Semidefinite Programming and Universal Rigidity

Yinyu Ye

Department of Management Science and Engineering, and
Institute of Computational and Mathematical Engineering
Stanford University

Based on joint work mostly with Abdo Alfakih, Pratik Biswas, Anthony So, Nicole Taheri, and Zhisu Zhu

Rigidity Workshop, Cornell, February 5, 2010

Outline

Introduction to Semidefinite Programming

SDP Solution Rank Theorems

Sensor Network Localization and Graph Realization

SDP Relaxation and Localizability

Semidefinite Programming Problem (SDP)

Find a real $n \times n$ symmetric matrix $X \in \mathcal{S}^n$ that satisfies

$$A_i \bullet X = b_i, \quad \forall i = 1, \ldots, m, \quad X \succeq \mathbf{0}$$

where $A_1, \ldots, A_m$ are given real symmetric matrices with real scalars $(b_1, \ldots, b_m)$,

$$A \bullet X = \sum_{i,j} a_{ij} x_{ij},$$

and $\succeq$ represents the matrix inequality of positive semi-definiteness.

Semidefinite Programming Problem (SDP)

Find a real $n \times n$ symmetric matrix $X \in \mathcal{S}^n$ that satisfies

$$A_i \bullet X = b_i, \quad \forall i = 1, \ldots, m, \quad X \succeq \mathbf{0}$$

where $A_1, \ldots, A_m$ are given real symmetric matrices with real scalars $(b_1, \ldots, b_m)$,

$$A \bullet X = \sum_{i,j} a_{ij} x_{ij},$$

and $\succeq$ represents the matrix inequality of positive semi-definiteness.

$$x_1 + x_2 + x_3 = 1,$$
$$\begin{pmatrix} x_1 & x_2 \\ x_2 & x_3 \end{pmatrix} \succeq \mathbf{0}.$$

Semidefinite Programming Optimization Problem

$$
\begin{array}{llll}
(SDP) & \inf & C \bullet X & \\
& \text{subject to} & A_i \bullet X = b_i & i = 1, \ldots, m, \\
& & X \succeq \mathbf{0}. &
\end{array}
$$

where C is a given (objective) symmetric matrix. The dual problem to (SDP) can be written as:

$$
\begin{array}{lll}
(SDD) & \sup & \mathbf{b} \cdot \mathbf{y} \\
& \text{subject to} & S = C - \sum_i^m y_i A_i \succeq \mathbf{0},
\end{array}
$$

where $\mathbf{y} \in \mathbf{R}^m$.

Duality Theorems for SDPs

Theorem

(Weak duality theorem in SDP) Let $\mathcal{F}_p$ and $\mathcal{F}_d$, the feasible sets for the primal and dual, be non-empty. Then,

$$C \bullet X \geq \mathbf{b} \cdot \mathbf{y}, \quad \forall \quad X \in \mathcal{F}_p, \ (\mathbf{y}, S) \in \mathcal{F}_d.$$

Duality Theorems for SDPs

Theorem

(Weak duality theorem in SDP) Let $\mathcal{F}_p$ and $\mathcal{F}_d$, the feasible sets for the primal and dual, be non-empty. Then,

$$C \bullet X \geq \mathbf{b} \cdot \mathbf{y}, \quad \forall \quad X \in \mathcal{F}_p, \ (\mathbf{y}, S) \in \mathcal{F}_d.$$

Theorem

(Strong duality theorem in SDP) Let $\mathcal{F}_p$ and $\mathcal{F}_d$ be non-empty and at least one of them has an interior. Then, X is optimal for (SDP) and $(\mathbf{y}, S)$ optimal for (SDD) if and only if the following conditions hold:

i) $X \in \mathcal{F}_p$ *and* $(\mathbf{y}, S) \in \mathcal{F}_d$;

ii) $C \bullet X = \mathbf{b} \cdot \mathbf{y}$ *or* $X \bullet S = 0$ *or* $XS = \mathbf{0}$.

Outline

SDP Solution Rank

- There are optimal solutions of X^* and S^* such that the ranks of X^* and S^* are maximal, respectively.

SDP Solution Rank

- There are optimal solutions of X^* and S^* such that the ranks of X^* and S^* are maximal, respectively.
- But the sum of the two ranks should be no more than n. If the sum is n, then we say the problem admits a strictly complementary solution pair.

SDP Solution Rank

- There are optimal solutions of X^* and S^* such that the ranks of X^* and S^* are maximal, respectively.
- But the sum of the two ranks should be no more than n. If the sum is n, then we say the problem admits a strictly complementary solution pair.
- There are optimal solutions of X^* and S^* such that the ranks of X^* and S^* are minimal, respectively.

SDP Solution Rank

- There are optimal solutions of X^* and S^* such that the ranks of X^* and S^* are maximal, respectively.
- But the sum of the two ranks should be no more than n. If the sum is n, then we say the problem admits a strictly complementary solution pair.
- There are optimal solutions of X^* and S^* such that the ranks of X^* and S^* are minimal, respectively.
- In applications, we like to find either a max-rank or min-rank solution or/and to prove it exist.

SDP Solution Rank

- There are optimal solutions of X^* and S^* such that the ranks of X^* and S^* are maximal, respectively.
- But the sum of the two ranks should be no more than n. If the sum is n, then we say the problem admits a strictly complementary solution pair.
- There are optimal solutions of X^* and S^* such that the ranks of X^* and S^* are minimal, respectively.
- In applications, we like to find either a max-rank or min-rank solution or/and to prove it exist.

If there is S^* such that $\text{rank}(S^*) \geq n - d$, then the rank of any X^* is bounded above by d.

SDP Solution Uniqueness: Zhu 2010

Theorem

Let X^* be a max-rank solution of (SDP) and let $X^* = P^T P$ where $P \in \mathbf{R}^{r \times n}$. Then, X^* is the unique solution for (SDP) if and only if the null space of the linear space spanned by $PA_iP^T,\ i = 1, \ldots, m$, is $\{\mathbf{0}\}$.

Corollary

If all (SDP) solutions share the same rank, then (SDP) has a unique solution.

SDP Computational Complexity and Solution Rank

- Let the SDP problem have a finite solution pair. Then, the SDP interior-point algorithm finds an ϵ-approximate solution where solution time is linear in $\log(1/\epsilon)$ and polynomial in m and n.

SDP Computational Complexity and Solution Rank

- Let the SDP problem have a finite solution pair. Then, the SDP interior-point algorithm finds an ϵ-approximate solution where solution time is linear in $\log(1/\epsilon)$ and polynomial in m and n.
- The approximate solution pair $(X(\epsilon), S(\epsilon))$ computed by the interior-point algorithm converges to a max-rank solution pair for the primal and dual (Güler and Y 1993, Y 1995).

SDP Computational Complexity and Solution Rank

- Let the SDP problem have a finite solution pair. Then, the SDP interior-point algorithm finds an ϵ-approximate solution where solution time is linear in $\log(1/\epsilon)$ and polynomial in m and n.
- The approximate solution pair $(X(\epsilon), S(\epsilon))$ computed by the interior-point algorithm converges to a max-rank solution pair for the primal and dual (Güler and Y 1993, Y 1995).
- Barvinok (1995) showed that if the problem is solvable, then there exists a solution X whose rank r satisfies $r(r+1) \leq 2m$, and it is essentially tight.

SDP Computational Complexity and Solution Rank

- Let the SDP problem have a finite solution pair. Then, the SDP interior-point algorithm finds an ϵ-approximate solution where solution time is linear in $\log(1/\epsilon)$ and polynomial in m and n.
- The approximate solution pair $(X(\epsilon), S(\epsilon))$ computed by the interior-point algorithm converges to a max-rank solution pair for the primal and dual (Güler and Y 1993, Y 1995).
- Barvinok (1995) showed that if the problem is solvable, then there exists a solution X whose rank r satisfies $r(r+1) \leq 2m$, and it is essentially tight.
- Given any solution, a such low-rank solution can be found in polynomial time based on Carathéodory's theorem (Pataki 1999, Alfakih/Wolkowicz 1999).

More on SDP Solution Rank

- We are interested in finding a fixed low–rank (say d) solution to the SDP problem.

More on SDP Solution Rank

- We are interested in finding a fixed low–rank (say d) solution to the SDP problem.
- However, there are some issues:
  - Such a solution may not exist!
  - Even if it does, one may not be able to find it efficiently.

More on SDP Solution Rank

- We are interested in finding a fixed low–rank (say d) solution to the SDP problem.
- However, there are some issues:
  - Such a solution may not exist!
  - Even if it does, one may not be able to find it efficiently.
- Suppose we consider an approximate low-rank solution to

$$A_i \bullet X = b_i, \quad \forall i = 1, \ldots, m, \quad X \succeq \mathbf{0},$$

where $A_i \succeq \mathbf{0}$ for all i.

Approximate Low-Rank SDP Solution

More precisely, find an $\hat{X} \succeq 0$ with rank at most d that satisfies the SDP constraints approximately:

$$\beta(m,n,d) \cdot b_i \le A_i \bullet \hat{X} \le \alpha(m,n,d) \cdot b_i \quad \forall\ i = 1, \ldots, m.$$

Approximate Low-Rank SDP Solution

More precisely, find an $\hat{X} \succeq 0$ with rank at most d that satisfies the SDP constraints approximately:

$$\beta(m,n,d) \cdot b_i \leq A_i \bullet \hat{X} \leq \alpha(m,n,d) \cdot b_i \quad \forall\ i = 1, \ldots, m.$$

Here, $\alpha(\cdot) \geq 1$ and $\beta(\cdot) \in (0,1]$ are called the distortion factors.

Approximate Low-Rank SDP Solution

More precisely, find an $\hat{X} \succeq 0$ with rank at most d that satisfies the SDP constraints approximately:

$$\beta(m,n,d) \cdot b_i \leq A_i \bullet \hat{X} \leq \alpha(m,n,d) \cdot b_i \quad \forall\ i = 1, \ldots, m.$$

Here, $\alpha(\cdot) \geq 1$ and $\beta(\cdot) \in (0,1]$ are called the distortion factors.

Clearly, the closer are both to 1, the better the approximation quality.

Approximation Result, So, Y and Zhang 2007

Theorem

For any $d \geq 1$*, there exists an* $\hat{X} \succeq \mathbf{0}$ *with* $rank(\hat{X}) \leq d$ *such that*

$$\alpha(m,n,d) = \begin{cases} 1 + \dfrac{18\ln(2m)}{d} & \text{for } 1 \leq d \leq 18\ln(2m) \\ 1 + \sqrt{\dfrac{18\ln(2m)}{d}} & \text{for } d > 18\ln(2m) \end{cases},$$

$$\beta(m,n,d) = \begin{cases} \dfrac{1}{e(2m)^{2/d}} & \text{for } 1 \leq d \leq 4\ln(2m) \\ \max\left\{\dfrac{1}{e(2m)^{2/d}}, 1 - \sqrt{\dfrac{4\ln(2m)}{d}}\right\} & \text{for } d > 4\ln(2m) \end{cases}$$

Some Remarks

- There is always a low-rank approximate SDP solution with respect to a bounded distortion. As the allowable rank increases, the distortion bounds become smaller and smaller.

Some Remarks

- There is always a low-rank approximate SDP solution with respect to a bounded distortion. As the allowable rank increases, the distortion bounds become smaller and smaller.
- Moreover, there exists an efficient randomized algorithm for finding such an $\hat{X}$.

Some Remarks

- There is always a low-rank approximate SDP solution with respect to a bounded distortion. As the allowable rank increases, the distortion bounds become smaller and smaller.
- Moreover, there exists an efficient randomized algorithm for finding such an $\hat{X}$.
- The distortion factors are independent of n.

Some Remarks

- There is always a low-rank approximate SDP solution with respect to a bounded distortion. As the allowable rank increases, the distortion bounds become smaller and smaller.
- Moreover, there exists an efficient randomized algorithm for finding such an $\hat{X}$.
- The distortion factors are independent of n.
- The factors are sharp; but they can be improved if we only consider one–sided inequalities.

Some Remarks

- There is always a low-rank approximate SDP solution with respect to a bounded distortion. As the allowable rank increases, the distortion bounds become smaller and smaller.
- Moreover, there exists an efficient randomized algorithm for finding such an $\hat{X}$.
- The distortion factors are independent of n.
- The factors are sharp; but they can be improved if we only consider one–sided inequalities.
- This result contains as special cases several well-known results in the literature.

Outline

Introduction to Semidefinite Programming

SDP Solution Rank Theorems

Sensor Network Localization and Graph Realization

SDP Relaxation and Localizability

Sensor Network Localization and Graph Realization

Given a graph $G = (V, E)$ and sets of non–negative weights, say $\{d_{ij} : (i, j) \in E\}$ on edges, the goal is to compute a realization of G in the Euclidean space $\mathbf{R}^d$ for a given low dimension d. That is,

Sensor Network Localization and Graph Realization

Given a graph $G = (V, E)$ and sets of non–negative weights, say $\{d_{ij} : (i,j) \in E\}$ on edges, the goal is to compute a realization of G in the Euclidean space $\mathbf{R}^d$ for a given low dimension d. That is,

- to place the vertexes of G in $\mathbf{R}^d$ such that

Sensor Network Localization and Graph Realization

Given a graph $G = (V, E)$ and sets of non–negative weights, say $\{d_{ij} : (i, j) \in E\}$ on edges, the goal is to compute a realization of G in the Euclidean space $\mathbf{R}^d$ for a given low dimension d. That is,

- to place the vertexes of G in $\mathbf{R}^d$ such that
- the Euclidean distance between a pair of adjacent vertexes (i, j) equals to (or bounded by) the prescribed weight $d_{ij} \in E$.

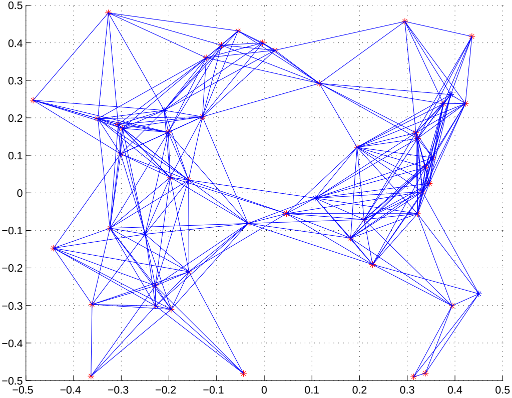
Sensor Network Localization and Graph Realization

Given a graph $G = (V, E)$ and sets of non–negative weights, say $\{d_{ij} : (i, j) \in E\}$ on edges, the goal is to compute a realization of G in the Euclidean space $\mathbf{R}^d$ for a given low dimension d. That is,

- to place the vertexes of G in $\mathbf{R}^d$ such that
- the Euclidean distance between a pair of adjacent vertexes (i, j) equals to (or bounded by) the prescribed weight $d_{ij} \in E$.

The positions of a few vertexes are known and they are called anchors for SNL.

50-node 2-D Sensor Localization



Quadratic Equality and Inequality Systems

Given graph $G = (V, E)$ and $d_{ij} \in E$, find $\mathbf{x}_j \in \mathbf{R}^d$ such that

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 \quad (\leq) = (\geq) \quad d_{ij}^2, \; \forall \; (i, j) \in E, \; i < j.$$

Quadratic Equality and Inequality Systems

Given graph $G = (V, E)$ and $d_{ij} \in E$, find $\mathbf{x}_j \in \mathbf{R}^d$ such that

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 \quad (\leq) = (\geq) \quad d_{ij}^2, \ \forall\ (i,j) \in E,\ i < j.$$

Or given $\mathbf{a}_k \in \mathbf{R}^d$, $d_{ij} \in N_x$, and $\hat{d}_{kj} \in N_a$, find $\mathbf{x}_i \in \mathbf{R}^d$ such that

$$\begin{array}{lll} \|\mathbf{x}_i - \mathbf{x}_j\|^2 & (\leq) = (\geq) & d_{ij}^2, \ \forall\ (i,j) \in N_x,\ i < j, \\ \|\mathbf{a}_k - \mathbf{x}_j\|^2 & (\leq) = (\geq) & \hat{d}_{kj}^2, \ \forall\ (k,j) \in N_a; \end{array}$$

that is, edge (ij) (or (kj)) connects sensors i and j (or anchor k and sensor j) with the Euclidean length equal to d_{ij} (or $\hat{d}_{kj}$).

Key Questions Related to SNL

Let the "bar system" be:

$$\begin{aligned} \|\mathbf{x}_i - \mathbf{x}_j\|^2 &= d_{ij}^2, \ \forall\ (i,j) \in N_x,\ i < j, \\ \|\mathbf{a}_k - \mathbf{x}_j\|^2 &= \hat{d}_{kj}^2, \ \forall\ (k,j) \in N_a, \end{aligned}$$

Key Questions Related to SNL

Let the "bar system" be:

$$\begin{aligned} \|\mathbf{x}_i - \mathbf{x}_j\|^2 &= d_{ij}^2, \ \forall\ (i,j) \in N_x,\ i < j, \\ \|\mathbf{a}_k - \mathbf{x}_j\|^2 &= \hat{d}_{kj}^2, \ \forall\ (k,j) \in N_a, \end{aligned}$$

- Does the system have a localization or configuration for all $\mathbf{x}_j$'s?

Key Questions Related to SNL

Let the "bar system" be:

$$\begin{aligned} \|\mathbf{x}_i - \mathbf{x}_j\|^2 &= d_{ij}^2, \ \forall\ (i,j) \in N_x,\ i < j, \\ \|\mathbf{a}_k - \mathbf{x}_j\|^2 &= \hat{d}_{kj}^2, \ \forall\ (k,j) \in N_a, \end{aligned}$$

- Does the system have a localization or configuration for all $\mathbf{x}_j$'s?
- Is the localization/configuration unique, and it can be (approximately) certified?

Key Questions Related to SNL

Let the "bar system" be:

$$\begin{aligned} \|\mathbf{x}_i - \mathbf{x}_j\|^2 &= d_{ij}^2, \ \forall\ (i,j) \in N_x,\ i < j, \\ \|\mathbf{a}_k - \mathbf{x}_j\|^2 &= \hat{d}_{kj}^2, \ \forall\ (k,j) \in N_a, \end{aligned}$$

- Does the system have a localization or configuration for all $\mathbf{x}_j$'s?
- Is the localization/configuration unique, and it can be (approximately) certified?
- Is the system partially localizable with an (approximate) certification?

Outline

Matrix Representation

Find $Y = X^T X$, where X is $d \times n$, such that

$$\begin{array}{rcl} (\mathbf{0}; \mathbf{e}_i - \mathbf{e}_j)(\mathbf{0}; \mathbf{e}_i - \mathbf{e}_j)^T \bullet \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix} & = & d_{ij}^2, \ \forall\ i, j \in N_x,\ \ i < j, \\ (\mathbf{a}_k; -\mathbf{e}_j)(\mathbf{a}_k; -\mathbf{e}_j)^T \bullet \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix} & = & \hat{d}_{kj}^2, \ \forall\ k, j \in N_a. \end{array}$$

where $\mathbf{e}_j$ is the vector of all zeros except 1 at the jth position.

SDP Relaxation of SNL, Biswas and Y 2004

Relax $Y = X^T X$ to $Y \succeq X^T X$;

SDP Relaxation of SNL, Biswas and Y 2004

Relax $Y = X^T X$ to $Y \succeq X^T X$;or equivalently

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

Then, we face a standard SDP (feasibility) problem.

$$\begin{array}{rcl} (\mathbf{0}; \mathbf{e}_i - \mathbf{e}_j)(\mathbf{0}; \mathbf{e}_i - \mathbf{e}_j)^T \bullet Z & = & d_{ij}^2, \ \forall\ i,j \in N_x,\ i < j, \\ (\mathbf{a}_k; -\mathbf{e}_j)(\mathbf{a}_k; -\mathbf{e}_j)^T \bullet Z & = & \hat{d}_{kj}^2, \ \forall\ k,j \in N_a, \\ Z & \succeq & \mathbf{0}. \end{array}$$

Properties of the SDP Relaxation

- If the network is connected and it has at least one anchor, then the solution set must be bounded.

Properties of the SDP Relaxation

- If the network is connected and it has at least one anchor, then the solution set must be bounded.
- A solution matrix Z has rank at least d.

Properties of the SDP Relaxation

- If the network is connected and it has at least one anchor, then the solution set must be bounded.
- A solution matrix Z has rank at least d.
- It's d if and only if $Y = X^T X$ and it produces one localization to the original SNL problem.

Properties of the SDP Relaxation

- If the network is connected and it has at least one anchor, then the solution set must be bounded.
- A solution matrix Z has rank at least d.
- It's d if and only if $Y = X^T X$ and it produces one localization to the original SNL problem.
- If there exists a dual solution with rank n, then the rank of any primal solution Z must be d.

The Dual of the SDP Relaxation

$$
\begin{array}{ll}
\text{minimize} & I \bullet V + \sum_{i<j \in N_x} w_{ij} d_{ij}^2 + \sum_{k,j \in N_a} \hat{w}_{kj} \hat{d}_{kj}^2 \\
\text{subject to} & \begin{pmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \sum_{i<j \in N_x} w_{ij} (\mathbf{0}; \mathbf{e}_i - \mathbf{e}_j)(\mathbf{0}; \mathbf{e}_i - \mathbf{e}_j)^T \\
& + \sum_{k,j \in N_a} w_{kj} (\mathbf{a}_k; -\mathbf{e}_j)(\mathbf{a}_k; -\mathbf{e}_j)^T = S \succeq 0,
\end{array}
$$

where variable matrix $V \in \mathcal{S}^d$, variable w_{ij} is the (stress) weight on edge between $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\hat{w}_{kj}$ is the (stress) weight on edge between $\mathbf{a}_k$ and $\mathbf{x}_j$.

The Dual of the SDP Relaxation

$$
\begin{array}{ll}
\text{minimize} & I \bullet V + \sum_{i<j \in N_x} w_{ij} d_{ij}^2 + \sum_{k,j \in N_a} \hat{w}_{kj} \hat{d}_{kj}^2 \\
\text{subject to} & \begin{pmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \sum_{i<j \in N_x} w_{ij} (\mathbf{0}; \mathbf{e}_i - \mathbf{e}_j)(\mathbf{0}; \mathbf{e}_i - \mathbf{e}_j)^T \\
& + \sum_{k,j \in N_a} w_{kj} (\mathbf{a}_k; -\mathbf{e}_j)(\mathbf{a}_k; -\mathbf{e}_j)^T = S \succeq 0,
\end{array}
$$

where variable matrix $V \in \mathcal{S}^d$, variable w_{ij} is the (stress) weight on edge between $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\hat{w}_{kj}$ is the (stress) weight on edge between $\mathbf{a}_k$ and $\mathbf{x}_j$.

- The dual is strictly feasible if the network is connected and has at least one anchor.

The Dual of the SDP Relaxation

$$\begin{array}{ll} \text{minimize} & I \bullet V + \sum_{i<j \in N_x} w_{ij} d_{ij}^2 + \sum_{k,j \in N_a} \hat{w}_{kj} \hat{d}_{kj}^2 \\ \text{subject to} & \begin{pmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \sum_{i<j \in N_x} w_{ij} (\mathbf{0}; \mathbf{e}_i - \mathbf{e}_j)(\mathbf{0}; \mathbf{e}_i - \mathbf{e}_j)^T \\ & + \sum_{k,j \in N_a} w_{kj} (\mathbf{a}_k; -\mathbf{e}_j)(\mathbf{a}_k; -\mathbf{e}_j)^T = S \succeq 0, \end{array}$$

where variable matrix $V \in \mathcal{S}^d$, variable w_{ij} is the (stress) weight on edge between $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\hat{w}_{kj}$ is the (stress) weight on edge between $\mathbf{a}_k$ and $\mathbf{x}_j$.

- The dual is strictly feasible if the network is connected and has at least one anchor.
- The max-rank of any optimal dual stress matrix S is n.

The Dual of the SDP Relaxation

$$\begin{array}{ll} \text{minimize} & I \bullet V + \sum_{i<j \in N_x} w_{ij} d_{ij}^2 + \sum_{k,j \in N_a} \hat{w}_{kj} \hat{d}_{kj}^2 \\ \text{subject to} & \begin{pmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \sum_{i<j \in N_x} w_{ij} (\mathbf{0}; \mathbf{e}_i - \mathbf{e}_j)(\mathbf{0}; \mathbf{e}_i - \mathbf{e}_j)^T \\ & + \sum_{k,j \in N_a} w_{kj} (\mathbf{a}_k; -\mathbf{e}_j)(\mathbf{a}_k; -\mathbf{e}_j)^T = S \succeq 0, \end{array}$$

where variable matrix $V \in \mathcal{S}^d$, variable w_{ij} is the (stress) weight on edge between $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\hat{w}_{kj}$ is the (stress) weight on edge between $\mathbf{a}_k$ and $\mathbf{x}_j$.

- The dual is strictly feasible if the network is connected and has at least one anchor.
- The max-rank of any optimal dual stress matrix S is n.
- It's n if the problem admits strict complementarity.

Unique Localizability and Uniquely Localizable Problem I

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

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 = d_{ij}^2,\ \forall\ i,j \in N_x,\ i < j,$$
$$\|(\mathbf{a}_k; \mathbf{0}) - \mathbf{x}_j\|^2 = \hat{d}_{kj}^2,\ \forall\ k,j \in N_a.$$

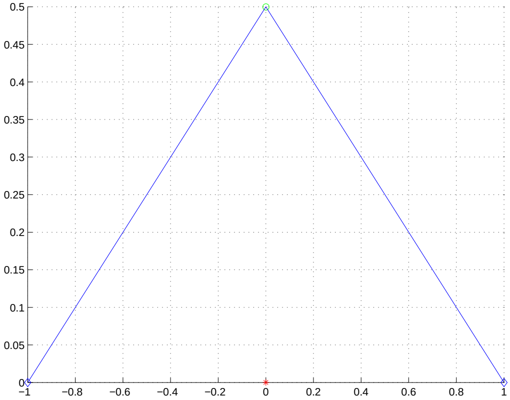
Unique Localizability and Uniquely Localizable Problem I

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

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 = d_{ij}^2,\ \forall\ i, j \in N_x,\ \ i < j,$$
$$\|(\mathbf{a}_k; \mathbf{0}) - \mathbf{x}_j\|^2 = \hat{d}_{kj}^2,\ \forall\ k, j \in N_a.$$

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

One sensor-Two anchors: Not localizable



Unique Localizability and Uniquely Localizable Problem II

- It's necessary to have $d+1$ anchors in general positions for any network to be UL.

Unique Localizability and Uniquely Localizable Problem II

- It's necessary to have $d+1$ anchors in general positions for any network to be UL.
- It's also necessary for every sensor to have at least $d+1$ edges.

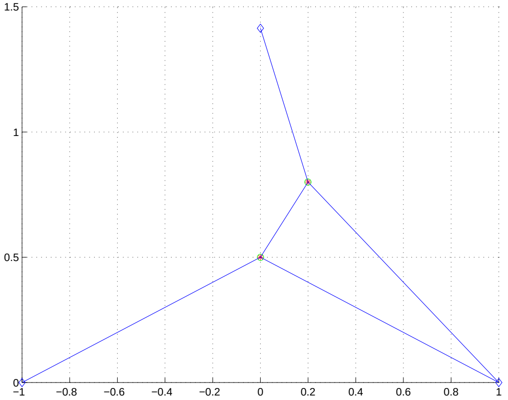
Unique Localizability and Uniquely Localizable Problem II

- It's necessary to have $d+1$ anchors in general positions for any network to be UL.
- It's also necessary for every sensor to have at least $d+1$ edges.
- For a UL instance, if there exists a dual stress matrix with rank n, then it is called strongly-localizable (SL).

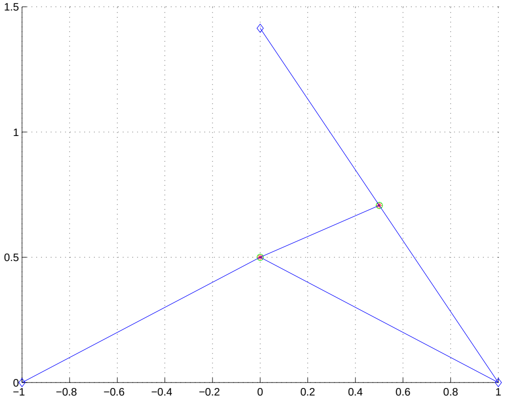
Unique Localizability and Uniquely Localizable Problem II

- It's necessary to have $d+1$ anchors in general positions for any network to be UL.
- It's also necessary for every sensor to have at least $d+1$ edges.
- For a UL instance, if there exists a dual stress matrix with rank n, then it is called strongly-localizable (SL).
- UL is stronger than the notion of *Global Rigidity*, and it is most similar to *Universal Rigidity*.

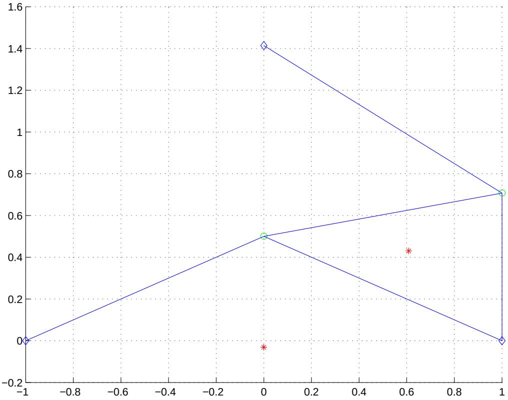
Two sensor-Three anchors: Strongly Localizable



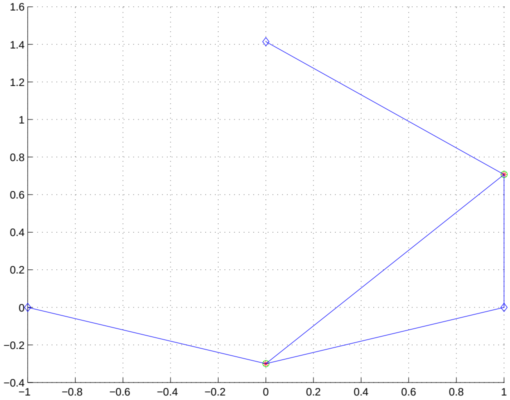
Two sensor-Three anchors: Localizable but not Strongly



Two sensor-Three anchors: Not localizable



Two sensor-Three anchors: Strongly Localizable



An Equivalence Theorem, So and Y 2005

Theorem

The following statements are equivalent:

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

An Equivalence Theorem, So and Y 2005

Theorem

The following statements are equivalent:

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

Moreover, the localization of a uniquely localizable instance can be computed approximately in a time polynomial in n, d, *and the accuracy* $\log(1/\epsilon)$.

An Equivalence Theorem, So and Y 2005

Theorem

The following statements are equivalent:

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

Moreover, the localization of a uniquely localizable instance can be computed approximately in a time polynomial in n, d, and the accuracy $\log(1/\epsilon)$.

The existence of a rank n stress matrix (or strong localizability) is sufficient but not necessary for UL.

Localize All Localizable Points, So and Y 2005

Theorem

If an SNL problem (graph) contains a subproblem (subgraph) that is strongly localizable, then the solution submatrix corresponding to the subproblem in the SDP has rank d. Thus, the SDP relaxation computes a solution that localize all possibly localizable sensor points.

Localize All Localizable Points, So and Y 2005

Theorem

If an SNL problem (graph) contains a subproblem (subgraph) that is strongly localizable, then the solution submatrix corresponding to the subproblem in the SDP has rank d. Thus, the SDP relaxation computes a solution that localize all possibly localizable sensor points.

Diagonals of the "co-variance" matrix

$$\hat{Y} - \hat{X}^T \hat{X},$$

can be a certification for uniquely localizable sensor points. That is, $\hat{Y}_{jj} - \|\hat{\mathbf{x}}_j\|^2 = 0$ if any only if the jth sensor point has a unique localization.

Uniquely-Localizable Graphs

Theorem

Let the SNL problem have at least $d+1$ anchors and they, together with all sensors, are in general positions.

- *If G is complete or every edge length is specified, then the sensor network is uniquely-localizable (Schoenberg 1942).*

Uniquely-Localizable Graphs

Theorem

Let the SNL problem have at least $d + 1$ anchors and they, together with all sensors, are in general positions.

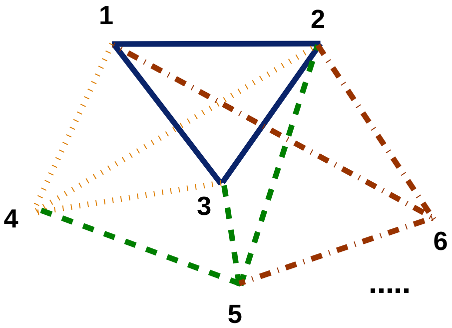
- *If G is complete or every edge length is specified, then the sensor network is uniquely-localizable (Schoenberg 1942).*
- *The $(d + 1)$-lateration graph is uniquely-localizable (So 2006 and Zhu, So and Y 2009). Moreover, it is near the sparsest graph (with only $(d + 1)n$ edges) that is uniquely-localizable.*

$(d+1)$-Lateration Graph

A $(d+1)$-lateration ordering for a graph G is an ordering of the vertexes $1, \cdots, d+1, d+2, \cdots, n$ such that the first $d+1$ vertexes form the complete graph, and every vertex $j > d+1$ has $d+1$ edges connected to its preceding vertexes on the sequence.

$(d+1)$-Lateration Graph

A $(d+1)$-lateration ordering for a graph G is an ordering of the vertexes $1, \cdots, d+1, d+2, \cdots, n$ such that the first $d+1$ vertexes form the complete graph, and every vertex $j > d+1$ has $d+1$ edges connected to its preceding vertexes on the sequence.



Universal Rigidity Theorems in Generic Position

Recall that an SNL problem is strongly localizable if there exists a max-rank n PSD stress matrix.

Universal Rigidity Theorems in Generic Position

Recall that an SNL problem is strongly localizable if there exists a max-rank n PSD stress matrix.

Let's consider a bar framework $(G, P \in \mathbf{R}^{d\times(n+d+1)})$ of $n+d+1$ vertexes in generic positions of $\mathbf{R}^d$. Then

Universal Rigidity Theorems in Generic Position

Recall that an SNL problem is strongly localizable if there exists a max-rank n PSD stress matrix.

Let's consider a bar framework $(G, P \in \mathbf{R}^{d\times(n+d+1)})$ of $n+d+1$ vertexes in generic positions of $\mathbf{R}^d$. Then

- The existence of a max-rank PSD stress matrix implies that the framework is universally rigid (Connelly 1999, Alfakih 2010).

Universal Rigidity Theorems in Generic Position

Recall that an SNL problem is strongly localizable if there exists a max-rank n PSD stress matrix.

Let's consider a bar framework $(G, P \in \mathbf{R}^{d\times(n+d+1)})$ of $n+d+1$ vertexes in generic positions of $\mathbf{R}^d$. Then

- The existence of a max-rank PSD stress matrix implies that the framework is universally rigid (Connelly 1999, Alfakih 2010).
- The framework is universally rigid if and only if there exists a max-rank PSD stress matrix (Gortler and Thurston, 2009).

Universal Rigidity Theorems in General Position

Let's consider a bar framework $(G, P \in \mathbf{R}^{d\times(n+d+1)})$ of $n+d+1$ vertexes in general positions of $\mathbf{R}^d$. Then

Universal Rigidity Theorems in General Position

Let's consider a bar framework $(G, P \in \mathbf{R}^{d\times(n+d+1)})$ of $n+d+1$ vertexes in general positions of $\mathbf{R}^d$. Then

- The existence of a max-rank PSD stress matrix implies that the framework is universally rigid (Alfakih and Y 2010).

Universal Rigidity Theorems in General Position

Let's consider a bar framework $(G, P \in \mathbf{R}^{d\times(n+d+1)})$ of $n+d+1$ vertexes in general positions of $\mathbf{R}^d$. Then

- The existence of a max-rank PSD stress matrix implies that the framework is universally rigid (Alfakih and Y 2010).
- A framework problem that contains a spanning $(d+1)$-lateration graph is universally rigid if and only if there exists a max-rank PSD stress matrix (Alfakih, Taheri and Y 2010).

Universal Rigidity Theorems in General Position

Let's consider a bar framework $(G, P \in \mathbf{R}^{d\times(n+d+1)})$ of $n + d + 1$ vertexes in general positions of $\mathbf{R}^d$. Then

- The existence of a max-rank PSD stress matrix implies that the framework is universally rigid (Alfakih and Y 2010).
- A framework problem that contains a spanning $(d + 1)$-lateration graph is universally rigid if and only if there exists a max-rank PSD stress matrix (Alfakih, Taheri and Y 2010).
- Given configuration/position matrix P and the lateration order, such a PSD stress matrix can be computed exactly in strongly polynomial time.

Proof Sketch I

Let the extended position matrix

$$A = \begin{pmatrix} P \\ \mathbf{e}^T \end{pmatrix} \in \mathbf{R}^{(d+1)\times(n+d+1)}.$$

Proof Sketch I

Let the extended position matrix

$$A = \begin{pmatrix} P \\ \mathbf{e}^T \end{pmatrix} \in \mathbf{R}^{(d+1)\times(n+d+1)}.$$

Recall that symmetric matrix S is a stress matrix if and only if

$$\text{orthogonality:} \quad AS = \mathbf{0}, \tag{1}$$

and

$$\text{purity:} \quad S_{ij} = 0, \ \forall (i,j) \notin E. \tag{2}$$

Proof Sketch II

- We start a PSD matrix satisfies orthogonality condition (1), say

$$S^0 = I - A^T(AA^T)^{-1}A,$$

where the columns of A are ordered according to the lateration order, and we call it a "prestress" matrix.

Proof Sketch II

- We start a PSD matrix satisfies orthogonality condition (1), say
$$S^0 = I - A^T(AA^T)^{-1}A,$$
where the columns of A are ordered according to the lateration order, and we call it a "prestress" matrix.
- We modify S^0 column (row) by column (row), starting from the last column (row) backward, by zeroing entries not in E from solving a linear equation system of $d+1$ variables in each step.

Proof Sketch II

- We start a PSD matrix satisfies orthogonality condition (1), say
$$S^0 = I - A^T(AA^T)^{-1}A,$$
where the columns of A are ordered according to the lateration order, and we call it a "prestress" matrix.
- We modify S^0 column (row) by column (row), starting from the last column (row) backward, by zeroing entries not in E from solving a linear equation system of $d+1$ variables in each step.
- In the modification process, we maintain the PSD, rank-n, and the orthogonality conditions of the "prestress" matrix.

Proof Sketch II

- We start a PSD matrix satisfies orthogonality condition (1), say
$$S^0 = I - A^T(AA^T)^{-1}A,$$
where the columns of A are ordered according to the lateration order, and we call it a “prestress” matrix.
- We modify S^0 column (row) by column (row), starting from the last column (row) backward, by zeroing entries not in E from solving a linear equation system of $d+1$ variables in each step.
- In the modification process, we maintain the PSD, rank-n, and the orthogonality conditions of the “prestress” matrix.
- In at most n steps, we reach a “prestress” matrix that finally satisfies purity condition (2).

Conclusions and Open Research Questions

- SDP seems to provide an efficient computation and analysis model for certain applications of the Rigidity Theory.

Conclusions and Open Research Questions

- SDP seems to provide an efficient computation and analysis model for certain applications of the Rigidity Theory.
- Does any UL graph contain at least one $d+1$ clique?

Conclusions and Open Research Questions

- SDP seems to provide an efficient computation and analysis model for certain applications of the Rigidity Theory.
- Does any UL graph contain at least one $d + 1$ clique?
- Sufficient and necessary conditions based on the max-rank condition of primal SDP matrix solutions?

Conclusions and Open Research Questions

- SDP seems to provide an efficient computation and analysis model for certain applications of the Rigidity Theory.
- Does any UL graph contain at least one $d + 1$ clique?
- Sufficient and necessary conditions based on the max-rank condition of primal SDP matrix solutions?
- Sufficient and necessary conditions of UL for points in general positions?