多項式最適化問題の半正定値計画緩和
— 疎性の活用 —

最適化：モデリングとアルゴリズム
2007 年 3 月 22-24 日
統計数理研究所

小島政和
東京工業大学

Outline

1. POPs (Polynomial Optimization Problems)
2. How do we formulate structured sparsity?
3. Sparse SDP relaxation of POPs
4. Numerical results
5. Concluding remarks.

Outline

1. POPs (Polynomial Optimization Problems)
2. How do we formulate structured sparsity?
3. Sparse SDP relaxation of POPs
4. Numerical results
5. Concluding remarks

Notation and Symbols

$\mathbb{R}^n$: the n-dim Euclidean space.
$\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$: a vector variable.
$f_j(\boldsymbol{x})$: a multivariate polynomial in $\boldsymbol{x} \in \mathbb{R}^n$ $(j = 0, 1, \ldots, m)$.

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

Example: $n = 3$

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

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

[1] Lasserre, “Global optimization with polynomials and the problems of moments”, *SIAM J. on Optim.* (2001).
[2] Parrilo, “Semidefinite programming relaxations for semialgebraic problems”, *Math. Prog.* (2003).
[3] Henrion-Lasserre, GloptiPoly.
[4] Prajna-Parachristodoulou-Parrilo, SOSTOOLS.

- [1,3]$\Rightarrow$ a sequence of SDP relaxations — primal approach.
- [2,4]$\Rightarrow$ a sequence of SOS relaxations — dual approach.

(a) Lower bounds for the optimal value.
(b) Convergence to global optimal solutions in theory.
(c) Each relaxed problem can be solved as an SDP; its size gets larger rapidly along “the sequence” as we require a higher accuracy.
(d) Expensive to solve large scale POPs in practice.
$\Rightarrow$ Exploiting Sparsity.

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

[1] Lasserre, “Global optimization with polynomials and the problems of moments”, *SIAM J. on Optim.* (2001).

[2] Parrilo, “Semidefinite programming relaxations for semialgebraic problems”, *Math. Prog.* (2003).

[3] Henrion-Lasserre, GloptiPoly.

[4] Prajna-Parachristodoulou-Parrilo, SOSTOOLS.

Exploiting sparsity to solve larger scale problem in practice

[5] Kojima-Kim-Waki, “Sparsity in SOS Polynomials”, *Math. Prog.* (2005).

[6] Waki-Kim-Kojima-Muramatsu, "SOS and SDP Relaxations for POPs with Structured Sparsity", *SIAM J. on Optim* (2006).

[7] Waki-Kim-Kojima-Muramatsu, SparsePOP (2005).

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

[1] Lasserre, “Global optimization with polynomials and the problems of moments”, *SIAM J. on Optim.* (2001).
[2] Parrilo, “Semidefinite programming relaxations for semialgebraic problems”, *Math. Prog.* (2003).
[3] Henrion-Lasserre, GloptiPoly.
[4] Prajna-Parachristodoulou-Parrilo, SOSTOOLS.

Extension to polynomial SDP and SOCP

[8] Kojima, “SOS relaxations of poly. SDPs” (2003).
[9] Hol-Scherer,“SOS relaxations of poly. SDPs” (2004).
[10] Henrion-Lasserre, “Convergent relaxations of poly. mat. inequalities & static output feedback ” (2006).
[11] Kojima-Muramatsu, "An Extension of SOS Relaxations to POPs over Symmetric Cones ", to appear in *Math. Prog.*

Outline

1. POPs (Polynomial Optimization Problems)

2. How do we formulate structured sparsity?

3. Sparse SDP relaxation of constrained POPs
4. Numerical results
5. Concluding remarks

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

How do we exploit sparsity in POP?

$\Downarrow$

The answer depends on which methods we use to solve POP.

POP

$\Downarrow$ SDP relaxation (Lasserre 2001)

SDP $\Leftarrow$ Primal-Dual IPM (Interior-Point Method)

We will assume a structured sparsity (correlative sparsity):
(a) The size of SDP gets smaller.
(b) SDP satisfies "a similar structured sparsity" under which Primal-Dual IPM works efficiently.

- Characterized in terms of a sparse Cholesky factorization
- Characterized in terms of a sparse chordal graph structure
- Useful to solve large-scale sparse POPs in practice

POP min. $f_0(\boldsymbol{x})$ s.t. $f_j(\boldsymbol{x}) \geq 0$ or $= 0$ $(j = 1, \ldots, m)$.

Example: $f_0(\boldsymbol{x}) = \sum_{k=1}^{n} (-x_k^2)$
$f_j(\boldsymbol{x}) = 1 - x_j^2 - 2x_{j+1}^2 - x_n^2$ $(j = 1, \ldots, n-1)$.

$\boldsymbol{H} f_0(\boldsymbol{x})$: the $n \times n$ Hes. mat. of $f_0(\boldsymbol{x})$,

$\boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})$: the $m \times n$ Jacob. mat. of $\boldsymbol{f}_*(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T$,

$\boldsymbol{R}$: the csp matrix, the $n \times n$ sparsiity pattern matrix of
$\boldsymbol{I} + \boldsymbol{H} f_0(\boldsymbol{x}) + \boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})$ (no cancellation in '$+$').

$[\boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})]_{ij} \neq 0$ iff x_i and x_j are in a common constraint.

Example with n = 6:

$$\text{the csp matrix } \boldsymbol{R} = \begin{pmatrix} \star & \star & 0 & 0 & 0 & \star \\ \star & \star & \star & 0 & 0 & \star \\ 0 & \star & \star & \star & 0 & \star \\ 0 & 0 & \star & \star & \star & \star \\ 0 & 0 & 0 & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \end{pmatrix}$$

POP min. $f_0(\boldsymbol{x})$ s.t. $f_j(\boldsymbol{x}) \geq 0$ or $= 0 \ (j = 1, \ldots, m)$.

Example: $f_0(\boldsymbol{x}) = \sum_{k=1}^{n} (-x_k^2)$
$f_j(\boldsymbol{x}) = 1 - x_j^2 - 2x_{j+1}^2 - x_n^2 \ (j = 1, \ldots, n-1)$.

$\boldsymbol{H} f_0(\boldsymbol{x})$: the $n \times n$ Hes. mat. of $f_0(\boldsymbol{x})$,

$\boldsymbol{J f}_*(\boldsymbol{x})$: the $m \times n$ Jacob. mat. of $\boldsymbol{f}_*(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T$,

$\boldsymbol{R}$: the csp matrix, the $n \times n$ sparsiity pattern matrix of $\boldsymbol{I} + \boldsymbol{H} f_0(\boldsymbol{x}) + \boldsymbol{J f}_*(\boldsymbol{x})^T \boldsymbol{J f}_*(\boldsymbol{x})$ (no cancellation in '$+$').

$[\boldsymbol{J f}_*(\boldsymbol{x})^T \boldsymbol{J f}_*(\boldsymbol{x})]_{ij} \neq 0$ iff x_i and x_j are in a common constraint.

POP : c-sparse (correlatively sparse) $\Leftrightarrow$ The $n \times n$ csp matrix $\boldsymbol{R} = (R_{ij})$ allows a symbolic sparse Cholesky factorization (under a row & col. ordering like a symmetric min. deg. ordering).

POP min. $f_0(\boldsymbol{x})$ s.t. $f_j(\boldsymbol{x}) \geq 0$ or $= 0$ $(j = 1, \ldots, m)$.

Example: $f_0(\boldsymbol{x}) = \sum_{k=1}^{n} (-x_k^2)$ ——— c-sparse
$f_j(\boldsymbol{x}) = 1 - x_j^2 - 2x_{j+1}^2 - x_n^2$ $(j = 1, \ldots, n-1)$.

$\boldsymbol{H} f_0(\boldsymbol{x})$: the $n \times n$ Hes. mat. of $f_0(\boldsymbol{x})$,
$\boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})$: the $m \times n$ Jacob. mat. of $\boldsymbol{f}_*(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T$,
$\boldsymbol{R}$: the csp matrix, the $n \times n$ sparsiity pattern matrix of
$\boldsymbol{I} + \boldsymbol{H} f_0(\boldsymbol{x}) + \boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})$ (no cancellation in '$+$').

$[\boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J} \boldsymbol{f}_*(\boldsymbol{x})]_{ij} \neq 0$ iff x_i and x_j are in a common constraint.

Example with n = 6:

$$\text{the csp matrix } \boldsymbol{R} = \begin{pmatrix} \star & \star & 0 & 0 & 0 & \star \\ \star & \star & \star & 0 & 0 & \star \\ 0 & \star & \star & \star & 0 & \star \\ 0 & 0 & \star & \star & \star & \star \\ 0 & 0 & 0 & \star & \star & \star \\ \star & \star & \star & \star & \star & \star \end{pmatrix}$$

Outline

1. POPs (Polynomial Optimization Problems)
2. How do we formulate structured sparsity?
3. Sparse SDP relaxation of POPs
4. Numerical results
5. Concluding remarks

Sparse (SDP) relaxation = Lasserre (2001) + c-sparsity

POP min. $f_0(\boldsymbol{x})$ s.t. $f_j(\boldsymbol{x}) \geq 0$ or $= 0$ $(j = 1, \ldots, m)$, c-sparse.

$$\Downarrow$$

A sequence of c-sparse SDP relaxation problems depending on the relaxation order $r = 1, 2, \ldots$;

- (a) Under a moderate assumption, opt. sol. of SDP $\rightarrow$ opt sol. of POP as $r \rightarrow \infty$.
- (b) $r = \lceil$"the max. deg. of poly. in POP"$/2\rceil + 0 \sim 3$ is usually large enough to attain opt sol. of POP in practice.
- (c) Such an r is unknown in theory except $\exists$ special cases.
- (d) Additional method for all opt. sol. of POP, but expensive.
- (e) The size of SDP increases rapidly as $r \rightarrow \infty$.

POP min. $f_0(\boldsymbol{x})$ s.t. $f_j(\boldsymbol{x}) \geq 0$ or $= 0$ $(j = 1, \ldots, m)$, c-sparse.

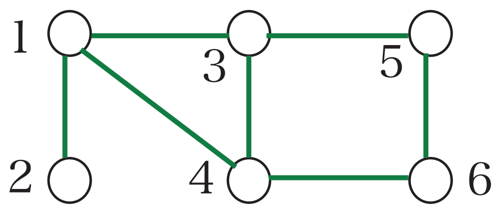
$\boldsymbol{R}$: the csp matrix, the $n \times n$ sparsiity pattern matrix of $\boldsymbol{I} + \boldsymbol{H} f_0(\boldsymbol{x}) + \boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})$ (no cancellation in '$+$').

POP : c-sparse (correlatively sparse) $\Leftrightarrow$ The $n \times n$ csp matrix $\boldsymbol{R} = (R_{ij})$ allows a symbolic sparse Cholesky factorization.

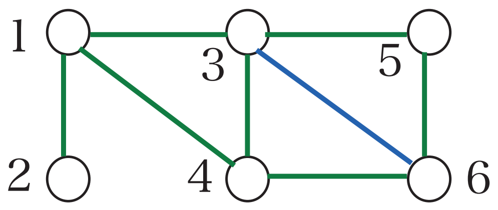
$$E = \{\{i, j\} \in N \times N : \boldsymbol{R}_{ij} = \star,\ i \neq j\} \Updownarrow$$

POP : c-sparse (correlatively sparse) $\Leftrightarrow$ The csp graph $G(N, E)$ has a sparse chordal extension $G(N, \bar{E})$; $E \subseteq \bar{E}$.

$G(N, E)$: not chordal



$G(N, \bar{E})$: chordal



$\forall$ cycle having more than 3 edges has a chord.

- The added edge $\{3, 6\}$ is corresponding to a fill-in.
- The maximal cliques $= \{1, 2\}, \{1, 3, 4\}, \{3, 4, 6\}, \{3, 5, 6\}$.

POP min. $f_0(\boldsymbol{x})$ s.t. $f_j(\boldsymbol{x}) \geq 0$ or $= 0$ $(j = 1, \ldots, m)$, c-sparse.

$\boldsymbol{R}$: the csp matrix, the $n \times n$ sparsiity pattern matrix of

$$\boldsymbol{I} + \boldsymbol{H} f_0(\boldsymbol{x}) + \boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})$$ (no cancellation in '$+$').

POP : c-sparse (correlatively sparse) $\Leftrightarrow$ The $n \times n$ csp matrix $\boldsymbol{R} = (R_{ij})$ allows a symbolic sparse Cholesky factorization.

Two steps to derive a sparse SDP relaxation of POP

(a) Using the max. cliques J_q $(q \in L)$ of $G(N, \bar{E})$, convert POP into an equivalent poly.SDP with the csp graph $G(N, \bar{E})$.

(b) Linearize poly.SDP $\Rightarrow$ SDP with a similar sparsity to poly.SDP.

$\forall$ int. $s \geq 0$, $\boldsymbol{u}_s(x_i : i \in J_q)$: the col. vect. of monomials with deg. at most s in var. x_i $(i \in J_q)$; *e.g.*, if $J_q = \{1, 4\}$,

$$\begin{aligned}
s = 0 &\Rightarrow \boldsymbol{u}_0(x_i : i \in J_q) = 1, \\
s = 1 &\Rightarrow \boldsymbol{u}_1(x_i : i \in J_q) = (1, x_1, x_4)^T, \\
s = 3 &\Rightarrow \boldsymbol{u}_3(x_i : i \in J_q) = (1, x_1, x_4, x_1^2, x_1x_4, x_4^2, x_1^3, x_1^2x_4, x_1x_4^2, x_4^3)^T.
\end{aligned}$$

POP min. $f_0(\boldsymbol{x})$ s.t. $f_j(\boldsymbol{x}) \geq 0$ or $= 0$ $(j = 1, \ldots, m)$, c-sparse.

$\boldsymbol{R}$: the csp matrix, the $n \times n$ sparsiity pattern matrix of

$$\boldsymbol{I} + \boldsymbol{H} f_0(\boldsymbol{x}) + \boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})^T \boldsymbol{J}\boldsymbol{f}_*(\boldsymbol{x})$$ (no cancellation in '$+$').

POP : c-sparse (correlatively sparse) $\Leftrightarrow$ The $n \times n$ csp matrix $\boldsymbol{R} = (R_{ij})$ allows a symbolic sparse Cholesky factorization.

Two steps to derive a sparse SDP relaxation of POP

(a) Using the max. cliques J_q $(q \in L)$ of $G(N, \bar{E})$, convert POP into an equivalent poly.SDP with the csp graph $G(N, \bar{E})$.

(b) Linearize poly.SDP $\Rightarrow$ SDP with a similar sparsity to poly.SDP.

$\forall$ int. $s \geq 0$, $\boldsymbol{u}_s(x_i : i \in J_q)$: the col. vect. of monomials with deg. at most s in var. x_i $(i \in J_q)$; *e.g.*, if $J_q = \{1, 4\}$,

$$s = 1 \Rightarrow \boldsymbol{u}_1(x_i : i \in J_q) = (1, x_1, x_4)^T,$$

$$s = 1 \Rightarrow \boldsymbol{u}_1(x_i : i \in J_q)\boldsymbol{u}_1(x_i : i \in J_q)^T = \begin{pmatrix} 1 & x_1 & x_4 \\ x_1 & x_1^2 & x_1 x_4 \\ x_4 & x_1 x_4 & x_4^2 \end{pmatrix}$$

Example of Sparse SDP Relaxation

POP: min $\sum_{i=1}^{4}(-x_i^3)$ s.t. $-a_i \times x_i^2 - x_4^2 + 1 \geq 0 \ (i = 1, 2, 3)$.

$$\text{csp matrix } \boldsymbol{R} = \begin{pmatrix} \star & 0 & 0 & \star \\ 0 & \star & 0 & \star \\ 0 & 0 & \star & \star \\ \star & \star & \star & \star \end{pmatrix}$$

No fill-in in the Cholesky factorization $\Rightarrow$ c-sparse.

Example of Sparse SDP Relaxation

POP: min $\sum_{i=1}^{4}(-x_i^3)$ s.t. $-a_i \times x_i^2 - x_4^2 + 1 \geq 0 \ (i = 1, 2, 3)$.

$\Updownarrow$ with the relaxation order $r = 2 \geq r_0 = \lceil 3/2 \rceil = 2$

poly.SDP:

min $\sum_{i=1}^{4}(-x_i^3)$

s.t. $(-a_i \times x_i^2 - x_4^2 + 1)(1, x_i, x_4)^T(1, x_i, x_4) \succeq \boldsymbol{O} \quad i = 1, 2, 3,$

$(1, x_j, x_4, x_j^2, x_j x_4, x_4^2)^T(1, x_j, x_4, x_j^2, x_j x_4, x_4^2) \succeq \boldsymbol{O} \ j = 1, 2, 3.$

$$\text{csp matrix } \boldsymbol{R} = \begin{pmatrix} \star & 0 & 0 & \star \\ 0 & \star & 0 & \star \\ 0 & 0 & \star & \star \\ \star & \star & \star & \star \end{pmatrix}$$

No fill-in in the Cholesky factorization $\Rightarrow$ c-sparse.

Example of Sparse SDP Relaxation

POP: min $\sum_{i=1}^{4}(-x_i^3)$ s.t. $-a_i \times x_i^2 - x_4^2 + 1 \geq 0 \ (i = 1,2,3)$.

$\Updownarrow$ with the relaxation order $r = 2 \geq r_0 = \lceil 3/2 \rceil = 2$

poly.SDP:

min $\sum_{i=1}^{4}(-x_i^3)$

s.t. $(-a_i \times x_i^2 - x_4^2 + 1)(1, x_i, x_4)^T(1, x_i, x_4) \succeq \boldsymbol{O} \quad i = 1,2,3,$

$(1, x_j, x_4, x_j^2, x_j x_4, x_4^2)^T(1, x_j, x_4, x_j^2, x_j x_4, x_4^2) \succeq \boldsymbol{O} \ j = 1,2,3.$

Represent poly.SDP as

min $\sum_{\boldsymbol{\alpha}\in\mathcal{A}_0} g_0(\boldsymbol{\alpha})\boldsymbol{x}^{\boldsymbol{\alpha}}$ s.t. $\sum_{\boldsymbol{\alpha}\in\mathcal{A}_j} \boldsymbol{G}_j(\boldsymbol{\alpha})\boldsymbol{x}^{\boldsymbol{\alpha}} \succeq \boldsymbol{O} \ j = 1,\ldots,6,$

where $\mathcal{A}_j \subset \mathbb{Z}_+^4$ and $\boldsymbol{x}^\alpha = x_1^{\alpha_1}x_2^{\alpha_2}x_3^{\alpha_3}x_4^{\alpha_4}$; $\boldsymbol{x}^{(1,2,1,0)} = x_1 x_2^2 x_3$.

$\Downarrow$ Linearize by replacing each $\boldsymbol{x}^{\boldsymbol{\alpha}}$ by an indep. var. $y_{\boldsymbol{\alpha}}$; $\boldsymbol{x}^0$ by 1

SDP min $\displaystyle\sum_{\boldsymbol{\alpha}\in\mathcal{A}_0} g_0(\boldsymbol{\alpha})y_{\boldsymbol{\alpha}}$ s.t. $\displaystyle\sum_{\boldsymbol{\alpha}\in\mathcal{A}_j} \boldsymbol{G}_j(\boldsymbol{\alpha})y_{\boldsymbol{\alpha}} \succeq \boldsymbol{O} \ j = 1,\ldots,6,$

which forms an SDP relaxation of POP.

Outline

1. POPs (Polynomial Optimization Problems)
2. How do we formulate structured sparsity?
3. Sparse SDP relaxation of POPs
4. Numerical results
5. Concluding remarks

Numerical result on unconstrained optimization

Unconstrained POP: min. $f(\boldsymbol{x})$, $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$.

Broyden tridiagonal function with min.val.$= 0$

$f(\boldsymbol{x}) = \sum_{i=1}^{n} \left((3-2x_i)x_i - x_{i-1} - 2x_{i+1} + 1\right)^2, \ x_0 = x_{n+1} = 0.$

Generalized Rosenbrock function with min.val.$= 0$

$f(\boldsymbol{x}) = \sum_{i=2}^{n} \left(100(x_i - x_{i-1}^2)^2 + (1-x_i)^2\right), \ x_1 \geq 0.$.

		B. tridiagonal		G. Rosenbrock	
n	r	approx.opt.val	cpu	approx.opt.val	cpu
600	2	1.0e-7	9.3	3.9e-7	3.4
800	2	2.2e-7	12.6	2.1e-7	5.1
1000	2	2.6e-7	16.0	4.5e-7	5.9

An optimal control problem from Coleman et al. 1995

$$\left.\begin{array}{ll} \text{min} & \dfrac{1}{M}\displaystyle\sum_{i=1}^{M-1}(y_i^2 + x_i^2) \\ \text{s.t.} & y_{i+1} = y_i + \dfrac{1}{M}(y_i^2 - x_i), \quad (i = 1, \ldots, M-1), \quad y_1 = 1. \end{array}\right\}$$

Numerical results on sparse relaxation

M	r	# of variables	ϵ_{obj}	ϵ_{feas}	cpu
600	2	1198	3.4e-08	2.2e-10	3.4
800	2	1598	5.9e-08	1.6e-10	3.8
1000	2	1998	6.3e-08	2.7e-10	5.0

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

ϵ_{feas} = the maximum error in the equality constraints,

cpu : cpu time in sec. to solve an SDP relaxation problem.

alkyl.gms : a benchmark problem from globalib

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

	Sparse			Dense (Lasserre)		
r	ϵ_{obj}	ϵ_{feas}	cpu	ϵ_{obj}	ϵ_{feas}	cpu
2	1.0e-02	7.1e-01	1.8	7.2e-3	4.3e-2	14.4
3	5.6e-10	2.0e-08	23.0	out of	memory	

ϵ_{obj} = approx.opt.val. – lower bound for opt.val.
ϵ_{feas} = the maximum error in the equality constraints

Some other benchmark problems from globalib

		Sparse			Dense (Lasserre)	
problem	n	ϵ_{obj}	ϵ_{feas}	cpu	ϵ_{obj}	cpu
ex3_1_1	8	6.3e-9	4.7e-4	5.5	0.7e-8	597.8
st_bpaf1b	10	3.8e-8	2.8e-8	1.0	4.6e-9	1.7
st_e07	10	0.0	8.1e-5	0.4	0.0	3.0
ex2_1_3	13	5.1e-09	3.5e-9	0.5	1.6e-9	7.7
ex9_1_1	13	0.0	4.5e-6	1.5	0.0	7.7
ex9_2_3	16	0.0	5.7e-6	2.3	0.0	49.7
ex2_1_8	24	1.0e-5	0.0	304.6	3.4e-6	1946.6
ex5_2_2	9	1.0e-2	3.2e+1	1.8	1.6e-5	2.6

- ex5_2_2 — Dense is better.
- Sparse attains approx. opt. solutions with the same quality as Dense except ex5_2_2.
- Sparse is much faster than Dense in large dim. cases.

Discretization of Mimura's ODE with 2 unknowns $u,\ v : [0,5] \to \mathbb{R}$

$$
\begin{aligned}
u_{xx} &= -(\delta_1/9)(35 + 16u - u^2)u + (\delta_1)(kuv), \\
v_{xx} &= (\delta_2)((1 + (2/5)v)v - kuv), \\
& \underline{u_x(0) = u_x(5) = v_x(0) = v_x(5) = 0,}
\end{aligned}
$$

where $k = 1$, $\delta_1 = 20$ and $\delta_2 = 1/4$. Discretize:

$x_i = i\Delta x \ (i = 0, 1, 2, \dots),\ u_x(x_i) \approx (u(x_{i+1}) - u(x_{i-1}))/(2\Delta x)$.

Discretized system of polynomials with $\Delta x = 1$:

$$
\begin{aligned}
f_1(\boldsymbol{u},\boldsymbol{v}) &= 76.8u_1 + u_3 + 35.6u_1^2 - 20.0u_1v_1 - 2.22u_2^3, \\
f_2(\boldsymbol{u},\boldsymbol{v}) &= -1.25v_1 + v_2 + 0.25u_1v_1 - 0.1v_1^2, \\
f_3(\boldsymbol{u},\boldsymbol{v}) &= u_1 + 75.8u_2 + u_3 + 35.6u_2^2 - 20.0u_2v_2 - 2.22u_2^3, \\
f_4(\boldsymbol{u},\boldsymbol{v}) &= v_1 - 2.25v_2 + v_3 + 0.25u_2v_2 - 0.1v_2^2, \\
f_5(\boldsymbol{u},\boldsymbol{v}) &= u_2 + 75.8u_3 + u_4 + 35.6u_3^2 - 20.0u_3v_3 - 2.22u_3^2, \\
f_6(\boldsymbol{u},\boldsymbol{v}) &= v_2 - 2.25v_3 + v_4 + 0.25u_3v_3 - 0.1v_3^2, \\
f_7(\boldsymbol{u},\boldsymbol{v}) &= u_3 + 76.8u_4 + 35.6u_4^2 - 20.0u_4v_4 - 2.22u_4^3, \\
f_8(\boldsymbol{u},\boldsymbol{v}) &= v_3 - 1.25v_4 + 0.25u_4v_4 - 0.1v_4^2.
\end{aligned}
$$

Here $u_i = u(x_i)$, $v_i = v(x_i)$ $(i = 0, 1, 2, 3, 4, 5)$,
$u_0 = u_1$, $u_5 = u_4$, $v_0 = v_1$ and $v_5 = v_4$.

$\Rightarrow$ c-sparse

Discretization of Mimura's ODE with 2 unknowns $u, \ v : [0,5] \to \mathbb{R}$

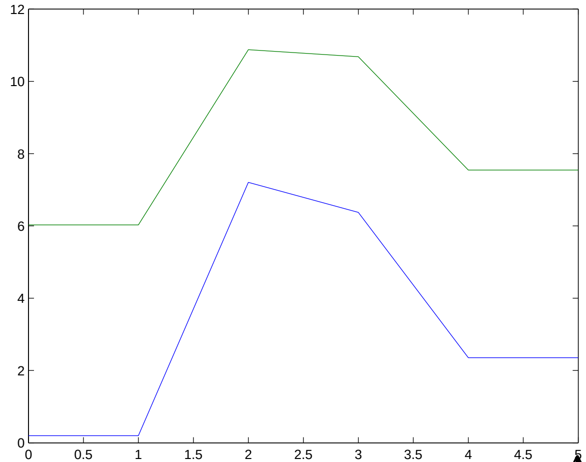
$$u_{xx} = -(\delta_1/9)(35 + 16u - u^2)u + (\delta_1)(kuv),$$
$$v_{xx} = (\delta_2)((1 + (2/5)v)v - kuv),$$
$$u_x(0) = u_x(5) = v_x(0) = v_x(5) = 0,$$

where $k = 1$, $\delta_1 = 20$ and $\delta_2 = 1/4$. Discretize:
$x_i = i\Delta x \ (i = 0, 1, 2, \dots), \ u_x(x_i) \approx (u(x_{i+1}) - u(x_{i-1}))/(2\Delta x)$.

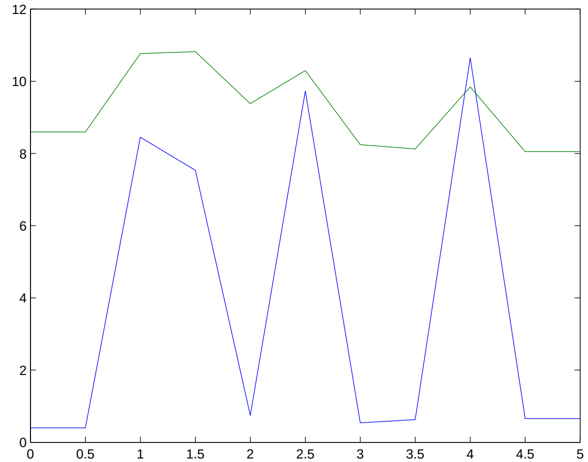
- Numerical results on SparsePOP

Δx	n	obj.funct.	r	$\boldsymbol{A}$ in SeDuMi	cpu
1.0	8	$\sum r_i u(x_i) \uparrow$	3	[1,084, 18,732]	11.3
0.5	18	$\sum r_i u(x_i) \uparrow$	3	[3,025, 48,285]	57.8

Here $r_i \in (0,1)$: random numbers.



$\Delta x = 1.0$



$\Delta x = 0.5$

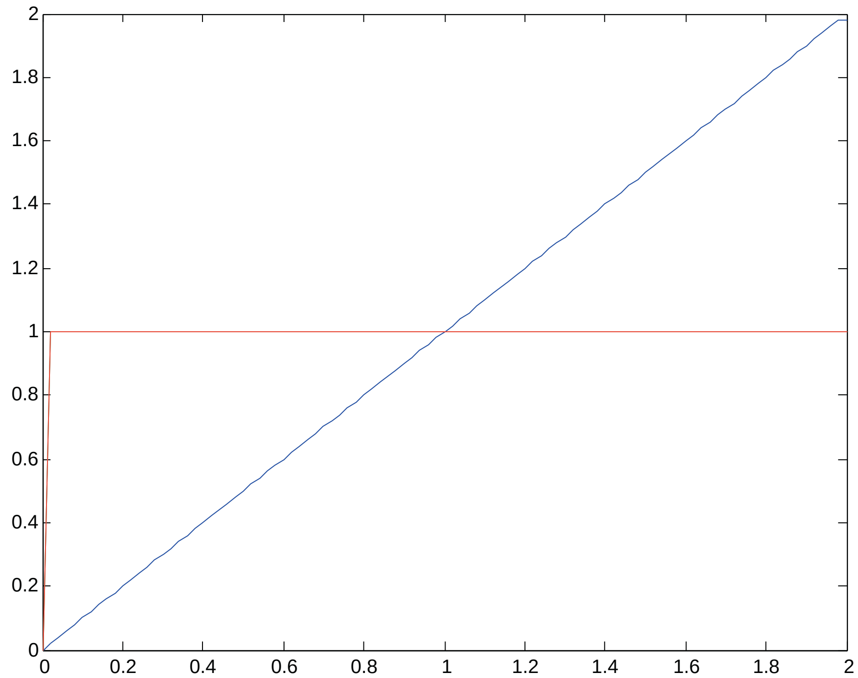
Discretization of DAE with 3 unknowns $y_1, y_2, y_3 : [0, 2] \to \mathbb{R}$

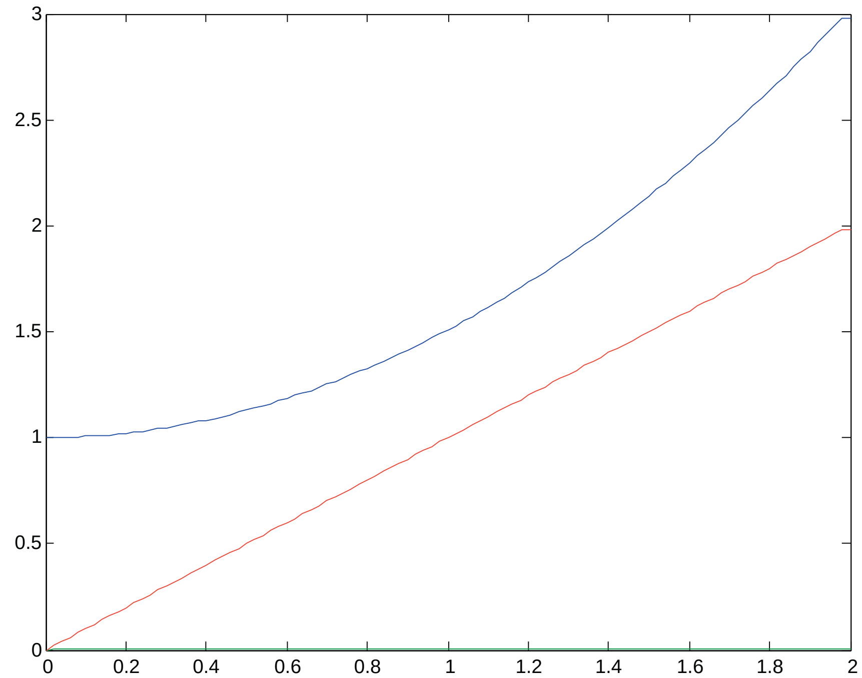
$y_1' = y_3,\ 0 = y_2(1 - y_2),\ 0 = y_1 y_2 + y_3(1 - y_2) - t,\ y_1(0) = y_1^0.$

2 solutions : $y(t) = (t, 1, 1)$ and $y(t) = (y_1^0 + t^2, 0, t)$.

- Numerical results on SparsePOP — c-sparse

y_1^0	Δt	n	obj.funct.	r	$\boldsymbol{A}$ in SeDuMi	cpu
0	0.02	297	$\sum y_2(t_i) \uparrow$	2	[3,557, 25,413]	30.9
1	0.02	297	$\sum y_1(t_i) \uparrow$	2	[3,557, 25,413]	33.9





Outline

1. POPs (Polynomial Optimization Problems)
2. How do we formulate structured sparsity?
3. Sparse SDP relaxation of POPs
4. Numerical results

5. Concluding remarks

- Lasserre's (dense) relaxation
  — Theoretical convergence but expensive in practice.
- Sparse relaxation (Waki-Kim-Kojima-Muramatsu)
  = Lasserre's (dense) relaxation + sparsity
  — Very powerful in practice
  and theoretical convergence (Lasserre)
- There remain many issues to be studied further.
  - Exploiting sparsity.
  - Large-scale SDPs.
  - Numerical difficulty in solving SDP relaxations of POPs.

Thank you!