Universal Rigidity Theory and Semidefinite Programming for Sensor Network Localization

Yinyu Ye

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

Joint work with Biswas, So, Alfakih, Taheri, ...

September 23, 2011
Conference on Discrete Geometry and Optimization

Sensor Network Localization or Graph Realization

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

Sensor Network Localization or Graph Realization

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

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

Sensor Network Localization or Graph Realization

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

- to position 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 $d_{ij} \in E$.

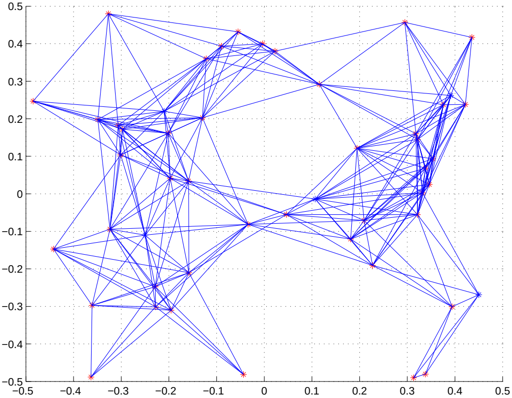
Sensor Network Localization or Graph Realization

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

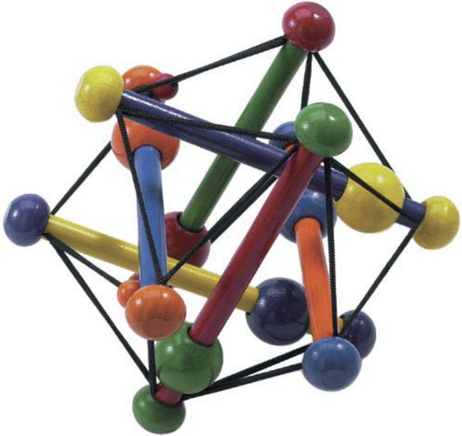
- to position the vertexes of G in $\mathbf{R}^d$ such that
- the Euclidean distance between a pair of adjacent vertexes (i,j) equals to (or bounded by) the prescribed weight $d_{ij} \in E$.

Sometimes, the positions of a few vertexes are known and they are called anchors.

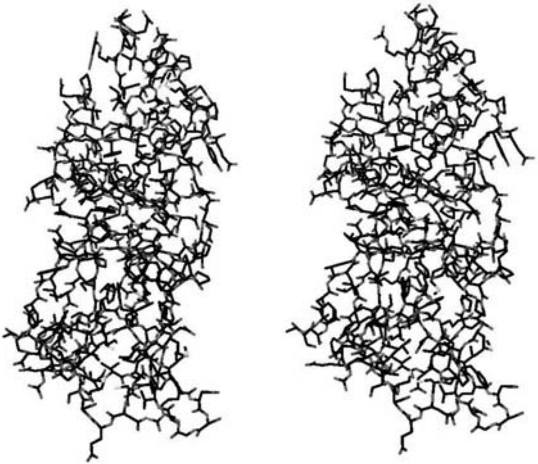
50-vertex 2-D Sensor Network Localization



3-D Tensegrity Network; a Toy Example by Anstreicher



Molecular Conformation: 1F39(1534 atoms) with 85% of distances below 6Å and 10% noises



Quadratic Equality and Inequality Systems

Given network (G, D), 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 network (G, D), 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 anchors $\mathbf{a}_k \in \mathbf{R}^d$, $d_{ij} \in N_x$, and $\hat{d}_{kj} \in N_a$, find $\mathbf{x}_i \in \mathbf{R}^d$ such that

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

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

Consider a bar SNL problem:

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

Key Questions Related to SNL

Consider a bar SNL problem:

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

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

Key Questions Related to SNL

Consider a bar SNL problem:

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

- Does the network have a solution or localization for all $\mathbf{x}_j$'s?
- Given a localization $\mathbf{x}_i = \mathbf{p}_i \in \mathbf{R}^d, \ i = 1, \ldots, n$, is the localization unique (up to a rigid motion when anchor free), and can the uniqueness be computationally certified?

Key Questions Related to SNL

Consider a bar SNL problem:

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

- Does the network have a solution or localization for all $\mathbf{x}_j$'s?
- Given a localization $\mathbf{x}_i = \mathbf{p}_i \in \mathbf{R}^d, \ i = 1, \ldots, n$, is the localization unique (up to a rigid motion when anchor free), and can the uniqueness be computationally certified?
- Is the network partially localizable with a certification?

Rigidity Notions for SNL, Connelly ...

- Global Rigidity (GR): the network has a unique localization $\mathbf{p}_i$s in $\mathbf{R}^d$.

Rigidity Notions for SNL, Connelly ...

- Global Rigidity (GR): the network has a unique localization $\mathbf{p}_i$s in $\mathbf{R}^d$.
- Universal Rigidity (UR): the network has a unique localization in all dimensions.

Rigidity Notions for SNL, Connelly ...

- Global Rigidity (GR): the network has a unique localization $\mathbf{p}_i$s in $\mathbf{R}^d$.
- Universal Rigidity (UR): the network has a unique localization in all dimensions.
- Strong Rigidity (SR): let $P = [\mathbf{p}_1, \ldots, \mathbf{p}_n]$ be a localization and $\mathbf{e}$ be the vector of all ones, and extended matrix

$$A = \begin{bmatrix} P \\ e^T \end{bmatrix}.$$

Then, there is a rank $n - d - 1$ and positive semidefinite stress matrix for SNL such that

$$S_{ij} = 0, \ \forall (i,j) \notin E(G), \quad \text{and} \quad AS = \mathbf{0}.$$

Rigidity Notions for SNL, Connelly ...

- Global Rigidity (GR): the network has a unique localization $\mathbf{p}_i$s in $\mathbf{R}^d$.
- Universal Rigidity (UR): the network has a unique localization in all dimensions.
- Strong Rigidity (SR): let $P = [\mathbf{p}_1, \ldots, \mathbf{p}_n]$ be a localization and $\mathbf{e}$ be the vector of all ones, and extended matrix

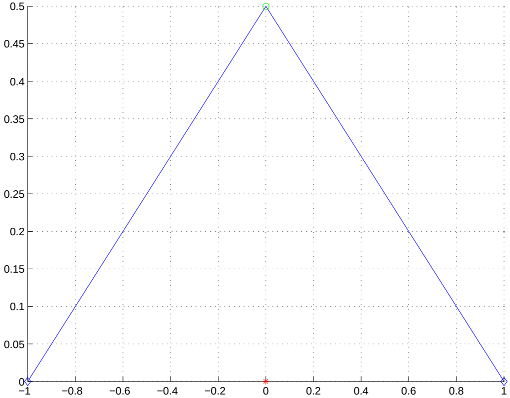
$$A = \begin{bmatrix} P \\ e^T \end{bmatrix}.$$

Then, there is a rank $n - d - 1$ and positive semidefinite stress matrix for SNL such that

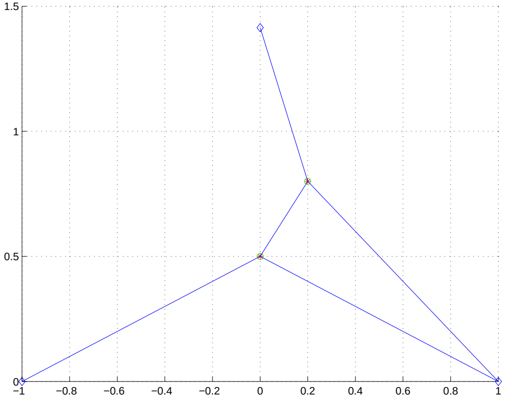
$$S_{ij} = 0, \ \forall (i,j) \notin E(G), \quad \text{and} \quad AS = \mathbf{0}.$$

Similar rigidity notions hold for SNL with anchors.

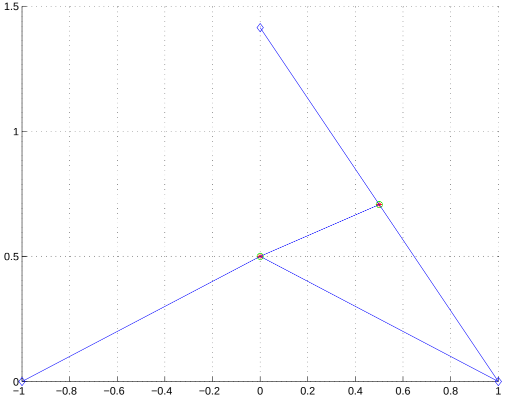
UR (not even GR when with anchors in 2-D)



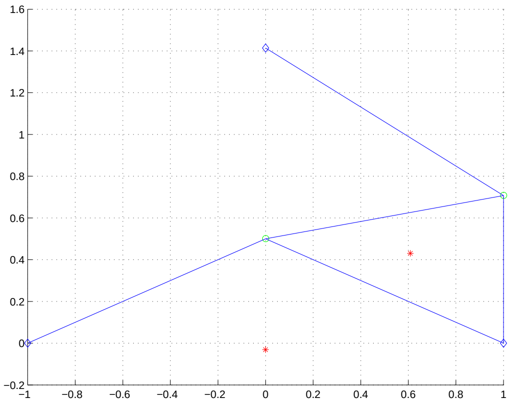
UR and SR



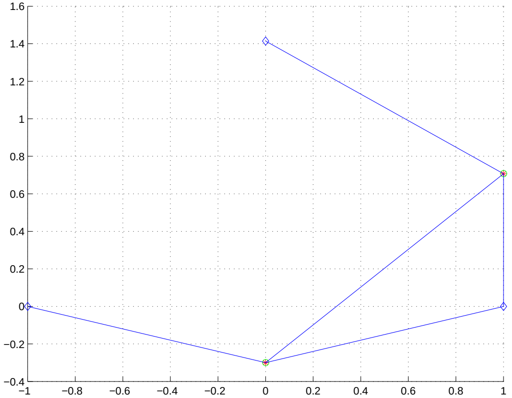
UR but not SR



GR but not UR



UR and SR



Relation of Rigidity Notions

- NP-hardness related to GR (Saxe 79).

Relation of Rigidity Notions

- NP-hardness related to GR (Saxe 79).
- UR implies GR.

Relation of Rigidity Notions

- NP-hardness related to GR (Saxe 79).
- UR implies GR.
- SR implies UR when P is in generic positions (Connelly 99, also see Alfakih 10).

Relation of Rigidity Notions

- NP-hardness related to GR (Saxe 79).
- UR implies GR.
- SR implies UR when P is in generic positions (Connelly 99, also see Alfakih 10).
- It's necessary to have $d + 1$ anchors in general positions to be UR for SNL with anchors, and then SR implies UR (So and Y, 05).

Semidefinite Programming Problem (SDP)

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

where $C, A_1, \ldots, A_m$ are given dimension n real symmetric matrices with real scalars $\mathbf{b} = [b_1, \ldots, b_m]$,

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

and $\succeq$ represents positive semi-definiteness.

Semidefinite Programming Problem (SDP)

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

where $C, A_1, \dots, A_m$ are given dimension n real symmetric matrices with real scalars $\mathbf{b} = [b_1, \dots, b_m]$,

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

and $\succeq$ represents positive semi-definiteness.
The dual problem to (SDP):

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

where variables $\mathbf{y} = [y_1, \dots, y_m] \in \mathbf{R}^m$.

SDP Solution Rank

- There are optimal solutions of X^* or S^* such that the rank of X^* or S^* is minimal.

SDP Solution Rank

- There are optimal solutions of X^* or S^* such that the rank of X^* or S^* is minimal.
- There are optimal solutions of X^* or S^* such that the rank of X^* or S^* is maximal.

SDP Solution Rank

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

SDP Solution Rank

- There are optimal solutions of X^* or S^* such that the rank of X^* or S^* is minimal.
- There are optimal solutions of X^* or S^* such that the rank of X^* or S^* is maximal.
- But the sum of the two ranks should be no more than n when they are complementary. If the sum is n, then the problem admits a strictly complementary solution pair.
- 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^* or S^* such that the rank of X^* or S^* is minimal.
- There are optimal solutions of X^* or S^* such that the rank of X^* or S^* is maximal.
- But the sum of the two ranks should be no more than n when they are complementary. If the sum is n, then the problem admits a strictly complementary solution pair.
- 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 Computational Complexity and Solution Rank

- Let the SDP problem have a finite complementary 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 complementary 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 solution pair $(X(\epsilon), S(\epsilon))$ computed by the interior-point algorithm converges to a max-rank solution pair for both the primal and dual (Güler and Y 93, Y 95).

SDP Computational Complexity and Solution Rank

- Let the SDP problem have a finite complementary 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 solution pair $(X(\epsilon), S(\epsilon))$ computed by the interior-point algorithm converges to a max-rank solution pair for both the primal and dual (Güler and Y 93, Y 95).
- But finding a min-rank SDP solution is strongly NP-Hard.

Matrix Representation of SNL

Find $Y = X^T X$, where $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ is $d \times n$, such that

$$(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T \bullet Y \quad = \quad d_{ij}^2, \; \forall \; i,j \in N_x, \;\; i < j,$$

Matrix Representation of SNL

Find $Y = X^T X$, where $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ is $d \times n$, such that

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

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

SDP Relaxations of SNL, Biswas and Y 04, Biswas, Toh and Y 06

Relax $Y = X^T X$ to $Y \succeq X^T X \succeq \mathbf{0}$;

SDP Relaxations of SNL, Biswas and Y 04, Biswas, Toh and Y 06

Relax $Y = X^T X$ to $Y \succeq X^T X \succeq \mathbf{0}$; or equivalently

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

Then, we face a standard SDP (feasibility) problem for SNL

$$\begin{aligned} (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T \bullet Y &= d_{ij}^2, \ \forall\ i, j \in N_x, \ \ i < j, \\ Y &\succeq \mathbf{0}. \end{aligned}$$

SDP Relaxations of SNL, Biswas and Y 04, Biswas, Toh and Y 06

Relax $Y = X^T X$ to $Y \succeq X^T X \succeq \mathbf{0}$; or equivalently

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

Then, we face a standard SDP (feasibility) problem for SNL

$$\begin{aligned} (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T \bullet Y &= d_{ij}^2, \ \forall\ i,j \in N_x,\ \ i<j, \\ Y &\succeq \mathbf{0}. \end{aligned}$$

or SNL with anchors

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

Properties of the SDP Relaxation

Given a framework/network (G, P), we have

- $Y = A^T A$ ($X = P,\ Y = P^T P$) is an SDP solution for the SDP relaxation of SNL (SNL with anchors).

Properties of the SDP Relaxation

Given a framework/network (G, P), we have

- $Y = A^T A$ ($X = P, \ Y = P^T P$) is an SDP solution for the SDP relaxation of SNL (SNL with anchors).
- If P is in general positions, then the SDP solution $Y = A^T A$ has rank $d + 1$ so that any dual solution matrix has rank no more than $n - d - 1$.

Properties of the SDP Relaxation

Given a framework/network (G, P), we have

- $Y = A^T A$ ($X = P,\ Y = P^T P$) is an SDP solution for the SDP relaxation of SNL (SNL with anchors).
- If P is in general positions, then the SDP solution $Y = A^T A$ has rank $d + 1$ so that any dual solution matrix has rank no more than $n - d - 1$.
- For SNL with anchors

Properties of the SDP Relaxation

Given a framework/network (G, P), we have

- $Y = A^T A$ $(X = P,\ Y = P^T P)$ is an SDP solution for the SDP relaxation of SNL (SNL with anchors).
- If P is in general positions, then the SDP solution $Y = A^T A$ has rank $d + 1$ so that any dual solution matrix has rank no more than $n - d - 1$.
- For SNL with anchors
  - Any solution matrix Z has rank at least d so that any dual solution matrix has rank no more than n.

Properties of the SDP Relaxation

Given a framework/network (G, P), we have

- $Y = A^T A$ ($X = P,\ Y = P^T P$) is an SDP solution for the SDP relaxation of SNL (SNL with anchors).
- If P is in general positions, then the SDP solution $Y = A^T A$ has rank $d + 1$ so that any dual solution matrix has rank no more than $n - d - 1$.
- For SNL with anchors
  - Any solution matrix Z has rank at least d so that any dual solution matrix has rank no more than n.
  - It's d if and only if $Y = X^T X$ and then X is a localization.

Properties of the SDP Relaxation

Given a framework/network (G, P), we have

- $Y = A^T A$ ($X = P,\ Y = P^T P$) is an SDP solution for the SDP relaxation of SNL (SNL with anchors).
- If P is in general positions, then the SDP solution $Y = A^T A$ has rank $d + 1$ so that any dual solution matrix has rank no more than $n - d - 1$.
- For SNL with anchors
  - Any solution matrix Z has rank at least d so that any dual solution matrix has rank no more than n.
  - It's d if and only if $Y = X^T X$ and then X is a localization.
  - If there exists a dual solution matrix with rank n, then the rank of any primal solution Z must be d.

The Dual of the SDP Relaxations

$$\begin{array}{ll} \text{minimize} & \sum_{i<j\in N_x} w_{ij} d_{ij}^2 \\ \text{subject to} & \sum_{i<j\in N_x} w_{ij}(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T = S \succeq 0, \end{array}$$

The Dual of the SDP Relaxations

$$
\begin{array}{ll}
\text{minimize} & \sum_{i<j\in N_x} w_{ij} d_{ij}^2 \\
\text{subject to} & \sum_{i<j\in N_x} w_{ij}(\mathbf{e}_i-\mathbf{e}_j)(\mathbf{e}_i-\mathbf{e}_j)^T = S \succeq 0,
\end{array}
$$

or

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

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 $\mathbf{x}_j$.

Benefits of SDP Relaxation, So and Y 05, Biswas, Toh and Y 06

Whether or not a network (G, D) (with or without anchors) is UR can be (numerically) certified in polynomial time.

Benefits of SDP Relaxation, So and Y 05, Biswas, Toh and Y 06

Whether or not a network (G, D) (with or without anchors) is UR can be (numerically) certified in polynomial time.

Whether or not a network (G, D) (with or without anchors) is SR can be (numerically) certified in polynomial time.

Benefits of SDP Relaxation, So and Y 05, Biswas, Toh and Y 06

Whether or not a network (G, D) (with or without anchors) is UR can be (numerically) certified in polynomial time.

Whether or not a network (G, D) (with or without anchors) is SR can be (numerically) certified in polynomial time.

http://www.stanford.edu/~yyye/URFrameworkTest.m
http://www.math.nus.edu.sg/~mattohkc/disco.html
http://www.stanford.edu/~yyye/Col.html

An Equivalence Theorem, Biswas and Y 04, So and Y 05

Theorem

The following statements are equivalent for SNL with anchors:

1. *The sensor network is UR;*
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, Biswas and Y 04, So and Y 05

Theorem

The following statements are equivalent for SNL with anchors:

1. *The sensor network is UR;*
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 UR instance can be computed approximately in a time polynomial in n, d*, and the accuracy* $\log(1/\epsilon)$.

Identify the Largest UR Subnetwork, So and Y 05

Theorem

If a network with anchors contains a subnetwork that is UR, then the SDP solution submatrix corresponding to the subnetwork has rank d. Thus, the SDP relaxation method finds the localization of the largest UR subnetwork for SNL with anchors.

Certification: Diagonals of the positive semidefinite matrix

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

can be used to certify the UR subnetwork; that is, $\hat{Y}_{jj} - \|\hat{\mathbf{x}}_j\|^2 = 0$ if any only if the jth sensor point is in the UR subnetwork.

The Dual Matrix Theorem, So and Y 07

Theorem

Any optimal dual solution matrix is a positive semidefinite stress matrix for SNL or SNL with anchors. Therefore, a max-rank positive semidefinite stress matrix can be computed approximately in a time polynomial in n, d, and the accuracy $\log(1/\epsilon)$.

UR Theorems in Generic Position, Gortler and Thurston 09

Theorem

Let the network possess a localization P in generic positions of $\mathbf{R}^d$. Then, the network is UR if and only if there exists a max-rank positive semidefinite stress matrix, that is, the network is SR.

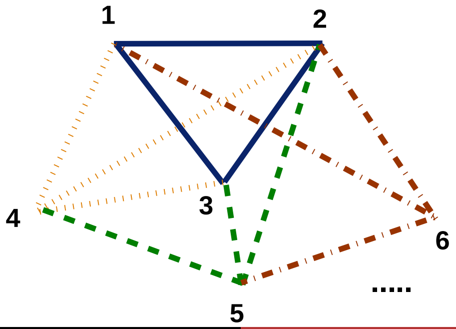
UR Theorems in General Position I

Let the network possess a localization P in general positions of $\mathbf{R}^d$. Then the network is UR if the graph contains a spanning $(d+1)$-lateration graph for SNL with or without anchors (So 06 and Zhu, So and Y 09).

UR Theorems in General Position I

Let the network possess a localization P in general positions of $\mathbf{R}^d$. Then the network is UR if the graph contains a spanning $(d+1)$-lateration graph for SNL with or without anchors (So 06 and Zhu, So and Y 09).

A $(d+1)$-lateration graph:



UR Theorems in General Position II

Let the network possess a localization P in general positions of $\mathbf{R}^d$. Then

UR Theorems in General Position II

Let the network possess a localization P in general positions of $\mathbf{R}^d$. Then

- The existence of a max-rank stress matrix implies that the network is UR, that is, SR implies UR (Alfakih and Y 10).

UR Theorems in General Position II

Let the network possess a localization P in general positions of $\mathbf{R}^d$. Then

- The existence of a max-rank stress matrix implies that the network is UR, that is, SR implies UR (Alfakih and Y 10).
- A network that contains a spanning $(d+1)$-lateration graph is UR if and only if it is SR (Alfakih, Taheri and Y 10), and the same result holds for SNL with anchors.

UR Theorems in General Position II

Let the network possess a localization P in general positions of $\mathbf{R}^d$. Then

- The existence of a max-rank stress matrix implies that the network is UR, that is, SR implies UR (Alfakih and Y 10).
- A network that contains a spanning $(d+1)$-lateration graph is UR if and only if it is SR (Alfakih, Taheri and Y 10), and the same result holds for SNL with anchors.
- Given localization matrix P and the lateration order, such a max-rank stress matrix can be computed exactly in strongly polynomial time (Alfakih, Taheri and Y 10).

Proof Sketch I

Recall 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

Recall 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

- SNL has benefited a lot from Rigidity Notion and Theory, and SDP seems also to provide an efficient computation and analysis model for applications related to Rigidity Theory.

Conclusions and Open Research Questions

- SNL has benefited a lot from Rigidity Notion and Theory, and SDP seems also to provide an efficient computation and analysis model for applications related to Rigidity Theory.
- A network is UR if and only if it is SR under the general position assumption?

Conclusions and Open Research Questions

- SNL has benefited a lot from Rigidity Notion and Theory, and SDP seems also to provide an efficient computation and analysis model for applications related to Rigidity Theory.
- A network is UR if and only if it is SR under the general position assumption?
- Other checkable sufficient and necessary conditions for UR networks?

Conclusions and Open Research Questions

- SNL has benefited a lot from Rigidity Notion and Theory, and SDP seems also to provide an efficient computation and analysis model for applications related to Rigidity Theory.
- A network is UR if and only if it is SR under the general position assumption?
- Other checkable sufficient and necessary conditions for UR networks?
- SNL based on other metric measurements: angles, path-distances, time-series data, etc.