum_{(i,k)\in N_a} \omega_{ik} (-a_k; e_i)^T (-a_k; e_i) \\
&(4.7) && + \begin{pmatrix} u_{11}+u_{12} & u_{12} & \mathbf{0} \\ u_{12} & u_{22}+u_{12} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} + \sum_{(i,j)\in N_x} S^{(i,j)} = \mathbf{0}, \\
& && S^{(i,j)}_{(1,2,i,j)} \succeq 0, \text{ and } S^{(i,j)}_{kl} = 0 \ \forall k \notin \{i,j\} \text{ or } \ l \notin \{i,j\}, \ \forall (i,j) \in N_x.
\end{aligned}
$$

We have the following complementarity result.

PROPOSITION 4.7. *Let Z be a solution to* (2.2) *and* $\{S^{(i,j)}\}$ *be an optimal solution to the dual. Then*

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

In particular, if $\mathrm{Rank}(S^{(i,j)}_{(1,2,i,j)})$ *is 2 for all* $(i,j) \in N_x$*, then* $\mathrm{Rank}\left(Z_{(1,2,i,j)}\right)$ *is 2 for all* $(i,j) \in N_x$ *so that* (2.2) *produces a unique localization for the sensor network in* R^2*.*

By using duality we can solve the dual problem and simultaneously yield a primal solution from the complementarity proposition. We 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 [34].

5. Computational results and comparison to other approaches. Now we address the question: Will the improvement in the speed of the ESDP relaxation compensate the loss in relaxation quality? In this section, we first present some computational results of the ESDP relaxation model. Then we compare the model with different kinds of approaches, including the full SDP approach (1.1) of [7], the SOCP approach [30], the SOS approach [26], and the domain-decomposition approach of [25, 29].

5.1. Computational results of the ESDP relaxation. In our numerical simulation, we follow [7]. We randomly generate the true positions of n points in a square of 1 by 1, then randomly select m points to be anchors, and compute every edge length $\bar{d}_{ij}$. We select only those edges whose edge length is less than the given radio range rd and add a multiplicative random noise to every selected edge length,

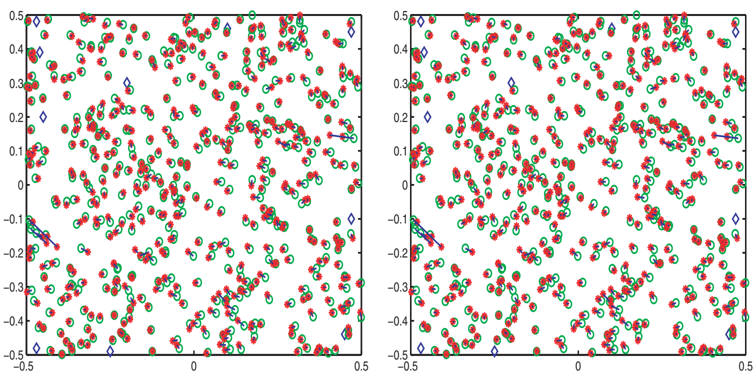
$$d_{ij} = \bar{d}_{ij}(1 + nf \cdot randn(1)),$$

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 still be many points within the radio range for a sensor or anchor. Thus, in order to maintain the sparsity of the graph, we set a limit 7 on the number of selected edges connected to every sensor or anchor, and they are *randomly* chosen.

In our computational experiments we also implement the steepest-descent local search refinement proposed in [28, 29] for solving noisy problems. All test problems are solved by SeDuMi 1.05 [32] of Matlab7.0 on a DELL D420 laptop with 1.99 GB memory and 1.06 GHz CPU.

Table 5.1
Noisy test problems and the SDP solution time comparison.

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



(a) Graphic result of the full SDP model

(b) Graphic result of the dual ESDP model

Fig. 5.1. *Comparision of graphical localization results generated by the full SDP and dual ESDP on a* 10% *noisy problem.*

The first set of test problems has noisy factor $nf = 0.1$ throughout. Table 5.1 contains a computational comparison of ESDP to the full SDP relaxation [7]. Here three models, the full SDP model (up to 400 points), the ESDP model, and the dual of the ESDP model, are all solved by SeDuMi 1.05. In order to see the efficiency of the ESDP model itself, the solution time (in seconds) in Table 5.1 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 speedup was remarkable, how about the localization quality? Figure 5.1 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 a few test problems have been placed on the public site [37]. We welcome the reader to test them and draw their own conclusions.)

Next we compare our approach to the SOS approach, the SOCP approach, and the domain-decomposition approach. We will use the same examples presented in these papers.

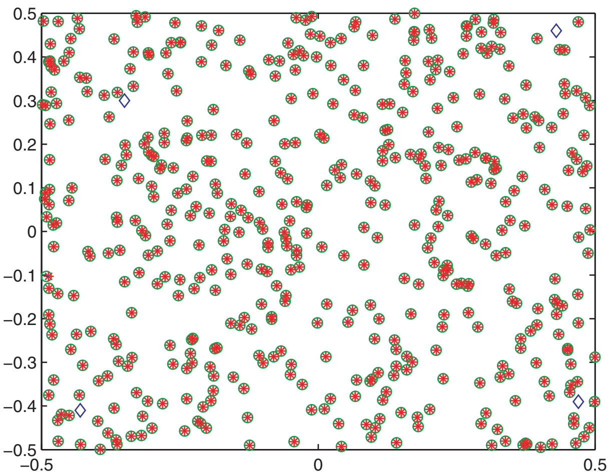


FIG. 5.2. *Graphical localization result of the ESDP model on the problem of Nie* [26]*, 500 sensors,* 4 *anchors,* $rd = 0.3$*,* $nf = 0$*, and RMSD* $= 1e - 6$*.*

5.2. Computational comparison with the SOS method. The SOS method is an SDP relaxation which applies to solving the problem

$$\min \quad f(x) = \sum_{(i,j)\in N_x} (\|x_i - x_j\|_2^2 - d_{ij}^2)^2 + \sum_{(i,k)\in N_a} (\|x_i - a_k\|_2^2 - \bar{d}_{ik}^2)^2, \tag{5.1}$$

where the objective function is a polynomial.

Recent study [26] has shown that by exploiting the sparsity in SOS relaxation one can get faster computing speed than the SDP relaxation (1.1) and sometimes higher accuracy as well. The author demonstrated that this structure can help save computation time significantly. In [26], the author used the model of 500 sensors and 4 anchors with a radio range of 0.3 and no noises in distance measurements.

The author of [26] reported that it took totally about 1 hour and 25 minutes on a 0.98 GB RAM and 1.46 GHz 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 the SOS approach in this case, and the solution quality is comparable to that of the SOS method; see Figure 5.2.

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 [30], 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 5.2. To solve the SOCP relaxation model, two methods are proposed in [30]: one directly uses Matlab SeDuMi, and the other uses a smoothing coordinate gradient descent (SCGD) method coded in FORTRAN 77. The latter is highly parallelizable, similar to the distributed methods of [25, 29].

From Table 5.2, we see that the ESDP approach is much faster than the SOCP approach when both use 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 5.3) of ESDP is much better than that reported in [30] for both SeDuMi of SOCP and SCGD of SOCP. Figure 5.3 shows the graphical result of test

Table 5.2
ESDP times are taken on DELL D420 (1.99 GB and 1.06 GHz), and SOCP times are reported from [30] on a HP DL360 (1 G memory and 3 GHz).

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

Table 5.3
Input parameters for the test problems, the corresponding ESDP dimensions, and ESDP computational results.

Test problem #	n	nf	rd	SeDuMi SDP dim	CPU time	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

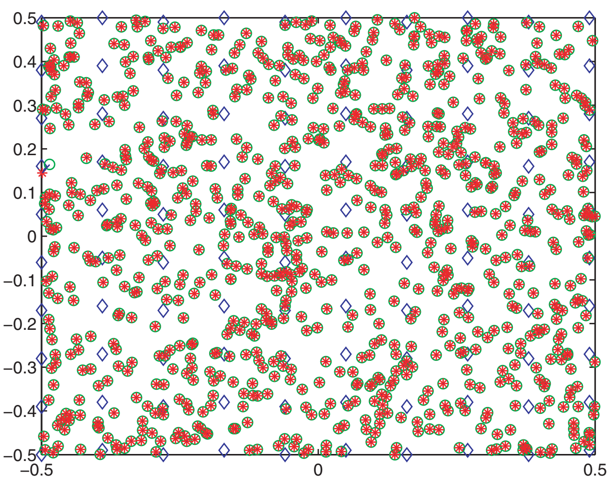


Fig. 5.3. *Graphical localization result of the ESDP model on test problem* 2 *in Table* 5.2.

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 [30].

In Table 5.2, "ESDP time" denotes the *total* solution running time, including Matlab data preparation and SeDuMi input setup time. By comparing Tables 5.1 and 5.2, one can see that, for ESDP, the Matlab data input and SeDuMi setup 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.

Table 5.3 contains more detailed statistical results on this test, where "SeDuMi SDP dim" represents problem dimensions solved by SeDuMi, "CPU time" denotes the total ESDP solution time in seconds (including Matlab data preparation and SeDuMi input setup time), "obj" denotes the SDP objective value, and RMSD is the localization quality defined by (4.3).

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 [29] and SpaceLoc of [25] 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 accurately. 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 independent localization problem. However, when the quantity of anchors is small or most of them are located on the boundary, such as the problems in Table 5.1, these approaches would fail at the beginning, simply because they are reduced to solving a nearly full-size SDP problem.

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 a full-size 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 sub-SDP approaches indeed have a great potential to save computation time in solving sensor network localization problems, and the efficiency of the model is considerable. At the same time, they retain some of the 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 in localization quality, we still lack some powerful theorems to illustrate why the model works. This is a major issue that needs to be answered. Second, since, in our ESDP model, the decision matrix has its special structure, applying a tailored interior-point method (such as SCGD for the SOCP approach) may save more computational time. 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 with the NSDP model for solving the Max-Cut problem and will 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 Appl., 310 (2000), pp. 149–165.
[2] A. M.-C So and Y. Ye, *Theory of semidefinite programming for sensor network localization*, Math. Program., 109 (2007), pp. 367–384.
[3] A. Y. Alfakih, A. Khandani, and H. Wolkowicz, *Solving Euclidean distance matrix completion problems via semidefinite programming*, Comput. Optim. Appl., 12 (1999), pp. 13–30.
[4] A. Y. Alfakih, *On rigidity and realizability of weighted graphs*, Linear Algebra Appl., 325 (2001), pp. 57–70.
[5] M. Bădoiu, *Approximation algorithm for embedding metrics into a two-dimensional space*, in SODA '03: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2003, pp. 434–443.

[6] A. I. Barvinokdoi, *Problems of distance geometry and convex properties of quadratic maps*, Discrete Comput. Geom., 13 (1995), pp. 189–202.
[7] P. Biswas and Y. Ye, *Semidefinite programming for ad hoc wireless sensor network localization*, in IPSN 2004 Proceedings of the Third International Symposium on Information Processing in Sensor Networks, ACM, New York, 2004, pp. 46–54.
[8] J. Blair and B. Peyton, *An introduction to chordal graphs and clique trees*, Inst. Math. Appl., 56 (1993), pp. 1–30.
[9] B. Grone, C. R. Johnson, E. M. Sa, and H. Wolkowicz, *Positive definite completions of partial Hermitian matrices*, Linear Algebra Appl., 58 (1984), pp. 109–124.
[10] D. Goldfarb and K. Scheinberg, *Interior point trajectories in semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 871–886.
[11] O. Güler and Y. Ye, *Convergence behavior of interior-point algorithms*, Math. Program., 60 (1993), pp. 215–228.
[12] B. Hendrickson, *The molecule problem: Exploiting structure in global optimization*, SIAM J. Optim., 5 (1995), pp. 835–857.
[13] L. Hogben, *Graph theoretic methods for matrix completion problems*, Linear Algebra Appl., 328 (2001), pp. 161–202.
[14] C. R. Johnson and R. L. Smith, *The positive definite completion problem relative to a subspace*, Linear Algebra Appl., 307 (2000), pp. 1–14(14).
[15] J. Aspnes, D. Goldenberg, and Y. R. Yang, *On the computational complexity of sensor network localization*, in Algorithmic Aspects of Wireless Sensor Networks: First International Workshop, ALGOSENSORS 2004, Turku, Finland, July 16, 2004. Proceedings, Lecture Notes in Comput. Sci. 3121, Springer-Verlag, New York, 2004, pp. 32–44.
[16] J. A. Costa, N. Patwari, and A. O. Hero, III, *Distributed weighted-multidimensional scaling for node localization in sensor networks*, ACM Trans. Sen. Netw., 2 (2006), pp. 39–64.
[17] J. Dahl, V. Roychowdhury, and L. Vandenberghe, *Maximum-Likelihood Estimation of Multivariate Normal, Graphical Models: Large-Scale Numerical Implementation and Topology*, Working Paper, Electrical Engineering Department, UCLA, 2006.
[18] M. Kojima, S. Kim, and H. Waki, *Sparsity in sums of squares of polynomials*, Math. Program., 103 (2005), pp. 45–62.
[19] 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, New York, 2004, p. 106.
[20] M. Laurent, *Matrix completion problems*, Encyclopedia Optim., 3 (2001), pp. 221–229.
[21] L. Doherty, K. S. J. Pister, and L. El Ghaoui, *Convex position estimation in wireless sensor networks*, in INFOCOM 2001 Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, 2001, pp. 1655–1663.
[22] J. J. Moré and Z. Wu, *Global continuation for distance geometry problems*, SIAM J. Optim., 7 (1997), pp. 814–836.
[23] M. Bâdoiu, E. D. Demaine, M. T. Hajiaghayi, and P. Indyk, *Low-dimensional embedding with extra information*, Discrete Comput. Geom., 36 (2006), pp. 609–632.
[24] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, *Exploiting sparsity in semidefinite programming via matrix completion* I: *General framework*, SIAM J. Optim., 11 (2000), pp. 647–674.
[25] M. W. Carter, H. H. Jin, M. A. Saunders, and Y. Ye, *SpaseLoc: An adaptive subproblem algorithm for scalable wireless sensor network localization*, SIAM J. Optim., 17 (2006), pp. 1102–1128.
[26] J. Nie, *Sum of squares method for sensor network localization*, Comput. Optim. Appl., to appear.
[27] G. Pataki, *On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eignevalues*, Math. Oper. Res., 23 (1998), pp. 339–358.
[28] P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang, *Semidefinite programming approaches for sensor network localization with noisy distance measurements*, IEEE Trans. Automat. Sci. Eng., 3 (2006), pp. 360–371.
[29] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, *Semidefinite programming based algorithms for sensor network localization*, ACM Trans. Sen. Netw., 2 (2006), pp. 188–220.
[30] P. Tseng, *Second-order cone programming relaxation of sensor network localization*, SIAM J. Optim., 18 (2007), pp. 156–185.
[31] A. M.-C. So and Y. Ye, *A semidefinite programming approach to tensegrity theory and realizability of graphs*, in SODA '06: Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm, ACM, New York, 2006, pp. 766–775.

[32] J. F. Sturm, *Using SeDuMi* 1.02, *a MATLAB toolbox for optimization over symmetric cones*, Optim. Methods Softw., 11–12 (1999), pp. 625–653.

[33] S. Boyd, P. Diaconis, and L. Xiao, *Fastest mixing Markov chain on a graph*, SIAM Rev., 46 (2004), pp. 667–689.

[34] S. J. Benson, Y. Ye, and X. Zhang, *Solving large-scale sparse semidefinite programs for combinatorial optimization*, SIAM J. Optim., 10 (1999), pp. 443–461.

[35] T. Eren, O. K. 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 INFOCOM 2004 Proceedings of the Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 4, 2004, pp. 2673–2684.

[36] 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, ON N2L 3G1, Canada.

[37] http://www.stanford.edu/~yyye/