Semidefinite Programming Based Algorithms for Sensor Network Localization

Pratik Biswas
Dept. of Electrical Engineering , Stanford University, pbiswas@stanford.edu
Tzu-Chen Liang
Dept. of Aeronautics And Astronautics, Stanford University, tzuchen@stanford.edu
Ta-Chung Wang
Dept. of Aeronautics And Astronautics, Stanford University, tcwang@stanford.edu
and
Yinyu Ye
Dept of Management Science and Engineering and by courtesy, Electrical Engineering, Stanford University, yinyu-ye@stanford.edu

An SDP relaxation based method is developed to solve the localization problem in sensor networks using incomplete and inaccurate distance information. The problem is set up to find a set of sensor positions such that given distance constraints are satisfied. The nonconvex constraints in the formulation are then relaxed in order to yield a semidefinite program which can be solved efficiently.

The basic model is extended in order to account for noisy distance information. In particular, a maximum likelihood based formulation and an interval based formulation are discussed. The SDP solution can then also be used as a starting point for steepest descent based local optimization techniques that can further refine the SDP solution.

We also describe the extension of the basic method to develop an iterative distributed SDP method for solving very large scale semidefinite programs that arise out of localization problems for large dense networks and are intractable using centralized methods.

The performance evaluation of the technique with regard to estimation accuracy and computation time is also presented by the means of extensive simulations.

Our SDP scheme also seems to be applicable to solving other Euclidean geometry problems where points are locally connected.

Categories and Subject Descriptors: G.1.6 [**Optimization**]: Convex programming; G.4 [**Mathematical Software**]: Algorithm design and analysis

General Terms: Algorithms

Additional Key Words and Phrases: Semidefinite Programming, Sensor Network Localization, Distributed Methods

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. 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 is a very expensive solution to this requirement.

Instead, techniques to estimate node positions are being developed that rely just on the measurements of distances and angles between neighboring nodes. Depending on the accuracy of these measurements and processor, power and memory constraints at each of the nodes, there is some degree of error in the distance information. Furthermore, it is assumed that we already know the positions of a few anchor nodes. The problem of finding the positions of all the nodes given a few anchor nodes and relative distance and angle information between the nodes is called the position estimation or localization problem. Section 2 briefly discusses some of the previous work in this area.

Section 3 describes an SDP relaxation based method (first discussed in [Biswas and Ye 2004]) for the position estimation problem in sensor networks. The optimization problem is set up so as to minimize the error in sensor positions for fitting the distance measures. The basic idea behind the technique is to convert the nonconvex quadratic distance constraints into convex constraints by introducing a relaxation to remove the quadratic term in the formulation. Similar relaxations were developed for solving other distance geometry problems, see, e.g., [Alfakih et al. 1999] and [Laurent 2001].

Section 4 extends the basic distance geometry model to effectively use noisy distance constraints as well. Nonlinear optimization problems using lower and upper bound or interval based constraints as well as equality constraints are developed and the SDP relaxation is used to convert the problem into a convex optimization problem. Ideas from maximum likelihood estimation are also utilized to setup optimization problems that minimize the expected error in estimation.

Section 5 describes a steepest descent based local search method that can further refine the solutions obtained from the SDP. Infact, the SDP solution turns out to be an excellent starting point for the local optimization and provides good convergence.

In Section 6, we present an iterative distributed semidefinite programming method for solving localization problems that arise from networks consisting of a large number of sensors. Using the distributed method, we can solve very large scale semidefinite programs which are intractable for the centralized methods.

The SDP based techniques are evaluated in terms of estimation error by performing extensive simulations using various configurations of radio range, measurement noise, number of anchors etc. The performance is highly satisfactory compared to other techniques as well. These results are presented in Section 7.

2. RELATED APPROACHES

A great deal of research has been done on the topic of position estimation in ad-hoc networks ([Ganesan et al. 2002], [Hightower and Borriello 2001]). Most techniques use distance or angle measurements from a fixed set of reference or anchor nodes; see [Doherty et al. 2001], [Niculescu and Nath 2001], [Savarese et al. 2002], [Savvides et al. 2001], [Savvides et al. 2002], [Shang et al. 2004]; or employ a grid of beacon nodes with known positions; see [Bulusu et al. 2000], [Howard et al. 2001]. Also see [Moré and Wu 1997] for solving another distance geometry problem.

[Niculescu and Nath 2001] describe the "DV-Hop" and related "DV-Distance" and Euclidean approaches which is quite effective in dense and regular topologies. The anchor nodes flood their position information to all the nodes in the network. Each node then estimates its own position by performing a triangulation using this information. For more irregular topologies however, the accuracy can deteriorate to the radio range. The authors also investigate the use of angle information ("DV-Bearing") in for localization [Niculescu and Nath 2003] using the information forwarding techniques described above.

[Savarese et al. 2002] present a 2 phase algorithm in which the start-up phase involves finding the rough positions of the nodes using a technique similar to the "DV-Hop" approach. The refinement phase improves the accuracy of the estimated positions by performing least squares triangulations using its own estimates and the estimates of the nodes in its own neighborhood. This method can accurately estimate points within one third of the radio range.

When the number of anchor nodes is high, the "iterative multilateration" technique proposed by [Savvides et al. 2001] yields good results. Nodes that are connected to 3 or more anchors compute their position by triangulation and upgrade themselves to anchor nodes. Now their position information can also be used by the other unknown nodes for their position estimation in the next iteration.

[Howard et al. 2001] and [Priyantha et al. 2001] have discussed the use of spring based relaxations that initially try to find a graph embedding that resembles the actual configuration and then modify the embedding to approach the actual one using a mass-spring based optimization to correct and balance errors.

While the above methods can be run in a distributed fashion, there also exist some centralized methods that offer more precise location determination by using global information.

[Shang et al. 2003] demonstrate the use of a data analysis technique called "multidimensional scaling" (MDS) in estimating positions of unknown nodes. Firstly, using basic connectivity or distance information, a rough estimate of relative node distances is made. Then classical MDS (which basically involves using a Singular Value decomposition) is used to obtain a relative map of the node positions. Finally an absolute map is obtained by using the known node positions. This technique works well with few anchors and reasonably high connectivity. For instance, for a connectivity level of 12 and 2% anchors, the error is about half of the radio range. The centralized version of the above mentioned technique does not work well when the network topology is irregular. So an alternative distributed MDS and patching technique is explored in [Shang and Ruml 2004; Shang et al. 2004]. Here local clusters with regular topologies are solved separately and then stitched together subsequently.

One approach closely related to ours is described in [Doherty et al. 2001] wherein the proximity constraints between nodes which are within 'hearing distance' of each other are modeled as convex constraints. Then a feasibility problem can be solved by efficient convex programming techniques.

It will be helpful to first introduce some notations to describe this technique. The trace of a given matrix A, denoted by $\mathrm{Trace}(A)$, is the sum of the entries on the main diagonal of A. We use I, e and $\mathbf{0}$ to denote the identity matrix, the vector of all ones and the vector of all zeros, whose dimension will be clear in the context.

The inner product of two vector p and q is denoted by $\langle p, q\rangle$. The 2-norm of a vector x, denoted by $\|x\|$, is defined by $\sqrt{\langle x, x\rangle}$. A positive semidefinite matrix X is represented by $X \succeq 0$.

Suppose 2 nodes x_1 and x_2 are within radio range R of each other, the proximity constraint can be represented as a convex second order cone constraint of the form

$$\|x_1 - x_2\|_2 \leq R. \tag{1}$$

This can be formulated as a matrix linear inequality ([Boyd et al. 1994]):

$$\begin{pmatrix} I_2R & x_1 - x_2 \\ (x_1 - x_2)^T & R \end{pmatrix} \succeq 0. \tag{2}$$

Alternatively, if the exact distance $r_{1,2} \leq R$ is known, we could set the constraint

$$\|(x_1 - x_2)\|_2 \leq r_{1,2}. \tag{3}$$

The second-order cone method for solving Euclidean metric problems can be also found in [Xue and Ye 1997] where its superior polynomial complexity efficiency is presented.

However, this technique yields good results only if the anchor nodes are placed on the outer boundary, since the estimated positions of their convex optimization model all lie within the convex hull of the anchor nodes. So if the anchor nodes are placed in the interior of the network, the position estimation of the unknown nodes will also tend to the interior, yielding highly inaccurate results. For example, with just 5 anchors in a random 200 node network, the estimation error is almost twice the radio range.

One may ask why not add, if $r_{1,2}$ is known, another 'bounding away' constraint

$$\|(x_1 - x_2)\|_2 \geq r_{1,2}. \tag{4}$$

These two constraints are much tighter and would yield more accurate results. The problem is that the latter is not a convex constraint, so that the efficient convex optimization techniques cannot apply. The SDP relaxation method presented in this paper attempts to formulate tighter convex constraints similar to 4. Then we relax it to linear matrix inequalities similar to (2).

The techniques described above also differ in their implementation in terms of being centralized or distributed. For centralized techniques, the available distance information between all the nodes must be present on a single computer. The distributed approach has the advantage that the techniques can be executed on the sensor nodes themselves thus removing the need to relay all the information to a central computer. This also affects the scalability and accuracy of the problems under consideration. Distributed techniques can handle much larger networks whereas centralized approaches yield higher accuracy by using global information. Our distributed approach is designed to employ the advantages of both paradigms.

3. SEMIDEFINITE PROGRAMMING METHODS

We would like to first introduce some notations and the mathematical formulation of ad hoc sensor network localization problem. For two symmetric matrices A and B, $A \succeq B$ means $A - B \succeq 0$, i.e. $A - B$ is a positive semidefinite matrix. We use I_d, e and $\mathbf{0}$ to denote the $d \times d$ identity matrix, the vector of all ones and the vector of all zeros, whose dimensions will be clear in the context. e_i is a vector with all zeros except its ith entry, which is one. The 2-norm of a vector x is denoted as $\|x\|$.

The following notations are related to the ad hoc sensor network localization problem. In an ad hoc sensor network in $\Re^2$ with m anchors and n sensors, an anchor is a node whose location a_k in $\Re^2$, $k = 1, 2, ..., m$, is known, and a sensor is a node whose location has yet to be decided and denoted by x_j in $\Re^2$, $j = 1, 2, ..., n$. Note that for the sake of uniformity, we will be referring to anchor nodes strictly with the index k, and the unknown nodes with indices i, j.

For a pair of sensors x_j and x_i, their Euclidean distance is denoted as d_{ji}. Similarly, for a sensor x_j and anchor a_k, their Euclidean distance is denoted as d_{jk}. In general, not all pairs of distances are known, so the pairs of nodes for which mutual distances are known are denoted as $(j, i) \in N_x$ for sensor/sensor and $(j, k) \in N_a$ for sensor/anchor pairs respectively. In this section, the basic idea of the SDP formulation will be explained using exact distance data, that is, not corrupted by noise. Extensions to noisy data will be discussed in following sections.

So, mathematically, the localization problem in $\Re^2$ can be stated as: given m anchor locations $a_k, k = 1, 2, ..., m$ and some distance measurements $d_{ji}, (j, i) \in N_x, d_{jk}, (j, k) \in N_a$, find $x_j, j = 1, 2, ..., n$, the locations of n sensors, such that

$$\begin{aligned} \|x_j - x_i\|^2 &= d_{ji}^2, \forall (j, i) \in N_x \\ \|x_j - a_k\|^2 &= d_{jk}^2, \forall (j, k) \in N_a. \end{aligned}$$

3.1 SDP Formulation

Let matrix $X = [x_1,\ x_2,\ ...,\ x_n]$. Then, the problem can be written in matrix form,

$$\begin{array}{ll} \text{find} & X \in \Re^{2\times n}, Y \in \Re^{n\times n} \\ \text{s.t} & (e_i - e_j)^T Y (e_i - e_j) = d_{ji}^2, \forall (j, i) \in N_x \\ & \begin{pmatrix} a_k \\ -e_j \end{pmatrix}^T \begin{pmatrix} I_2 & X \\ X^T & Y \end{pmatrix} \begin{pmatrix} a_k \\ -e_j \end{pmatrix} = d_{jk}^2, \forall (j, k) \in N_a \\ & Y = X^T X. \end{array} \tag{5}$$

Our method is to relax problem (5) to a semidefinite program: Change the constraint $Y = X^T X$ in (5) to $Y \succeq X^T X$. This matrix inequality is equivalent to (e.g., [Boyd et al. 1994])

$$Z := \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix} \succeq 0.$$

Then, the problem can be written as a standard SDP (feasibility) problem:

$$\begin{array}{c} \text{find } Z \in \Re^{(n+2)\times(n+2)} \\ \text{s.t. } (1; 0; \mathbf{0})^T Z (1; 0; \mathbf{0}) = 1 \\ (0; 1; \mathbf{0})^T Z (0; 1; \mathbf{0}) = 1 \\ (1; 1; \mathbf{0})^T Z (1; 1; \mathbf{0}) = 2 \\ (0; e_i - e_j)^T Z (0; e_i - e_j) = d_{ji}^2, \forall (j, i) \in N_x \\ (a_k; -e_j)^T Z (a_k; -e_j) = d_{jk}^2, \forall (j, k) \in N_a \\ Z \succeq 0 \end{array} \tag{6}$$

where $Z = \begin{pmatrix} I_2 & X \\ X^T & Y \end{pmatrix}$, e_j is a vector with all zeros except its ith entry, which is one, $(a_k; -e_j)$ is a column vector in $\Re^{n+2}$ with a_k stacked on top of $-e_j$.

3.2 SDP Model Analyses

The matrix of Z of (6) has $2n + n(n+1)/2$ unknown variables. Consider the case that among $\{k, i, j\}$, there are $2n + n(n+1)/2$ pairs in N_x and N_a, Then we have at least $2n + n(n+1)/2$ linear equalities among the constraints. Moreover, if these equalities are linearly independent, then Z has a unique solution. Therefore, we can show

PROPOSITION 1. *If there are $2n + n(n+1)/2$ distance pairs each of which has an accurate distance measure. Then (6) has a unique feasible solution*

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

then we must have $\bar{Y} = (\bar{X})^T \bar{X}$ and $\bar{X}$ equal true positions of the unknown sensors. That is, the SDP relaxation solves the original problem exactly.

PROOF. Let X^* be the true locations of the n points, and

$$Z^* = \begin{pmatrix} I & X^* \\ (X^*)^T & (X^*)^T X^* \end{pmatrix}.$$

Then Z^* is a feasible solution for (6).

On the other hand, since $\bar{Z}$ is the unique solution to satisfy the $2n + n(n+1)/2$ equalities, we must have $\bar{Z} = Z^*$ so that $\bar{Y} = (X^*)^T X^* = \bar{X}^T \bar{X}$. □

We present a simple case to show what it means for the system has a unique solution. Consider $n = 1$ and $m = 3$. The accurate distance measures from unknown b_1 to known a_1, a_2 and a_3 are d_{11}, d_{21} and d_{31}, respectively. Therefore, the three linear equations are

$$\begin{aligned} y - 2x^T a_1 &= (d_{11})^2 - \|a_1\|^2 \\ y - 2x^T a_2 &= (d_{21})^2 - \|a_2\|^2 \\ y - 2x^T a_3 &= (d_{31})^2 - \|a_3\|^2. \end{aligned}$$

This system has a unique solution if it has a solution and the matrix

$$\begin{pmatrix} 1 & 1 & 1 \\ a_1 & a_2 & a_3 \end{pmatrix}$$

is nonsingular. This essentially means that the three points a_1, a_2 and a_3 are not on the same line, and then $\bar{x} = b_1$ can be uniquely determined. Here, the SDP method reduces to the so-called triangular method. Proposition 1 and the example show that the SDP relaxation method has the advantage of the triangular method in solving the original problem.

Conditions under which the problem is uniquely localizable and its connections to rigidity theory are explored in more detail in [So and Ye 2004] in detail. In particular, we repeat a condition of uniquely localizability,

DEFINITION 1. *Problem (5) is uniquely localizable if it has a unique feasible solution $\bar{X}$ in $\Re^{2\times n}$ and there is no x_j in $\Re^h, j = 1, ..., n$, where $h > 2$ (excluding the* *case appending all zeros to* $\bar{X}$*), such that*

$$\begin{aligned} \|x_j - x_i\|^2 &= d_{ji}^2, \forall (i,j) \in N_x \\ \|x_j - (a_k; \mathbf{0})\|^2 &= d_{jk}^2, \forall (j,k) \in N_a. \end{aligned} \quad (7)$$

The latter condition in the definition says that the problem cannot be localized in a higher dimensional space where anchor points are augmented to $(a_k; \mathbf{0}) \in \Re^h, j = 1, ..., m$. The importance of [So and Ye 2004] is that it firstly states that if the problem is uniquely localizable, then the relaxation problem (6) solves (5) exactly. To tell whether a problem is uniquely localizable or not before solving it is not easy. But once we solve the SDP relaxation and observe whether $Y = X^T X$ in the solution, we immediately know if the problem is uniquely localizable or not. It should be kept in mind however, that the assumption in the above analyses is that the distance measures are exact. We will deal with noisy data in the following sections.

In particular, each individual trace

$$\bar{Y}_{jj} - \|\bar{x}_j\|^2 \quad (8)$$

helps us to detect errors in estimation and isolate exactly the sensors which fail to be estimated given the incomplete distance information.

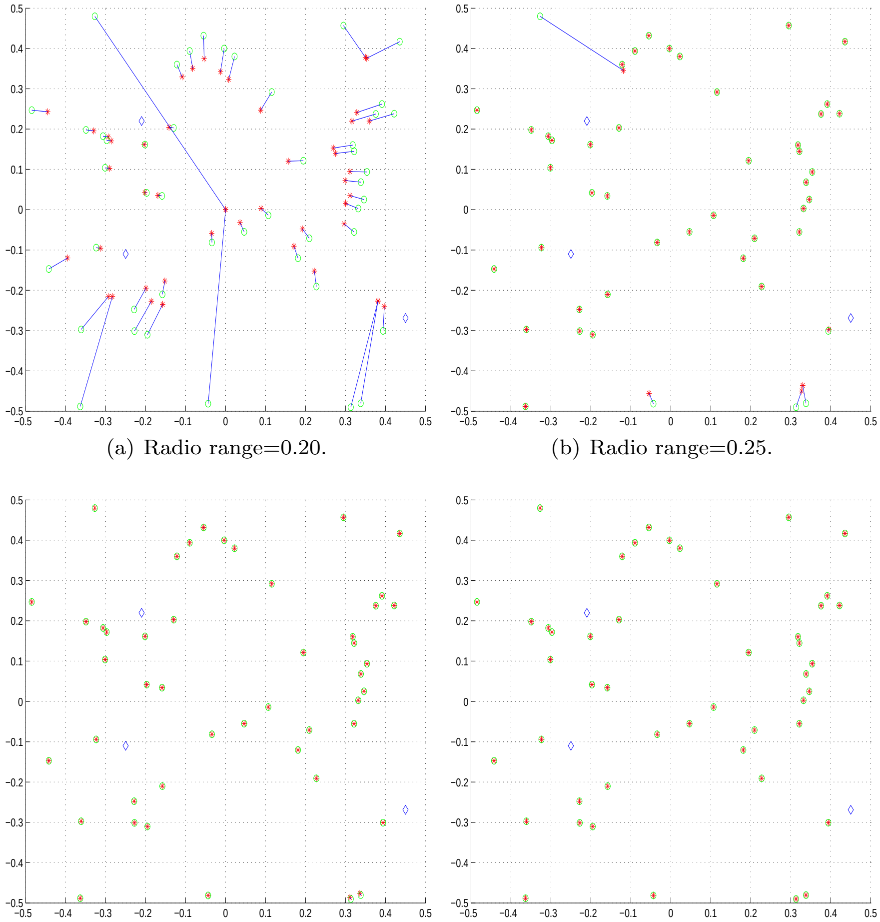
Example 1. *For the purpose of our examples, we generated random networks of points uniformly distributed in a square area of* 1×1. *Distances were computed between points that are within the radio range of each other. (The radio range indicates that the distance values between any two nodes are known to the solver if they are below the range; otherwise they are unknown.) The original and the estimated sensors were plotted. The (blue) diamond nodes refer to the positions of the anchors; (green) circle nodes to the original locations, A, of the unknown sensors; and (red) asterisk nodes to their estimated positions from* $\bar{X}$. *The discrepancies in the positions can be estimated by the offsets between the original and the estimated points as indicated by the solid lines. (The same setup and notations will be used in all examples of this article).*

The effect of variable radio ranges and as a result, connectivity, was observed in Figure 1. For a network of 50 points, the radio range was varied from 0.2 *to* 0.35. *In Figure 1(b), for the four sensors with large error estimation, their individual traces made up most of the total traces, which match where the real errors are accurately, see Figure 2 for the correlation between individual error and trace for each unknown sensor for cases in Figure 1(a) and 1(b).*

In comparison, for the same case as Figure 1, we computed the results from the [Doherty et al. 2001] method with the number of anchors 10 *and* 25, *and depicted their pictures in Figure 3. As we expected, the estimated positions were all in the convex hull of the anchors.*

3.3 Computational Complexity

A worst-case complexity result to solve the SDP relaxation can be derived from employing interior-point algorithms, e.g., [Benson et al. 2000].



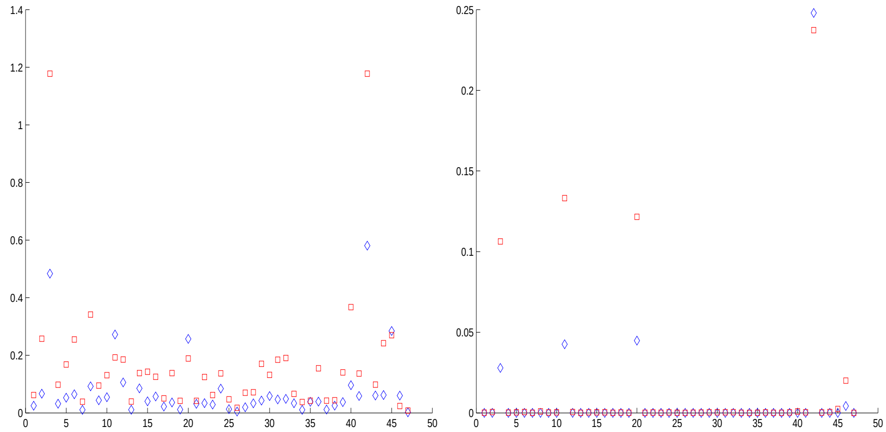
(a) Radio range=0.20. (b) Radio range=0.25.

(c) Radio range=0.30. (d) Radio range=0.35.

Fig. 1. Position estimations with 3 anchors, accurate distance measures and various radio ranges.

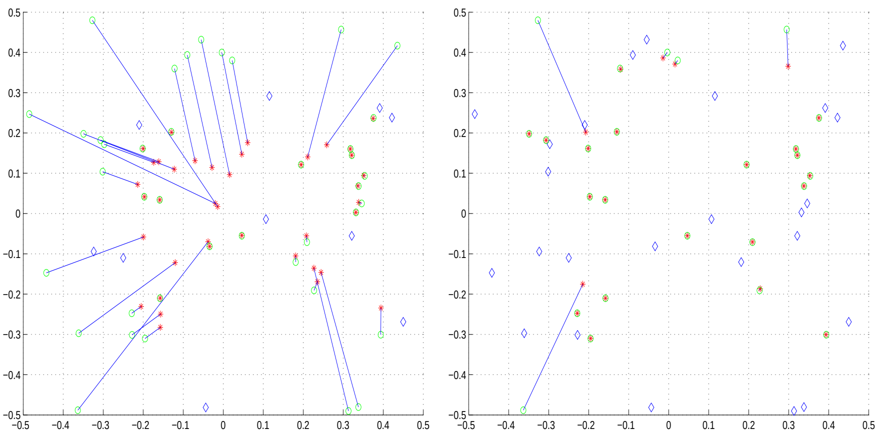
THEOREM 1. *Let $k = 3 + |N_x| + |N_a|$, the number of constraints. Then, the worst-case number of total arithmetic operations to compute an ϵ-solution of (6), meaning its objective value is at most $\epsilon(> 0)$ above the minimal one, is bounded by $O(\sqrt{n+k}(n^3 + n^2k + k^3)\log\frac{1}{\epsilon})$, in which $\sqrt{n+k}\log\frac{1}{\epsilon}$ represents the bound on the worst-case number of interior-point algorithm iterations.*

Practically, the number of interior-point algorithm iterations to compute a fairly accurate solution is a constant, $20-30$, for semidefinite programming, and k is bounded by $O(n^2)$. Thus, the worst case complexity is bounded by $O(n^6)$. However, in practice, as the number of points increases, the required radio range and number of constraints required to solve for all positions scales more typically with $O(n)$. Therefore the number of operations is typically bounded by $O(n^3)$ in solving a localization problem with n sensors. This issue is explored in further detail through



(a) Error and trace correlation in Figure 1(a). (b) Error and trace correlation in Figure 1(b).

Fig. 2. Diamond: the offset distance between estimated and true positions, Box: the square root of individual trace $\bar{Y}_{jj} - \|\bar{x}_j\|^2$.



(a) 10 anchors (b) 25 anchors

Fig. 3. Position estimations by Doherty et al., Radio range=0.30, accurate distance measures and various number of anchors

simulations in Section 7.

4. EXTENSIONS FOR NOISY DISTANCE INFORMATION

For the localization problem with measurement noises, the story can be quite different. In general there is no feasible solution to satisfy the constraints in (5). We develop two formulations to deal with noisy data: one minimizes the maximum likelihood estimation error and the other solves the distance feasibility problem with upper and lower bound or *confidence interval* measures. Both formulations admit SDP relaxations similar to the one discussed earlier, and yield highly satisfactory computational results.

4.1 Maximum Likelihood Estimation and its SDP relaxation

One approach, called maximum likelihood estimation, has been presented in [Liang et al. 2004]. Let $d : \Re^2 \times \Re^2 \rightarrow \Re$ be the distance function between a sensor/anchor or sensor/sensor pair. Suppose there are some measurement errors between x_j/a_k and x_j/x_i denoted by ω_{jk} and ω_{ji}, respectively,

$$\begin{aligned} d_{jk} &= d(x_j, a_k) + \omega_{jk}, \forall (j,k) \in N_a \\ d_{ji} &= d(x_j, x_i) + \omega_{ji}, \forall (j,i) \in N_x \end{aligned}$$

where we assume each $\omega_{jk} \sim \mathcal{N}(0, \sigma_{jk}^2)$ and $\omega_{ji} \sim \mathcal{N}(0, \sigma_{ji}^2)$, where $\mathcal{N}(0, \sigma^2)$ is a normal random variable with mean zero and variance σ^2, and they are independent.

Let the maximum likelihood function p to estimate X, using all distance measure information, be

$$\begin{aligned} & p((d_{jk}, (j,k) \in N_a; d_{ji}, (j,i) \in N_x), X) = \\ & \prod_{j,k;(j,k)\in N_a} \frac{1}{2\pi^{\frac{1}{2}}\sigma_{jk}} \exp\left(-\frac{1}{2\sigma_{jk}^2} (d_{jk} - d(x_j, a_k))^2 \right) \\ & \prod_{j,i;(j,i)\in N_x} \frac{1}{2\pi^{\frac{1}{2}}\sigma_{ji}} \exp\left(-\frac{1}{2\sigma_{ji}^2} (d_{ji} - d(x_j, x_i))^2 \right) \end{aligned}$$

and the maximum likelihood estimation be

$$X_{ml} = \arg\max_X p((d_{jk}, (j,k) \in N_a; d_{ji}, (j,i) \in N_x), X).$$

Then, X_{ml} can be written explicitly as

$$X_{ml} = \arg\min_X \begin{pmatrix} \sum_{j,k;(j,k)\in N_a} \frac{1}{\sigma_{jk}^2} (d_{jk} - d(x_j, a_k))^2 \\ + \sum_{j,i;(j,i)\in N_x} \frac{1}{\sigma_{ji}^2} (d_{ji} - d(x_j, x_i))^2 . \end{pmatrix} \tag{9}$$

Hence the following optimization problem solves the maximum likelihood estimation problem

$$\begin{array}{ll} \min & \sum_{j,k;(j,k)\in N_a} \frac{1}{\sigma_{jk}^2} \epsilon_{jk} + \sum_{j,i;(j,i)\in N_x} \frac{1}{\sigma_{ji}^2} \epsilon_{ji} \\ \text{s.t.} & (\|x_j - x_i\| - d_{ji})^2 = \epsilon_{ji}, \forall (j,i) \in N_x \\ & (\|x_j - a_k\| - d_{jk})^2 = \epsilon_{jk}, \forall (j,k) \in N_a. \end{array} \tag{10}$$

The objective function described above is a special case of a class of cost functions described in weighted multidimensional scaling literature, see [Cox and Cox 2001]. In fact, weighted multidimensional scaling has been applied to the sensor network localization problem in [Costa et al. 2005], and the MLE function described above is a specific case of the general cost function described therein. The key lies on how to solve this (nonconvex) problem. In what follows, we show that this cost minimization problem admits a similar SDP relaxation problem and can be solved by solving and rounding its SDP relaxation. We use the MLE cost function for illustrating our relaxation technique. It should be borne in mind, however, that more complicated cost functions of the type defined in [Costa et al. 2005] can also be dealt with in the SDP framework.

If the variance of distance measurements are not known, any reasonable assumption can be applied. Although problem (10) is not a convex optimization problem, one can construct its SDP relaxation problem:

$$\begin{array}{ll} \min & \sum_{j,k;(j,k)\in N_a} \frac{1}{\sigma_{jk}^2}\epsilon_{jk} + \sum_{j,i;(j,i)\in N_x} \frac{1}{\sigma_{ji}^2}\epsilon_{ji} \\ & (-d_{ji};1)^T D_{ji} (-d_{ji};1) = \epsilon_{ji}, \forall (j,i)\in N_x \\ & (-d_{jk};1)^T D_{jk} (-d_{jk};1) = \epsilon_{jk}, \forall (j,k)\in N_a \\ & (0;e_j-e_i)^T Z (0;e_j-e_i) = v_{ji}, \forall (j,i)\in N_x \\ \text{s. t.} & (a_k;-e_j)^T Z (a_k;-e_j) = v_{jk}, \forall (j,k)\in N_a \\ & D_{ji}\succeq 0, \forall (j,i)\in N_x \\ & D_{jk}\succeq 0, \forall (j,k)\in N_a \\ & Z\succeq 0 \end{array} \tag{11}$$

where $Z = \begin{pmatrix} I_2 & X \\ X^T & Y \end{pmatrix}$ and

$D_{ji} = \begin{pmatrix} 1 & u_{ji} \\ u_{ji} & v_{ji} \end{pmatrix}$, $\forall (j,i)\in N_x, D_{jk} = \begin{pmatrix} 1 & u_{jk} \\ u_{jk} & v_{jk} \end{pmatrix}$, $\forall (j,k)\in N_a$.

One particular case is when we assume multiplicative noise (as is the case with our simulations), that is,

$$d_{ji} = \tilde{d}_{ji}\cdot(1+\mathcal{N}(0,\sigma^2)),$$

where $\tilde{d}_{ji}$ is the actual distance and d_{ji} is the measured distance. Then, $\sigma_{jk}^2 = \tilde{d}_{jk}^2\sigma^2, \quad \forall (j,k)\in N_a$ and $\sigma_{ji}^2 = \tilde{d}_{ji}^2\sigma^2, \quad \forall (j,i)\in N_x$.

Since the true distances are not known, we may approximate the variances using the measured distances d_{ij} and d_{jk}. Therefore, in this case, the objective value will be

$$\sum_{j,k;(j,k)\in N_a} \frac{1}{d_{jk}^2}\epsilon_{jk} + \sum_{j,i;(j,i)\in N_x} \frac{1}{d_{ij}^2}\epsilon_{ji}. \tag{12}$$

We remark that the SDP approach is extensible to a larger class of distance measure, formulations and cost functions. Other types of constraints are also easily integrated in to the SDP framework. For example, the use of angle information between sensors has been exploited to perform localization using SDP in [Biswas et al. 2005]. Another example is presented in more detail next where instead of single distance measures between a pair of sensors, the information we have between them are in the form of intervals or ranges, that is, the distance between the 2 sensors will lie in a certain confidence interval. Therefore, for multimodal scenarios with different types of distance and angle measures, the SDP approach can provide a global framework with which to attack the localization problem.

4.2 Interval Approach and its SDP relaxation

Our second approach to deal with noises is solving an SDP feasibility problem with upper and lower bound distance measures. Often, the mutual distance measures that can be obtained are in terms of interval instead of a single value, i.e. noisy distance measures may be represented in a confidence interval form of an upper bound $\bar{d}_{kj}$ and lower bound $\underline{d}_{kj}$ between a_k and x_j, or upper bound $\bar{d}_{ij}$ and lower bound $\underline{d}_{ij}$ between x_i and x_j.

Then, the quadratic model can be defined by:

$$\begin{array}{ll} \text{Find } X & \\ \text{subject to} & (\underline{d}_{ij})^2 \le \|x_i - x_j\|^2 \le (\bar{d}_{ij})^2, \ \forall (j,i) \in N_x, \\ & (\underline{d}_{kj})^2 \le \|a_k - x_j\|^2 \le (\bar{d}_{kj})^2, \ \forall (j,k) \in N_a. \end{array} \tag{13}$$

Therefore, in this formulation , we end up with inequality constraints instead of equality constraints. The SDP relaxation is

$$\begin{array}{ll} \text{find} & Z \in \Re^{(n+2)\times(n+2)} \\ & \underline{d}_{ij}^2 \le (0; e_i - e_j)^T Z (0; e_i - e_j) \le \bar{d}_{ji}^2, \forall (j,i) \in N_x \\ \text{s.t.} & \underline{d}_{kj}^2 \le (a_k; -e_j)^T Z (a_k; -e_j) \le (\bar{d}_{kj})^2, \forall (j,k) \in N_a \\ & Z \succeq 0 \end{array} \tag{14}$$

where $Z = \begin{pmatrix} I_2 & X \\ X^T & Y \end{pmatrix}$.

If the distance measurements are exactly correct (in the case of inequality constraints, this would imply that both the upper and lower bounds are the same and we are essentially solving an equality constrained problem) and the sensor network is uniquely localizable, then all three formulations (6), (11) and (14) solves the true sensor locations. If the bounds are not the same as is the case of noisy data, the model will return (due to the property of SDP interior-point algorithms) a central solution that is the "mean solution" of all feasible SDP solutions. This is desirable when distance noises exist, since a localization in the center of the possible location range is a unbiased estimate.

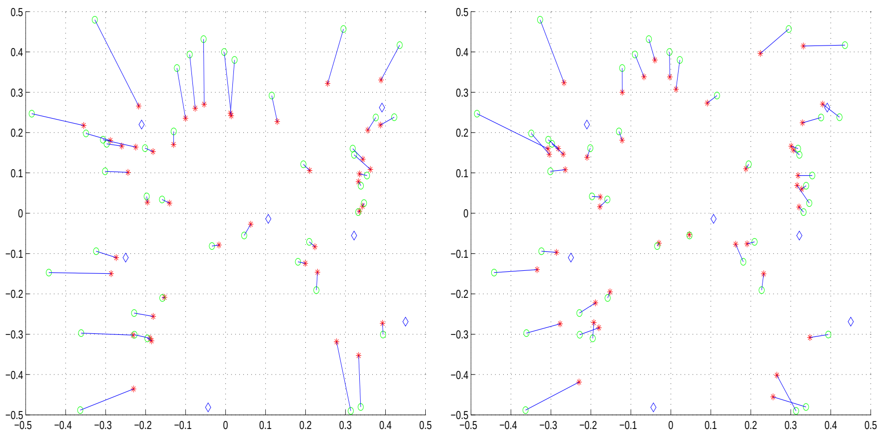
4.3 The High-Rank Property of SDP Relaxations

The solution of (11) and (14) may not be satisfactory when the distance data is extremely noisy. Due to the relaxation, the matrix rank constraint is removed and therefore the SDP solution may be lifted to a higher rank (higher dimensional space) in which the objective function is lower than it would be had the solution been constrained to be in the original dimension. In other words, the relaxation $Y \succeq X^T X$ is no longer exact. So the matrix $\begin{pmatrix} I_2 & X \\ X^T & Y \end{pmatrix}$ has a rank higher than 2.

Like in other SDP applications (e.g., combinatorial optimization), a main SDP research topic is how to round the higher-dimension (higher rank) SDP solution into a lower-dimension (desired low rank) solution. One way is to ignore the augmented dimensions and use the projection x^* as a suboptimal solution, which is the case in [Biswas and Ye 2004]. The other is the eigenvalue decomposition similar to multi-dimension scaling. An alternative way is to use randomized rounding or local search. One particular choice of local search is the gradient-based descent method that will be discussed in the next section.

We are currently also investigating the use of different objective functions with a regularization term, intelligent weighting of constraints and optimum anchor placement that will ensure a lower-dimension (lower-rank) solution from the SDP. This will eliminate the need for rounding and refinement procedures such as the gradient local search method. We use the following examples to describe the effect of noise on position estimation and to illustrate the effect of one of the above mentioned ideas.

EXAMPLE 2. *In Figure 4, the estimation results for a random network of 50 sensors obtained from setting* nf *to* 0.10*, the number of anchors to 7, and varying the radio range are shown. Even with 10% error measurement, the position estimation for the sensors near anchor nodes is still fairly accurate.*



(a) Radio range=0.30. (b) Radio-range=0.40.

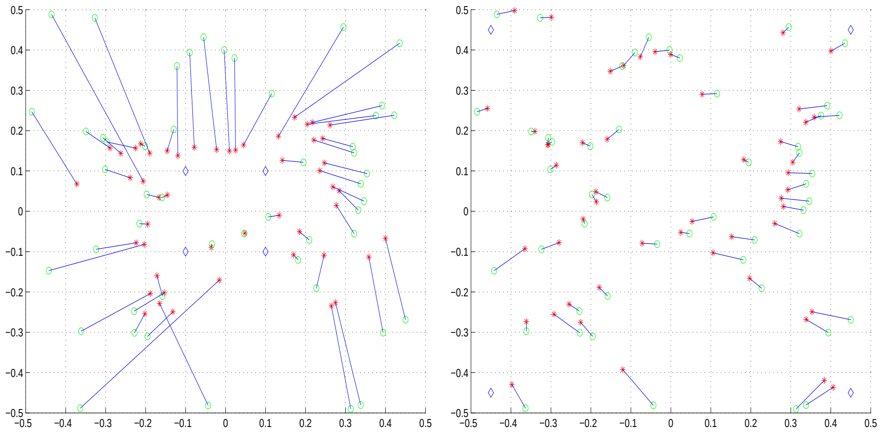
Fig. 4. Position estimations with 7 anchors, noisy factor=0.1, and various radio range.

One important point to note from the example in the noisy case is that the anchor placement is crucial. Placing the anchor points at the perimeter of the network greatly reduces the estimation error caused by noisy distance measures. In sharp contrast, points outside the convex hull of the anchor points tended to be estimated quite poorly. When the distance measures used are exact, the SDP relaxation finds the exact position estimates irrespective of the anchor placement. This is the major improvement that the SDP relaxation has over the previous convex optimization approaches [Doherty et al. 2001]. However, the introduction of noise in the distance measures diminishes the degree of improvement. We are currently investigating this in further detail in order to develop formulations that are less sensitive to anchor placement and to also develop efficient anchor placement strategies.

EXAMPLE 3. *An example of this occurrence is demonstrated in Figure 5 where for the same network of points and different anchor positions, the estimations are drastically different.*

5. A GRADIENT LOCAL SEARCH METHOD

The examples discussed in Figures 4 and 5 are uniquely localizable so that all the sensors should be located correctly if $nf = 0$. However, the figure shows that quite



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

Fig. 5. Position estimations with 4 anchors, noisy factor=0.1, Radio range=0.3 and different anchor placement

few of localizations are far from their actual locations. Since this network is uniquely localizable, the localization errors are completely due to the distance measurement errors. A higher dimensional localization is then found from the SDP relaxation, and after its projection on the 2-dimensional space, the estimation errors can be high.

As we mentioned before, one local search method to improve the SDP solution is moving every sensor location along the opposite of its gradient direction of the sum of error square function, which will for sure reduce error function value. Let us begin from (9) and simplicity, assume the case of multiplicative noise discussed in (12). The maximum likelihood estimation is an unconstrained optimization problem if all constraints are substituted into the objective function, say $f(X)$. Let the gradient be ∂f_{x_j} for a certain sensor x_j. It is important to notice that ∂f_{x_j} only relates to the sensors and anchors that are connected to (within the radio range) x_j, and they are local information, so that ∂f_{x_j} for every sensor x_j can be solved distributively.

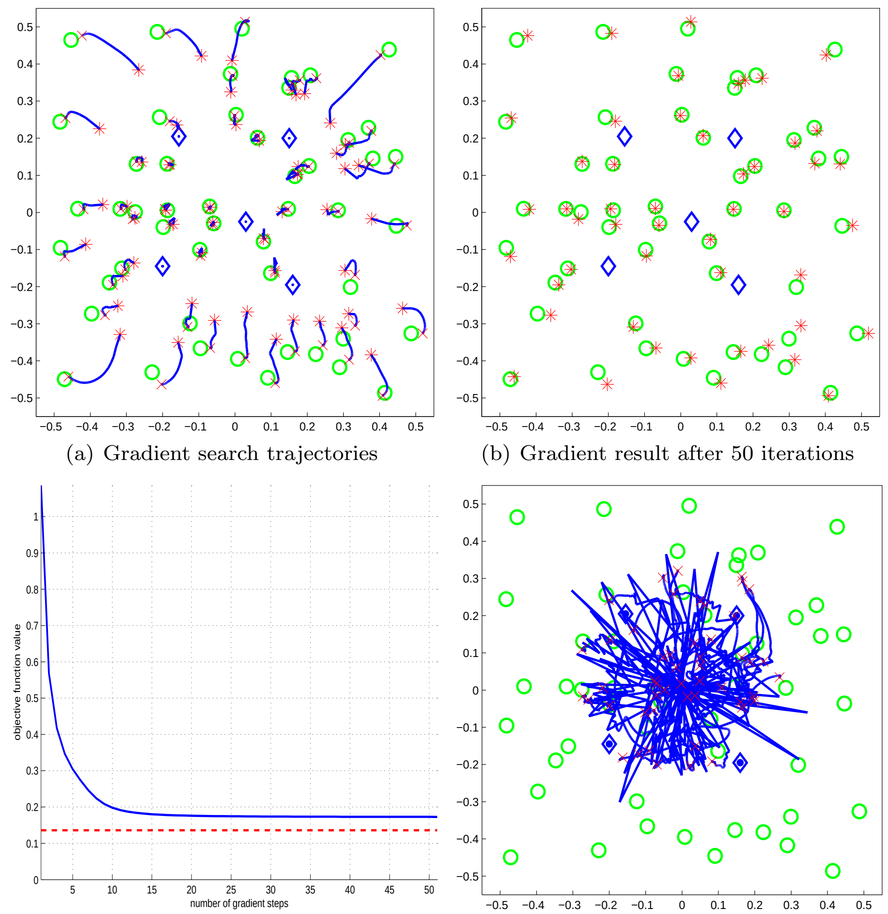
The following update rule is applied to improve the iterative solution:

$$x_j \longleftarrow x_j - \alpha \cdot \partial f_{x_j}^T \quad \text{for } j = 1 \text{ to } n, \tag{15}$$

where α is the step size. In each gradient step, the method calculates the gradient of each sensor and updates its location by this rule.

Example 4. *Consider the example presented in Figure 6. For a network of 45 sensors and 5 anchor nodes, a radio range of 0.3, 10% noise, the SDP result is indicated by the red stars in 6(a). The SDP localization is used as the initial solution. Figure 6(a) also shows the update trajectories in* 50 *iterations. It can be observed clearly that most sensors are moving toward their actual locations marked by the green circles. The final localizations after* 50 *gradient steps are plotted in Figure 6(b). which is a much more accurate localization.*

In Figure 6(c) the objective function values vs. number of gradient steps curve is



(a) Gradient search trajectories

(b) Gradient result after 50 iterations

(c) The sum of error squares vs. number of gradient search steps

(d) Result of gradient search with random initial starting point

Fig. 6. Refinement through gradient method for 50 anchor network, 10 % noise

plotted in a solid line. One can see that in the first 15 *steps the objective function value drops rapidly, and then the trajectory tends to be flat. This demonstrates that the gradient search method does improve the overall localization result. A natural question is how good the new localization is. To answer this question we need a lower bound of the objective function value. One trivial lower bound of the objective function value is* 0*; but a better one is the SDP relaxation objective value, since the SDP problem is a relaxation of the original 2-dimensional localization problem. In this case, the SDP objective value is about* 0.136*, plotted in Figure 6(c) in a dashed line, and the gradient search method finds a 2-dimensional localization with objective function value about* 0.173*. Thus an error gap* 0.037 *of the suboptimality is obtained, which is less than 30% of the error lower bound value*

The gradient-based descent method is a local search method and can be proved to find the global optimal solution only for convex optimization. The ad hoc sensor *network localization problem is NOT a convex optimization problem. Hence a pure gradient search method with randomly initial localization obviously does not work. It is quite important that the SDP relaxation finds a well-positioned localization, even is not accurate, which can be used as a good initial solution to start the local search.*

To see this, another experiment is performed. We set the initial sensor locations at the origin and update them by the same rule (15). The updated trajectories are shown in Figure 6(d). None of these sensors converges to their actual positions. So we can see that the use of the SDP solution as the initial localization makes all the difference.

The idea of local refinement after finding an approximate solution is not new. Many other localization algorithms (see [Savarese et al. 2002] and [Shang et al. 2004]) perform such a post refinement step after obtaining a rough global map. It should be also pointed out, that using majorization techniques developed in [Groenen 1993], the cost function can be minimized in a manner that the function value drops at every iteration. This idea is used in [Costa et al. 2005] for the sensor network localization problem. Therefore the problem of non-convergence can be avoided. But this does not prevent the method from converging to a possible local minimizer, even though advanced search techniques may reduce the extent of this possibility.

Although this might imply that SDP is an expensive initialization technique to avoid non-convergence when this problem can be tackled by other means, we argue that the gradient descent technique is merely an optional refinement technique in our case. As our results will demonstrate, the original formulations yield estimates that are already quite accurate and the gradient method offers minor improvement. In fact, our localization results using the interval formulation with upper and lower bounds does not involve the post processing gradient step and in spite of this, for high noise, it delivers better estimation accuracy than the MLE formulation followed by post processing by the gradient descent method; see Section 7. Furthermore, the SDP solution establishes a lower bound on the error function value so that the final localization quality can be certified for every instance of the problem; see the final gap between the blue curve and red dash line in Figure 6(c). No previous methods were able to generate simultaneously with the solution this kind of problem specific "proof of quality" . Note that unlike a Cramer Rao lower bound as discussed in [Patwari and III 2002], the SDP lower bound does not need to make any assumptions on the noise model or distribution.

The effects of the gradient method on estimation accuracy seem more pronounced when the anchors are in the interior. As mentioned before, all estimates are erroneously pulled towards the origin in such cases. This is indeed the case in the example presented in Figure 6. We believe that the addition of regularization terms in the objective function that penalize such crowding at the center can improve the performance substantially. Preliminary results have been promising and a complete explanation will be reported elsewhere in the future.

6. A DISTRIBUTED SDP METHOD

Unfortunately, the existing SDP solvers have very poor scalability. They can only handle SDP problems with the dimension and the number of constraints up to few thousands, where in the SDP sensor localization model the number of constraints is in the order of $O(n^2)$, where n is the number of sensors. The difficulty is that each iteration of interior-point algorithm SDP solvers needs to factorize and solve a dense matrix linear system whose dimension is the number of constraints. While we could solve localization problems with 50 sensors in few seconds, we have tried to use several off-the-shell codes to solve localization problems with 200 sensors and often these codes quit either due to memory shortage or having reached the maximum computation time.

We describe an iterative distributed SDP computation scheme, first demonstrated in [Biswas and Ye 2003] to overcome this difficulty. We first partition the anchors into many clusters according to their physical positions, and assign some sensors into these clusters if a sensor has a direct connection to one of the anchors. We then solve semidefinite programs *independently* at each cluster, and fix those sensors' positions which have high accuracy measures according the SDP computation. These positioned sensors become 'ghost anchors' and are used to decide the remaining un-positioned sensors. The distributed scheme then repeats.

The distributed scheme is highly scalable and we have solved randomly generated sensor networks of $4,000$ sensors in few minutes for a sequential implementation (that is, the cluster SDP problems are solved sequentially on a single processor), while the solution quality remains as good as that of using the centralized method for solving small networks. A parallel implementation for solving the SDP at each cluster would be even more efficient.

We remark that our distributed or decomposed computation scheme should be applicable to solving other Euclidean geometry problems where points are locally connected.

A round of the distributed computation method is straightforward and intuitive:

(1) Partition the anchors into a number of clusters according to their geographical positions. In our implementation, we partition the entire sensor area into a number of equal-sized squares and those anchors in a same square form a regional cluster.

(2) Each (unpositioned) sensor sees if it has a direct connection to an anchor (within the communication range to an anchor). If it does, it becomes an unknown sensor point in the cluster to which the anchor belongs. Note that a sensor may be assigned into multiple clusters and some sensors are not assigned into any cluster.

(3) For each cluster of anchors and unknown sensors, formulate the error minimization problem for that cluster, and solve the resulting SDP model if the number of anchors is more than 2. Typically, each cluster has less than 100 sensors and the model can be solved efficiently.

(4) After solving each SDP model, check the individual trace error(8) for each unknown sensor in the model. If it is below a predetermined small tolerance, label the sensor as *positioned* and its estimation $\bar{x}_j$ becomes an " anchor". If a sensor

is assigned in multiple clusters, we choose the $\bar{x}_j$ that has the smallest individual trace. This is done so as to choose the best estimation of the particular sensor from the estimations provided by solving the different clusters.

(5) Consider positioned sensors as anchors and return to Step 1 to start the next round of estimation.

Note that the solution of the SDP problem in each cluster can be carried out at the cluster level so that the computation is highly distributive. The only information that needs to be passed among the neighboring clusters is which of the unknown sensors become positioned after a round of SDP solutions.

In solving the SDP model for each cluster, even if the number of sensors is below 100, the total number of constraints could be in the range of thousands. However, many of those "bounding away" constraints, i.e., the constraints between two remote points, are inactive or redundant at the optimal solution. Therefore, we adapt an iterative active constraint generation method. First, we solve the problem including only partial equality constraints and completely ignoring the bounding-away inequality constraints to obtain a solution. Secondly we verify the equality and inequality constraints and add those violated at the current solution into the model, and then resolve it with a "warm-start" solution. We can repeat this process until all of the constraints are satisfied. Typically, only about $O(n+m)$ constraints are active at the final solution so that the total number of constraints in the model can be controlled at $O(n+m)$.

However, in the case with measurement noises, this method needs to be modified. Firstly, the trace error(8) may be no longer a reliable error measure. Instead, we compute the local error measure for each point x_j,

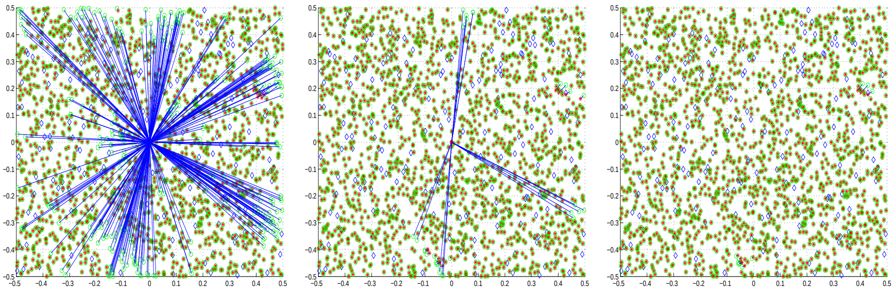
$$LDME_j = \frac{\sum_{i \in N_x^j} (\|x_i - x_j\| - d_{ij})^2 + \sum_{k \in N_a^j} (\|a_k - x_j\| - d_{kj})^2}{|N_x^j| + |N_a^j|}.$$

where $k \in N_a^k$ if $(j,k) \in N_a$ and $i \in N_x^i$ if $(j,i) \in N_x$, that is, the indices of the anchors and unknown points which have distance measures with point x_j.

If the local error measure is below a certain threshold for a point, it is updated to anchor status in the next iteration. Therefore in the noisy case, there are more iterations, it takes much longer time to converge and the results are inaccurate for a large number of points as can be expected. Therefore, once again, we employ the gradient method once a good starting point for local optimization is obtained.

In the gradient based approach, we stop the SDP iterations after a fixed number of steps. Then, the solution produced by the decomposed SDP strategy serves as the starting solution for the gradient-based search method to minimize the objective function of the entire sensor network. Since the computational complexity of the gradient vector is very low so that the gradient-based method, although applied to the entire network without decomposition, can be completed extremely fast.

EXAMPLE 5. *For the distributed case, simulations were performed on networks of* 2,000 *to* 4,000 *sensor points. The first simulation is carried out for solving a network localization with* 2,000 *sensors, where the iterative distributed SDP method terminates in three rounds, see Figure 7 . When a sensor is not positioned, its estimation is typically at the origin. In this simulation, the entire sensor region is* *partitioned into* 7×7 *equal-sized squares, that is,* 49 *clusters, and the radio range is set at* $.06$*.*



(a) First iteration (b) Second iteration (c) Final iteration

Fig. 7. Position estimations for the 2,000 node sensor network, 200 anchors, noisy-factor=0, radio-range=.06, and the number of clusters=49.

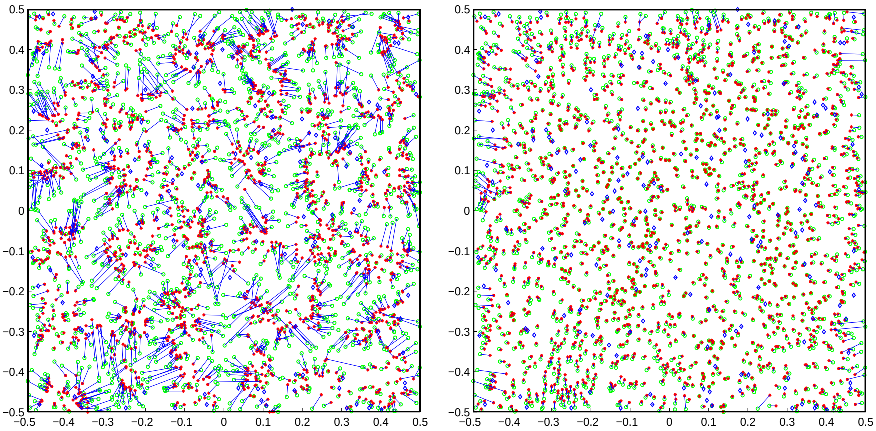
As can be seen from the Figure 7 , it is usually the outlying sensors at the boundary or the sensors which do not have many anchors within the radio range that are not estimated in the initial stages of the method. Gradually, as the number of well estimated sensors or 'ghost' anchors grows, more and more of these points are estimated. The total solution time for the three round computation, excluding the computation of $\hat{d}_{ji}$*, is about two minutes.*

It is interesting to note that the erroneous points are concentrated within particular regions. This clearly indicates that the clustering approach prevents the propagation of errors to other clusters. This is because the estimated points within a cluster are used to estimate other points only if their estimation error is below a threshold.

The second simulation solves a network localization with $4,000$ sensors, where the iterative distributed SDP method terminates in five rounds. In this simulation, the entire sensor region is partitioned into 10×10 equal-sized squares, that is, 100 clusters, and the radio range is set at .035. The total solution time for the five round computation, excluding computing $\hat{d}_{ji}$, is about four minutes.

Example 6. *Now consider a localization problem with 1800 sensors, 200 anchors, radio range 0.05 but also introduce a noise factor of 0.1. The sensor network is decomposed into 36 equal-sized domains. Figure 8(a) shows the SDP solution after 3 iterations. One can see that the SDP algorithm fails to find accurate solution in certain small areas. But after 50 gradient-based search steps, the final localization, shown in Figure 8(b), is improved.*

Our program is implemented with MATLAB and it uses SEDUMI[Sturm 1999] or DSDP2[Benson et al. 2000] as the SDP solver. It costs 165 seconds of SEDUMI or 63 seconds of DSDP2 to get the SDP localization in Figure 8(a) on a PC. Then, after 50 gradient search steps (costs less than 20 seconds), the objective function is reduced from 12.81 to 0.230. It can be seen from Figure 8(b) that most of the



(a) SDP solution after 5 iterations (b) Gradient solutions after 50 iterations

sensors are located very close to their true positions, although few of them, most of which are close to the boundary of the network, are solved inaccurately.

The current clustering approach assumes that the anchor nodes are more or less uniformly distributed over the entire space. So by dividing the entire space into smaller sized square clusters, the number of anchors in each cluster is also more or less the same.

However this may or may not be the case in a real scenario. A better approach would be to create clusters more intelligently based on local connectivity information. Keeping this in mind, we try and find for each sensor its immediate neighborhood, that is, points within radio range of it. It can be said that such points are within one hop of each other. Higher degrees of connectivity between different points can also be evaluated by calculating the minimum number of hops between the 2 points. Using the hop information, we propose to construct clusters which are not necessarily of any particular geometric configuration but are defined by its connectivity with neighborhood points. Such clusters would yield much more efficient SDP models and faster and more accurate estimations. These ideas have been discussed in further detail in [Biswas et al. 2005].

Our distributed algorithm has also been used to develop adaptive rule-based procedures for performing cluster formation and localization for very large sensor networks. Significant improvements have been reported in [Carter et al. 2005],[Jin 2005] with regard to the accuracy and scalability of the original distributed algorithm.

7. EXPERIMENTAL RESULTS

Simulations were performed on networks of 50 nodes randomly placed in a square region of size 1×1 centered at the origin. The distances between the nodes was calculated. If the distance between 2 nodes was less than a specified radio range R between $[0, 1]$, a random error was added to it.

$$d_{ji} = \tilde{d}_{ji} \cdot (1 + \mathcal{N}(0,1) * nf),$$

where $\tilde{d}_{ji}$ is the actual distance between the 2 nodes, nf (noise factor) is a given number between $[0,1]$ that is used to control the amount of noise variance and $N(0,1)$ is a standard normal random variable.

Upper and lower bounds for the distance measures were generated in the following manner,

$$\underline{d}_{ji} = \tilde{d}_{ji} \cdot (1 - |\mathcal{N}(0,1) * nf|) \quad \bar{d}_{ji} = \tilde{d}_{ji} \cdot (1 + |\mathcal{N}(0,1) * nf|),$$

which gives more or less the 68% confidence interval for the interval SDP method (14).

The average estimation error was defined by

$$\frac{1}{n} \cdot \sum_{j=1}^{n} \|\bar{x}_j - \tilde{x}_j\| \tag{16}$$

respectively, where n is the total number of unknown points, x_j comes from the SDP solution and $\tilde{x}_j$ is the true position of the jth node. The error is also expressed in terms of $\%R$. In that case, it is the average estimation error described above multiplied by $100/R$.

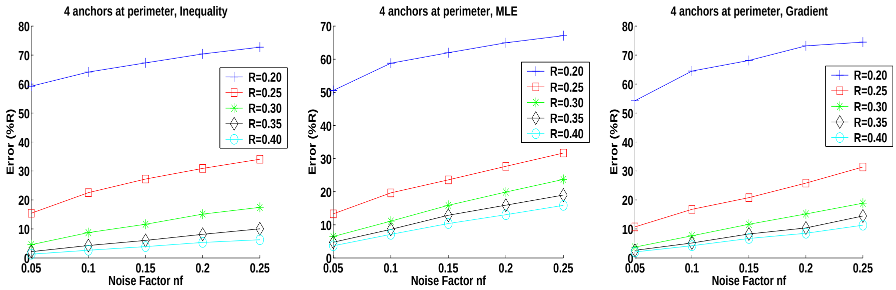
Connectivity indicates how many of the nodes, on average, are within the radio range of a node. Therefore, by varying the radio range, the connectivity and the number of available distance measures can be varied. The average case performance of the algorithm with varying connectivity, noise variance in distance measurements and number of anchors was assessed by running extensive simulations on randomly generated networks. For each configuration of a particular radio range, number of anchors and noise variance, the algorithms were run on 25 independently generated networks of points and the average error computed.

The effect of anchor placement in the perimeter of the network as opposed to the interior was discussed in Section 4.3 and illustrated in Figure 5. For the first part of our simulations, we place four anchors,one each on the corners of a square network. The estimation error is reduced significantly by doing this. We argue that this assumption that anchors are placed on the perimeter of the network is reasonable in a range of position estimation scenarios since during deployment of a network, we should be aware of the overall area in which it it to be deployed. The placement of powerful anchor nodes on the perimeter of this area is a feasible assumption. Other anchors may be deployed randomly in the interior like the rest of the nodes.

For all the graphs shown below, the notation R will be used for radio range, nf for noise factor(a number between 0 and 1 that is used to control the noise variance), and m for the number of anchors. The results of 3 different methods are compared, that is, using inequality or interval formulation (14), the maximum likelihood estimation formulation (11), and the gradient local improvement method (15) where the MLE solution is used as the starting point for the gradient method for all cases considered.

7.1 Effect of Varying Radio Range and Noise Factor

Figure 8 shows the variation of average estimation error(normalized by the radio range R) when the noise in the angle measurements increases and with different radio ranges. Figure 9 shows the variation of estimation error when the radio range is increased and with different measurement noises. Also the error is calculated as a percentage of the radio range. The absolute error is also presented in cases where deemed necessary. Figure 10 shows the absolute error variation as calculated from 16 with varying radio range. 4 anchors are placed near the corners of the square network. There are no more anchors placed in the interior.



(c) Interval Constraints (d) Max. Likelihood (e) Gradient

Fig. 8. Variation of estimation error with measurement noise, 4 anchors,

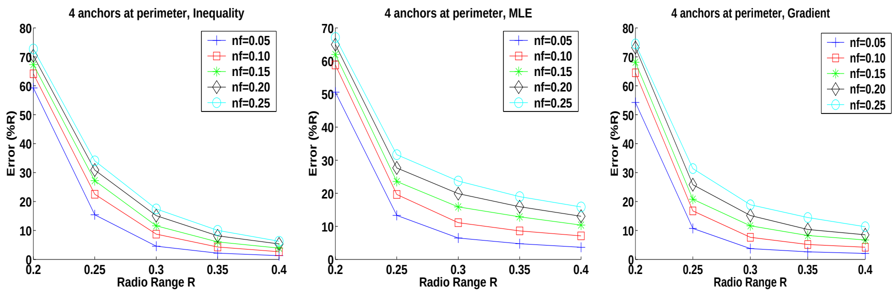


Fig. 9. Variation of estimation error with radio range, 4 anchors

When the radio range R is too low, there is not enough information between the points for the estimation to be effective (60-70$\%R$). The error decreases consistently for all methods as the radio range and consequently, the connectivity is increased. This holds for the high measurement noise cases as well. For $nf = 0.2$ and $R = 0.3$, the error is about $10 - 20\%R$ for all methods.

Observing the results, many insights into the relative advantages of the different algorithms in different scenarios can be obtained. For lower radio ranges and low measurement noise, the MLE and gradient methods together outperform the interval method. On the other hand, the inequality method is more accurate and less sensitive to noise than the other 2 methods together when the connectivity and measurement noise are high. The gradient method also offers more improvement over the MLE method for high noise when the connectivity is high. This is an

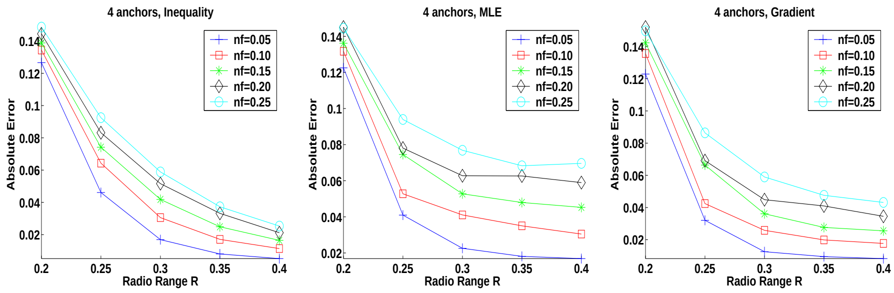


Fig. 10. Variation of absolute estimation error with radio range

encouraging fact since distance measurements maybe available in terms of intervals or ranges than exact values. This and the improvement offered by the gradient method over the MLE method for high noise and connectivity cases could also motivate the study of gradient type methods that solve local improvement problems using inequality constraints. The choice of a suitable objective function that improves estimation accuracy and allows efficient computation of gradients using interval constraints is a topic of further research.

Overall, this indicates that if the measurement technology is accurate over lower ranges, the MLE and gradient methods might be be used. But if the technology provides distance ranges over longer distances, the interval method should be used. This also makes perfect intuitive sense since a small subset of accurate distances corresponds closely to solving a set of equality constraints. When there are too many such constraints and there is also high uncertainty in them, the results are poorer. If there are many uncertain constraints, it makes more sense to solve for a set of interval or inequality constraints. There are enough constraints to limit the solution space and the sensitivity to erroneous measurements is less severe.

7.2 Effect of Number of Anchors

Figure 11 shows the variation of estimation error by increasing the radio range while varying the number of anchors from 4 to 8. 4 anchors were placed at the perimeter and the rest in the interior, and the nf was fixed at 0.1. For the same networks, Figure 12 shows the variation of estimation error by increasing the measurement noise(nf) while varying the number of anchors from 4 to 8, where the radio range is fixed at 0.3. We remind the reader that the number of anchors is denoted by m.

The improvement from having a higher number of anchors diminishes with higher radio range. But when the measurement noise is high, the presence of more anchors tends to reduce the estimation error.

7.3 Effect of Random Anchor Placement

In order to demonstrate the effect that anchor placement has over the estimation result, simulation results are also presented for random anchor placement. That is, we no longer operate under the assumption that anchors are placed in the perimeter. These cases are also instructive because they behave differently with varying noise, radio range and number of anchors.

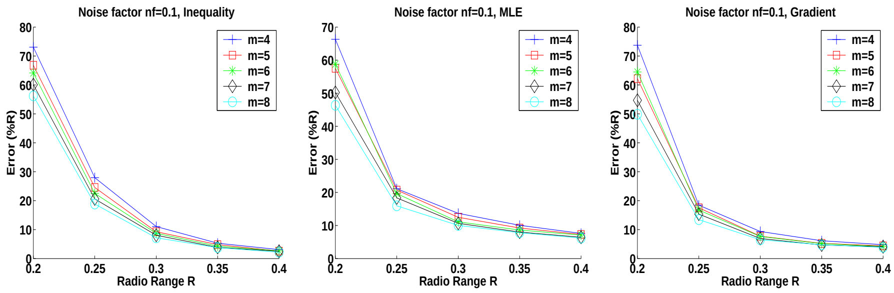


Fig. 11. Variation of estimation error with more anchors and varying radio range

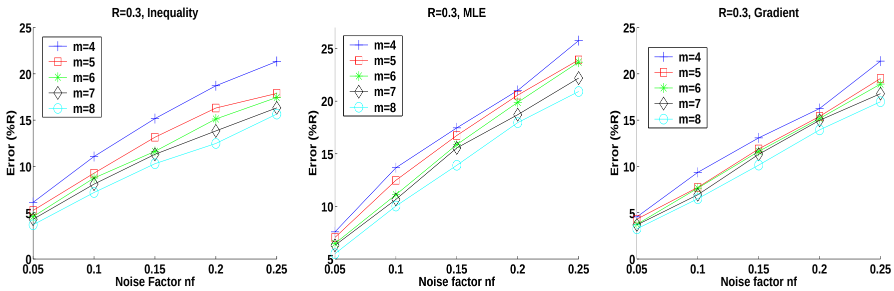
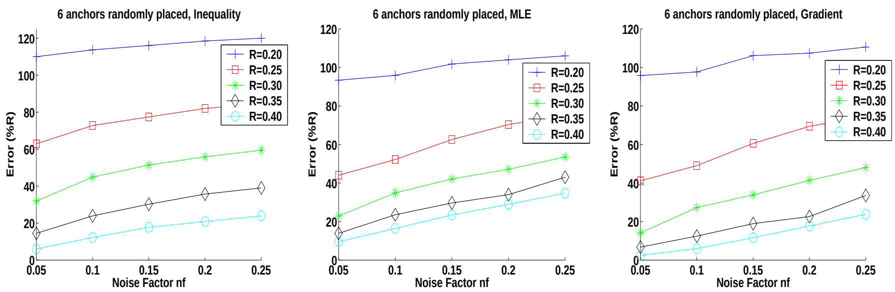


Fig. 12. Variation of estimation error with more anchors and varying measurement noise

—**Varying Radio Range and Noise Factor**
Figure 13 shows the variation of average estimation error(normalized by the radio range R) when the noise in the distance measurements increases and with different radio ranges. Figure 14 shows the variation of estimation error when the radio range is increased and with different measurement noises.



(a) Inequality Constraints (b) Max. Likelihood (c) Gradient

Fig. 13. Variation of estimation error with measurement noise, 6 anchors randomly placed

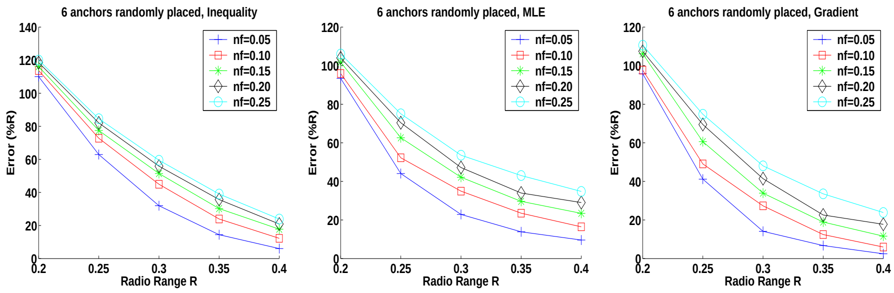


Fig. 14. Variation of estimation error with radio range, 6 anchors randomly placed

Clearly, in the case of random anchor placement, the estimation error is far higher for a particular radio range R and measurement noise nf. The improvement in estimation with greater radio range/connectivity is more pronounced. This indicates that even with random anchor placement, the problem of bad estimation for outlying points when the measurement noise can be mitigated by having a high connectivity. For example, even for $nf = 0.25$, when the radio range is 0.4, the estimation error is only about 20%R.

—**Varying Number of anchors**
Figure 15 shows the variation of estimation error by increasing the radio range while varying the number of anchors from 6 to 10. The nf is fixed at 0.1. For the same networks, Figure 16 shows the variation of estimation error by increasing the measurement noise(nf) while varying the number of anchors from 6 to 10. The radio range is fixed at 0.3. Again, in these figures, the number of anchors is denoted by m.

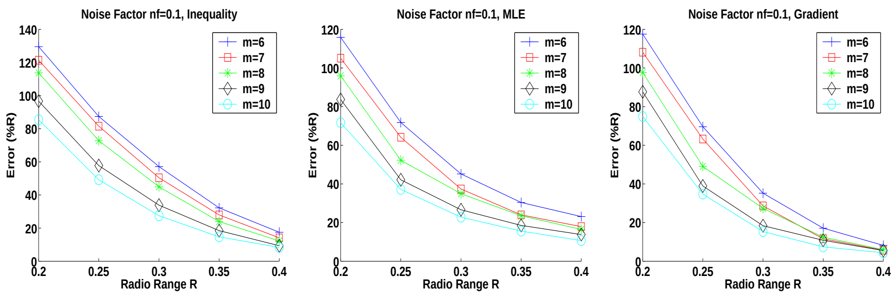


Fig. 15. Variation of estimation error with more anchors and varying radio range, Random anchor placement

The graphs show that in contrast to the case when 4 anchors are placed on the perimeter, adding more anchors in the random placement case offers a distinct advantage in terms of reducing the estimation error. In fact, selecting the optimum number of anchors and intelligent anchor placement are future research topics that merit deeper investigation.

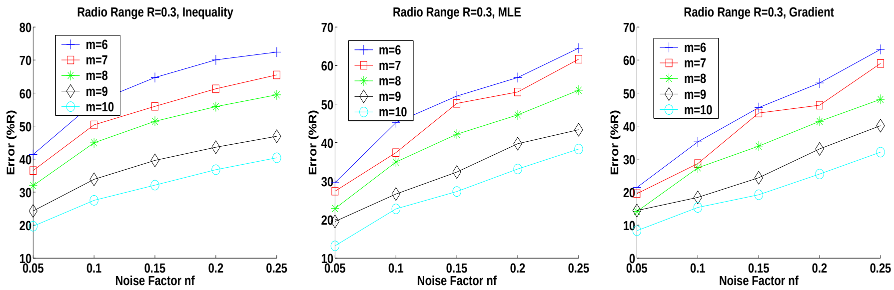


Fig. 16. Variation of estimation error with more anchors and varying measurement noise, Random anchor placement

7.4 Computational Effort

Our simulation program is implemented with MATLAB and it uses SEDUMI[Sturm 1999] as the SDP solver. The simulations were performed on a Pentium IV 2.0 GHz and 512 MB RAM PC. The computational results presented here were generated using the interior-point algorithm SDP solvers SeDuMi of [Sturm 1999] and DSDP2.0 of [Benson et al. 2000] with their interfaces to MATLAB on a Pentium 1.2 GHz and 500 MB RAM PC. DSDP is faster due to the fact that the data structure of the problem is more suitable for DSDP. However SeDuMi is often more accurate and robust.

—**Number of constraints and solution time**
The number of constraints and solution time was analyzed. Figure 17 shows how the number of constraints grows with the number of points for the case considered above for the interval formulation. Figure 18 illustrates how the solution time increases as the number of points in the network increases. Note that the gradient solution time just takes into account the time taken for the local gradient optimization. The MLE SDP solution needs to be computed before the gradient method can be applied. Therefore, the total time would be the combined time for the MLE and gradient methods. However, in our graphs, we do not show the combined times. Instead, times for each of the individual steps are shown in order to also compare the relative times taken by each of the steps. It should be noted that a smaller radio range will be required for denser networks. Since the networks get denser with more points, the connectivity increases as well. However, beyond a certain value, increasing the connectivity only adds more redundant constraints to the problem and increases the computational effort required. So we also progressively decrease the radio range with the number of points so as to keep connectivity within a certain range. The radio range for 40 points is set at 0.4. For each increase of 20 points, the radio range is decreased by 0.05.
From the results, it can be observed that the number of constraints and solution time is highly dependent not only on the number of points but upon the radio range selected as well. For example, the connectivity for 100 points and a radio range 0.25 is higher than that for 120 points and radio range 0.2, therefore number of constraints and solution time and are also higher. But overall, it can be said

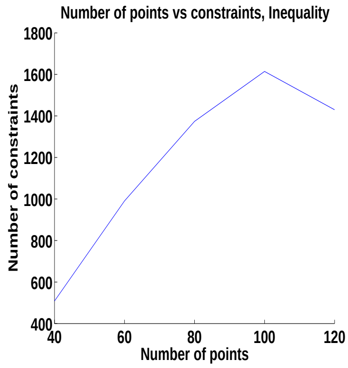


Fig. 17. Number of constraints vs. Number of points

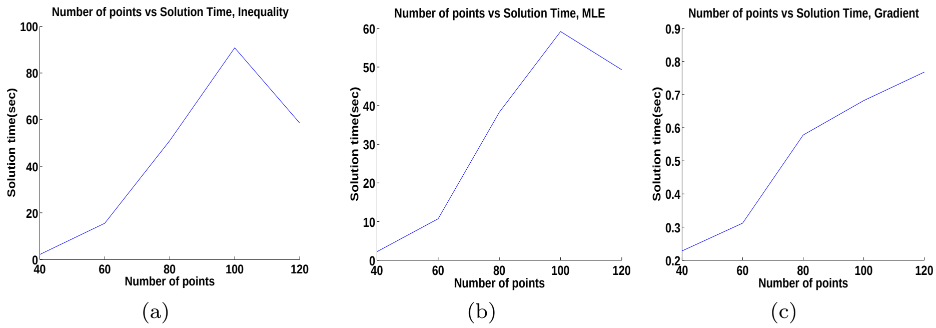
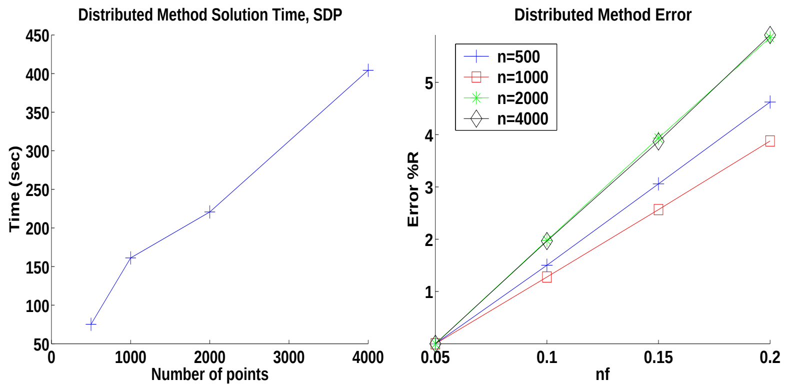


Fig. 18. Solution Time vs. Number of points

that the number of constraints does not scale with $O(n^2)$ but more typically with $O(n)$. Therefore as discussed in Section 3.3, the computational effort is more typically $O(n^3)$ as opposed to the worst case $O(n^6)$. Also, the inequality method is always more time-consuming simply because for every pair of points, instead of one constraint, we have both upper and a lower bound constraints. But we believe that the inequality method time can be reduced further by using a combined semidefinite programming and linear programming formulation.

—**Distributed Method**
The solution time and error for the distributed method for larger networks is presented in Figure 19(a) and 19(b) respectively.
As can be seen from the results, a combination of the SDP and gradient method is particularly useful in these cases in reducing the error and is also very computationally efficient. Even for a 4000 point network, the combined solution time is about 400 seconds for a sequential implementation. The estimation error is also extremely low because the networks have a high connectivity and this approach exploits this fact while keeping the solution time low. In fact, the 1000 point network which has a radio range of 0.1 has the highest connectivity, so the error is the least. For the 2000 point network with radio range 0.05, the connectivity



(a) (b)

Fig. 19. Distributed Method Simulations

is lower, so the relative average estimation error is higher. But a higher connectivity is also reflected in the increase in solution time from going to the 500 to 1000 points. This case has a sharper rate of increase than the other transitions.

7.5 Comparison with existing methods

Partial comparison with existing methods is also presented. For this purpose, Multidimensional Scaling, another centralized approach is used. The comparisons are made against the simulation results reported in [Shang et al. 2004] with networks of 200 points uniformly distributed in a square area of $10r \times 10r$ and the radio range varied from $1.25r$ to $2.5r$. Only 5% measurement noise, that is, $nf = 0.05$, was reported in [Shang et al. 2004]. For detailed results obtained by MDS, refer to the experimental results section in [Shang et al. 2004] for random uniform networks. While their results are also presented for localization using only simple connectivity data, our comparisons will only be against the results presented for localization using distance data as well. The results from the paper show that MDS outperforms other methods such as [Niculescu and Nath 2001] and so is a good choice for comparison.

Figure 20 shows the results for the above mentioned setup as reported in [Shang et al. 2004] for 4 different MDS based methods(MDS-MAP(C), MDS-MAP(C,R), MDS-MAP(P), MDS-MAP(P,R)) which differ in whether a central method is used or alternatively broken into smaller patches and also whether a post processing refinement step is applied or not. The radio range and number of anchors is varied.

—**Random anchor Placement**
Figure 21 shows the results of the SDP based relaxation for the same setup when the anchors are randomly placed.
The results show that in the random anchor placement case, the error is quite high when the radio range is low, where our estimation is inferior to the MDS method. It is interesting to observe again that the impact of having a different number of anchors is strong. For 6-10 anchors, the anchors cover more of the square space and so the estimation is better. Also, for higher radio ranges, the

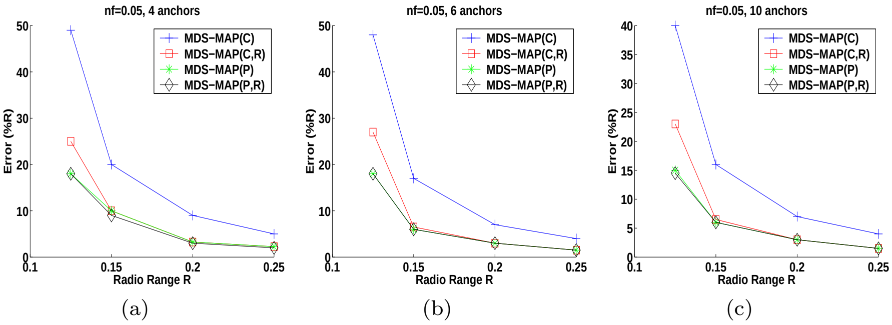


Fig. 20. MDS results, Estimation error for 200 point networks with varying radio range, random anchor placement

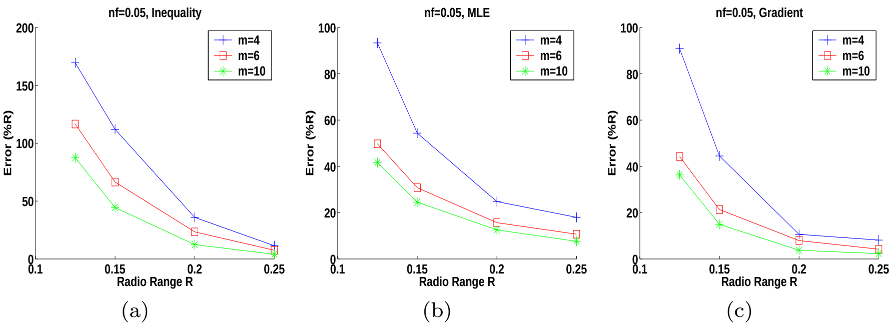


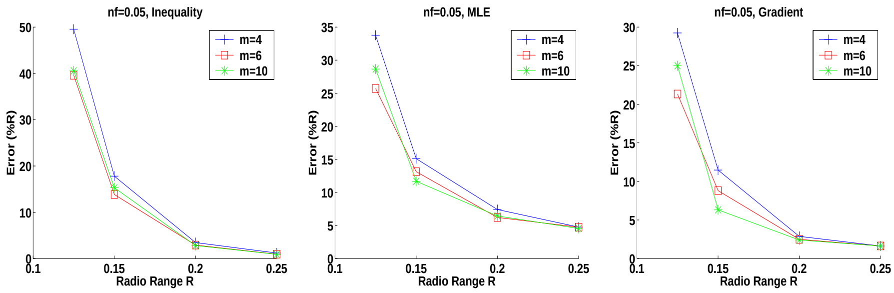
Fig. 21. Comparison with MDS, Estimation error for 200 point networks with varying radio range, random anchor placement

error reduces quite substantially so that the estimation is only a few $2 - 3\%R$ more than the reported MDS result.

—**Anchors placed in the perimeter**

Figure 22 shows the results obtained by SDP for the case when the setup is the same except that 4 anchors are placed at the perimeter. Remaining anchors are placed inside randomly.

When 4 anchors are placed in the perimeter, the estimation once again improves dramatically for our SDP methods, especially for lower radio ranges. The high radio range performance is comparable to the performance of the MDS methods (between $0 - 5\%R$). Only in the lower radio range regime, the MDS estimation error reportedly drops to about $10 - 20\%R$ after a local refinement, as opposed to $20 - 30\%R$ for our SDP based methods. Therefore, the refined MDS methods appear to have slightly better performance for uniform networks with low radio range and low noisy factor. Overall, however, the performances of the competing techniques are quite similar for this particular problem. Given that the estimation error of our methods grows slowly as the noise factor increases (see Figure 13), it is our belief that with development of more suitable objective functions,



(a) (b) (c)

Fig. 22. Comparison with MDS, Estimation error for 200 point networks with varying radio range, 4 anchors in perimeter

intelligent anchor placement schemes and better ways of rounding the higher dimension solution to the 2 dimensional space, the SDP methods could be made to outperform existing localization methods for different kinds of sensor networks over a wider range of noises.

8. CONCLUDING REMARKS

The paper presents an SDP relaxation based method to solve the sensor network localization problem based on incomplete and inaccurate distance information. Formulations that account for noisy distance data and distributed processing techniques are also developed in order to make the algorithm more robust and scalable. Experimental results are presented to show that the algorithm performs well even in highly noisy environments.

More theoretical analysis of the dependence of the algorithm on the anchor placement, measurement noise and the objective function is required in order to develop optimum anchor placement schemes, provide tighter bounds on the estimation error and improve estimation accuracy.

The intelligent inclusion of angle constraints (possibly obtained using Angle of Arrival methods) can yield much better solutions using fewer anchors and less distance information. Formulations that extend the model for a combination of distance and angle information as well as pure angle information are being developed. Some results are presented in [Biswas et al. 2005].

The overall distance geometry model and SDP relaxation are applicable to any other problem in distance geometry with mutual distance and angle information between two points. Applications to problems such as molecule conformation and Euclidean ball packing are being explored. The case of molecule conformation in 3-D has been dealt with in detail in [Biswas et al. 2005].

REFERENCES

Alfakih, A. Y., Khandani, A., and Wolkowicz, H. 1999. Solving euclidean distance matrix completion problems via semidefinite programming. *Comput. Optim. Appl. 12,* 1-3, 13–30.

Benson, S. J., Ye, Y., and Zhang, X. 2000. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization 10(2)*, 443–461.

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

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

Biswas, P. and Ye, Y. 2003. A distributed method for solving semidefinite programs arising from ad hoc wireless sensor network localization. Tech. rep., Dept of Management Science and Engineering, Stanford University. October.

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

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

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

Carter, M., Jin, H., Saunders, M., and Ye, Y. 2005. Spaseloc: An adaptable subproblem algorithm for scalable wireless network localization. *Submitted to the SIAM J. on Optimization*.

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

Cox, T. and Cox, M. A. A. 2001. *Multidimensional Scaling.* Chapman Hall/CRC, London.

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

Ganesan, D., Krishnamachari, B., Woo, A., Culler, D., Estrin, D., and Wicker, S. 2002. An empirical study of epidemic algorithms in large scale multihop wireless networks.

Groenen, P. 1993. *The majorization approach to multidimensional scaling: some problems and extensions*. DSWO Press, Leiden University.

Hightower, J. and Borriello, G. 2001. Location systems for ubiquitous computing. *Computer 34,* 8, 57–66.

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

Jin, H. H. 2005. Scalable sensor localization algorithms for wireless sensor networks. Ph.D. thesis, Department of Mechanical and Industrial Engineering, University of Toronto.

Laurent, M. 2001. Matrix completion problems. *The Encyclopedia of Optimization 3*, 221–229.

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

Moré, J. and Wu, Z. 1997. Global continuation for distance geometry problems. *SIAM Journal on Optimization. 7*, 814–836.

Niculescu, D. and Nath, B. 2001. Ad hoc positioning system (APS). In *GLOBECOM (1)*. 2926–2931.

Niculescu, D. and Nath, B. 2003. Ad hoc positioning system (APS) using AoA. In *INFOCOM*. San Francisco, CA.

Patwari, N. and III, A. O. H. 2002. Location estimation accuracy in wireless sensor networks. In *Proceedings of Asilomar Conference on Signals and Systems*. Pacific Grove, CA.

Priyantha, N. B., Miu, A. K. L., Balakrishnan, H., and Teller, S. 2001. The cricket compass for context-aware mobile applications. In *Mobile Computing and Networking*. 1–14.

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

Savvides, A., Han, C.-C., and Strivastava, M. B. 2001. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Mobile Computing and Networking*. 166–179.

Savvides, A., Park, H., and Srivastava, M. B. 2002. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM Press, 112–121.

Shang, Y. and Ruml, W. 2004. Improved MDS-based localization. In *Proceedings of the 23rd Conference of the IEEE Communicatons Society (Infocom 2004)*.

Shang, Y., Ruml, W., Zhang, Y., and Fromherz, M. P. 2003. Localization from mere connectivity. In *Proceedings of the fourth ACM international symposium on Mobile ad hoc networking and computing*. ACM Press, 201–212.

Shang, Y., Ruml, W., Zhang, Y., and Fromherz, M. P. J. 2004. Localization from connectivity in sensor networks. *IEEE Transactions on Parallel and Distributed Systems 15,* 11, 961–974.

So, A. M.-C. and Ye, Y. 2004. Theory of semidefinite programming relaxation for sensor network localization. Tech. rep., Dept of Management Science and Engineering, Stanford University. April.

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

Xue, G. and Ye, Y. 1997. An efficient algorithm for minimizing a sum of euclidean norms with applications. *SIAM J. on Optimization 7,* 4, 1017–1036.