Exploiting Sparsity in SOS and SDP Relaxations of Polynomial Optimization Problems

FoCM2005 Optimization Workshop
Santander, Spain, July 4 - 6, 2005

M. Kojima Tokyo Inst. of Tech. Tokyo
S. Kim Ewha Women’s Univ. Seoul
M. Muramatsu Univ. of Electro-Communications Tokyo
H. Waki Tokyo Inst. of Tech. Tokyo
Outline
1. POPs (Polynomial Optimization Problems)
2. Rough sketch of SOS and SDP relaxations of POPs
3. Exploiting structured sparsity --- unconstrained case
4. Exploiting structured sparsity --- constrained case
5. Numerical results
6. Concluding remarks
Outline
1. POPs (Polynomial Optimization Problems)
2. Rough sketch of SOS and SDP relaxations of POPs
3. Exploiting structured sparsity --- unconstrained case
4. Exploiting structured sparsity --- constrained case
5. Numerical results
6. Concluding remarks
The $n$-dim Euclidean space.

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

$f_p(x)$ : a multivariate polynomial in $x \in \mathbb{R}^n$ ($p = 0, 1, \ldots, m$).

**POP:** min $f_0(x)$ sub.to $f_p(x) \geq 0$ ($p = 1, \ldots, m$).

**Example:** $n = 3$

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

\[x_1(x_1 - 1) = 0 \text{ (0-1 integer)}, \]
\[x_2 \geq 0, \quad x_3 \geq 0, \quad x_2x_3 = 0 \text{ (complementarity)}.
\]

- Various problems can be described as POPs.
- A unified theoretical model for global optimization in non-linear and combinatorial optimization problems.
Outline
1. POPs (Polynomial Optimization Problems)
2. Rough sketch of SOS and SDP relaxations of POPs
3. Exploiting structured sparsity --- unconstrained case
4. Exploiting structured sparsity --- constrained case
5. Numerical results
6. Concluding remarks
POP: \[ \min f_0(x) \quad \text{sub.to} \quad f_p(x) \geq 0 \quad (p = 1, \ldots, m). \]


- [1] \(\implies\) SDP relaxation — primal approach.
- [1] and [2] are dual to each other.

(a) Lower bounds for the optimal value.

(b) Convergence to global optimal solutions in theory.

(c) Large-scale SDPs require enormous computation.

(d) SDP[1] + “Exploiting structured sparsity”\(\implies\) Sparse SDP relaxation
POP: \[ \min f_0(x) \quad \text{sub.to} \quad f_p(x) \geq 0 \quad (p = 1, \ldots, m). \]

Basic idea (practical point of view)

(a) Linearization (Lifting) \[ \implies \text{relaxation}. \]
(b) Strengthen the relaxation by adding valid poly. matrix inequalities (before (a)) \[ \implies \text{a poly. SDP equiv. to POP}. \]

Represent a polynomial \( f \) as \[ f(x) = \sum_{\alpha \in \mathcal{G}} c(\alpha) x^\alpha, \]
where \[ \mathcal{G} = \text{a finite subset of } \mathbb{Z}_+^n \equiv \{ z \in \mathbb{R}_+^n : z_i \text{ is an integer } \geq 0 \}, \]
\[ x^\alpha = x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n} \quad \text{for } \forall x \in \mathbb{R}_+^n \text{ and } \forall \alpha \in \mathbb{Z}_+^n. \]

Replacing each \( x^\alpha \) by a single variable \( y_\alpha \in \mathbb{R} \), we have the linearization of \( f(x) \): \[ F(y) = F((y_\alpha : \alpha \in \mathcal{G})) = \sum_{\alpha \in \mathcal{G}} c(\alpha) y_\alpha. \]

Example

\[ f(x_1, x_2) = 2x_1 - 3x_1^2 + 4x_1 x_2^3 \]
\[ = 2x^{(1,0)} - 3x^{(2,0)} + 4x^{(1,3)} \]
\[ \Downarrow \quad \text{(a) Linearization} \]
\[ F(y_{(1,0)}, y_{(2,0)}, y_{(1,3)}) = 2y_{(1,0)} - 3y_{(2,0)} + 4y_{(1,3)}. \]
POP: \( \text{min } f_0(x) \) sub.to \( f_p(x) \geq 0 \) \((p = 1, \ldots, m)\).  

Basic idea (practical point of view)

(a) Linearization (Lifting) \( \implies \) relaxation.

(b) Strengthen the relaxation by adding valid poly. matrix inequalities (before (a)) \( \implies \) a poly. SDP equiv. to POP.

For \( \forall \) finite \( \mathcal{G} \subset \mathbb{Z}^n_+ \equiv \{z \in \mathbb{R}^n_+ : z_i \text{ is an integer } \geq 0\} \), let \( u(x; \mathcal{G}) \) denote a column vector consisting of \( x^\alpha \) \((\alpha \in \mathcal{G})\). Then

(i) \( \text{rank } 1 \text{ sym. matrix } u(x; \mathcal{G})u(x; \mathcal{G})^T \succeq O \) for \( \forall x \in \mathbb{R}^n \).

(ii) \( f_p(x)u(x; \mathcal{G})u(x; \mathcal{G})^T \succeq O \) if \( f_p(x) \geq 0 \).

Example of (ii). \( n = 2 \). \( \mathcal{G} = \{(0,0), (1,0)\} \).

\[
(1-x_1x_2) \begin{pmatrix} 1 \\ x_1 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \end{pmatrix}^T \succeq O \iff \begin{pmatrix} 1-x_1x_2 & x_1-x_1^2x_2 \\ x_1-x_1^2x_2 & x_1^2-x_1^3x_2 \end{pmatrix} \succeq O
\]

\( \Downarrow \) (a) Linearization

\( \Downarrow \) (a) Linearization

\( 1-y_{(1,1)} \geq 0 \)

\[
\begin{pmatrix} 1-y_{(1,1)} & y_{(1,0)}-y_{(2,1)} \\ y_{(1,0)}-y_{(2,1)} & y_{(2,0)}-y_{(3,1)} \end{pmatrix} \preceq O
\]

LMI is stronger!
POP: \( \min f_0(x) \ \text{sub.to} \ f_p(x) \geq 0 \ (p = 1, \ldots, m) \).

Basic idea (practical point of view)

(a) **Linearization (Lifting) \( \implies \) relaxation.**
(b) Strengthen the relaxation by adding valid poly. matrix inequalities (before (a)) \( \implies \) a poly. SDP equiv. to POP.

For \( \forall \text{ finite } \mathcal{G} \subset \mathbb{Z}_+^n \equiv \{ z \in \mathbb{R}_+^n : z_i \text{ is an integer } \geq 0 \} \), let \( u(x; \mathcal{G}) \) denote a column vector consisting of \( x^\alpha \ (\alpha \in \mathcal{G}) \). Then

(i) rank 1 sym.matrix \( (u(x; \mathcal{G})u(x; \mathcal{G})^T \geq O \) for \( \forall x \in \mathbb{R}^n \).
(ii) \( f_p(x)u(x; \mathcal{G})u(x; \mathcal{G})^T \geq O \) if \( f_p(x) \geq 0 \).

Let \( \mathcal{G}_p \ (p = 1, \ldots, q > m) \) be finite subset of \( \mathbb{Z}_+^n \); \( 0 \in \mathcal{G}_p \).

Polynomial SDP(\( \mathcal{G}_p \))

\[
\begin{align*}
\min & \quad f_0(x) \\
\text{sub.to} & \quad f_p(x)u(x, \mathcal{G}_p)u(x, \mathcal{G}_p)^T \geq O \ (p = 1, \ldots, m) \ \Leftrightarrow \ (ii) \\
& \quad u(x, \mathcal{G}_p)u(x, \mathcal{G}_p)^T \geq O \ (p = m + 1, \ldots, q) \ \Leftrightarrow \ (i)
\end{align*}
\]

Apply (a) \( \Rightarrow \) **Linear SDP(\( \mathcal{G}_p \)) = SDP relaxation of POP**

- \( \{ \mathcal{G}_p^k \} \); opt.val. of **L.SDP(\( \mathcal{G}_p^k \)) \( \rightarrow \) opt.val. of POP (Lasserre01).
- Expensive \( \Rightarrow \) Exploit sparsity of \( f_p(x) \ (p = 0, \ldots, m) \).
Outline
1. POPs (Polynomial Optimization Problems)
2. Rough sketch of SOS and SDP relaxations of POPs
3. Exploiting structured sparsity --- unconstrained case
4. Exploiting structured sparsity --- constrained case
5. Numerical results
6. Concluding remarks
\[ \mathcal{P}: \min_{x \in \mathbb{R}^n} f(x), \text{ where } f \text{ is a polynomial with } \deg f = 2r \]

\[ H : \text{the sparsity pattern of the Hessian matrix of } f(x) \]

\[ H_{ij} = \begin{cases} \star & \text{if } i = j \text{ or } \partial^2 f(x)/\partial x_i \partial x_j \neq 0, \\ 0 & \text{otherwise}. \end{cases} \]

\[ f(x) : \text{correlatively sparse } \iff \exists \text{ sparse Cholesky fact. of } H. \]

(a) Sparse C.fact. is characterized as a sparse chordal graph \( G(N, E') \); \( N = \{1, \ldots, n\}, E' \supset E = \{(i, j) : H_{ij} = \star\} \).

(b) Let \( C_1, C_2, \ldots, C_\ell \subset N \) be the max. cliques of a chordal extension \( G(N, E') \) of \( G(N, E) \), where \( E' = E \& \text{ "fill in"} \).

**Sparse relaxation = Linearization of**

\[ \min f(x) \text{ s.t. } u(x, \mathcal{G}_p)u(x, \mathcal{G}_p)^T \succeq O \ (p = 1, 2, \ldots, \ell), \]

where \( \mathcal{G}_p \subset \{z \in \mathbb{Z}_+^n : z_i = 0 \ (i \not\in C_p)\} \ (p = 1, 2, \ldots, \ell) \).

**Dense relaxation (Lasserre) = Linearization of**

\[ \min f(x) \text{ s.t. } u(x, \mathcal{G})u(x, \mathcal{G})^T \succeq O, \text{ where } \mathcal{G} \subset \mathbb{Z}_+^n. \]
\[ \mathcal{P}: \min_{x \in \mathbb{R}^n} f(x), \text{ where } f \text{ is a polynomial with } \deg f = 2r \]

\[ H : \text{ the sparsity pattern of the Hessian matrix of } f(x) \]
\[ H_{ij} = \begin{cases} * & \text{if } i = j \text{ or } \partial^2 f(x)/\partial x_i \partial x_j \neq 0, \\ 0 & \text{otherwise}. \end{cases} \]

\[ f(x) : \text{correlatively sparse } \Leftrightarrow \exists \text{ sparse Cholesky fact. of } H. \]

G. Rosenbrock func: \[ f(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_{i+1})^2. \]

Dense relaxation (Lasserre) = Linearization of \[ \min f(x) \text{ s.t. } u(x, \mathcal{G})u(x, \mathcal{G})^T \succeq O, \]
where \[ u(x, \mathcal{G}) = (1, x_1, \ldots, x_n, x_1^2, x_1x_2, \ldots, x_n^2, x_2x_3, \ldots, x_n^2)^T \]
consisting of all monomials in \( x_1, \ldots, x_n \) with degree \( \leq 2 \).

- The size of \( u(x, \mathcal{G})u(x, \mathcal{G})^T = \binom{n + 2}{2}; \geq 20,000 \text{ if } n = 200. \)
- Difficult to use Dense relaxation for larger POPs in practice.
\[ \mathcal{P}: \min_{x \in \mathbb{R}^n} f(x), \text{ where } f \text{ is a polynomial with } \deg f = 2r \]

\[ H: \text{ the sparsity pattern of the Hessian matrix of } f(x) \]

\[ H_{ij} = \begin{cases} * & \text{if } i = j \text{ or } \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \neq 0, \\ 0 & \text{otherwise}. \end{cases} \]

\[ f(x): \text{ correlatively sparse } \iff \exists \text{ sparse Cholesky fact. of } H. \]

G. Rosenbrock func: \[ f(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_{i+1})^2. \]

- The Hessian matrix is sparse (tridiagonal).

Sparse relaxation = Linearization of

\[ \min f(x) \text{ s.t. } \begin{pmatrix} 1 \\ x_i \\ x_{i+1} \\ x_i^2 \\ x_{i+1}^2 \\ x_i x_{i+1} \\ x_i^2 x_{i+1} \\ x_{i+1}^2 \end{pmatrix}^T \begin{pmatrix} 1 \\ x_i \\ x_{i+1} \\ x_i^2 \\ x_{i+1}^2 \\ x_i x_{i+1} \\ x_i^2 x_{i+1} \\ x_{i+1}^2 \end{pmatrix} \succeq O \ (i = 1, 2, \ldots, n - 1), \]

- Much smaller than Dense relaxation; the size is linear in \( n \).
Outline
1. POPs (Polynomial Optimization Problems)
2. Rough sketch of SOS and SDP relaxations of POPs
3. Exploiting structured sparsity --- unconstrained case
4. Exploiting structured sparsity --- constrained case
5. Numerical results
6. Concluding remarks

• This part is complicated, so we present only a basic idea in 3 steps 1), 2) and 3).
POP: \[ \min f_0(x) \quad \text{sub.to} \quad f_p(x) \geq 0 \quad (p = 1, \ldots, m). \]

Let \( \mathcal{G}_p \) \((p = 1, 2, \ldots, q > m)\) be finite subset of \( \mathbb{Z}_+^n; \ 0 \in \mathcal{G}_p. \)

\begin{center}
\text{Relaxation = Linearization of Polynomial SDP(\( \mathcal{G}_p \))}
\end{center}

\[
\begin{align*}
\min & \quad f_0(x) \\
\text{sub.to} & \quad f_p(x)u(x, \mathcal{G}_p)u(x, \mathcal{G}_p)^T \geq O \quad (p = 1, \ldots, m) \quad \text{--- (a)} \\
& \quad u(x, \mathcal{G}_p)u(x, \mathcal{G}_p)^T \geq O \quad (p = m + 1, \ldots, q) \quad \text{--- (b)}
\end{align*}
\]

1) In (a), take \( u(x, \mathcal{G}_p) \) so that it shares all \( x_i \)'s with \( f_p(x) \).

For example,

\[
-x_1^2 + 2x_5^3 - 2 \geq 0 \Rightarrow (-x_1^2 + 2x_5^3 - 2) \begin{pmatrix} 1 \\ x_1 \\ x_5 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_5 \end{pmatrix}^T \geq O,
\]

\[
x_3^2 + 3x_3 - 2 \geq 0 \Rightarrow (x_3^2 + 3x_3 - 2) \begin{pmatrix} 1 \\ x_3 \\ x_3^2 \end{pmatrix} \begin{pmatrix} 1 \\ x_3 \\ x_3^2 \end{pmatrix}^T \geq O.
\]
POP: \( \min f_0(x) \) sub.to \( f_p(x) \geq 0 \) \( (p = 1, \ldots, m) \).

Let \( \mathcal{G}_p \) \( (p = 1, 2, \ldots, q > m) \) be finite subset of \( \mathbb{Z}^n_+ \); \( 0 \in \mathcal{G}_p \).

**Relaxation = Linearization of Polynomial SDP(\( \mathcal{G}_p \))**

\[
\begin{align*}
\min & \quad f_0(x) \\
\text{sub.to} & \quad f_p(x) u(x, \mathcal{G}_p) u(x, \mathcal{G}_p)^T \geq O \quad (p = 1, \ldots, m) \quad \text{— (a)} \\
& \quad u(x, \mathcal{G}_p) u(x, \mathcal{G}_p)^T \geq O \quad (p = m + 1, \ldots, q) \quad \text{— (b)}
\end{align*}
\]

1) In (a), take \( u(x, \mathcal{G}_n) \) so that it shares all \( x_i \)'s with \( f_n(x) \).
2) Let \( H \) be the correlative sparsity pattern of \( f_0(x) \) and (a);

\[
H_{ij} = \begin{cases} 
\ast & \text{if } i = j \text{ or } \partial^2 f_0(x)/\partial x_i \partial x_j \neq 0, \\
\ast & \text{if } x_i \text{ and } x_j \text{ involved in } f_p(x) \text{ for some } p, \\
0 & \text{otherwise.}
\end{cases}
\]

In (b), choose \( u(x, \mathcal{G}_p) \) taking account of the correlative sparsity pattern \( H \) as in the unconstrained case.

3) Expand \( \mathcal{G}_p \) in (a) as long as the sparsity is maintained.
- Balance degrees of poly. mat. inequalities in (a) and (b).
- Let \( r \) denote the max degree of monomials in \( u(x, \mathcal{G}_p) \)s.
- As \( r \uparrow \), a better approx. sol. but the size \( \uparrow \).
Outline
1. POPs (Polynomial Optimization Problems)
2. Rough sketch of SOS and SDP relaxations of POPs
3. Exploiting structured sparsity --- unconstrained case
4. Exploiting structured sparsity --- constrained case
5. Numerical results
6. Concluding remarks

- Balance degrees of poly. mat. inequalities in (a) and (b).
- Let $r$ denote the max degree of monomials in $u(x, G_p)s$.
- As $r \uparrow$, a better approx. sol. but the size $\uparrow$.

$r =$ relaxation order
Software

- MATLAB for constructing sparse and dense SDP relaxation problems
- SeDuMi to solve SDPs.

Hardware

- 2.4GHz Xeon cpu with 6.0GB memory.
G. Rosenbrock function:

\[ f(x) = \sum_{i=2}^{n} (100(x_i - x_{i-1}^2)^2 + (1 - x_i)^2) \]

- Two minimizers on \( \mathbb{R}^n \): \( x_1 = \pm 1, x_i = 1 \) (\( i \geq 2 \)).
- Sparse can not handle multiple minimizers effectively.
- Perturb the function or add \( x_1 \geq 0 \Rightarrow \) unique minimizer.

<table>
<thead>
<tr>
<th>cpu in sec.</th>
<th>( \epsilon_{\text{obj}} )</th>
<th>( n )</th>
<th>( \epsilon_{\text{obj}} )</th>
<th>cpu in sec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse</td>
<td>( \epsilon_{\text{obj}} )</td>
<td>( n )</td>
<td>( \epsilon_{\text{obj}} )</td>
<td>Sparse</td>
</tr>
<tr>
<td>0.2</td>
<td>5.1e-04</td>
<td>10</td>
<td>2.5e-08</td>
<td>0.2</td>
</tr>
<tr>
<td>0.3</td>
<td>1.8e-03</td>
<td>15</td>
<td>6.5e-08</td>
<td>0.2</td>
</tr>
<tr>
<td>2.2</td>
<td>3.1e-03</td>
<td>200</td>
<td>5.2e-07</td>
<td>2.2</td>
</tr>
<tr>
<td>4.6</td>
<td>5.9e-03</td>
<td>400</td>
<td>2.5e-06</td>
<td>3.7</td>
</tr>
<tr>
<td>8.6</td>
<td>8.3e-03</td>
<td>800</td>
<td>5.5e-06</td>
<td>6.8</td>
</tr>
</tbody>
</table>

\[ \epsilon_{\text{obj}} = \frac{|\text{the lower bound for opt. value} - \text{the approx. opt. value}|}{\max\{1, |\text{the lower bound for opt. value}|\}}. \]
An optimal control problem from Coleman et al. 1995

$$\min \frac{1}{M} \sum_{i=1}^{M-1} (y_i^2 + x_i^2)$$

s.t. \( y_{i+1} = y_i + \frac{1}{M}(y_i^2 - x_i), \quad (i = 1, \ldots, M - 1), \quad y_1 = 1. \)

Numerical results on sparse relaxation ($r = 2$)

<table>
<thead>
<tr>
<th>(M)</th>
<th># of variables</th>
<th>(\epsilon_{\text{obj}})</th>
<th>(\epsilon_{\text{feas}})</th>
<th>cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>600</td>
<td>1198</td>
<td>3.4e-08</td>
<td>2.2e-10</td>
<td>3.4</td>
</tr>
<tr>
<td>700</td>
<td>1398</td>
<td>2.5e-08</td>
<td>8.1e-10</td>
<td>3.3</td>
</tr>
<tr>
<td>800</td>
<td>1598</td>
<td>5.9e-08</td>
<td>1.6e-10</td>
<td>3.8</td>
</tr>
<tr>
<td>900</td>
<td>1798</td>
<td>1.4e-07</td>
<td>6.8e-10</td>
<td>4.5</td>
</tr>
<tr>
<td>1000</td>
<td>1998</td>
<td>6.3e-08</td>
<td>2.7e-10</td>
<td>5.0</td>
</tr>
</tbody>
</table>

$$\epsilon_{\text{obj}} = \frac{|\text{the lower bound for opt. value} - \text{the approx. opt. value}|}{\max\{1, |\text{the lower bound for opt. value}|\}},$$

$$\epsilon_{\text{feas}} = \text{the maximum error in the equality constraints},$$

cpu : cpu time in sec. to solve an SDP relaxation problem.
alkyl.gms: a benchmark problem from globallib

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

<table>
<thead>
<tr>
<th>problem</th>
<th>(n)</th>
<th>(r)</th>
<th>(\epsilon_{\text{obj}})</th>
<th>(\epsilon_{\text{feas}})</th>
<th>cpu</th>
<th>(\epsilon_{\text{obj}})</th>
<th>(\epsilon_{\text{feas}})</th>
<th>cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>alkyl</td>
<td>14</td>
<td>2</td>
<td>4.1e-03</td>
<td>2.7e-01</td>
<td>0.9</td>
<td>6.3e-06</td>
<td>1.8e-02</td>
<td>17.6</td>
</tr>
<tr>
<td>alkyl</td>
<td>14</td>
<td>3</td>
<td>5.6e-10</td>
<td>2.0e-08</td>
<td>6.9</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

\(r = \text{relaxation order},\)

\[
\epsilon_{\text{obj}} = \frac{|\text{the lower bound for opt. value} - \text{the approx. opt. value}|}{\max\{1, |\text{the lower bound for opt. value}|\}},
\]

\(\epsilon_{\text{feas}} = \text{the maximum error in the equality constraints},\)

\(\text{cpu} : \text{cpu time in sec. to solve an SDP relaxation problem.}\)
Some other benchmark problems from globallib

<table>
<thead>
<tr>
<th>problem</th>
<th>n</th>
<th>r</th>
<th>$\varepsilon_{\text{obj}}$</th>
<th>$\varepsilon_{\text{feas}}$</th>
<th>cpu</th>
<th>$\varepsilon_{\text{obj}}$</th>
<th>$\varepsilon_{\text{feas}}$</th>
<th>cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>ex3_1_1</td>
<td>8</td>
<td>3</td>
<td>6.3e-09</td>
<td>4.7e-04</td>
<td>5.5</td>
<td>0.7e-08</td>
<td>2.5e-03</td>
<td>597.8</td>
</tr>
<tr>
<td>st_bpaflb</td>
<td>10</td>
<td>2</td>
<td>3.8e-08</td>
<td>2.8e-08</td>
<td>1.0</td>
<td>4.6e-09</td>
<td>7.2e-10</td>
<td>1.7</td>
</tr>
<tr>
<td>st_e07</td>
<td>10</td>
<td>2</td>
<td>0.0e+00</td>
<td>8.1e-05</td>
<td>0.4</td>
<td>0.0e+00</td>
<td>8.8e-06</td>
<td>3.0</td>
</tr>
<tr>
<td>ex2_1_3</td>
<td>13</td>
<td>2</td>
<td>5.1e-09</td>
<td>3.5e-09</td>
<td>0.5</td>
<td>1.6e-09</td>
<td>1.5e-09</td>
<td>7.7</td>
</tr>
<tr>
<td>ex9_1_1</td>
<td>13</td>
<td>2</td>
<td>0.0</td>
<td>4.5e-06</td>
<td>1.5</td>
<td>0.0</td>
<td>9.2e-07</td>
<td>7.7</td>
</tr>
<tr>
<td>ex9_2_3</td>
<td>16</td>
<td>2</td>
<td>0.0e+00</td>
<td>5.7e-06</td>
<td>2.3</td>
<td>0.0e+00</td>
<td>7.5e-06</td>
<td>49.7</td>
</tr>
<tr>
<td>ex2_1_8</td>
<td>24</td>
<td>2</td>
<td>1.0e-05</td>
<td>0.0e+00</td>
<td>304.6</td>
<td>3.4e-06</td>
<td>0.0e+00</td>
<td>1946.6</td>
</tr>
<tr>
<td>ex5_2_2_c1</td>
<td>9</td>
<td>2</td>
<td>1.0e-2</td>
<td>3.2e+01</td>
<td>1.8</td>
<td>1.6e-05</td>
<td>2.1e-01</td>
<td>2.6</td>
</tr>
<tr>
<td>ex5_2_2_c1</td>
<td>9</td>
<td>3</td>
<td>6.4e-04</td>
<td>2.3e-01</td>
<td>295.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ex5_2_2_c2</td>
<td>9</td>
<td>2</td>
<td>1.0e-02</td>
<td>7.2e+01</td>
<td>2.1</td>
<td>1.3e-04</td>
<td>2.7e-01</td>
<td>3.5</td>
</tr>
<tr>
<td>ex5_2_2_c2</td>
<td>9</td>
<td>3</td>
<td>5.8e-04</td>
<td>8.9e-01</td>
<td>332.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

$r =$ relaxation order,

$$\varepsilon_{\text{obj}} = \frac{|\text{the lower bound for opt. value} - \text{the approx. opt. value}|}{\max\{1, |\text{the lower bound for opt. value}|\}},$$

$\varepsilon_{\text{feas}} =$ the maximum error in the equality constraints,

cpu : cpu time in sec. to solve an SDP relaxation problem.
Some other benchmark problems from globallib

<table>
<thead>
<tr>
<th>problem</th>
<th>n</th>
<th>r</th>
<th>$\epsilon_{\text{obj}}$</th>
<th>$\epsilon_{\text{feas}}$</th>
<th>cpu</th>
<th>$\epsilon_{\text{obj}}$</th>
<th>$\epsilon_{\text{feas}}$</th>
<th>cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sparse</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ex3_1.1</td>
<td>8</td>
<td>3</td>
<td>6.3e-09</td>
<td>4.7e-04</td>
<td>5.5</td>
<td>0.7e-08</td>
<td>2.5e-03</td>
<td>597.8</td>
</tr>
<tr>
<td>st_bpafl1b</td>
<td>10</td>
<td>2</td>
<td>3.8e-08</td>
<td>2.8e-08</td>
<td>1.0</td>
<td>4.6e-09</td>
<td>7.2e-10</td>
<td>1.7</td>
</tr>
<tr>
<td>st_e07</td>
<td>10</td>
<td>2</td>
<td>0.0e+00</td>
<td>8.1e-05</td>
<td>0.4</td>
<td>0.0e+00</td>
<td>8.8e-06</td>
<td>3.0</td>
</tr>
<tr>
<td>ex2_1.3</td>
<td>13</td>
<td>2</td>
<td>5.1e-09</td>
<td>3.5e-09</td>
<td>0.5</td>
<td>1.6e-09</td>
<td>1.5e-09</td>
<td>7.7</td>
</tr>
<tr>
<td>ex9_1.1</td>
<td>13</td>
<td>2</td>
<td>0.0</td>
<td>4.5e-06</td>
<td>1.5</td>
<td>0.0</td>
<td>9.2e-07</td>
<td>7.7</td>
</tr>
<tr>
<td>ex9_2.3</td>
<td>16</td>
<td>2</td>
<td>0.0e+00</td>
<td>5.7e-06</td>
<td>2.3</td>
<td>0.0e+00</td>
<td>7.5e-06</td>
<td>49.7</td>
</tr>
<tr>
<td>ex2_1.8</td>
<td>24</td>
<td>2</td>
<td>1.0e-05</td>
<td>0.0e+00</td>
<td>304.6</td>
<td>3.4e-06</td>
<td>0.0e+00</td>
<td>1946.6</td>
</tr>
<tr>
<td><strong>Dense (Lasserre)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ex5_2.2_c1</td>
<td>9</td>
<td>2</td>
<td>1.0e-2</td>
<td>3.2e+01</td>
<td>1.8</td>
<td>1.6e-05</td>
<td>2.1e-01</td>
<td>2.6</td>
</tr>
<tr>
<td>ex5_2.2_c1</td>
<td>9</td>
<td>3</td>
<td>6.4e-04</td>
<td>2.3e-01</td>
<td>295.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ex5_2.2_c2</td>
<td>9</td>
<td>2</td>
<td>1.0e-02</td>
<td>7.2e+01</td>
<td>2.1</td>
<td>1.3e-04</td>
<td>2.7e-01</td>
<td>3.5</td>
</tr>
<tr>
<td>ex5_2.2_c2</td>
<td>9</td>
<td>3</td>
<td>5.8e-04</td>
<td>8.9e-01</td>
<td>332.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

- **ex5_2.2_c1** and **ex5_2.2_c2** ($r = 2$) — Dense is better.
- **Sparse** attains approx. opt. solutions with the same quality as Dense except **ex5_2.2_c1** and **ex5_2.2_c2**.
- **Sparse** is much faster than Dense in large dim. and higher relaxation order cases.
Outline
1. POPs (Polynomial Optimization Problems)
2. Rough sketch of SOS and SDP relaxations of POPs
3. Exploiting structured sparsity --- unconstrained case
4. Exploiting structured sparsity --- constrained case
5. Numerical results
6. Concluding remarks
• Lasserre’s (dense) relaxation
  — theoretical convergence but expensive in practice.

• The proposed sparse relaxation
  = Lasserre’s (dense) relaxation + sparsity
  — no theoretical convergence but very powerful in practice.

• There remain many issues to be studied further.
  – Exploiting sparsity.
  – Large-scale SDPs.
  – Numerical difficulty in solving SDP relaxations of POPs.

This presentation material is available at

http://www.is.titech.ac.jp/~kojima/talk.html

Thank you!