Global Optimization Using Semidefinite Programming Relaxation

Masakazu Kojima

Tokyo Institute of Technology

Nonlinear Analysis and Convex Analysis

Tokyo Institute of Technology, Tokyo
March 27-31, 2009
Purpose of this talk —
Introduction to Semidefinite Programming Relaxation for Polynomial Optimization Problems

Contents

1. Global optimization of nonconvex problems
   1-1 Polynomial Optimization Problems (POPs)
   1-2 SemiDefinite Programs (SDPs)
2. SDP relaxation
3. Exploiting sparsity in SDP relaxation
4. Numerical results
5. Concluding remarks
Contents

1. Global optimization of nonconvex problems
   1-1 Polynomial Optimization Problems (POPs)
   1-2 SemiDefinite Programs (SDPs)
2. SDP relaxation
3. Exploiting sparsity in SDP relaxation
4. Numerical results
5. Concluding remarks
**OP**: Optimization problem in the $n$-dim. Euclidean space $\mathbb{R}^n$.

\[
\min_{x \in S \subseteq \mathbb{R}^n} f(x), \text{ sub.to } x \in S \subseteq \mathbb{R}^n, \text{ where } f : \mathbb{R}^n \to \mathbb{R}.
\]

We want to approximate a global optimal solution $x^*$;

\[
x^* \in S \text{ and } f(x^*) \leq f(x) \text{ for all } x \in S
\]

if it exists. But, impossible without any assumption.

Various assumptions

- continuity, differentiability, compactness, \ldots
- convexity $\Rightarrow$ local opt. sol. $\equiv$ global opt. sol.
  $\Rightarrow$ local improvement leads to a global opt. sol.
- Powerful software for convex problems $\ni$ LPs, SDPs, \ldots
OP: Optimization problem in the \( n \)-dim. Euclidean space \( \mathbb{R}^n \)

\[
\min \ f(x) \ \text{sub.to} \ x \in S \subseteq \mathbb{R}^n, \ \text{where} \ f : \mathbb{R}^n \rightarrow \mathbb{R}.
\]

We want to approximate a global optimal solution \( x^* \);

\[ x^* \in S \text{ and } f(x^*) \leq f(x) \text{ for all } x \in S \]

if it exists. But, impossible without any assumption.

Various assumptions

- continuity, differentiability, compactness, \ldots
- convexity \( \Rightarrow \) local opt. sol. \( \equiv \) global opt. sol.
  \( \Rightarrow \) local improvement leads to a global opt. sol.
- Powerful software for convex problems \( \ni \) LPs, SDPs, \ldots

What can we do beyond convexity?

- We still need proper models and assumptions
  - Polynomial Optimization Problems (POPs) — this talk
- Main tool is SDP relaxation — this talk
  Powerful in theory but expensive in practice
- Exploiting sparsity in large scale SDPs & POPs — this talk
Contents

1. Global optimization of nonconvex problems
   1-1 Polynomial Optimization Problems (POPs)
   1-2 SemiDefinite Programs (SDPs)

2. SDP relaxation

3. Exploiting sparsity in SDP relaxation

4. Numerical results

5. Concluding remarks
\( \mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n \): a vector variable

\( f_j(\mathbf{x}) \): a real-valued polynomial in \( x_1, \ldots, x_n \) \( (j = 0, 1, \ldots, m) \)

| POP: \( \min f_0(\mathbf{x}) \) sub.to \( f_j(\mathbf{x}) \geq 0 \) or \( = 0 \) \( (j = 1, \ldots, m) \) |

Example. \( n = 3, \mathbf{x} = (x_1, x_2, x_3) \in \mathbb{R}^3 \): a vector variable

\[
\begin{align*}
\min & \quad f_0(\mathbf{x}) \equiv x_1^3 - 2x_1x_2^2 + x_1^2x_2x_3 - 4x_3^2 \\
\text{sub.to} & \quad f_1(\mathbf{x}) \equiv -x_1^2 + 5x_2x_3 + 1 \geq 0, \\
& \quad f_2(\mathbf{x}) \equiv x_1^2 - 3x_1x_2x_3 + 2x_3 + 2 \geq 0, \\
& \quad f_3(\mathbf{x}) \equiv -x_1^2 - x_2^2 - x_3^2 + 1 \geq 0. \\
\end{align*}
\]

\( x_1(x_1 - 1) = 0 \) (0-1 integer cond.),

\( x_2 \geq 0, \ x_3 \geq 0, \ x_2x_3 = 0 \) (comp. cond.).

- Various problems (including 0-1 integer programs) \( \Rightarrow \) POP

- POP serves as a unified theoretical model for global optimization in nonlinear and combinatorial optimization problems.
Contents

1. Global optimization of nonconvex problems
   1-1 Polynomial Optimization Problems (POPs)
   1-2 SemiDefinite Programs (SDPs)

2. SDP relaxation

3. Exploiting sparsity in SDP relaxation

4. Numerical results

5. Concluding remarks
SDP is an extension of Linear Program (LP)

**LP:**

minimize \(-x_1 - 2x_2 - 5x_3\)

subject to

\[2x_1 + 3x_2 + x_3 = 7, \quad x_1 + x_2 \geq 1,\]

\[x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.\]

**SDP:**

minimize \(-x_1 - 2x_2 - 5x_3\)

subject to

\[2x_1 + 3x_2 + x_3 = 7, \quad x_1 + x_2 \geq 1,\]

\[x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0,\]

\[
\begin{pmatrix}
  x_1 & x_2 \\
  x_2 & x_3
\end{pmatrix} \succeq O \text{ (positive semidefinite)}.
\]

- **common:** a linear objective function in \(x_1, x_2, x_3\)
- **common:** linear equality/inequality constraints in \(x_1, x_2, x_3\)
- **difference:** SDP can have positive semidefinite constraints
- **difference in their feasible regions:**
  - polyhedral set VS nonpolyhedral convex set
- **common:** the primal-dual interior-point method
Contents

1. Global optimization of nonconvex problems
   1-1 Polynomial Optimization Problems (POPs)
   1-2 SemiDefinite Programs (SDPs)
2. SDP relaxation — Lasserre 2001
3. Exploiting sparsity in SDP relaxation
4. Numerical results
5. Concluding remarks

Three ways of describing the SDP relaxation by Lasserre:
- Sum of squares of polynomials
- Linearization of polynomial SDPs
- Probability measure and its moments ⇒ this talk
\( \mu \): a probability measure on \( \mathbb{R}^n \). We assume \( n = 2 \) in this talk.

For every \( r = 0, 1, 2, \ldots \), define

\[
\mathbf{u}_r(x) = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2, x_1^3, \ldots, x_r^r) : \text{row vector (all monomials with degree } \leq r)\]

\[
M_r(y) = \int_{\mathbb{R}^2} \mathbf{u}_r(x)^T \mathbf{u}_r(x) d\mu \quad \text{(moment matrix, symmetric, positive semidefinite)}
\]

\[
y_{\alpha\beta} = \int_{\mathbb{R}^2} x_1^\alpha x_2^\beta d\mu = (\alpha, \beta)\text{-element depending on } \mu, \ y_{00} = 1
\]

Example with \( r = 2 \):

\[
M_r(y) = \begin{bmatrix}
y_{00} & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\
y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\
y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\
y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\
y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\
y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04}
\end{bmatrix}, \ y_{00} = 1
\]

\[y_{21} = \int_{\mathbb{R}^2} x_1^2 x_2 d\mu\]
\( \mu : \) a probability measure on \( \mathbb{R}^n \). We assume \( n = 2 \) in this talk. For every \( r = 0, 1, 2, \ldots \), define

\[
u_r(x) = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, \ldots, x_2^r) : \text{row vector}
\]

(all monomials with degree \( \leq r \))

\[
M_r(y) = \int_{\mathbb{R}^2} \nu_r(x)^T \nu_r(x) d\mu
\]

(moment matrix, symmetric, positive semidefinite)

\[
y_{\alpha\beta} = \int_{\mathbb{R}^2} x_1^\alpha x_2^\beta d\mu = (\alpha, \beta)\text{-element depending on } \mu, \ y_{00} = 1
\]

\[
\begin{array}{c}
\mu : \text{a probability measure on } \mathbb{R}^2 \\
\downarrow
\end{array}
\]

\[
y_{00} = 1, \ M_r(y) \succeq \mathbf{0} \text{ (positive semidefinite)} (r = 1, 2, \ldots)
\]

We will use this necessary cond. with a finite \( r \) for \( \mu \) to be a probability measure in relaxation of a POP \( \Rightarrow \) next slide.
**SDP relaxation (Lasserre ’01) of a POP — an example**

**POP:** \(\min f_0(x) = x_1^4 - 2x_1x_2 \)  
opt. val. \( \zeta^* : \text{unknown} \)

sub. to \( x \in S \equiv \{ x \in \mathbb{R}^2 : f_1(x) = 1 - x_1^2 - x_2^2 \geq 0, f_2(x) = x_1 \geq 0 \} \).

\[
\min \int f_0(x) d\mu 
\text{sub. to } \mu : \text{a prob. meas. on } S.
\]

\[
\downarrow y_{\alpha\beta} = \int_{\mathbb{R}^2} x_1^\alpha x_2^\beta d\mu
\]

\[
\min y_{40} - 2y_{11} \quad \Rightarrow \text{SDP relaxation, opt. val. } \zeta_r \leq \zeta^*
\]

sub. to “a certain moment cond. with a parameter \( r \) for \( \mu \) to be a probability measure on \( S \)” \( \Rightarrow \) next slide

- \( \zeta_r \leq \zeta_{r+1} \leq \zeta^* \), and \( \zeta_r \to \zeta^* \) as \( r \to \infty \) under a moderate assumption that requires \( S \) is bounded (Lasserre ’01).
- We can apply **SDP relaxation** to general POPs in \( \mathbb{R}^n \).
SDP relaxation (Lasserre ’01) of a POP — an example

\[ r = 2 \]

\[
\begin{align*}
\min y_{40} - 2y_{11} & \quad \text{s.t.} \\
1 - x_1^2 - x_2^2 & \geq 0 \Rightarrow \\
& \quad \int \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ x_2^2 \end{pmatrix}^T \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ x_2^2 \end{pmatrix} d\mu \succeq 0, \\
& \quad \int \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ x_2^2 \end{pmatrix}^T \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ x_2^2 \end{pmatrix} (1 - x_1^2 - x_2^2) d\mu \succeq 0, \\
& \quad y_{\alpha\beta} = \int_{\mathbb{R}^2} x_1^\alpha x_2^\beta d\mu
\end{align*}
\]
SDP relaxation (Lasserre ’01) of a POP — an example

\( r = 2 \)

\[
\begin{align*}
\min & \quad y_{40} - 2y_{11} \\
\text{s.t.} & \quad \begin{pmatrix}
    y_{10} & y_{20} & y_{11} \\
    y_{20} & y_{30} & y_{21} \\
    y_{11} & y_{21} & y_{12}
\end{pmatrix} \succeq O,
\end{align*}
\]

\[
\begin{pmatrix}
    1 - y_{20} - y_{02} & y_{10} - y_{30} - y_{12} & y_{01} - y_{21} - y_{03} \\
    y_{10} - y_{30} - y_{12} & y_{20} - y_{40} - y_{22} & y_{11} - y_{31} - y_{13} \\
    y_{01} - y_{21} - y_{03} & y_{11} - y_{31} - y_{13} & y_{02} - y_{22} - y_{04}
\end{pmatrix} \succeq O,
\]

(moment matrix)

\[
\begin{pmatrix}
    1 & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\
    y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\
    y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\
    y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\
    y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\
    y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04}
\end{pmatrix} \succeq O.
\]

We can apply SDP relaxation to general POPs in \( \mathbb{R}^n \).

Powerful in theory but very expensive in computation.

\( \Rightarrow \) Exploiting sparsity is crucial in practice.
Contents

1. Global optimization of nonconvex problems
   1-1 Polynomial Optimization Problems (POPs)
   1-2 SemiDefinite Programs (SDPs)

2. SDP relaxation

3. Exploiting sparsity in SDP relaxation
   Joint work by S. Kim, M. Kojima, M. Muramatsu, H. Waki

4. Numerical results

5. Concluding remarks
\[ \mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n : \text{a vector variable} \]
\[ f_j(\mathbf{x}) : \text{a real-valued polynomial w. deg } \leq q \ (j = 0, 1, \ldots, m) \]
POP: \[ \min f_0(\mathbf{x}) \text{ sub.to } f_j(\mathbf{x}) \geq \text{ or } = 0 \ (j = 1, \ldots, m) \]

\[ \mathcal{F}^* = \text{the set of all monomials with deg } \leq q; \ #\mathcal{F}^* = \binom{n + q}{q} \]
\[ \mathcal{F}^* \supseteq \mathcal{F}_j = \text{the set of monomials involved in } f_j \]

\begin{align*}
\min f_0 &= -6.3x_5x_8 + 5.04x_2 + 0.35x_3 + x_4 + 3.36x_6 \\
\text{subject to} \quad f_1 &= -0.820x_2 + x_5 - 0.820x_6 = 0 \\
f_2 &= -x_2x_9 + 10x_3 + x_6 = 0 \\
f_3 &= 0.98x_4 - x_7(0.01x_5x_{10} + x_4) = 0, \quad \text{lbdi} \leq x_i \leq \text{ubdi} \\
f_4 &= x_5x_{12} - x_2(1.12 + 0.132x_9 - 0.0067x_9^2) = 0 \\
f_5 &= x_8x_{13} - 0.01x_9(1.098 - 0.038x_9) - 0.325x_7 - 0.574 = 0 \\
f_6 &= x_{10}x_{14} + 22.2x_{11} - 35.82 = 0 \\
f_7 &= x_1x_{11} - 3x_8 - 1.33 = 0
\end{align*}

- \( n = 14 \) variables. polynomials with deg \( \leq q = 3 \); \( \#\mathcal{F}^* = 680 \)
- \( \forall f_j \) involves less than 6 monomials + structured sparsity
- \( H f_0(\mathbf{x}) : \text{Hessian mat.}, F(\mathbf{x}) = (f_1, \ldots, f_7)^T, DF(\mathbf{x}) : 7 \times 14 \) Jacobian mat.. Sparsity pattern of \( H f_0 + DF^TDF \Rightarrow \)
Sparsity pattern of $Hf_0 + DF^T DF$ with simultaneous row and column reordering (Reverse Cuthill-McKee ordering)

Structured sparsity
- Sparse (symbolic) Cholesky factorization
- Also, characterized by a sparse chordal graph structure
\( \mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n \): a vector variable

\( f_j(\mathbf{x}) \): a real-valued polynomial w. \( \deg \leq r \) \( (j = 0, 1, \ldots, m) \)

\text{POP: min} \ f_0(\mathbf{x}) \ \text{sub.to} \ f_j(\mathbf{x}) \geq \text{ or } = 0 \ (j = 1, \ldots, m)

\( \mathcal{F}^* = \) the set of all monomials with \( \deg \leq r \); \( \# \mathcal{F}^* = \binom{n + r}{r} \)

\( \mathcal{F}^* \supseteq \mathcal{F}_j = \) the set of monomials involved in \( f_j \)

\textbf{Structured sparsity condition}

(a) \( \mathcal{F}_j \) does not involve many monomials.

(b) \( \{\mathcal{F}_j : j = 0, \ldots, n\} \) satisfy a cond. characterized by a chordal graph.

Sparse SDP relaxation proposed in Waki-Kim-Kojima-Muramatsu 2007

\( \downarrow \)

Original SDP relaxation Lasserre 2001

\begin{itemize}
  \item SDP is “smaller”, and “more efficient” than dense SDP
  \item Theoretical convergence to the opt. val. of POP
\end{itemize}
1. Global optimization of nonconvex problems
   1-1 Polynomial Optimization Problems (POPs)
   1-2 SemiDefinite Programs (SDPs)
2. SDP relaxation
3. Exploiting sparsity in SDP relaxation
4. Numerical results
5. Concluding remarks

In both cases, the SDP relaxation problems were solved by a MATLAB software SeDuMi developed by Sturm.
P1: a POP alkyl from globalib — presented previously
\[
\begin{align*}
\text{min} & & -6.3x_5x_8 + 5.04x_2 + 0.35x_3 + x_4 + 3.36x_6 \\
\text{sub.to} & \ & -0.820x_2 + x_5 - 0.820x_6 = 0, \\
& & 0.98x_4 - x_7(0.01x_5x_{10} + x_4) = 0, \\
& & -x_2x_9 + 10x_3 + x_6 = 0, \\
& & x_5x_{12} - x_2(1.12 + 0.132x_9 - 0.0067x_9^2) = 0, \\
& & x_8x_{13} - 0.01x_9(1.098 - 0.038x_9) - 0.325x_7 = 0.574, \\
& & x_{10}x_{14} + 22.2x_{11} = 35.82, \\
& & x_{11} - 3x_8 = -1.33, \\
\end{align*}
\]
\(lbd_i \leq x_i \leq ubd_i \quad (i = 1, 2, \ldots, 14)\).

<table>
<thead>
<tr>
<th>Sparse</th>
<th>Dense (Lasserre)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\epsilon_{\text{obj}})</td>
<td>(\epsilon_{\text{feas}})</td>
</tr>
<tr>
<td>1.8e-9</td>
<td>9.6e-9</td>
</tr>
</tbody>
</table>

\(\epsilon_{\text{obj}} = \frac{|lbd. \text{ for opt.val.} - \text{approx.opt.val.}|}{\max\{1, |lbd. \text{ for opt.val.}|\}}\).

\(\epsilon_{\text{feas}} = \) the max. error in equalities, \(\text{cpu} : \) cpu time in second

Global optimality is guaranteed with high accuracy.
Unconstrained optimization problem

The generalized Rosenbrock function — poly. with deg = 4
\[
f_R(x) = 1 + \sum_{i=2}^{n} \left(100(x_i - x_{i-1}^2)^2 + (1 - x_i^2)^2\right)
\]

The chained singular function — poly. with deg = 4
\[
f_C(x) = \sum_{i \in J} \left((x_i + 10x_{i+1})^2 + 5(x_{i+2} - x_{i+3})^2 \right.
\]
\[+ \left. (x_{i+1} - 2x_{i+2})^4 + 10(x_i - 10x_{i+3})^4\right)
\]

Here \( J = \{1, 3, 5, \ldots, n - 3\} \), \( n \) is a multiple of 4.

\[P2: \min f_R(x) + f_C(x)\]
— unknown global optimal value and solution
\[Hf_R(x) + Hf_C(x) : \text{very sparse} \Rightarrow \text{next}\]
Sparsity pattern of $H_{fR} + H_{fC}$ ($n = 100$) with simultaneous row and column reordering (Reverse Cuthill-McKee ordering)

Structured sparsity

Sparse (symbolic) Cholesky factorization
P2 : \( \min f_R(x) + f_C(x) \) — deg. 4, sparse, unknown opt.val.

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \epsilon_{\text{obj}} )</th>
<th># =</th>
<th>( \epsilon_{\text{obj}} )</th>
<th># =</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>6e-9</td>
<td>214</td>
<td>1e-9</td>
<td>1,819</td>
</tr>
<tr>
<td>16</td>
<td>5e-9</td>
<td>294</td>
<td>1e-9</td>
<td>4,844</td>
</tr>
<tr>
<td>100</td>
<td>2e-9</td>
<td>1,974</td>
<td>out of mem</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>7e-11</td>
<td>19,974</td>
<td>1e-9</td>
<td>1311.1</td>
</tr>
<tr>
<td>2000</td>
<td>6e-12</td>
<td>39,974</td>
<td>out of mem</td>
<td></td>
</tr>
<tr>
<td>3000</td>
<td>out of mem</td>
<td></td>
<td>out of mem</td>
<td></td>
</tr>
</tbody>
</table>

\[ \epsilon_{\text{obj}} = \frac{|\text{lbd. for opt.val.} - \text{approx.opt.val.}|}{\max\{1, |\text{lbd. for opt.val.}|\}}. \]

\# = : the number of equalities of SDP,

\( \text{cpu} \) : cpu time in second

Global optimality is guaranteed with high accuracy.
Sensor network localization problem: Let $s = 2$ or $3$.

- $x^p \in \mathbb{R}^s$: unknown location of sensors ($p = 1, 2, \ldots, m$),
- $x^r = a^r \in \mathbb{R}^s$: known location of anchors ($r = m + 1, \ldots, n$),
- $d_{pq}^2 = \|x^p - x^q\|^2 + \epsilon_{pq}$ — given for $(p, q) \in \mathcal{N}$,
- $\mathcal{N} = \{(p, q) : \|x^p - x^q\| \leq \rho = \text{a given radio range}\}$

Here $\epsilon_{pq}$ denotes a noise.

$m = 5$, $n = 9$.
1, $\ldots$, 5: sensors
6, 7, 8, 9: anchors

Anchors’ positions are known. A distance is given for $\forall$ edge. Compute locations of sensors.

⇒ Some nonconvex QOPs
- SDP relaxation — FSDP by Biswas-Ye '06, ESDP by Wang et al '07, ... for $s = 2$.
- SOCP relaxation — Tseng '07 for $s = 2$.
- ...
Numerical results on 3 methods applied to a sensor network localization problem with

\[ m = \text{the number of sensors dist. randomly in } [0, 1]^2, \]

4 anchors located at the corner of \([0, 1]^2\),

\( \rho = \text{radio distance } = 0.1, \) no noise.

**FSDP** — Biswas-Ye ’06, powerful but expensive

**SFSDP** = FSDP + exploiting sparsity, equivalent to FSDP

**ESDP** — a further relaxation of FSDP, weaker than FSDP

<table>
<thead>
<tr>
<th>m</th>
<th>SeDuMi cpu time in second</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FSDP</td>
</tr>
<tr>
<td>500</td>
<td>389.1</td>
</tr>
<tr>
<td>1000</td>
<td>3345.2</td>
</tr>
<tr>
<td>2000</td>
<td>111.1</td>
</tr>
<tr>
<td>4000</td>
<td>182.1</td>
</tr>
</tbody>
</table>
$m = 1000$ sensors, 4 anchors located at the corner of $[0, 1]^2$, 
$\rho = \text{radio distance} = 0.1$, no noise

SFSDP = FSDP + exploiting sparsity
3 dim, 500 sensors, radio range = 0.3, noise $\leftarrow N(0,0.1)$;

$$(\text{estimated distance}) \hat{d}_{pq} = (1 + \epsilon_{pq}) d_{pq}(\text{true unknown distance})$$

$\epsilon_{pq} \leftarrow N(0, 0.1)$

$\text{SFSDP} = \text{FSDP} + \text{exploiting sparsity}$
3 dim, 500 sensors, radio range = 0.3, noise $\sim N(0,0.1)$;

(estimated distance) $\hat{d}_{pq} = (1 + \epsilon_{pq})d_{pq}$ (true unknown distance)

$\epsilon_{pq} \sim N(0,0.1)$

(SFSDP = FSDP + exploiting sparsity) + Gradient method
Contents

1. Global optimization of nonconvex problems
   1-1 Polynomial Optimization Problems (POPs)
   1-2 SemiDefinite Programs (SDPs)

2. SDP relaxation

3. Exploiting sparsity in SDP relaxation

4. Numerical results

5. Concluding remarks
Concluding remarks

- **Sparse SDP relaxation** (Waki-Kim-Kojima-Muramatsu)
  - = Lasserre’s (dense) SDP relaxation + exploiting sparsity
  - — powerful in practice and theoretical convergence

- Some important issues to be studied.
  - Exploiting sparsity further to solve larger scale and/or higher degree POPs.
  - Huge-scale SDPs.
  - Numerical difficulty in solving SDP relaxations of POPs.

*Thank you!*