# 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

## 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. $f(\boldsymbol{x})$ sub.to $\boldsymbol{x} \in S \subseteq \mathbb{R}^n$, where $f : \mathbb{R}^n \to \mathbb{R}$.

We want to approximate a global optimal solution $\boldsymbol{x}^*$;

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

if it exists. But, impossible without any assumption.

Various assumptions

- continuity, differentiability, compactness, . . .
- 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, . . .

OP : Optimization problem in the $n$-dim. Euclidean space $\mathbb{R}^n$

min. $f(\boldsymbol{x})$ sub.to $\boldsymbol{x} \in S \subseteq \mathbb{R}^n$, where $f : \mathbb{R}^n \to \mathbb{R}$.

We want to approximate a global optimal solution $\boldsymbol{x}^*$;

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

if it exists. But, impossible without any assumption.

Various assumptions

- continuity, differentiability, compactness, ...
- 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, ...

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

$\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ : a vector variable

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

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

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

$$
\begin{array}{rl}
\text{min} & f_0(\boldsymbol{x}) \equiv x_1^3 - 2x_1x_2^2 + x_1^2x_2x_3 - 4x_3^2 \\
\text{sub.to} & f_1(\boldsymbol{x}) \equiv -x_1^2 + 5x_2x_3 + 1 \geq 0, \\
& f_2(\boldsymbol{x}) \equiv x_1^2 - 3x_1x_2x_3 + 2x_3 + 2 \geq 0, \\
& f_3(\boldsymbol{x}) \equiv -x_1^2 - x_2^2 - x_3^2 + 1 \geq 0. \\
& x_1(x_1 - 1) = 0 \text{ (0-1 integer cond.)}, \\
& x_2 \geq 0, \ x_3 \geq 0, \ x_2x_3 = 0 \text{ (comp. cond.)}.
\end{array}
$$

- 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

## SDP is an extension of Linear Program (LP)

LP: minimize $-x_1 - 2x_2 - 5x_3$

subject to $2x_1 + 3x_2 + x_3 = 7,\ x_1 + x_2 \geq 1,$

$x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0.$

SDP: minimize $-x_1 - 2x_2 - 5x_3$

subject to $2x_1 + 3x_2 + x_3 = 7,\ x_1 + x_2 \geq 1,$

$x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0,$

$\begin{pmatrix} x_1 & x_2 \\ x_2 & x_3 \end{pmatrix} \succeq \boldsymbol{O}$ (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
- Probabilty measure and its moments $\Rightarrow$ 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

$$\boldsymbol{u}_r(\boldsymbol{x}) = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2, x_1^3, \ldots, x_2^r) \text{ : row vector}$$

(all monomials with degree $\leq r$)

$$\boldsymbol{M}_r(\boldsymbol{y}) = \int_{\mathbb{R}^2} \boldsymbol{u}_r(\boldsymbol{x})^T \boldsymbol{u}_r(\boldsymbol{x}) d\mu$$

(moment matrix, symmetric, positive semidefinite)

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

Example with $r = 2$:

$$y_{21} = \int_{\mathbb{R}^2} x_1^2 x_2 d\mu$$

$$\boldsymbol{M}_r(\boldsymbol{y}) = \begin{pmatrix} y_{00} & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{pmatrix}, \quad y_{00} = 1$$

$\mu$ : a probability measure on $\mathbb{R}^n$ . We assume $n = 2$ in this talk.
For every $r = 0, 1, 2, \ldots$ , define

$$\boldsymbol{u}_r(\boldsymbol{x}) = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2, x_1^3, \ldots, x_2^r)$$ : row vector

(all monomials with degree $\leq r$)

$$\boldsymbol{M}_r(\boldsymbol{y}) = \int_{\mathbb{R}^2} \boldsymbol{u}_r(\boldsymbol{x})^T \boldsymbol{u}_r(\boldsymbol{x}) d\mu$$ ( moment matrix, symmetric, positive semidefinite )

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

$\mu$ : a probability measure on $\mathbb{R}^2$

$\Downarrow$

$y_{00} = 1$, $\boldsymbol{M}_r(\boldsymbol{y}) \succeq \boldsymbol{O}$ (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

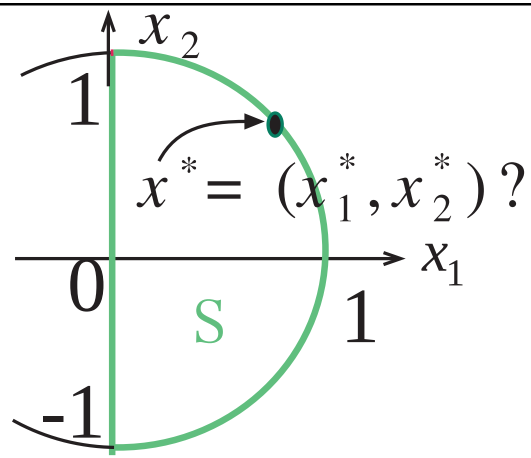
POP: min $\quad f_0(\boldsymbol{x}) = x_1^4 - 2x_1x_2 \qquad$ opt. val. $\zeta^*$ : unknown

sub. to $\quad \boldsymbol{x} \in S \equiv \left\{ \boldsymbol{x} \in \mathbb{R}^2 : \begin{array}{l} f_1(\boldsymbol{x}) = 1 - x_1^2 - x_2^2 \geq 0 \\ f_2(\boldsymbol{x}) = x_1 \geq 0 \end{array} \right\}.$

$\Updownarrow$

min $\quad \int f_0(\boldsymbol{x}) d\mu$

sub. to $\quad \mu$ : a prob. meas. on $S$.




$\Downarrow \; y_{\alpha\beta} = \int_{\mathbb{R}^2} x_1^\alpha x_2^\beta d\mu$

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

sub. to $\quad$ "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

$r = 2$

min $y_{40} - 2y_{11}$ s.t. $$\int \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}^T x_1 d\mu \succeq \boldsymbol{O}, \Leftarrow x_1 \geq 0$$

$$1 - x_1^2 - x_2^2 \geq 0 \Rightarrow \int \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}^T (1 - x_1^2 - x_2^2) d\mu \succeq \boldsymbol{O},$$

(moment matrix) $$\int \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ x_2^2 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ x_2^2 \end{pmatrix}^T d\mu \succeq \boldsymbol{O}.$$

$$\Downarrow \quad y_{\alpha\beta} = \int_{\mathbb{R}^2} x_1^\alpha x_2^\beta d\mu$$

## SDP relaxation (Lasserre '01) of a POP — an example

$r = 2$

$$\min\ y_{40} - 2y_{11} \text{ s.t. } \begin{pmatrix} y_{10} & y_{20} & y_{11} \\ y_{20} & y_{30} & y_{21} \\ y_{11} & y_{21} & y_{12} \end{pmatrix} \succeq \boldsymbol{O},$$

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

(moment matrix) $$\begin{pmatrix} 1 & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{pmatrix} \succeq \boldsymbol{O}.$$

- We can apply SDP relaxation to general POPs in $\mathbb{R}^n$.
- Poweful in theory but very expensive in computation
  $\Rightarrow$ Exploiting sparsity is crucial in practice.

## Contents

$\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ : a vector variable
$f_j(\boldsymbol{x})$ : a real-valued polynomial w. deg $\leq q$ $(j = 0, 1, \ldots, m)$

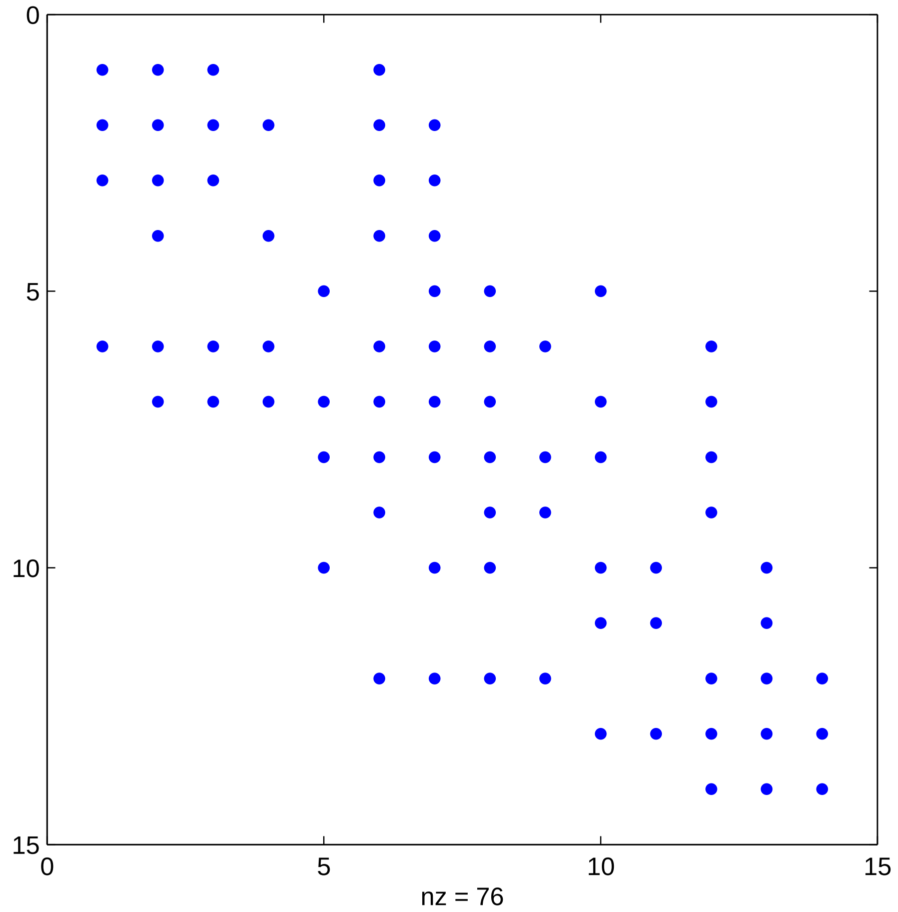
| |
|---|
| POP: min $f_0(\boldsymbol{x})$ sub.to $f_j(\boldsymbol{x}) \geq$ or $= 0$ $(j = 1, \ldots, m)$ |

$\mathcal{F}^*$ = the set of all monomials with deg $\leq q$; $\#\mathcal{F}^* = \begin{pmatrix} n+q \\ q \end{pmatrix}$
$\mathcal{F}^* \supseteq \mathcal{F}_j$ = the set of monomials involved in $f_j$

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

- $n = 14$ variables. polynomials with deg $\leq q = 3$; $\#\mathcal{F}^* = 680$
- $\forall f_j$ involves less than $6$ monomials + structured sparsity
- $Hf_0(\boldsymbol{x})$ : Hessian mat., $F(\boldsymbol{x}) = (f_1, \ldots, f_7)^T$, $DF(\boldsymbol{x})$ : $7 \times 14$ Jacobian mat.. Sparsity pattern of $Hf_0 + DF^TDF \Rightarrow$

## 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

$\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ : a vector variable
$f_j(\boldsymbol{x})$ : a real-valued polynomial w. deg $\leq r$ $(j = 0, 1, \ldots, m)$

| POP: min $f_0(\boldsymbol{x})$ sub.to $f_j(\boldsymbol{x}) \geq$ 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 $f_j$

**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.

Original SDP relaxation Lasserre 2001 ↓

$\Downarrow$ Sparse SDP relaxation proposed in Waki-Kim-Kojima-Muramatsu 2007

Sparse SDP

Dense SDP

- SDP is "smaller", and "more efficient" than dense SDP
- Theoretical convergence to the opt. val. of POP

## Contents



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

$$
\begin{aligned}
&\text{min} \quad -6.3x_5x_8 + 5.04x_2 + 0.35x_3 + x_4 + 3.36x_6 \\
&\text{sub.to} \; -0.820x_2 + x_5 - 0.820x_6 = 0, \\
&0.98x_4 - x_7(0.01x_5x_{10} + x_4) = 0, \; -x_2x_9 + 10x_3 + x_6 = 0, \\
&x_5x_{12} - x_2(1.12 + 0.132x_9 - 0.0067x_9^2) = 0, \\
&x_8x_{13} - 0.01x_9(1.098 - 0.038x_9) - 0.325x_7 = 0.574, \\
&x_{10}x_{14} + 22.2x_{11} = 35.82, \; x_1x_{11} - 3x_8 = -1.33, \\
&\text{lbd}_i \le x_i \le \text{ubd}_i \; (i = 1, 2, \ldots, 14).
\end{aligned}
$$

| Sparse | | | Dense (Lasserre) | | |
|---|---|---|---|---|---|
| $\epsilon_{\text{obj}}$ | $\epsilon_{\text{feas}}$ | cpu | $\epsilon_{\text{obj}}$ | $\epsilon_{\text{feas}}$ | cpu |
| 1.8e-9 | 9.6e-9 | 4.1 | out of | memory | |

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

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

- Global optimality is guaranteed with high accuracy.

Unconstrained optimization problem

The gneralized Rosenbrock function — poly. with deg $= 4$

$$f_R(\boldsymbol{x}) = 1 + \sum_{i=2}^{n} \left(100(x_i - x_{i-1}^2)^2 + (1 - x_i^2)^2\right)$$

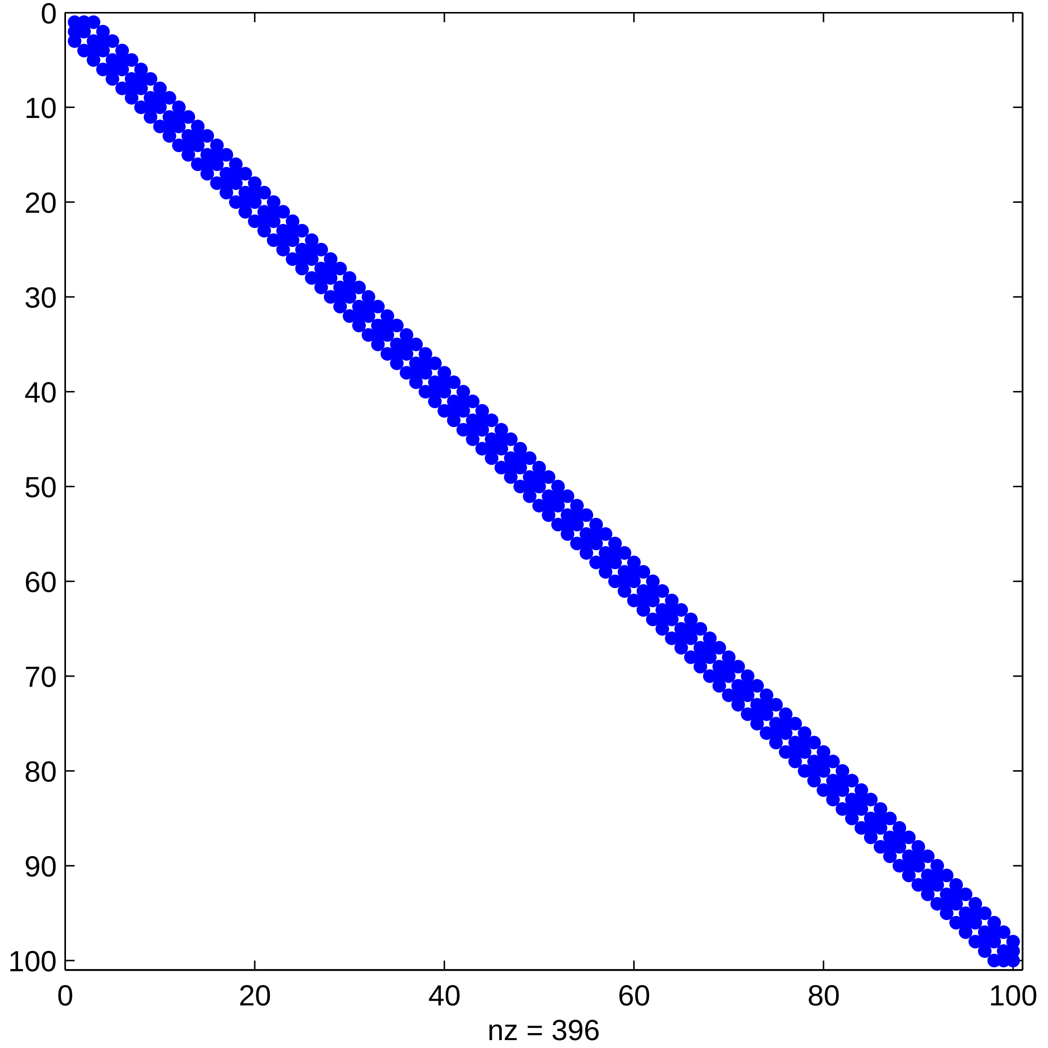
The chained singular function — poly. with deg $= 4$

$$\begin{aligned} f_C(\boldsymbol{x}) &= \sum_{i \in J} \Big((x_i + 10x_{i+1})^2 + 5(x_{i+2} - x_{i+3})^2 \\ &\quad + (x_{i+1} - 2x_{i+2})^4 + 10(x_i - 10x_{i+3})^4\Big) \end{aligned}$$

Here $J = \{1, 3, 5, \ldots, n-3\}$, $n$ is a mutiple of $4$.

P2 : min $f_R(\boldsymbol{x}) + f_C(\boldsymbol{x})$
— unknown global optimal value and solution
$H f_R(\boldsymbol{x}) + H f_C(\boldsymbol{x})$ : very sparse $\Rightarrow$ next

## Sparsity pattern of $Hf_R + Hf_C$ $(n = 100)$ with simultaneous row and column reordering (Reverse Cuthill-McKee ordering)




## Structured sparsity

- Sparse (symbolic) Cholesky factorization

P2 : min $f_R(\boldsymbol{x}) + f_C(\boldsymbol{x})$ — deg. 4, sparse, unknown opt.val.

| | Sparse | | | Dense (Lasserre) | | |
|---|---|---|---|---|---|---|
| $n$ | $\epsilon_{\text{obj}}$ | $\# =$ | cpu | $\epsilon_{\text{obj}}$ | $\# =$ | cpu |
| 12 | 6e-9 | 214 | 0.2 | 1e-9 | 1,819 | 64.1 |
| 16 | 5e-9 | 294 | 0.2 | 1e-9 | 4,844 | 1311.1 |
| 100 | 2e-9 | 1,974 | 1.2 | out of | mem | |
| 1000 | 7e-11 | 19,974 | 16.9 | | | |
| 2000 | 6e-12 | 39,974 | 45.1 | | | |
| 3000 | out of | mem | | | | |

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

$\# =$ : the number of equalities of SDP,

cpu : cpu time in second

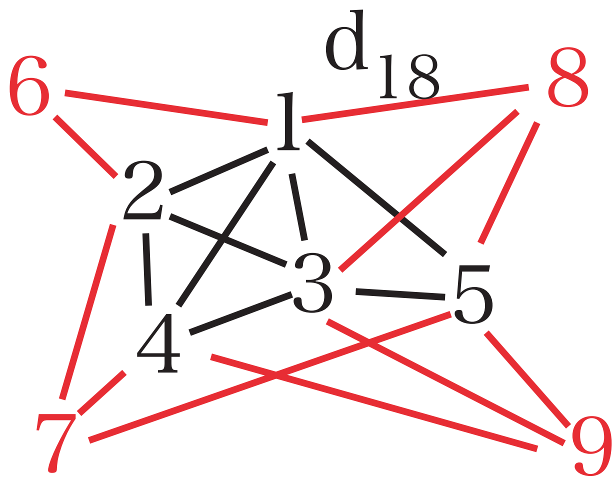
- Global optimality is guaranteed with high accuracy.

Sensor network localization problem: Let $s = 2$ or $3$.

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

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.

$\Rightarrow$ 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

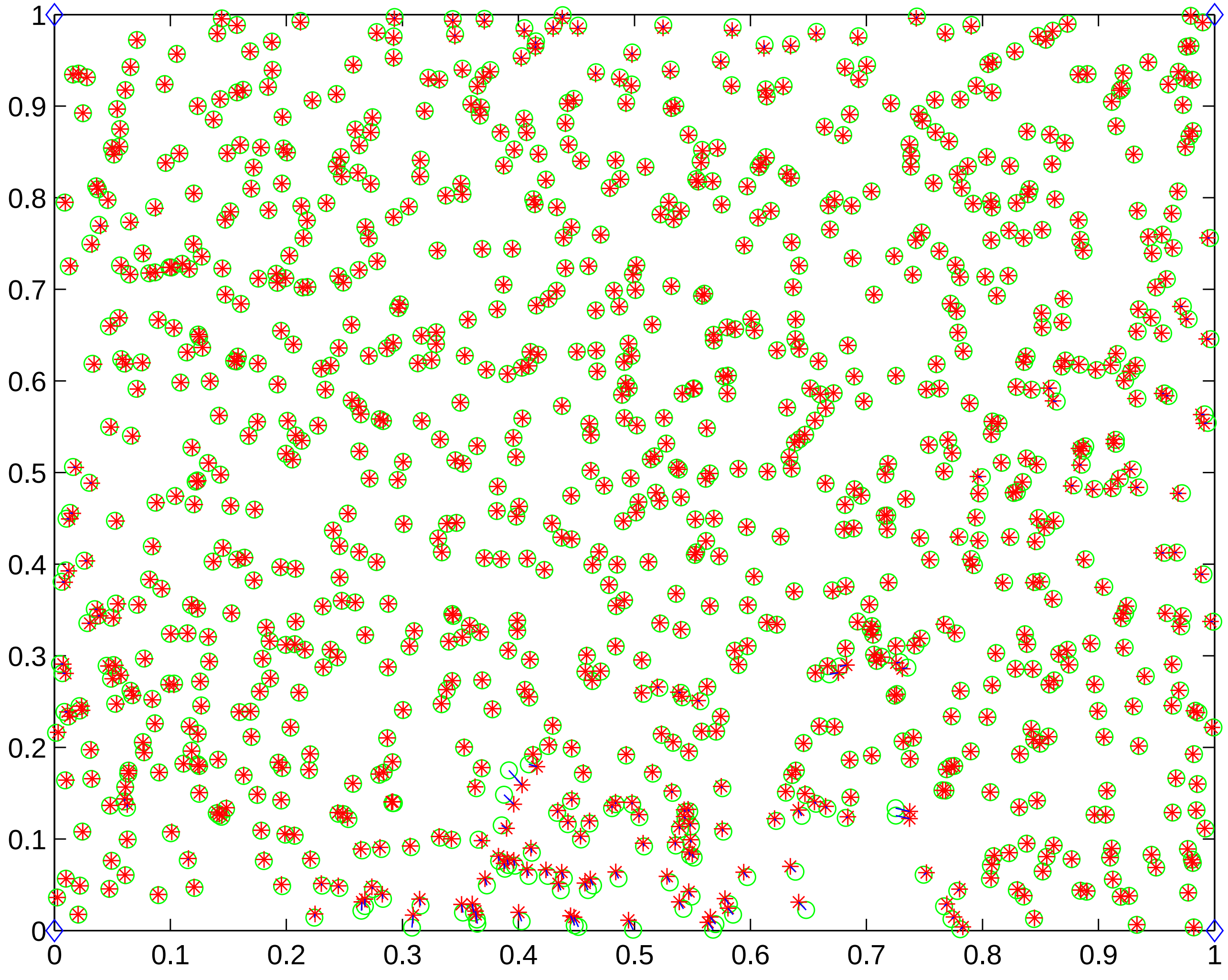
$m =$ the number of sensors dist. randomly in $[0, 1]^2$,
$4$ anchors located at the corner of $[0, 1]^2$,
$\rho =$ radio distance $= 0.1,$ no noise.

FSDP — Biswas-Ye '06, powerful but expensive
SFSDP = FSDP + exploiting sparsity, equivalent to FSDP
ESDP — a further relaxation of FSDP, weaker than FSDP

| m | SeDuMi cpu time in second | | |
|---|---|---|---|
| | FSDP | SFSDP | ESDP |
| 500 | 389.1 | 35.0 | 62.5 |
| 1000 | 3345.2 | 60.4 | 200.3 |
| 2000 | | 111.1 | 1403.9 |
| 4000 | | 182.1 | 11559.8 |

$m$ = 1000 sensors, 4 anchors located at the corner of $[0, 1]^2$,
$\rho =$ radio distance $= 0.1,$ no noise

SFSDP = FSDP + exploiting sparsity



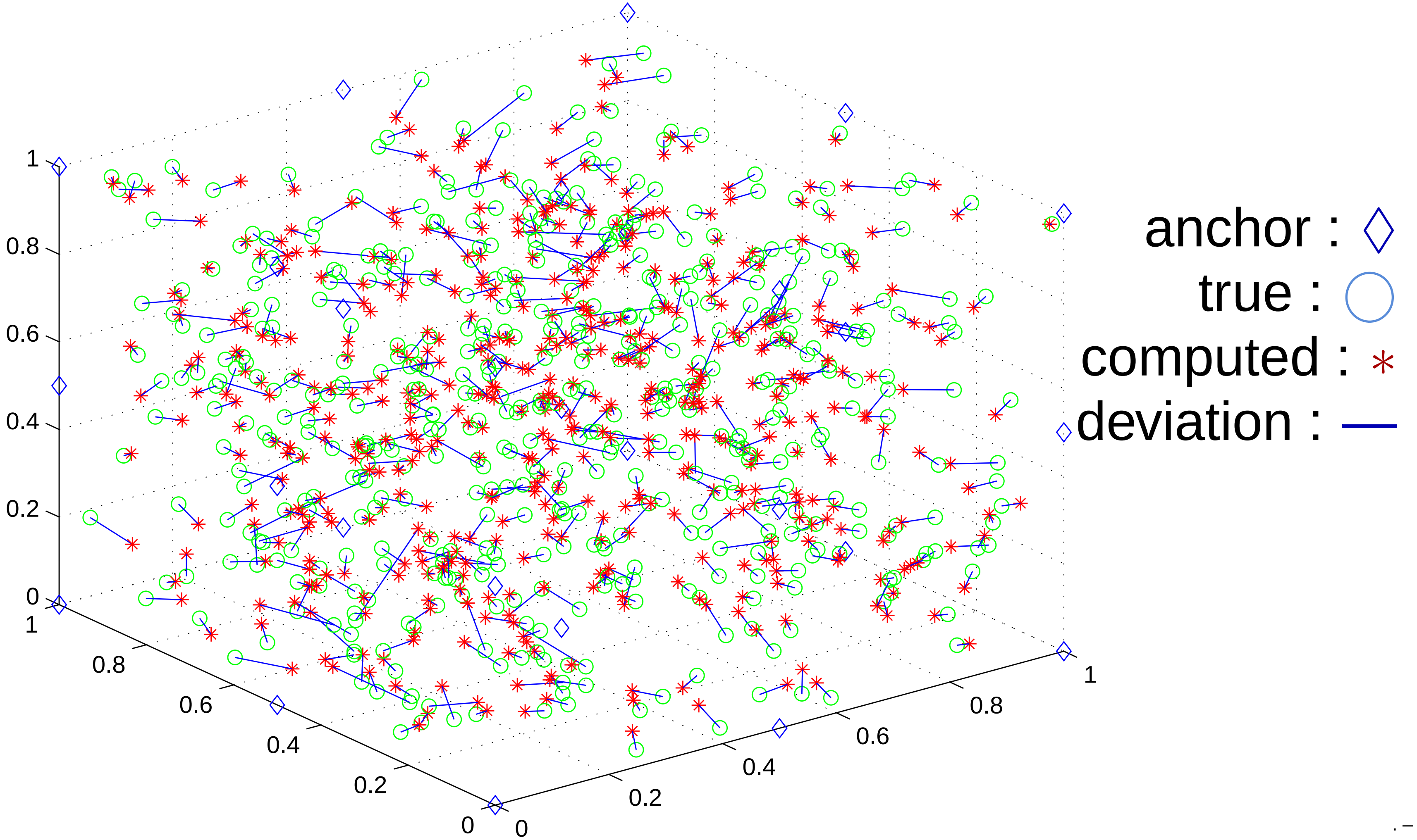

anchor : ◇
true : ○
computed : $*$

3 dim, 500 sensors, radio range = $0.3$, noise $\leftarrow$ N(0,0.1);

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

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

SFSDP = FSDP + exploiting sparsity

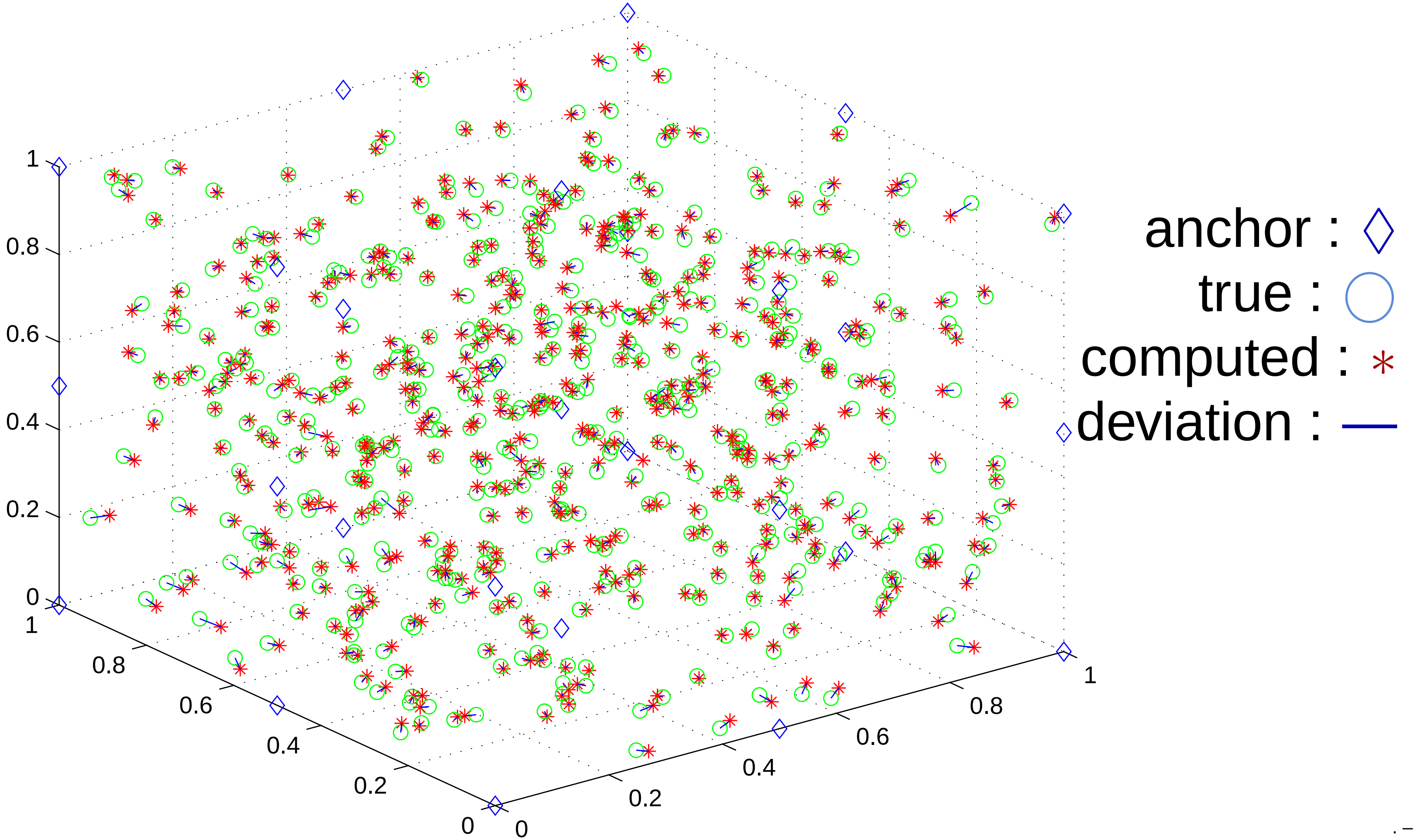


3 dim, 500 sensors, radio range = $0.3$, noise $\leftarrow$ N(0,0.1);

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

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

(SFSDP = FSDP + exploiting sparsity) + Gradient method



## Contents

## Concluding remarks

- Sparse SDP relaxation (Waki-Kim-Kojima-Muramatsu)
  = Lasserre's (dense) SDP relaxation + exploiting sparsity
  — poweful 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!