Theory and Computation of Semidefinite Programming for Sensor Network Localization

Yinyu Ye

Department of Management Science and Engineering and

by courtesy, Electrical Engineering

Stanford University

Stanford, CA 94305, U.S.A.

http//www.stanford.edu/~yyye

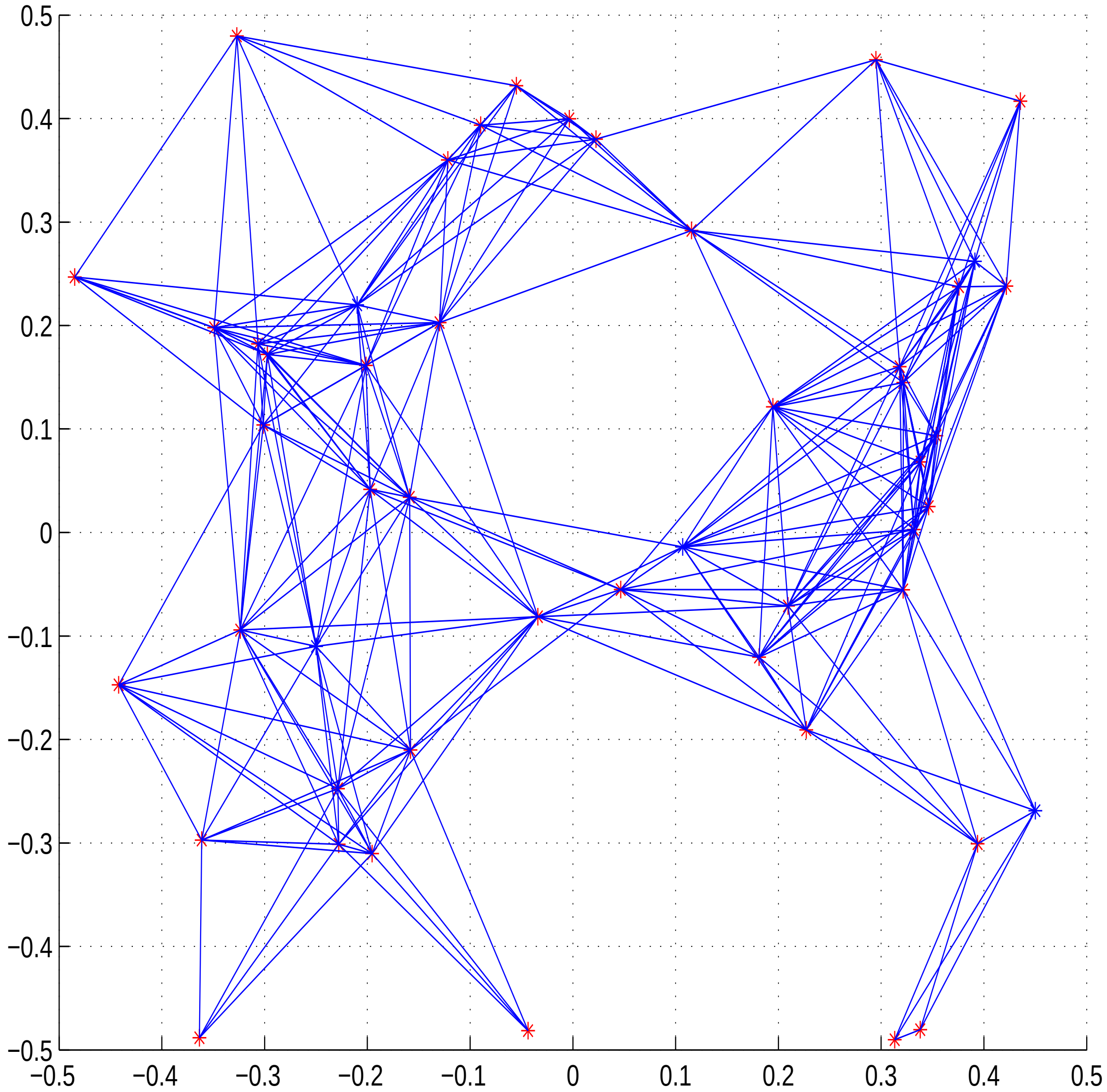
Outline

- Ad Hoc Wireless Sensor Network Localization
- SDP Relaxations and analyses
- SDP Decomposions and analyses
- Localization Demonstrations

Ad Hoc Wireless Sensor Network Localization

- **Input** m known points (anchors) $a_k \in \mathbf{R}^2$, $k = 1, ..., m$, and n unknown points (sensors or targets) $x_j \in \mathbf{R}^2$, $j = 1, ..., n$. For some pair of two points, we have a Euclidean distance measure $\hat{d}_{kj}$ between a_k and x_j, or distance measure $\hat{d}_{ij}$ between x_i and x_j.
- **Output** Position estimation for all unknown points, and confidence measures on reliability of each position estimation.
- **Objective** Robust, fast and accurate.

Figure 1: 50-Sensor Network with Radio Range .3



Related Work

- A great deal of research has been done on the topic of position estimation in ad-hoc networks, see Hightower and Boriello (2001) and Ganesan et al. (2002); Beacon grid: e.g., Bulusu and Heidemann (2000) and Howard et al. (2001); Distance measurement: e.g., Doherty et al. (2001), Niculescu and Nath (2001), Savarese et al. (2002), Savvides et al. (2001, 2002), Shang et al. (2003), Eren et al. (2004).
- Multidimensional scaling: Schoenberg and Young/Householder (1932) studied the case where all pairwise distances are given; Metric embedding: Johnson and Lindenstrauss (1984) and Bourgain (1985)
- Barvinok, Pataki, Alfakih/Wolkowicz and Laurent used SDP models to show that the problem is solvable in polynomial time if the dimension of the localization is not restricted. However, if we require the realization to be in $\mathbf{R}^d$ for some fixed d, then the problem becomes NP–complete (e.g., Saxe 1979,

Aspnes, Goldenberg, and Yang 2004).

- This talk: Using SDP to identify families of graph instances that admit polynomial time algorithms for computing a localization in the required dimension (SODA'05, ACM, IEEE, Math. Programing ...); and propose a further relaxation to improve solution efficiency.

Euclidean Distance Geometry Model

$$\|x_i - x_j\|^2 = d_{ij}^2, \ \forall\,(i,j) \in N_x, \ i < j,$$
$$\|a_k - x_j\|^2 = d_{kj}^2, \ \forall\,(k,j) \in N_a,$$

d_{ij}^2 (d_{kj}^2) connects x_i to x_j (a_k to x_j) with an edge whose length is d_{ij} (d_{kj}).

Does the system has a localization or realization of all x_j's? Is the localization unique? Is the localization reliable or trustworthy? Is the system partially localizable?

Convex Optimization Method: SOCP

$$\|x_i - x_j\|^2 \leq d_{ij}^2, \ \forall\, (i,j) \in N_x, \ i < j,$$
$$\|a_k - x_j\|^2 \leq d_{kj}^2, \ \forall\, (k,j) \in N_a.$$

Doherty et al. (2001) and Tseng (2005).

Global and Nonlinear Least Squares

$$\min \quad \sum_{i,j \in N_x} (\|x_i - x_j\|^2 - d_{ij}^2)^2 + \sum_{k,j \in N_a} (\|a_k - x_j\|^2 - d_{kj}^2)^2$$

$$\min \quad \sum_{i,j \in N_x} (\|x_i - x_j\| - d_{ij})^2 + \sum_{k,j \in N_a} (\|a_k - x_j\| - d_{kj})^2$$

For example, Moré and Wu (1997).

Matrix Representation

Let $X = [x_1\ x_2\ ...\ x_n]$ be the $2 \times n$ matrix that needs to be determined. Then

$\|x_i - x_j\|^2 = e_{ij}^T X^T X e_{ij}$ and $\|a_k - x_j\|^2 = (a_k; e_j)^T [I\ \ X]^T [I\ \ X](a_k; e_j)$,

where e_{ij} is the vector with 1 at the ith position, -1 at the jth position and zero everywhere else; and e_j is the vector of all zero except -1 at the jth position.

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

SDP Relaxation and Analyses

Change

$$Y = X^T X$$

to

$$Y \succeq X^T X.$$

This matrix inequality is equivalent to (e.g., Boyd et al. 1994)

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

SDP standard form

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

Find a symmetric matrix $Z \in \mathbf{R}^{(2+n)\times(2+n)}$ such that

$$\begin{aligned}
&Z_{1:2,1:2} = I \\
&(\mathbf{0}; e_{ij})(\mathbf{0}; e_{ij})^T \bullet Z = d_{ij}^2, \ \forall\, i,j \in N_x,\ i < j, \\
&(a_k; e_j)(a_k; e_j)^T \bullet Z = d_{kj}^2, \ \forall\, k,j \in N_a, \\
&Z \succeq 0.
\end{aligned}$$

Any matrix solution for the SDP relaxation has rank at least 2. If every sensor point is connected, directly or indirectly, to an anchor point, then the solution set must be bounded.

The dual of the SDP relaxation

$$
\begin{array}{ll}
\text{minimize} & I \bullet V + \sum_{i<j \in N_x} w_{ij} d_{ij}^2 + \sum_{k,j \in N_a} w_{kj} d_{kj}^2 \\
\text{subject to} & \begin{pmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \sum_{i<j \in N_x} w_{ij} (\mathbf{0}; e_{ij})(\mathbf{0}; e_{ij})^T \\
& + \sum_{k,j \in N_a} w_{kj} (a_k; e_j)(a_k; e_j)^T \succeq 0,
\end{array}
$$

where variable matrix $V \in \mathcal{M}^2$, varaible w_{ij} is the weight on edge from x_i to x_j, and w_{kj} is the weight on edge from a_k to x_j.

Note that the dual is always feasible since $V = 0$ and all $w.$ equal 0 is a feasible solution.

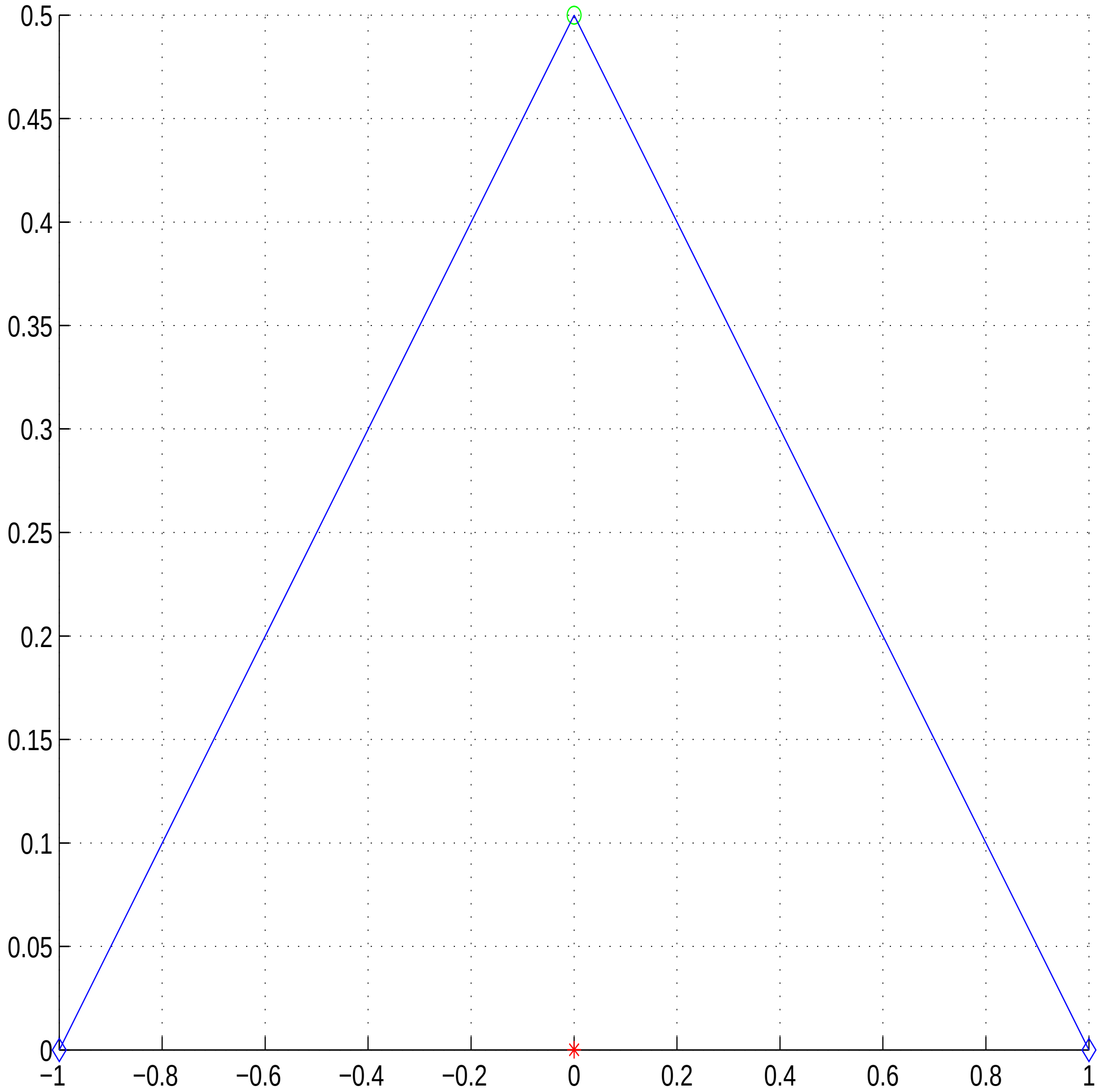
Localizable problem

A sensor network is localizable if there is a unique localization in $\mathbf{R}^2$ and there is no $x_j \in \mathbf{R}^h,\ j = 1, ..., n$, where $h > 2$, such that

$$\|x_i - x_j\|^2 = d_{ij}^2,\ \forall\, i, j \in N_x,\ i < j,$$
$$\|(a_k; \mathbf{0}) - x_j\|^2 = d_{kj}^2,\ \forall\, k, j \in N_a.$$

The latter says that the problem cannot be localized in a higher dimension space where anchor points are augmented to $(a_k; \mathbf{0}) \in \mathbf{R}^h$, $k = 1, ..., m$.

Figure 2: One sensor-Two anchors: Not localizable



When is the problem localizable?

Theorem 1. *The following statements are equivalent:*

1. *The sensor network is localizable;*
2. *The max-rank solution of the SDP relaxaion has rank 2;*
3. *The solution matrix has $Y = X^T X$ or Trace$(Y - X^T X) = 0$.*

When the dual has a solution with rank n, then the problem is strongly localizable.

Figure 3: Two sensor-Three anchors: Strongly Localizable

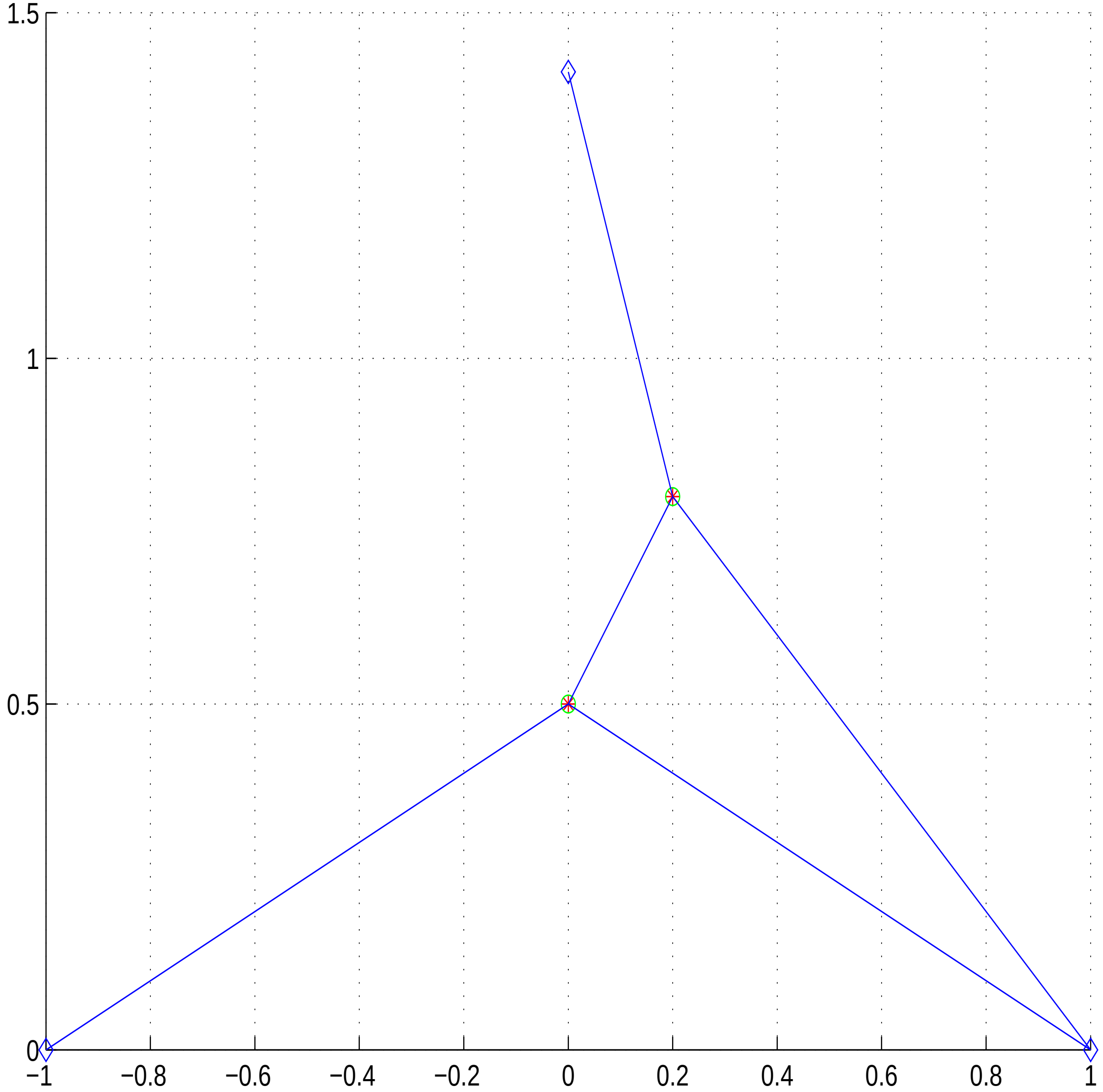


Figure 4: Two sensor-Three anchors: Localizable but not Strongly

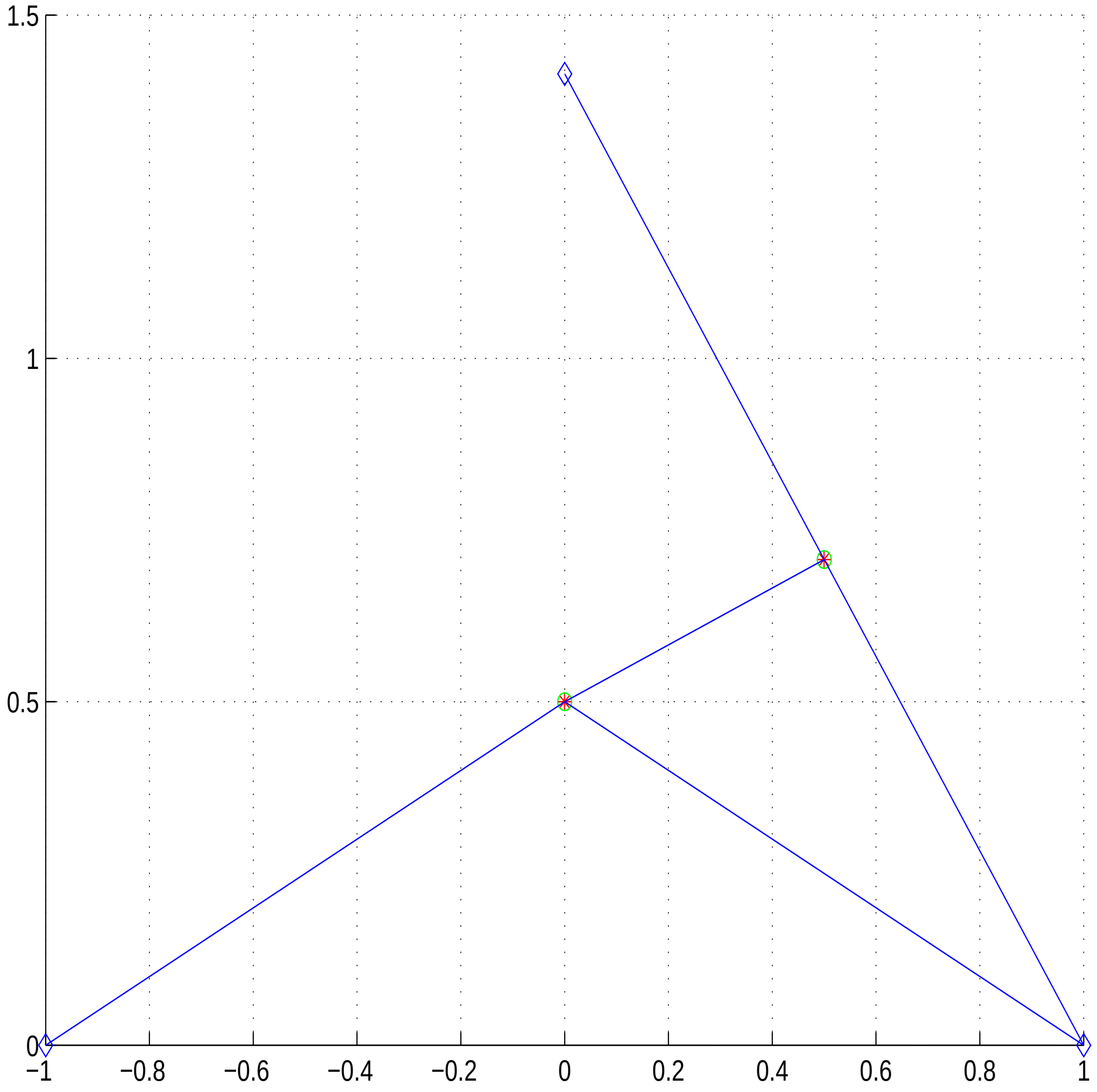


Figure 5: Two sensor-Three anchors: Not localizable

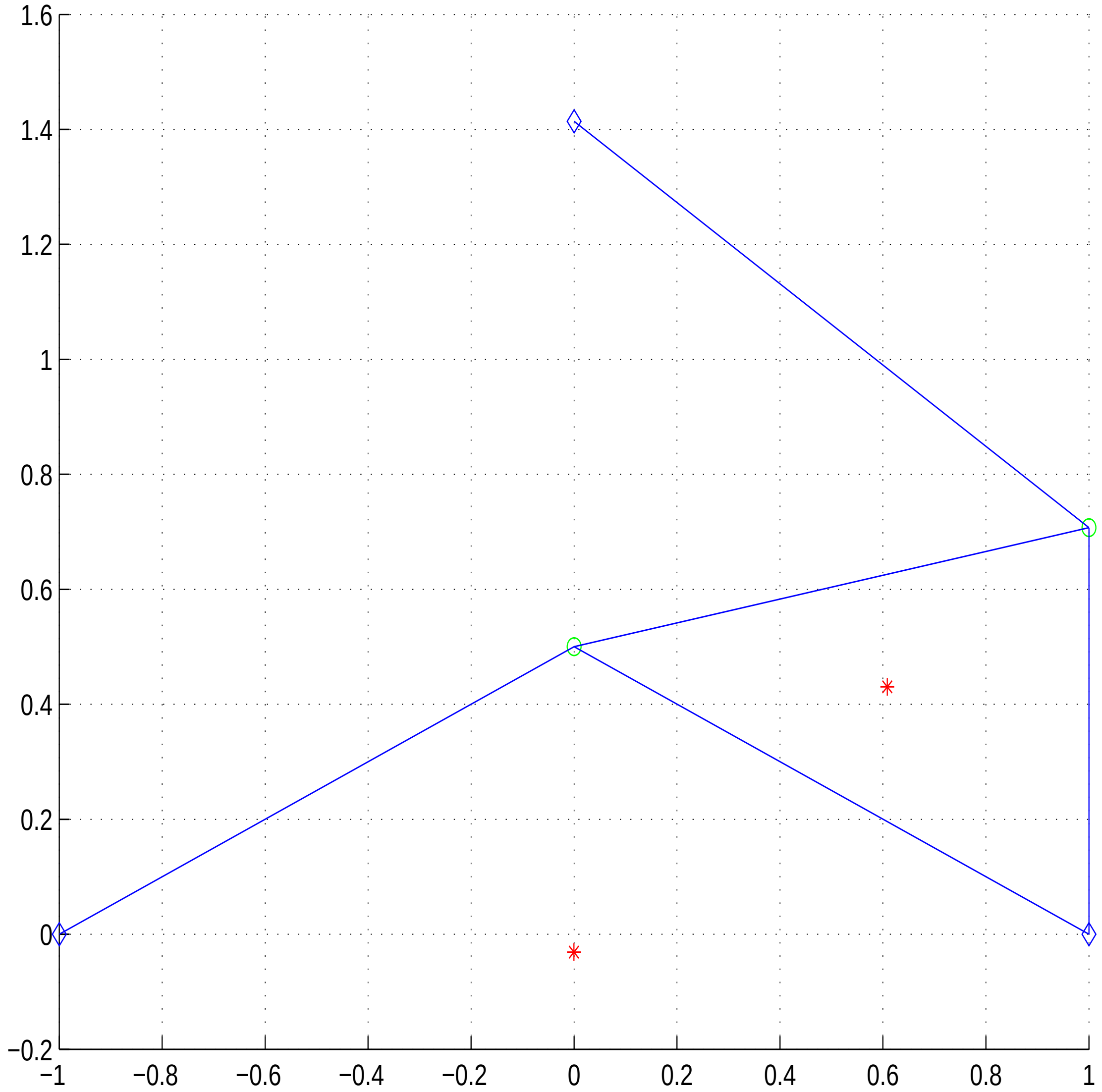
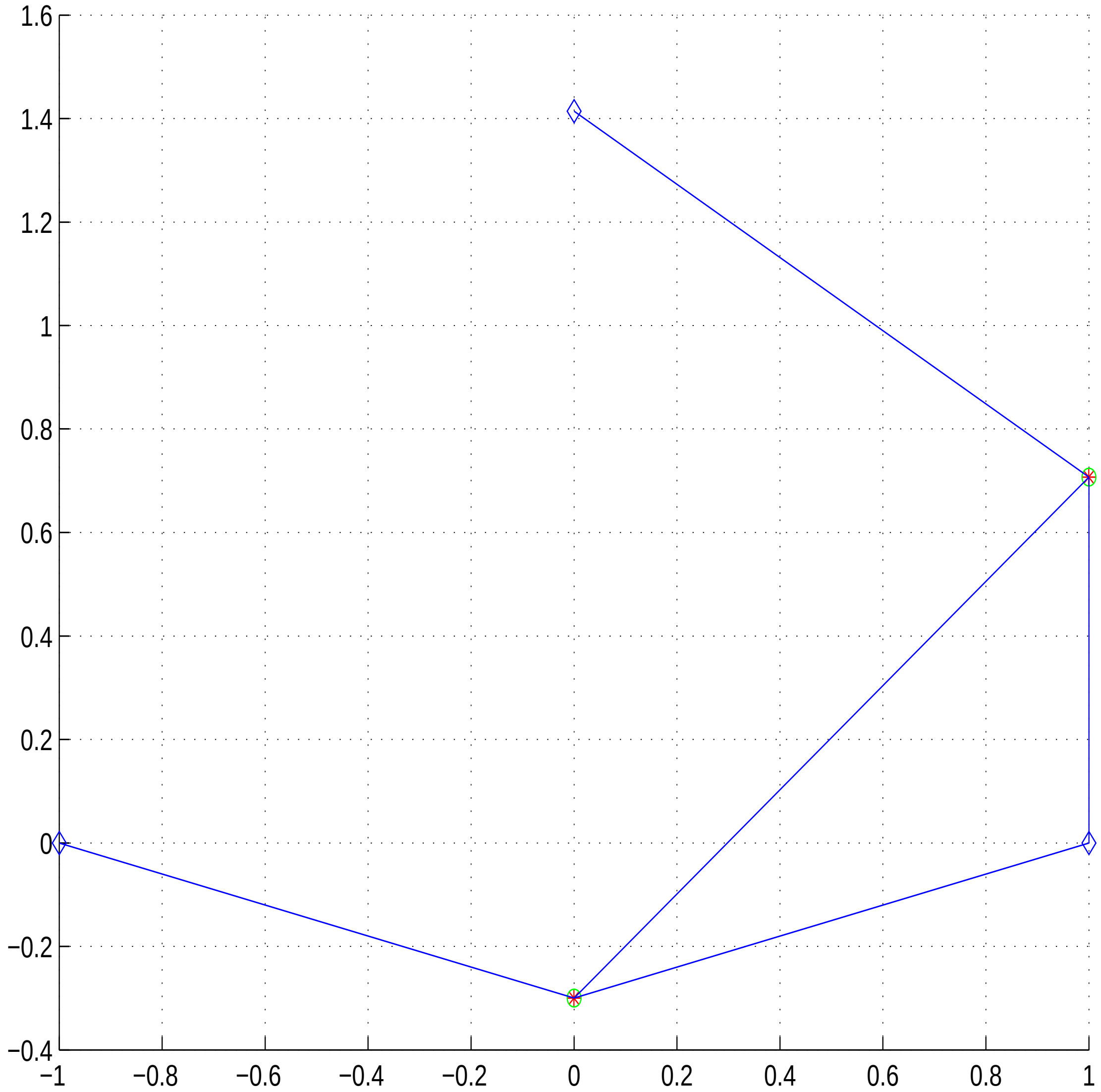


Figure 6: Two sensor-Three anchors: Stronly Localizable



Localize All Localizable Points

Theorem 2. *If a problem (graph) contains a subproblem (subgraph) that is localizable, then the submatrix solution corresponding to the subproblem in the SDP solution has rank 2. That is, the SDP relaxation computes a solution that localize all possibly localizable unknown sensor points.*

Implication: Trace,

$$\text{Trace}(\bar{Y} - \bar{X}^T \bar{X}) = \sum_{j=1}^{n} (\bar{Y}_{jj} - \|\bar{x}_j\|^2)$$

$\bar{Y}_{jj} - \|\bar{x}_j\|^2$ can be used as a measure to see whether jth sensor's estimated position is reliable or not.

Uncertainty Analysis and Confidence Measure

Alternatively, each x_j's can be viewed as uncertain points from the incomplete distance measures. Then the solution to the SDP problem provides the first and second moment estimation (Bertsimas and Ye 1998).

Generally, $\bar{x}_j$ is a point estimate of x_j and $\bar{Y}_{ij}$ is a point estimate $x_i^T x_j$.

Consequently,

$$\bar{Y}_{jj} - \|\bar{x}_j\|^2,$$

which is the individual variance estimation of sensor j, gives an interval estimation for its true position.

SDP Computation

SDP solvers used were SeDuMi (Sturm, 2001) and DSDP2.0 (Benson et al. 1998).

In our computational experiments:

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

Rounding the SDP solution

- When measurement noises exist or problem is not localizable, the SDP solution almost always has a high rank. How to round the high-rank solution into a low rank?

- Gradient-based local search: using the SDP relaxation solution as the initial point, we apply the steepest descent method to further reducing the estimation error:

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

or

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

- A checkable bound of suboptimality can be used to ensure the solution quality.

Figure 7: Gradient search trajectories: Two sensor-Three anchor Example

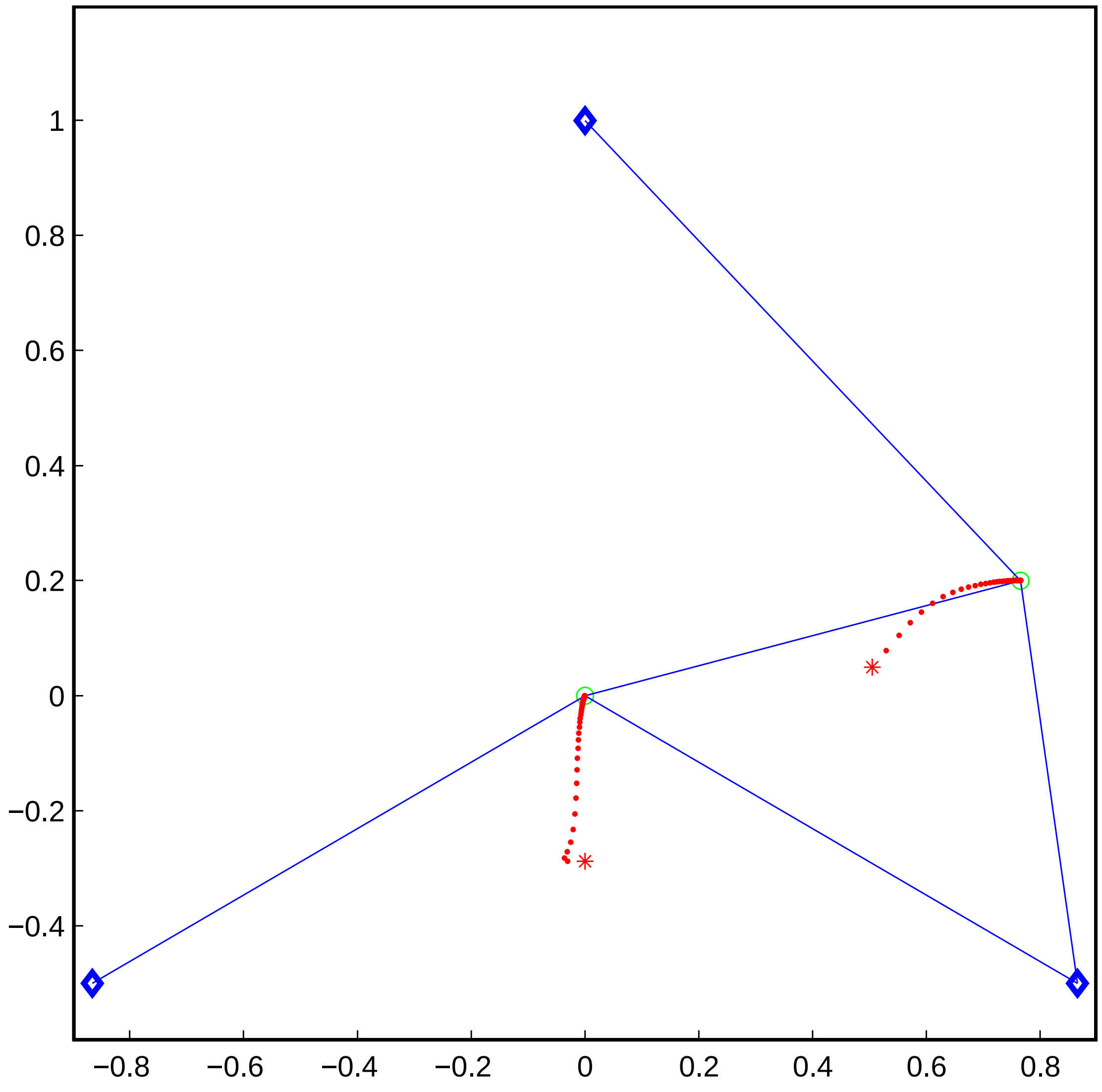


Figure 8: SDP/Gradient search trajectories: 10% Noise Example

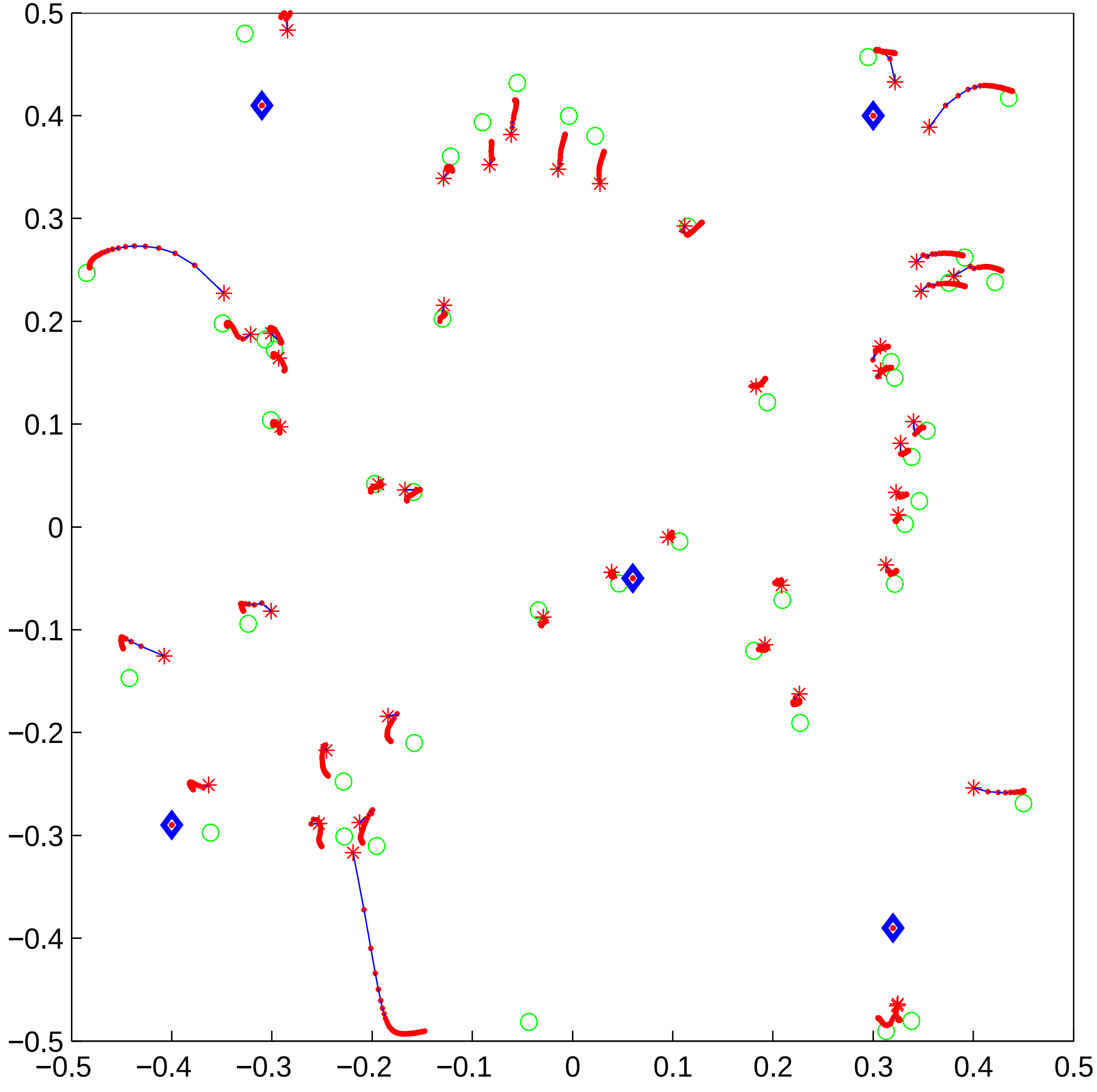


Figure 9: The objective value reduction: 10% Noise Example

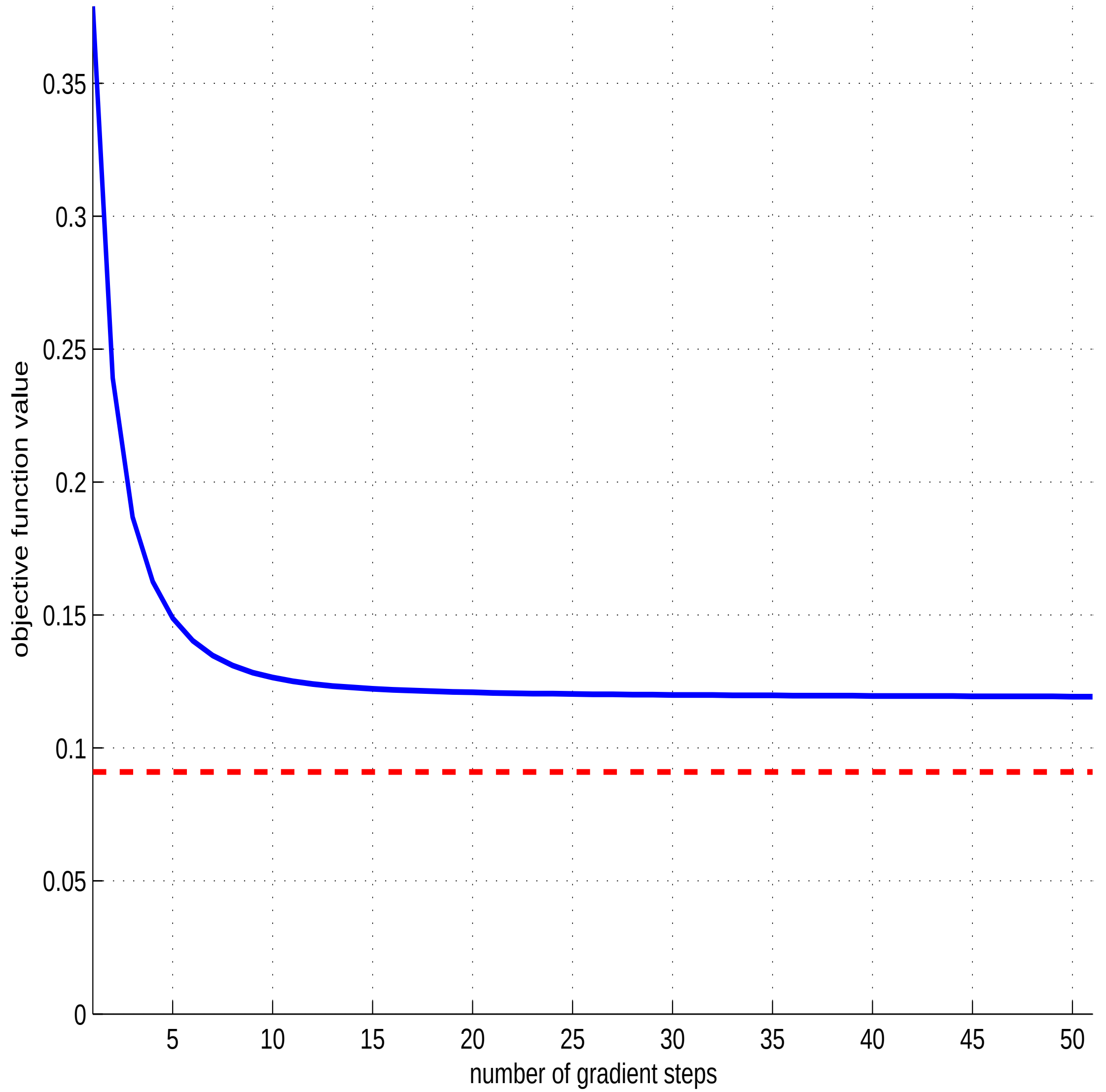


Figure 10: No-SDP/Gradient search trajectories: 10% Noise Example

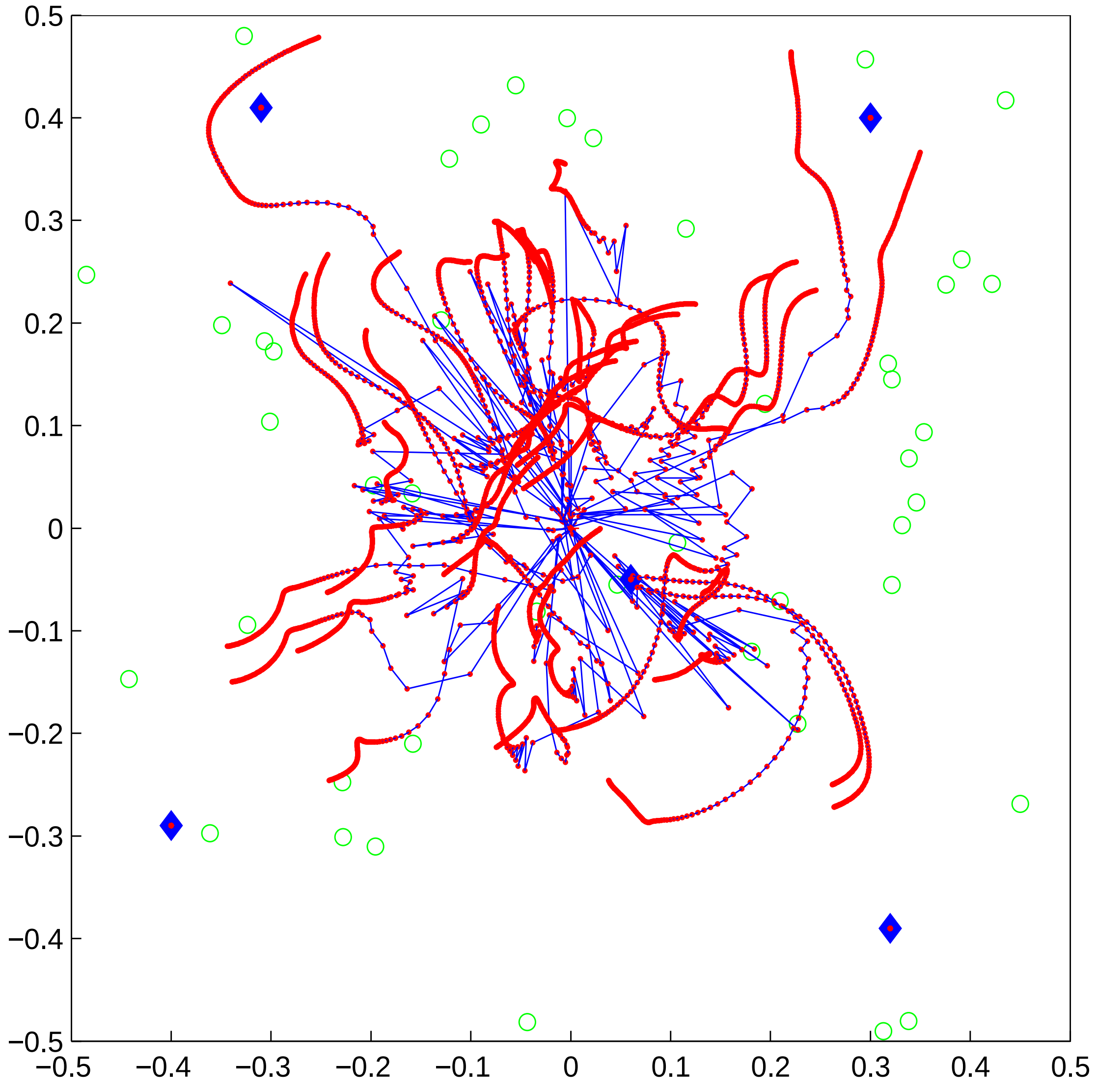
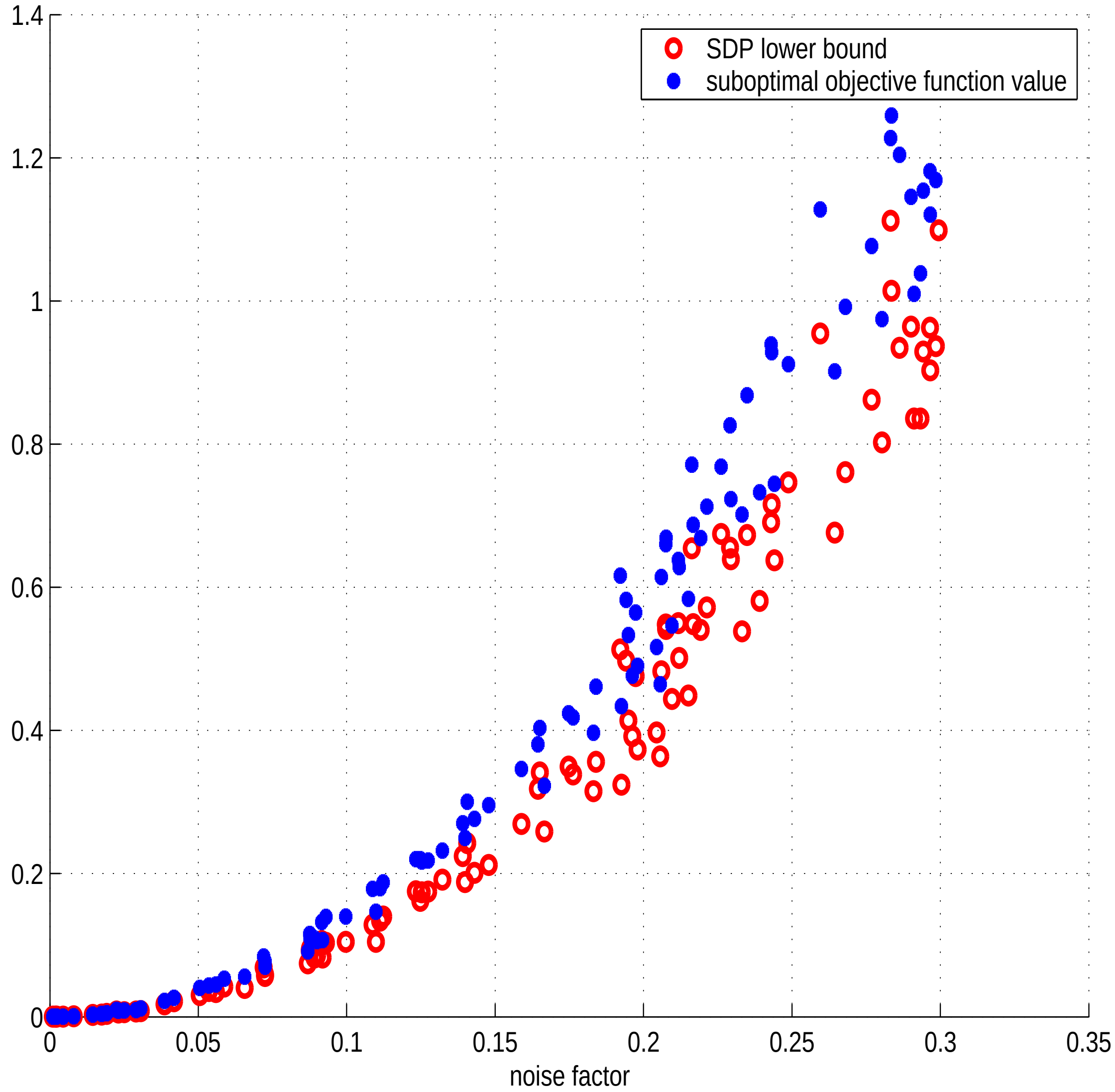


Figure 11: SDP lower bound and suboptimal objective function value vs noisy factor



NSDP Decomposition: Further Relaxation

Replace

$$(C1): \quad \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix} \succeq 0;$$

by

$$(C2): \quad \begin{pmatrix} I & x_i & x_{i_1} & \dots & x_{i_{d(i)}} \\ x_i^T & Y_{ii} & Y_{ii_1} & \dots & Y_{ii_{d(i)}} \\ x_{i_1}^T & Y_{i_1 i} & Y_{i_1 i_1} & \dots & Y_{i_1 i_{d(i)}} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i_{d(i)}}^T & Y_{i_{d(i)} i} & Y_{i_{d(i)} i_1} & & Y_{i_{d(i)} i_{d(i)}} \end{pmatrix} \succeq 0, \quad \forall i,$$

where $(i, i_{ik}) \in N_x$ and $d(i)$ is the degree of sensor node i.

ESDP Decomposition: Further Relaxation

Replace

$$(C1): \quad \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix} \succeq 0;$$

by

$$(C2): \quad \begin{pmatrix} I & x_i & x_j \\ x_i^T & Y_{ii} & Y_{ij} \\ x_j^T & Y_{ji} & Y_{jj} \end{pmatrix} \succeq 0, \quad \forall (i,j) \in N_x.$$

Analyses

A undirected graph is a chordal graph if every cycle of length greater than three has a chord.

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

Lemma 1. *(Hogben 2001) Every partial positive semi-definite matrix with undirected graph G has positive semi-definite completion if and only if G is chordal.*

The Equivalence Theorem for NSDP

Theorem 3. *(Wang, Zheng, Boyd and Ye [2006]) Suppose the undirected graph of sensor nodes with edge set* N_x *is chordal, then SDP and NSDP relaxations are equivalent.*

The Trace Theorem for ESDP

Theorem 4. *Let*

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

be a solution of ESDP computed by a path-following method. If the diagonal entry or individual trace

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

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 ESDP.*

Relations of ESDP and SOCP

Let

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

be a solution of ESDP. Then, X is also feasible for SOCP; and the reverse is not true.

ESDP Computational Results

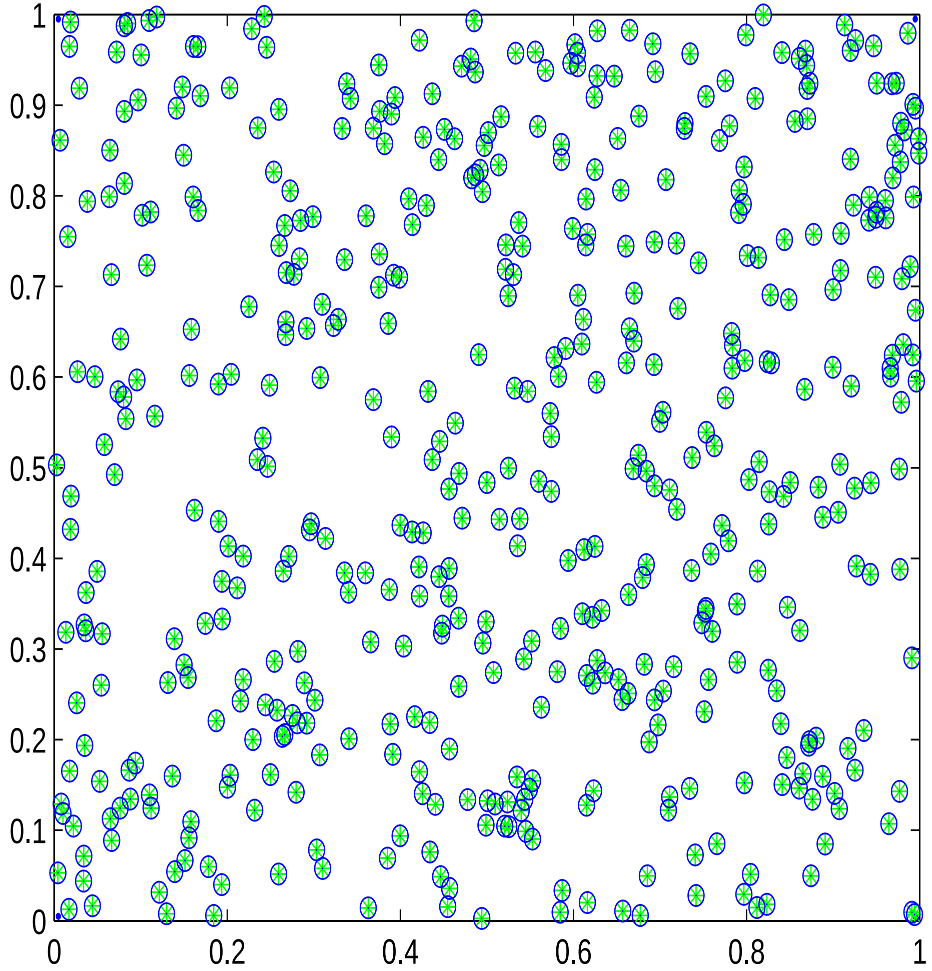
Test Problem #	n	nf	r	SDP dim
1	100	0	0.25	3815×7372
2	100	0.005	0.25	3815×7372
3	100	0.05	0.25	3815×7372
4	500	0	0.1	13685×27166
5	500	0.005	0.1	13685×27166
6	500	0.05	0.1	13685×27166
7	1000	0	0.06	28879×57004
8	1000	0.005	0.06	28879×57004
9	1000	0.05	0.06	28879×57004
10	4000	0	0.035	133759×261214

Table 1: Input parameters for the test problems and the corresponding SDP dimensions

Test Problem #	iter	CPUtime	obj	RMSD
1	13	4.65	2e-4	4.8e-5
2	17	6.17	6.9e-4	2.9e-4
3	17	5.1	3e-4	1e-3
4	14	21.73	6.5e-5	4e-6
5	15	21.87	1.5e-3	3e-4
6	17	24.35	0.035	1e-3
7	15	65.64	4e-3	9e-4
8	16	72.86	4.5e-3	0.03
9	17	73.26	0.024	0.03
10	17	1035.45	0.024	3e-4

Table 2: ESDP mumerical results where CPUtime are in seconds on a laptop with 256MB Memory and 1.4GHz CPU

Figure 12: Graphical localization result of the ESDP model on the problem of Nie, 500 sensors, 4 anchors, $r = 0.3$, RMSD=$1e-6$



ESDP and SOCP Comparisons

Test Problem #	n	nf	r	ESDP	SeDuMi of SOCP	SCGD of SOCP
1	500	0	0.1	21.73sec	5.5min	0.4min
2	500	0.005	0.1	21.87sec	5.4min	3.3min
3	500	0.05	0.1	24.35sec	4.6min	2.2min
4	1000	0	0.06	65.64sec	209.6min	1.3 min
5	1000	0.005	0.06	72.86sec	230.1min	6.8min
6	1000	0.05	0.06	73.26sec	176.6min	3.75min

Table 3: ESDP times are taken on a laptop (256MB and 1.4GHz), and SOCP times are reported from Tseng on a HP DL360 (1G Memory and 3GHz)

Related Problem: the Graph Realization Problem

Given a graph $G = (V, E)$ and a set of non–negative edge weights $\{d_{ij} : (i, j) \in E\}$, the goal is to compute a realization of G in the Euclidean space $\mathbf{R}^d$ for a given dimension d, i.e. to place the vertices of G in $\mathbf{R}^d$ such that the Euclidean distance between every pair of adjacent vertices (i, j) in E equals the prescribed weight d_{ij}.

d–Realizable Graphs

A graph is d–realizable if it can always be realized in $\mathbf{R}^d$ whenever it is realizable (the edge weights are Euclidean metric) for every instance of the graph.

- Connelly and Sloughter have recently given a complete characterization of the class of d–realizable graphs, where $d = 1, 2, 3$
- It is trivial to find a realization of an 1–realizable graph, since a graph is 1–realizable iff it is a forest.
- A polynomial time algorithm for realizing 2–realizable graphs exists: triangulation.
- Finding a corresponding algorithm for 3–realizable graphs is posed as an open question.

3–realizable graph II

Using the forbidden minor characterization of partial 3–trees, one can show that a graph is 3–realizable if it either

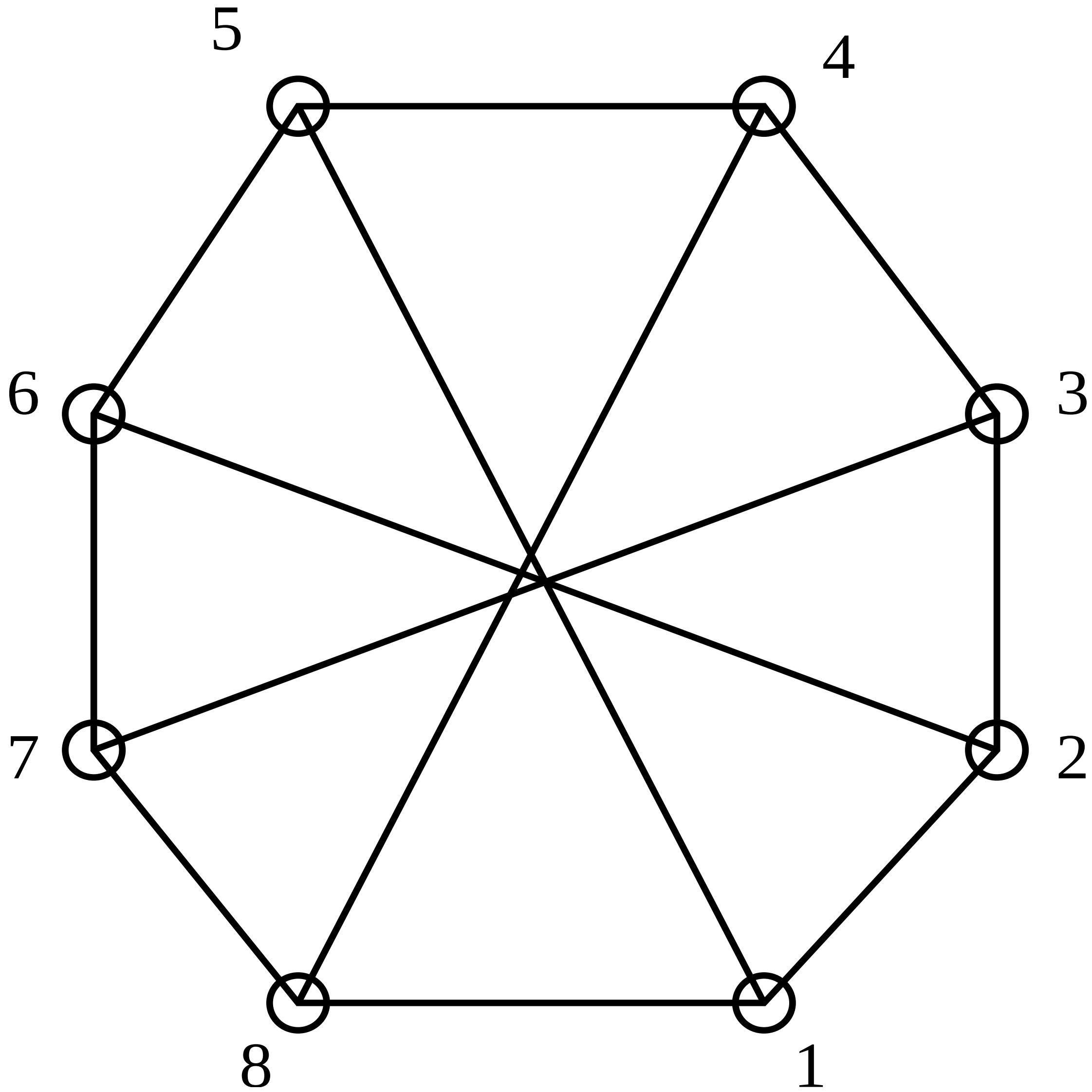
- contains an V_8 or an $C_5 \times C_2$ as a minor
- or does not contain either graphs as a minor.

Indeed, if it is the latter, then G is a partial 3–tree.

An k-tree is defined recursively as follows. The complete graph on k vertices is an k–tree. An k–tree with $n+1$ vertices (where $n \geq k$) can be constructed from an k–tree with n vertices by adding a vertex adjacent to all vertices of one of its k–vertex complete subgraphs, and only to those vertices.

A partial k–tree is a subgraph of an k–tree.

Figure 13: V-8



Our Result

We resolve the above open question by giving a polynomial time algorithm for (approximately) realizing 3–realizable graphs.

The main bottleneck in the proof is to show that two graphs, V_8 and $C_5 \times C_2$, are 3–realizable.

There exists a realization $\mathbf{p}$ of $H \in \{V_8, C_5 \times C_2\}$ such that the distance between a certain pair of non–adjacent vertices (i, j) is maximized. Such a realization induces a non–zero equilibrium stress on the graph H' obtained from H by adding the edge (i, j). Then use this equilibrium force to prove that H' must be in $\mathbf{R}^3$.

We show that the problem of computing the desired $\mathbf{p}$ can be formulated as an SDP. More interesting is that the optimal dual multipliers of our SDP give rise to a non–zero equilibrium stress.

Related Problem: the Kissing Number Problem

- Given a unit center sphere, the maximum number of unit spheres, in n dimensions, can touch or kiss the center sphere?
- General Solutions does not exist.
- Delsarte Method uses linear programming to provide an upper bound on the number of spheres.
- K(8) = 240, K(24) = 196650.
- K(4) = 24: proved using Delsarte Method by Oleg Musin only 3 years ago.
- For other dimensions, lower bounds have been provided by constructing a lattice structure. There also exists a bound using the Riemann zeta function, but is non-constructive.

The Kissing Problem as a Graph Realization

- Can be formulated as a SDP feasibility problem; but SDP solution may not provide proper rank.

$$\begin{aligned} (e_i - e_j)^T X (e_i - e_j) &\geq 4, \ \forall i \neq j, \\ e_i^T X e_i &= 4, \ \forall i \end{aligned}$$

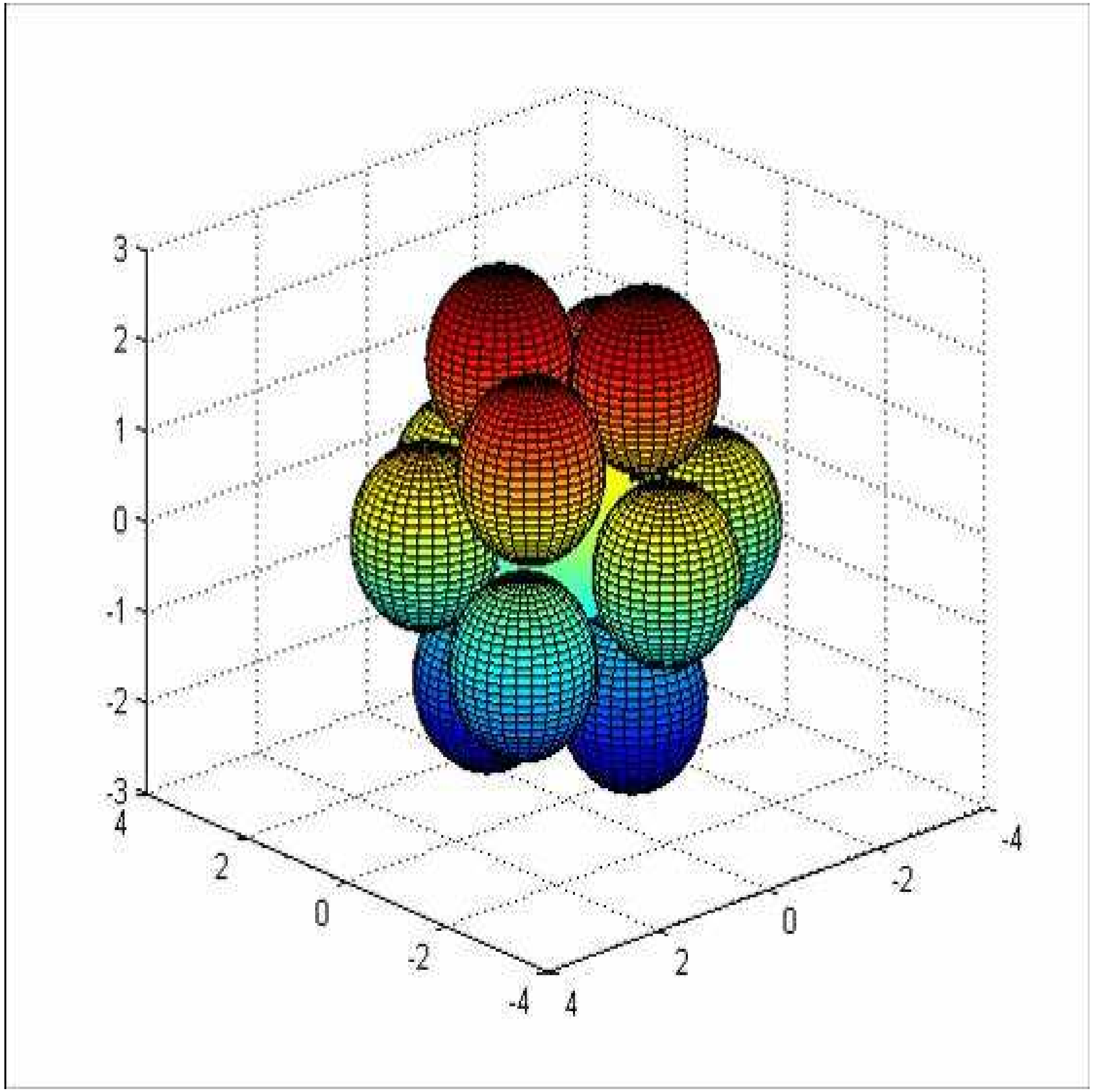
- Construct a nonzero SDP objective function to reduce the rank of a solution.

$$\begin{array}{lrl} \min & C \bullet X, & \\ \text{s.t.} & (e_i - e_j)^T X (e_i - e_j) & \geq 4, \ \forall i \neq j, \\ & e_i^T X e_i & = 4, \ \forall i \end{array}$$

Solving the 3-D Kissing Problem

This objective structure can be extended to dimension 3. For 12 spheres, SDP method provides the following realization

Figure 14: 12 Spheres in 3-D



The Kissing Problem and Coding

Given the number of points, find the largest r such that

$$\begin{aligned} (e_i - e_j)^T X (e_i - e_j) &\geq (1+r)^2,\ \forall i \neq j, \\ e_i^T X e_i &= (1+r)^2,\ \forall i \end{aligned}$$

has a rank 3 matrix solution.

Schoenberg's theorem on the Gegenbauer polynomial may be used to strenghten the above SDP formulation.

More Applications

- Global Position System (GPS)
- Molecular conformation
- Data dimension reduction
- Euclidean ball packing

More research topics

- Sensor Network Design: how many anchors need to be used? Where to place them?
- Why ESDP works well?
- Applicable to solving general SDPs?
- How to efficiently round an SDP solution matrix into a lower rank solution matrix (if it exists)?