Chordal Graphs, Semidefinite Optimization, and Sum-of-squares Matrices

Yang Zheng

Assistant Professor, ECE, UC San Diego
Scalable Optimization and Control (SOC) Lab

UCSD Optimization and Data Science Seminar
October 27, 2021

UC San Diego
JACOBS SCHOOL OF ENGINEERING
Electrical and Computer Engineering

SOC Lab

Acknowledgments

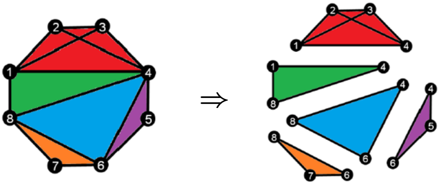
UNIVERSITY OF OXFORD

Imperial College London

Outline

1. Introduction: Chordal graphs and Matrix decomposition
2. Part I - Decomposition in sparse semidefinite optimization
3. Part II - Decomposition in PSD polynomial matrices
4. Conclusion

Introduction: Chordal graphs and Matrix decomposition



Matrix decomposition and chordal graphs

Matrix decomposition:

- A simple example

$$A = \underbrace{\begin{bmatrix} 3 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 3 \end{bmatrix}}_{\succeq 0} = \underbrace{\begin{bmatrix} 3 & 1 & 0 \\ 1 & 0.5 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\succeq 0} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 1 \\ 0 & 1 & 3 \end{bmatrix}}_{\succeq 0}$$

- This is true for any PSD matrix with such pattern, *i.e.*, sparse cone decomposition

$$\underbrace{\begin{bmatrix} * & * & 0 \\ * & * & * \\ 0 & * & * \end{bmatrix}}_{\succeq 0} = \underbrace{\begin{bmatrix} * & * & 0 \\ * & * & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\succeq 0} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & * & * \\ 0 & * & * \end{bmatrix}}_{\succeq 0}$$

where $*$ denotes a real scalar number (or block matrix).

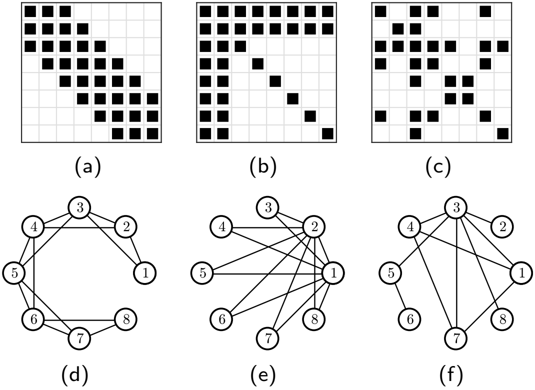
Benefits:

- Reduce computational complexity, and thus improve efficiency! $(3 \times 3 \rightarrow 2 \times 2)$

Matrix decomposition and chordal graphs

Matrix decomposition:

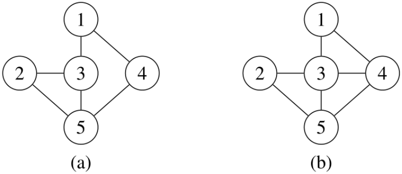
- Many other patterns admit similar decompositions, *e.g.*



- They can be commonly characterized by **chordal graphs**.

Chordal graphs

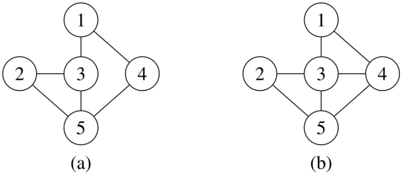
Chordal graphs: An undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is called *chordal* if every cycle of length greater than three has a chord.



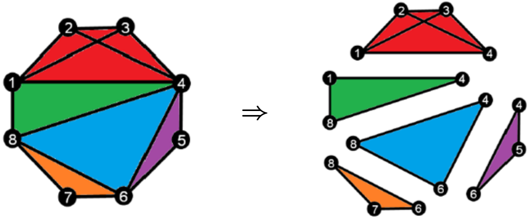
Notation: (Vandenberghe & Andersen, 2014)

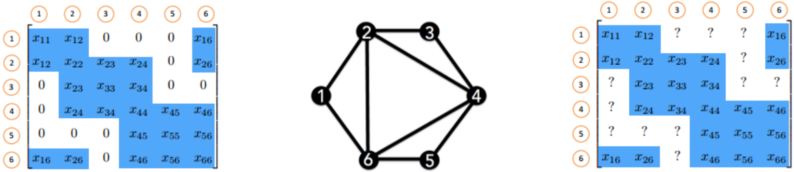
- *Chordal extension:* Any non-chordal graph can be chordal extended;
- *Maximal clique:* A clique is a set of nodes that induces a complete subgraph;
- *Clique decomposition:* A chordal graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ can be decomposed into a set of maximal cliques $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_t\}$.

Chordal graphs: An undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is called *chordal* if every cycle of length greater than three has a chord.



Clique decomposition:





Sparse positive semidefinite (PSD) matrices

$$\mathbb{S}^n(\mathcal{E},0) = \{X \in \mathbb{S}^n \mid X_{ij} = X_{ji} = 0, \forall (i,j) \notin \mathcal{E}\},$$
$$\mathbb{S}^n_+(\mathcal{E},0) = \{X \in \mathbb{S}^n(\mathcal{E},0) \mid X \succeq 0\}.$$

Positive semidefinite completable matrices

$$\mathbb{S}^n(\mathcal{E},?) = \{X \in \mathbb{S}^n \mid X_{ij} = X_{ji}, \text{ given if } (i,j) \in \mathcal{E}\},$$
$$\mathbb{S}^n_+(\mathcal{E},?) = \{X \in \mathbb{S}^n(\mathcal{E},?) \mid \exists M \succeq 0, M_{ij} = X_{ij}, \forall (i,j) \in \mathcal{E}\}.$$

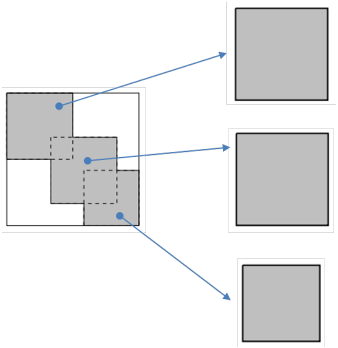
$\mathbb{S}^n_+(\mathcal{E},0)$ and $\mathbb{S}^n_+(\mathcal{E},?)$ are dual to each other.

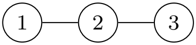
Two matrix decomposition theorems

Clique decomposition for PSD completable matrices (Grone, *et al.*, 1984)

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a chordal graph with maximal cliques $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_p\}$. Then,

$$X \in \mathbb{S}_+^n(\mathcal{E}, ?) \Leftrightarrow E_{\mathcal{C}_k} X E_{\mathcal{C}_k}^{\mathsf{T}} \in \mathbb{S}_+^{|\mathcal{C}_k|}, \qquad k = 1, \ldots, p.$$





$$\begin{bmatrix} X_{11} & X_{12} & ? \\ X_{21} & X_{22} & X_{23} \\ ? & X_{32} & X_{33} \end{bmatrix} \in \mathbb{S}_+^3(\mathcal{E}, ?)$$

$$\Updownarrow$$

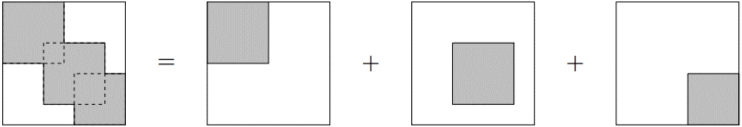
$$\begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \succeq 0$$

$$\begin{bmatrix} X_{22} & X_{23} \\ X_{32} & X_{33} \end{bmatrix} \succeq 0$$

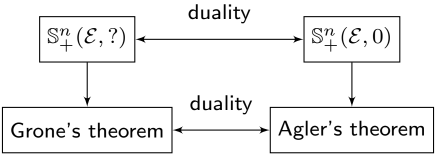
Clique decomposition for PSD matrices (Agler, Helton, McCullough, & Rodman, 1988; Griewank and Toint, 1984)

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a chordal graph with maximal cliques $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_p\}$. Then,

$$Z \in \mathbb{S}_+^n(\mathcal{E}, 0) \Leftrightarrow Z = \sum_{k=1}^{p} E_{\mathcal{C}_k}^{\mathsf{T}} Z_k E_{\mathcal{C}_k}, \; Z_k \in \mathbb{S}_+^{|\mathcal{C}_k|}$$



Sparse Cone Decomposition



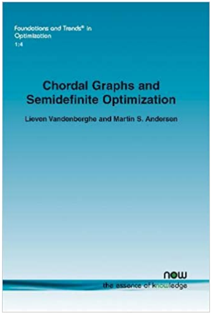
A growing number of applications

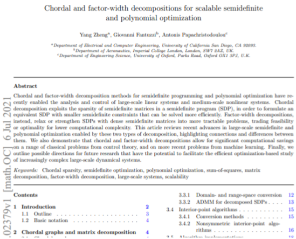
Control, machine learning, relaxation of QCQP, fluid dynamics, and beyond

Area	Topic	References
Control	Linear system analysis	Andersen et al. (2014b); Deroo et al. (2015); Mason & Papachristodoulou (2014); Pakazad et al. (2017b); Zheng et al. (2018c)
	Decentralized control	Deroo et al. (2014); Heinke et al. (2020); Zheng et al. (2020); Zheng et al. (2018d)
	Nonlinear system analysis	Schlosser & Korda (2020); Tacchi et al. (2019a); Zheng et al. (2019a); Mason (2015, Chapter 5)
	Model predictive control	Ahmadi et al. (2019); Hansson & Pakazad (2018)
Machine learning	Verification of neural networks	Batten et al. (2021); Dvijotham et al. (2020); Newton & Papachristodoulou (2021); Zhang (2020)
	Lipschitz constant estimation	Chen et al. (2020b); Latorre et al. (2020)
	Training of support vector machine	Andersen & Vandenberghe (2010)
	Geometric perception & coarsening	Chen et al. (2020a); Liu et al. (2019); Yang & Carlone (2020)
	Covariance selection	Dahl et al. (2008); Zhang et al. (2018)
	Subspace clustering	Miller et al. (2019a)
Relaxation of QCQP and POPs	Sensor network locations	Jing et al. (2019); Kim et al. (2009); Nie (2009)
	Max-Cut problem	Andersen et al. (2010a); Garstka et al. (2019); Zheng et al. (2020)
	Optimal power flow (OPF)	Andersen et al. (2014a); Dall'Anese et al. (2013); Jabr (2011); Jiang (2017); Molzahn & Hiskens (2014); Molzahn et al. (2013)
	State estimation in power systems	Weng et al. (2013); Zhang et al. (2017); Zhu & Giannakis (2014)
Others	Fluid dynamics	Arslan et al. (2021); Fantuzzi et al. (2018)
	Partial differential equations	Mevissen (2010); Mevissen et al. (2008, 2011, 2009)
	Robust quadratic optimization	Andersen et al. (2010b)
	Binary signal recovery	Fosson & Abuabiah (2019)
	Solving polynomial systems	Cifuentes & Parrilo (2016, 2017); Li et al. (2021); Mou et al. (2021); Tacchi et al. (2019b)
	Other problems	Baltean-Lugojan et al. (2019); Jeyakumar et al. (2016); Madani et al. (2017b); Pakazad et al. (2017a); Yang & Deng (2020)

This talk

Two survey papers





Chordal and factor-width decompositions for scalable semidefinite and polynomial optimization

Yang Zheng[a], Giovanni Fantuzzi[b], Antonis Papachristodoulou[c]

[a]Department of Electrical and Computer Engineering, University of California San Diego, CA 92093.
[b]Department of Aeronautics, Imperial College London, London, SW7 2AZ, UK.
[c]Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, U.K.

Abstract

Chordal and factor-width decomposition methods for semidefinite programming and polynomial optimization have recently enabled the analysis and control of large-scale linear systems and medium-scale nonlinear systems. Chordal decomposition exploits the sparsity of semidefinite matrices in a semidefinite program (SDP), in order to formulate an equivalent SDP with smaller semidefinite constraints that can be solved more efficiently. Factor-width decompositions, instead, relax or strengthen SDPs with dense semidefinite matrices into more tractable problems, trading feasibility or optimality for lower computational complexity. This article reviews recent advances in large-scale semidefinite and polynomial optimization enabled by these two types of decomposition, highlighting connections and differences between them. We also demonstrate that chordal and factor-width decompositions allow for significant computational savings on a range of classical problems from control theory, and on more recent problems from machine learning. Finally, we outline possible directions for future research that have the potential to facilitate the efficient optimization-based study of increasingly complex large-scale dynamical systems.

Keywords: Chordal sparsity, semidefinite optimization, polynomial optimization, sum-of-squares, matrix decomposition, factor-width decomposition, large-scale systems, scalability

Contents

1 Introduction 2
1.1 Outline 3
1.2 Basic notation 4
2 Chordal graphs and matrix decomposition 4
3.3.1 Domain- and range-space conversion 12
3.3.2 ADMM for decomposed SDPs 13
3.4 Interior-point algorithms 15
3.4.1 Conversion methods 15
3.4.2 Nonsymmetric interior-point algorithms 16

.02379v1 [math.OC] 6 Jul 2021

- **Part I: Decomposition in sparse semidefinite optimization**
  - Zheng, Y., Fantuzzi, G., Papachristodoulou, A., Goulart, P., & Wynn, A. (2020). Chordal decomposition in operator-splitting methods for sparse semidefinite programs. Mathematical Programming, 180(1), 489-532.
- **Part II: Decomposition in polynomial matrix inequalities (PMIs)**
  - Zheng, Y., & Fantuzzi, G. (2021). Sum-of-squares chordal decomposition of polynomial matrix inequalities. Mathematical Programming (accepted).

Part I: Decomposition in sparse semidefinite optimization

Semidefinite programs (SDPs)

$$\begin{aligned} \min \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i, i = 1, \ldots, m, \\ & X \succeq 0. \end{aligned}$$

$$\begin{aligned} \max_{y,\, Z} \quad & \langle b, y \rangle \\ \text{subject to} \quad & Z + \sum_{i=1}^{m} A_i\, y_i = C, \\ & Z \succeq 0. \end{aligned}$$

where $X \succeq 0$ means X is positive semidefinite.

- **Applications:** Control theory, fluid dynamics, polynomial optimization, *etc*.
- **Interior-point solvers:** SeDuMi, SDPA, SDPT3, MOSEK (suitable for small and medium-sized problems); *Modelling package:* YALMIP, CVX
- **Large-scale cases:** it is important to exploit the inherent structure
  - Low rank;
  - Algebraic symmetry;
  - **Chordal sparsity**
    - Second-order methods: Fukuda *et al.*, 2001; Nakata *et al.*, 2003; Burer 2003; Andersen *et al.*, 2010.
    - **First-order methods**: Madani *et al.*, 2015; Sun, Andersen, and Vandenberghe, 2014.

Aggregate sparsity pattern of matrices

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, A_1 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}, A_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \Longrightarrow \begin{bmatrix} * & * & 0 \\ * & * & * \\ 0 & * & * \end{bmatrix}$$

Primal SDP

$$\begin{aligned} \min \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_1, X \rangle = b_1 \\ & \langle A_2, X \rangle = b_2 \\ & X \succeq 0. \end{aligned}$$

Dual SDP

$$\begin{aligned} \max_{y,\, Z} \quad & \langle b, y \rangle \\ \text{subject to} \quad & y_1 A_1 + y_2 A_2 + Z = C, \\ & Z \succeq 0. \end{aligned}$$

$X \in \begin{bmatrix} * & * & ? \\ * & * & * \\ ? & * & * \end{bmatrix}$	Patterns of feasible solutions	$Z \in \begin{bmatrix} * & * & 0 \\ * & * & * \\ 0 & * & * \end{bmatrix}$
$X \in \mathbb{S}^3_+(\mathcal{E}, ?)$	Cone replacement	$Z \in \mathbb{S}^3_+(\mathcal{E}, 0)$

Apply the clique decomposition on $\mathbb{S}^3_+(\mathcal{E}, ?)$ **and** $\mathbb{S}^3_+(\mathcal{E}, 0)$

- Fukuda *et al.*, 2001; Nakata *et al.*, 2003; Andersen *et al.*, 2010; Madani *et al.*, 2015; Sun, Andersen, and Vandenberghe, 2014.

Cone decomposition of sparse SDPs

Primal SDP

$$\begin{aligned} \min \quad & \langle C, X\rangle \\ \text{subject to} \quad & \langle A_i, X\rangle = b_i, i = 1,\ldots,m \\ & \boxed{X \succeq 0}. \end{aligned}$$

Dual SDP

$$\begin{aligned} \max_{y,\,Z} \quad & \langle b, y\rangle \\ \text{subject to} \quad & \sum_{i=1}^{m} y_i A_i + Z = C, \\ & \boxed{Z \succeq 0}. \end{aligned}$$

Cone replacement

(Assuming an aggregate sparsity pattern $\mathcal{E}$)

$X \in \mathbb{S}^n_+(\mathcal{E}, ?)$ $\Downarrow$

$Z \in \mathbb{S}^n_+(\mathcal{E}, 0)$ $\Downarrow$

$$\begin{aligned} \min \quad & \langle C, X\rangle \\ \text{s.t.} \quad & \langle A_i, X\rangle = b_i, i = 1,\ldots,m \\ & \boxed{E_{\mathcal{C}_k} X E_{\mathcal{C}_k}^{\mathsf{T}} \succeq 0, k = 1,\ldots,p}. \end{aligned}$$

$$\begin{aligned} \max_{y,\,Z} \quad & \langle b, y\rangle \\ \text{s.t.} \quad & \sum_{i=1}^{m} y_i A_i + \sum_{k=1}^{p} E_{\mathcal{C}_k}^{\mathsf{T}} Z_k E_{\mathcal{C}_k} = C, \\ & \boxed{Z_k \succeq 0, k = 1,\ldots,p} \end{aligned}$$

- **A big sparse PSD cone** is equivalently replaced by a set of **coupled small** PSD cones;
- Our idea: **consensus** variables $\Rightarrow$ decouple the coupling constraints;

Decomposed SDPs for operator-splitting algorithms

Primal decomposed SDP

$$\begin{aligned}
\min_{X, X_k} \quad & \langle C, X \rangle \\
\text{s.t.} \quad & \langle A_i, X \rangle = b_i, \qquad i = 1, \ldots, m, \\
& \boxed{X_k = E_{\mathcal{C}_k} X E_{\mathcal{C}_k}^{\mathsf{T}}, k = 1, \ldots, p,} \\
& X_k \in \mathbb{S}_+^{|\mathcal{C}_k|}, \qquad k = 1, \ldots, p.
\end{aligned}$$

Dual decomposed SDP

$$\begin{aligned}
\max_{y, Z_k, V_k} \quad & \langle b, y \rangle \\
\text{s.t.} \quad & \sum_{i=1}^{m} A_i \, y_i + \sum_{k=1}^{p} E_{\mathcal{C}_k}^{\mathsf{T}} V_k E_{\mathcal{C}_k} = C, \\
& \boxed{Z_k - V_k = 0, \ k = 1, \ldots, p,} \\
& Z_k \in \mathbb{S}_+^{|\mathcal{C}_k|}, \quad k = 1, \ldots, p.
\end{aligned}$$

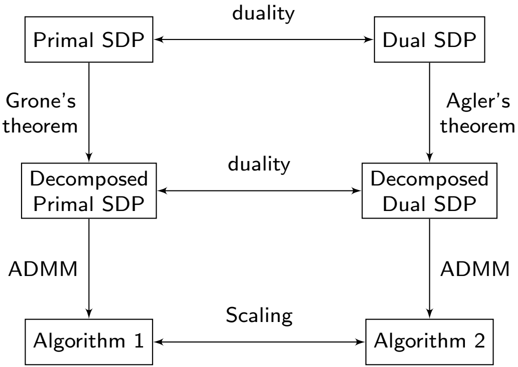
- A set of slack consensus variables has been introduced;
- The slack variables allow one to **separate the conic and the affine constraints** when using operator-splitting algorithms $\Rightarrow$ fast iterations:

projection on affine space
$+$ parallel projections on multiple small PSD cones
$\mathbb{S}_+^{|\mathcal{C}_k|}, k = 1, \ldots, p$

ADMM for primal and dual decomposed SDPs

Equivalence between the primal and dual cases

- ADMM steps in the dual form are scaled versions of those in the primal form.
- Extension to the homogeneous self-dual embedding exists.



Both algorithms only require conic projections onto small PSD cones. **Complexity depends on the largest maximal cliques, instead of the original dimension!**

Comparison with other first-order algorithms

Key difference: How to decouple the coupling constraints

Table 1: Comparison of first-order algorithms for solving SDPs. "Chordal Sparsity": whether the algorithm exploits chordal sparsity; "SDP Type": the types of SDP problems the algorithm considers; "Algorithm": the underlying first-order algorithm; "infeas./unbounded": whether the algorithm can detect infeasible or unbounded cases; "Solver": whether the code is open-source.

Reference	Chordal Sparsity	SDP Type	Algorithm	Infeas./ Unbounded	Solver
Wen et al. (2010)	✗	(3.2)	ADMM	✗	✗
Zhao et al. (2010)	✗	(3.2)	Augm. Lagrang.	✗	SDPNAL
O'Donoghue et al. (2016)	✗	(3.1)-(3.2)	ADMM	✓	SCS
Yurtsever et al. (2021)	✗	(3.1)[1]	SketchyCGAL	✗	CGAL
Lu et al. (2007)	✓	(3.1)	Mirror-Prox	✗	✗
Lam et al. (2012)	✓	OPF[2]	Primal-dual	✗	✗
Dall'Anese et al. (2013)	✓	OPF[2]	ADMM	✗	✗
Sun et al. (2014)	✓	Special[3]	Gradient proj.	✗	✗
Sun & Vandenberghe (2015)	✓	(3.1)-(3.2)	Spingarn	✗	✗
Kalbat & Lavaei (2015)	✓	Special[4]	ADMM	✗	✗
Madani et al. (2017a)	✓	General[5]	ADMM	✗	✗
Zheng et al. (2020)	✓	(3.1)-(3.2)	ADMM	✓	CDCS
Garstka et al. (2019)	✓	Quad. SDP[6]	ADMM	✓	COSMO

Note: 1. It requires an explicit trace constraint on X; 2. Special SDPs from the optimal power flow (OPF) problem; 3. Special SDPs from the matrix nearness problem; 4. Special SDPs with decoupled affine constraints; 5. General SDPs with inequality constraints; 6. A dual SDP (3.2) with a quadratic objective function.

CDCS

Cone decomposition conic solver

- An open source MATLAB solver for sparse conic programs (Julia interface);
- CDCS supports constraints on the following cones:
  - Free variables
  - non-negative orthant
  - second-order cone
  - the positive semidefinite cone.
- Input-output format: SeDuMi; Interface via YALMIP, SOSTOOLS.
- Syntax: `[x,y,z,info] = cdcs(At,b,c,K,opts);`

Download from `https://github.com/OxfordControl/CDCS`

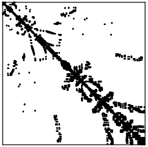
Numerical comparison

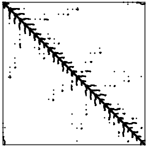
- SeDuMi (interior-point solver): default parameters, and low-accuracy solution 10^{-3}
- SCS (first-order solver)
- CDCS and SCS: stopping condition 10^{-3} (max. iterations 2000)
- All simulations were run on a PC with a 2.8 GHz Intel Core i7 CPU and 8GB of RAM.

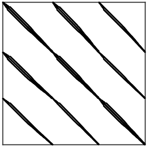
Large-scale sparse SDPs

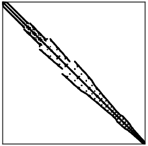
Instances from Andersen, Dahl, Vandenberghe, 2010

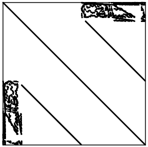
	rs35	rs200	rs228	rs365	rs1555	rs1907
Original cone size, n	2003	3025	1919	4704	7479	5357
Affine constraints, m	200	200	200	200	200	200
Number of cliques, p	588	1635	783	1244	6912	611
Maximum clique size	418	102	92	322	187	285
Minimum clique size	5	4	3	6	2	7

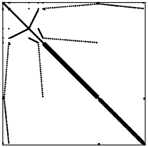

rs35


rs200


rs228


rs365


rs1555


rs1907

Large-scale sparse SDPs: Numerical results

	rs35			rs200		
	Time (s)	# Iter.	Objective	Time (s)	# Iter.	Objective
SeDuMi (high)	1 391	17	25.33	4 451	17	99.74
SeDuMi (low)	986	11	25.34	2 223	8	99.73
SCS (direct)	2 378	†2 000	25.08	9 697	†2 000	81.87
CDCS-primal	370	379	25.27	159	577	99.61
CDCS-dual	272	245	25.53	103	353	99.72
CDCS-hsde	208	198	25.64	54	214	99.77

	rs228			rs365		
	Time (s)	# Iter.	Objective	Time (s)	# Iter.	Objective
SeDuMi (high)	1 655	21	64.71	***	***	***
SeDuMi (low)	809	10	64.80	***	***	***
SCS (direct)	2 338	†2 000	62.06	34 497	†2 000	44.02
CDCS-primal	94	400	64.65	321	401	63.37
CDCS-dual	84	341	64.76	240	265	63.69
CDCS-hsde	38	165	65.02	151	175	63.75

	rs1555			rs1907		
	Time (s)	# Iter.	Objective	Time (s)	# Iter.	Objective
SeDuMi (high)	***	***	***	***	***	***
SeDuMi (low)	***	***	***	***	***	***
SCS (direct)	139 314	†2 000	34.20	50 047	†2 000	45.89
CDCS-primal	1 721	†2 000	61.22	330	349	62.87
CDCS-dual	317	317	69.54	271	252	63.30
CDCS-hsde	361	448	66.38	190	187	63.15

***: the problem could not be solved due to memory limitations.
†: maximum number of iterations reached.

Large-scale sparse SDPs: Numerical results

Average CPU time per iteration

	rs35	rs200	rs228	rs365	rs1555	rs1907
SCS (direct)	1.188	4.847	1.169	17.250	69.590	25.240
CDCS-primal	0.944	0.258	0.224	0.715	0.828	0.833
CDCS-dual	1.064	0.263	0.232	0.774	0.791	0.920
CDCS-hsde	1.005	0.222	0.212	0.733	0.665	0.891

- $20\times, 21\times, 26\times$, and $75\times$ faster than SCS, respectively, for problems rs200, rs365, rs1907, and rs1555.
- The computational benefit comes form the cone decomposition (projections onto small PSD cones)
- CDCS enables us to solve large, sparse conic problems with moderate accuracy that are beyond the reach of standard interior-point and/or other first-order methods

The conic projections in all Algorithms require $\mathcal{O}(\sum_{k=1}^{p} |\mathcal{C}_k|^3)$ flops. **Complexity is dominated by the largest maximal clique!**

Part II: Decomposition in PSD polynomial matrices

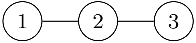
— sparsity-exploiting versions of the Hilbert-Artin, Reznick, Putinar, and Putinar-Vasilescu Positivstellensätze.

Positive (semi)-definite polynomial matrices

- Recall the simple example

$$A = \underbrace{\begin{bmatrix} 3 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 3 \end{bmatrix}}_{\succeq 0} = \underbrace{\begin{bmatrix} 3 & 1 & 0 \\ 1 & 0.5 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\succeq 0} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 1 \\ 0 & 1 & 3 \end{bmatrix}}_{\succeq 0}$$

- **How about positive (semi)-definite polynomial matrices?**



$$P(x) = \begin{bmatrix} p_{11}(x) & p_{12}(x) & 0 \\ p_{21}(x) & p_{22}(x) & p_{23}(x) \\ 0 & p_{32}(x) & p_{33}(x) \end{bmatrix} \succeq 0, \quad \forall x \in \mathcal{K}$$

$$\mathcal{K} = \mathbb{R}^n, \text{or}, \mathcal{K} = \{x \in \mathbb{R}^n \mid g_i(x) \geq 0, i = 1, \ldots, m\}$$

- **Point-wise:** the decomposition still holds, but can it be represented by polynomials or even better, by SOS matrices?

$$\underbrace{\begin{bmatrix} * & * & 0 \\ * & * & * \\ 0 & * & * \end{bmatrix}}_{\succeq 0} = \underbrace{\begin{bmatrix} * & * & 0 \\ * & * & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\succeq 0} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & * & * \\ 0 & * & * \end{bmatrix}}_{\succeq 0}, \qquad \forall x \in \mathcal{K}$$

Sum-of-squares (SOS) matrices

- Consider a symmetric matrix-valued polynomial

$$P(x) = \begin{bmatrix} p_{11}(x) & p_{12}(x) & \dots & p_{1r}(x) \\ p_{21}(x) & p_{22}(x) & \dots & p_{2r}(x) \\ \vdots & \vdots & \ddots & \vdots \\ p_{r1}(x) & p_{r2}(x) & \dots & p_{rr}(x) \end{bmatrix} \succeq 0, \forall x \in \mathbb{R}^n.$$

- The problem of checking whether $P(x)$ is positive semidefinite is NP-hard in general (even with $r = 1, d = 4$).
- **SOS representation:** We call $P(x)$ is an SOS matrix if

$$p(x, y) = y^{\mathsf{T}} P(x) y \text{ is SOS in } [x; y]$$

  A polynomial $q(x)$ is SOS if it can be written as $q(x) = \sum_{i=1}^{m} f_i(x)^2$.
- **SDP characterization (Parrilo et al.):** $P(x)$ is an SOS matrix if and only if there exists $Q \succeq 0$, such that

$$P(x) = (I_r \otimes v_d(x))^{\mathsf{T}} Q (I_r \otimes v_d(x)).$$

  where Q is called the Gram matrix, $v_d(x)$ is the standard monomial basis.

Negative result

There exists an n-variate $r \times r$ polynomial matrix $P(x)$ with chordal sparsity $\mathcal{G}$ that is strictly positive definite for all $x \in \mathbb{R}^n$, but cannot be written as the decomposition form with positive semidefinite polynomial matrices $S_k(x)$.

- **Example:**

$$P(x) = \begin{bmatrix} k+1+x^2 & x+x^2 & 0 \\ x+x^2 & k+2x^2 & x-x^2 \\ 0 & x-x^2 & k+1+x^2 \end{bmatrix} = \begin{bmatrix} x & 1 \\ x & x \\ 1 & -x \end{bmatrix} \begin{bmatrix} x & x & 1 \\ 1 & x & -x \end{bmatrix} + kI_3$$

- It is not difficult to show that

$$P(x) = \underbrace{\begin{bmatrix} a(x) & b(x) & 0 \\ b(x) & c(x) & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\succeq 0} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & d(x) & e(x) \\ 0 & e(x) & f(x) \end{bmatrix}}_{\succeq 0},$$

fails to exist when $0 \leq k < 2$.

- $P(x)$ is strictly positive definite if $0 < k < 2$.

Hilbert–Artin theorem

Sparse matrix version of the Hilbert–Artin theorem

Let $P(x)$ be an $m \times m$ positive semidefinite polynomial matrix whose sparsity graph is chordal and has maximal cliques $\mathcal{C}_1, \ldots, \mathcal{C}_t$. There exist an SOS polynomial $\sigma(x)$ and SOS matrices $S_k(x)$ of size $|\mathcal{C}_k| \times |\mathcal{C}_k|$ such that

$$\sigma(x)P(x) = \sum_{k=1}^{t} E_{\mathcal{C}_k}^{\mathsf{T}} S_k(x) E_{\mathcal{C}_k}.$$

- **Example:** $\sigma(x) = 1 + k + x^2$ suffices for the previous example

$$P(x) = \begin{bmatrix} k+1+x^2 & x+x^2 & 0 \\ x+x^2 & \dfrac{(1+x)^2x^2}{1+k+x^2} & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & \dfrac{k^2+k+3kx^2+(1-x)^2x^2}{1+k+x^2} & x-x^2 \\ 0 & x-x^2 & k+1+x^2 \end{bmatrix}$$

- PSD polynomial matrices are equivalent to SOS matrices when $n = 1$.

Reznick's Positivstellensatz

Sparse matrix version of Reznick's Positivstellensatz

Let $P(x)$ be an $m \times m$ homogeneous polynomial matrix whose sparsity graph is chordal and has maximal cliques $\mathcal{C}_1, \ldots, \mathcal{C}_t$. If P is strictly positive definite on $\mathbb{R}^n \setminus \{0\}$, there exist an integer $\nu \geq 0$ and homogeneous SOS matrices $S_k(x)$ of size $|\mathcal{C}_k| \times |\mathcal{C}_k|$ such that

$$\|x\|^{2\nu} P(x) = \sum_{k=1}^{t} E_{\mathcal{C}_k}^{\mathsf{T}} S_k(x) E_{\mathcal{C}_k}.$$

- **Corollary**: If P is strictly positive definite on $\mathbb{R}^n$ and its highest-degree homogeneous part $\sum_{|\alpha|=2d} P_\alpha x^\alpha$ is strictly positive definite on $\mathbb{R}^n \setminus \{0\}$, there exist an integer $\nu \geq 0$ and SOS matrices $S_k(x)$ of size $|\mathcal{C}_k| \times |\mathcal{C}_k|$ such that

$$(1 + \|x\|^2)^{\nu} P(x) = \sum_{k=1}^{t} E_{\mathcal{C}_k}^{\mathsf{T}} S_k(x) E_{\mathcal{C}_k}.$$

Reznick's Positivstellensatz

- **Non-trivial example:** Let $q(x) = x_1^2x_2^4 + x_1^4x_2^2 - 3x_1^2x_2^2 + 1$ be the Motzkin polynomial, and

$$P(x) = \begin{bmatrix} 0.01(1 + x_1^6 + x_2^6) + q(x) & -0.01x_1 & 0 \\ -0.01x_1 & x_1^6 + x_2^6 + 1 & -x_2 \\ 0 & -x_2 & x_1^6 + x_2^6 + 1 \end{bmatrix}.$$

- $P(x)$ is is strictly positive definite on $\mathbb{R}^2$, but is not SOS (since $\varepsilon(1 + x_1^6 + x_2^6) + q(x)$ is not SOS unless $\varepsilon \gtrsim 0.01006$ [Laurent 2009, Example 6.25]).
- Our theorem guarantees the following decomposition exists

$$(1 + \|x\|^2)^\nu P(x) = E_{\mathcal{C}_1}^\mathsf{T} S_1(x) E_{\mathcal{C}_1} + E_{\mathcal{C}_2}^\mathsf{T} S_2(x) E_{\mathcal{C}_2}.$$

- It suffices to use $\nu = 1$ and SOS matrices

$$S_1(x) = \begin{bmatrix} (1 + \|x\|^2)q(x) & 0 \\ 0 & 0 \end{bmatrix} + \frac{1 + \|x\|^2}{100} \begin{bmatrix} 1 + x_1^6 + x_2^6 & -x_1 \\ -x_1 & 100x_1^2 \end{bmatrix},$$

$$S_2(x) = (1 + \|x\|^2) \begin{bmatrix} 1 - x_1^2 + x_1^6 + x_2^6 & -x_2 \\ -x_2 & 1 + x_1^6 + x_2^6 \end{bmatrix}.$$

Putinar's Positivstellensatz

Consider $P(x) \succ 0, \forall x \in \mathcal{K}$ with $\mathcal{K} = \{x \in \mathbb{R}^n \mid g_i(x) \geq 0, i = 1, \ldots, m\}$, and

$$\sigma_0(x) + g_1(x)\sigma_1(x) + \cdots + g_q(x)\sigma_q(x) = r^2 - \|x\|^2.$$

Sparse matrix version of Putinar's Positivstellensatz

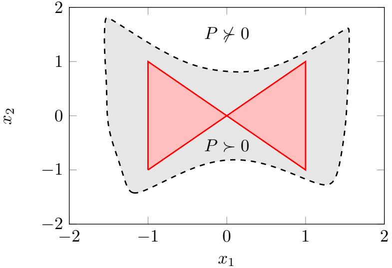
let $P(x)$ be a polynomial matrix whose sparsity graph is chordal and has maximal cliques $\mathcal{C}_1, \ldots, \mathcal{C}_t$. If P is strictly positive definite on $\mathcal{K}$ (satisfying the Archimedean condition), there exist SOS matrices $S_{j,k}(x)$ of size $|\mathcal{C}_k| \times |\mathcal{C}_k|$ such that

$$P(x) = \sum_{k=1}^{t} E_{\mathcal{C}_k}^{\mathsf{T}} \left(S_{0,k}(x) + \sum_{j=1}^{q} g_j(x) S_{j,k}(x) \right) E_{\mathcal{C}_k}.$$

- **Example:** Consider $\mathcal{K} = \{x \in \mathbb{R}^2 : \ g_1(x) := 1 - x_1^2 \geq 0, \ g_2(x) := x_1^2 - x_2^2 \geq 0\}$, and

$$P(x) := \begin{bmatrix} 1 + 2x_1^2 - x_1^4 & x_1 + x_1x_2 - x_1^3 & 0 \\ x_1 + x_1x_2 - x_1^3 & 3 + 4x_1^2 - 3x_2^2 & 2x_1^2x_2 - x_1x_2 - 2x_2^3 \\ 0 & 2x_1^2x_2 - x_1x_2 - 2x_2^3 & 1 + x_2^2 + x_1^2x_2^2 - x_2^4 \end{bmatrix}$$

Putinar's Positivstellensatz



- It guarantees the following decomposition holds for some SOS matrices $S_{i,j}(x)$

$$P(x) = \sum_{k=1}^{2} E_{\mathcal{C}_k}^{\mathsf{T}} \left[S_{0,k}(x) + g_1(x) S_{1,k}(x) + g_2(x) S_{2,k}(x) \right] E_{\mathcal{C}_k}$$

- Possible choices are

$$S_{0,1}(x) = I_2 + \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{bmatrix} x_1 & x_2 \end{bmatrix} \qquad S_{1,1}(x) = \begin{bmatrix} x_1 \\ 1 \end{bmatrix} \begin{bmatrix} x_1 & 1 \end{bmatrix}$$

$$S_{0,2}(x) = I_2 + \begin{bmatrix} x_1 \\ -x_2 \end{bmatrix} \begin{bmatrix} x_1 & -x_2 \end{bmatrix} \qquad S_{2,2}(x) = \begin{bmatrix} 2 \\ x_2 \end{bmatrix} \begin{bmatrix} 2 & x_2 \end{bmatrix}.$$

Application to robust semidefinite optimization

Consider a robust SDP program

$$B^* := \inf_{\lambda \in \mathbb{R}^\ell} \quad b^\mathsf{T} \lambda$$

$$\text{subject to} \quad P(x,\lambda) := P_0(x) - \sum_{i=1}^{\ell} P_i(x)\lambda_i \succeq 0 \quad \forall x \in \mathcal{K},$$

$$B^*_{d,\nu} := \inf_{\lambda,\, S_{j,k}} \quad b^\mathsf{T} \lambda$$

$$\text{subject to} \quad \sigma(x)^\nu P(x,\lambda) = \sum_{k=1}^{t} E_{\mathcal{C}_k}^\mathsf{T} \left(S_{0,k}(x) + \sum_{j=1}^{m} g_j(x) S_{j,k}(x) \right) E_{\mathcal{C}_k},$$

$$S_{j,k} \in \Sigma_{2d_j}^{|\mathcal{C}_k|} \quad \forall j = 0,\ldots,q,\ \forall k = 1,\ldots,t,$$

Convergence guarantees

- $\mathcal{K}$ is compact and satisfies the Archimedean condition, under some technical conditions, we fix $\sigma(x) = 1$ and $B^*_{d,0} \to B^*$ from above as $d \to \infty$.
- $\mathcal{K} \equiv \mathbb{R}^n$: under some technical conditions, we fix $\sigma(x) = 1 + \|x\|^2$ and $B^*_{d,\nu} \to B^*$ from above as $\nu \to \infty$ and $d = \nu + \lceil \frac{1}{2} \max\{\deg(P), \deg(g_1), \ldots, \deg(g_q)\} \rceil$.

Proof ideas: Hilbert–Artin theorem

Diagonalization with no fill-ins

If $P(x)$ is an $m \times m$ symmetric polynomial matrix with chordal sparsity graph, there exist an $m \times m$ permutation matrix T, an invertible $m \times m$ lower-triangular polynomial matrix $L(x)$, and polynomials $b(x)$, $d_1(x)$, $\ldots$, $d_m(x)$ such that

$$b^4(x)\, TP(x)T^{\mathsf{T}} = L(x)\text{Diag}\left(d_1(x), \ldots, d_m(x)\right) L(x)^{\mathsf{T}}.$$

Moreover, L has no fill-in in the sense that $L + L^{\mathsf{T}}$ has the same sparsity as TPT^{T}.

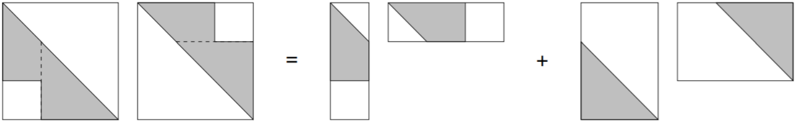


Figure: Decomposition follows by combining columns.

Figure from Prof. Lieven Vandenberghe's talk.

Proof ideas: Putinar's theorem

Scherer and Ho, 2006

Let $\mathcal{K}$ be a compact semialgebraic set that satisfies the Archimedean condition. If an $m \times m$ symmetric polynomial matrix $P(x)$ is strictly positive definite on $\mathcal{K}$, there exist $m \times m$ SOS matrices $S_0, \ldots, S_q$ such that

$$P(x) = S_0(x) + \sum_{i=1}^{q} S_i(x) g_i(x).$$

- Weierstrass polynomial approximation theorem + the above version of Putinar's Positivstellensatz

$$\begin{aligned} P(x) &= \begin{bmatrix} a(x) & b(x)^{\mathsf{T}} & 0 \\ b(x) & U(x) & V(x) \\ 0 & V(x) & W(x) \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} a(x) & b(x)^{\mathsf{T}} & 0 \\ b(x) & H(x) + 2\varepsilon I & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\succeq 0, \forall x \in \mathcal{K}} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & U(x) - H(x) - 2\varepsilon I & V(x) \\ 0 & V(x)^{\mathsf{T}} & W(x) \end{bmatrix}}_{\succeq 0, \forall x \in \mathcal{K}}. \end{aligned}$$

Experiment 1: global PMI

Define a set

$$\mathcal{F}_\omega = \{\lambda \in \mathbb{R}^2 : \; P_\omega(x,\lambda) \succeq 0 \quad \forall x \in \mathbb{R}^3\}.$$

$$P_\omega(x,\lambda) = \begin{bmatrix} \lambda_2 x_1^4 + x_2^4 & \lambda_1 x_1^2 x_2^2 & & & & & & \\ \lambda_1 x_1^2 x_2^2 & \lambda_2 x_2^4 + x_3^4 & \lambda_2 x_2^2 x_3^2 & & & & & \\ & \lambda_2 x_2^2 x_3^2 & \lambda_2 x_3^4 + x_1^4 & \lambda_1 x_1^2 x_3^2 & & & & \\ & & \lambda_1 x_1^2 x_3^2 & \lambda_2 x_1^4 + x_2^4 & \lambda_2 x_1^2 x_2^2 & & & \\ & & & \lambda_2 x_1^2 x_2^2 & \lambda_2 x_2^4 + x_3^4 & \ddots & & \\ & & & & \ddots & \ddots & \lambda_i x_2^2 x_3^2 & \\ & & & & & \lambda_i x_2^2 x_3^2 & \lambda_2 x_3^4 + x_1^4 & \end{bmatrix},$$

- Define two hierarchies of subsets of $\mathcal{F}_\omega$, indexed by a nonnegative integer ν, as

$$\mathcal{D}_{\omega,\nu} := \left\{\lambda \in \mathbb{R}^2 : \; \|x\|^{2\nu} P_\omega(x,\lambda) \text{ is SOS}\right\},$$

$$\mathcal{S}_{\omega,\nu} := \left\{\lambda \in \mathbb{R}^2 : \; \|x\|^{2\nu} P_\omega(x,\lambda) = \sum_{k=1}^{3\omega-1} E_{\mathcal{C}_k}^{\mathsf{T}} S_k(x) E_{\mathcal{C}_k}, S_k(x) \text{ is SOS}\right\}.$$

- We always have

$$\mathcal{S}_{\omega,\nu} \subseteq \mathcal{D}_{\omega,\nu} \subseteq \mathcal{F}_\omega$$

Experiment 1: global PMI

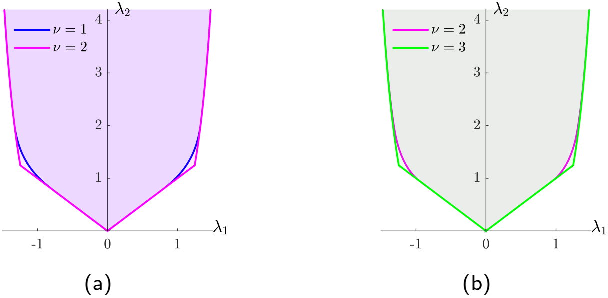


Figure: Inner approximations of the set $\mathcal{F}_2$ obtained with SOS optimization. (a) Sets $\mathcal{D}_{2,\nu}$ obtained using the standard SOS constraint; (b) Sets $\mathcal{S}_{2,\nu}$ obtained using the sparse SOS constraint. The numerical results suggest $\mathcal{S}_{2,3} = \mathcal{D}_{2,2} = \mathcal{F}_2$.

Experiment 1: global PMI

We consider

$$B^* := \inf_{\lambda} \quad \lambda_2 - 10\lambda_1$$
$$\text{subject to} \quad \lambda \in \mathcal{F}_\omega$$

Table: Upper bounds $B_{d,\nu}$ on the optimal value B^* and CPU time (seconds) by MOSEK

	Standard SOS						Sparse SOS					
	$\nu = 1$		$\nu = 2$		$\nu = 3$		$\nu = 2$		$\nu = 3$		$\nu = 4$	
ω	t	$B_{d,\nu}$	t	$B_{d,\nu}$	t	$B_{d,\nu}$	t	$B_{d,\nu}$	t	$B_{d,\nu}$	t	$B_{d,\nu}$
5	12	-8.68	25	-9.36	69	-9.36	0.58	-8.97	0.72	-9.36	1.29	-9.36
10	407	-8.33	886	-9.09	2910	-9.09	1.65	-8.72	0.82	-9.09	2.08	-9.09
15	2090	-8.26	OOM	OOM	OOM	OOM	2.76	-8.68	1.13	-9.04	2.79	-9.04
20	OOM	OOM	OOM	OOM	OOM	OOM	3.24	-8.66	1.54	-9.02	4.70	-9.02
25	OOM	OOM	OOM	OOM	OOM	OOM	2.85	-8.66	1.94	-9.02	4.59	-9.02
30	OOM	OOM	OOM	OOM	OOM	OOM	2.38	-8.65	2.40	-9.01	5.50	-9.01
35	OOM	OOM	OOM	OOM	OOM	OOM	2.66	-8.65	3.25	-9.01	6.17	-9.01
40	OOM	OOM	OOM	OOM	OOM	OOM	3.07	-8.65	3.14	-9.01	8.48	-9.01

Experiment 2: PMI locally

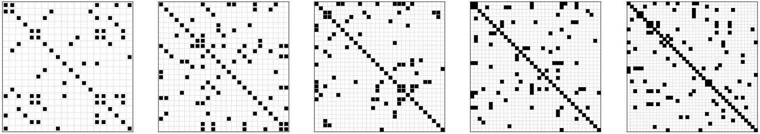
$$B^*_{m,d} := \max_{s_{2d}(x)} \int_{\mathcal{K}} s_{2d}(x)\,\mathrm{d}x$$

$$\text{subject to} \quad P(x) - s_{2d}(x) I \succeq 0 \quad \forall x \in \mathcal{K}.$$

- Set approximation: $\mathcal{P} = \{x \in \mathbb{R}^n \mid P(x) \succeq 0\} \subset \mathcal{K}$
- the unit disk: $\mathcal{K} = \{x \in \mathbb{R}^2 : 1 - x_1^2 - x_2^2 \geq 0\}$ and

$$P(x) = (1 - x_1^2 - x_2^2) I_m + (x_1 + x_1 x_2 - x_1^3) A + (2x_1^2 x_2 - x_1 x_2 - 2x_2^3) B,$$

A, B with chordal sparsity graphs, zero diagonal elements, and other entries from the uniform distribution on $(0, 1)$.



(a) $m = 20$ (b) $m = 25$ (c) $m = 30$ (d) $m = 35$ (e) $m = 40$

Figure: Chordal sparsity patterns for the polynomial matrix $P(x)$.

Experiment 2: PMI locally

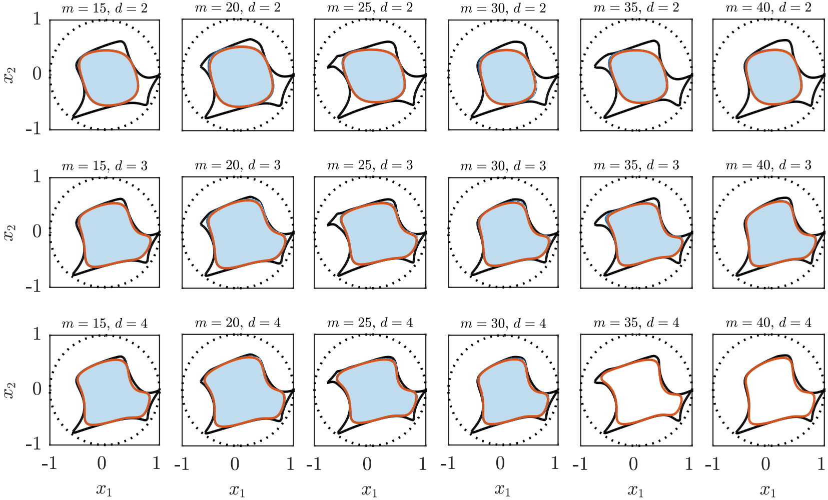


Figure: The real boundary of $\mathcal{P}$: a solid black line. Standard SOS: blue solid boundary and blue shading; the sparsity-exploiting SOS: red solid boundary, no shading.

Experiment 2: PMI locally

Table: Lower bounds and CPU time (seconds, by Mosek) using the standard SOS and the sparsity-exploiting SOS. The asymptotic value $B^*_{m,\infty}$ was found by integrating the minimum eigenvalue function of P over the unit disk $\mathcal{K}$.

	Standard SOS						Sparse SOS						
	$d=2$		$d=3$		$d=4$		$d=2$		$d=3$		$d=4$		
m	t	$B^{\text{sos}}_{m,d}$	t	$B^{\text{sos}}_{m,d}$	t	$B^{\text{sos}}_{m,d}$	t	$B^{\text{sos}}_{m,d}$	t	$B^{\text{sos}}_{m,d}$	t	$B^{\text{sos}}_{m,d}$	$B^*_{m,\infty}$
15	3.7	-2.07	24.8	-1.50	95.1	-1.36	0.95	-2.10	0.97	-1.52	1.94	-1.37	-1.15
20	13.3	-1.51	96.5	-1.03	375	-0.92	0.69	-1.58	1.06	-1.07	2.12	-0.95	-0.75
25	38.1	-2.47	326	-1.85	1308	-1.64	0.95	-2.50	1.28	-1.87	3.04	-1.66	-1.41
30	136	-2.13	963	-1.54	4031	-1.41	0.75	-2.21	1.35	-1.58	3.14	-1.43	-1.21
35	219	-2.46	2210	-1.82	OOM	OOM	0.77	-2.51	1.51	-1.84	3.01	-1.65	-1.40
40	550	-2.22	5465	-1.59	OOM	OOM	1.03	-2.24	2.07	-1.59	5.62	-1.47	-1.25

Experiment 2: PMI locally

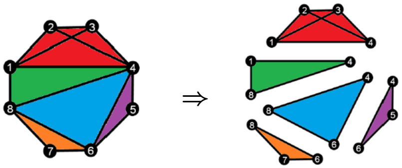
Table: Lower bounds $B^{\text{sos}}_{15,d}$ on the asymptotic value $B^*_{15,\infty} = -1.153$ for $m = 15$, calculated using the sparsity-exploiting SOS with $\nu = 0$ and the standard SOS. The CPU time (t, seconds) to compute these bounds using MOSEK is also reported.

	d	6	8	10	12	14
Sparse SOS	$B^{\text{sos}}_{15,d}$	-1.257	-1.219	-1.199	-1.195	-1.191
	t	13.3	85.1	309.3	818.3	2149
Standard SOS	$B^{\text{sos}}_{15,d}$	-1.252	-1.216	OOM	OOM	OOM
	t	1133	8250	OOM	OOM	OOM

Conclusion

Take-home message

- **Message 1: Chordal decomposition:** leading to sparse PSD cone decompositions



- **Message 2: Sparse SDPs can be solved 'fast'**

$$\begin{aligned} \min_{x, x_k} \quad & \langle c, x \rangle \\ \text{s.t.} \quad & Ax = b, \\ & \boxed{x_k = H_k x}, \qquad k = 1, \ldots, p, \\ & x_k \in \mathcal{S}_k, \qquad k = 1, \ldots, p, \end{aligned}$$

$$\sigma(x)P(x) = \sum_{k=1}^{t} E_{\mathcal{C}_k}^{\mathsf{T}} S_k(x) E_{\mathcal{C}_k}.$$

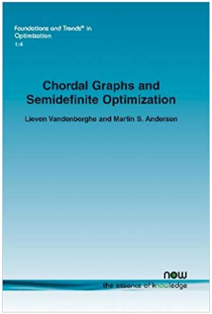
CDCS: an open-source first-order conic solver;

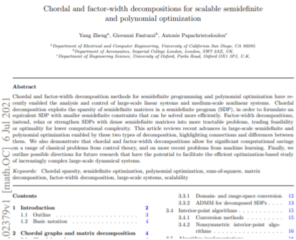
Download from `https://github.com/OxfordControl/CDCS`

- **Message 3: Sparse robust SDPs can be solved 'fast':** the Hilbert-Artin, Reznick, Putinar, and Putinar-Vasilescu Positivstellensätze.

Future work

- Decomposition and completion of polynomial matrices
- Moment interpretation of the PSD polynomial decomposition results
- Combining matrix decomposition with other structures
- Blending application-driven modeling with optimization
- Efficient software for modern computers





Chordal and factor-width decompositions for scalable semidefinite and polynomial optimization

Yang Zheng[a], Giovanni Fantuzzi[b], Antonis Papachristodoulou[c]

[a]Department of Electrical and Computer Engineering, University of California San Diego, CA 92093.
[b]Department of Aeronautics, Imperial College London, London, SW7 2AZ, UK.
[c]Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, U.K.

.02379v1 [math.OC] 6 Jul 2021

Abstract

Chordal and factor-width decomposition methods for semidefinite programming and polynomial optimization have recently enabled the analysis and control of large-scale linear systems and medium-scale nonlinear systems. Chordal decomposition exploits the sparsity of semidefinite matrices in a semidefinite program (SDP), in order to formulate an equivalent SDP with smaller semidefinite constraints that can be solved more efficiently. Factor-width decompositions, instead, relax or strengthen SDPs with dense semidefinite matrices into more tractable problems, trading feasibility or optimality for lower computational complexity. This article reviews recent advances in large-scale semidefinite and polynomial optimization enabled by these two types of decomposition, highlighting connections and differences between them. We also demonstrate that chordal and factor-width decompositions allow for significant computational savings on a range of classical problems from control theory, and on more recent problems from machine learning. Finally, we outline possible directions for future research that have the potential to facilitate the efficient optimization-based study of increasingly complex large-scale dynamical systems.

Keywords: Chordal sparsity, semidefinite optimization, polynomial optimization, sum-of-squares, matrix decomposition, factor-width decomposition, large-scale systems, scalability

Contents

1 **Introduction** 2
1.1 Outline 3
1.2 Basic notation 4
2 **Chordal graphs and matrix decomposition** 4
3.3.1 Domain- and range-space conversion 12
3.3.2 ADMM for decomposed SDPs 13
3.4 Interior-point algorithms 15
3.4.1 Conversion methods 15
3.4.2 Nonsymmetric interior-point algorithms 16

Thank you for your attention!

Q & A

- Zheng, Y., Fantuzzi, G., Papachristodoulou, A., Goulart, P., & Wynn, A. (2020). Chordal decomposition in operator-splitting methods for sparse semidefinite programs. Mathematical Programming, 1-44.
- Zheng, Y., & Fantuzzi, G. (2020). Sum-of-squares chordal decomposition of polynomial matrix inequalities. arXiv preprint arXiv:2007.11410. (Mathematical Programming, accepted)
- Zheng, Y., Fantuzzi, G., & Papachristodoulou, A. (2018, December). Decomposition and completion of sum-of-squares matrices. In 2018 IEEE Conference on Decision and Control (CDC) (pp. 4026-4031). IEEE.
- Zheng, Y., Fantuzzi, G., & Papachristodoulou, A. (2019, July). Sparse sum-of-squares (SOS) optimization: A bridge between DSOS/SDSOS and SOS optimization for sparse polynomials. In 2019 American Control Conference (ACC) (pp. 5513-5518). IEEE.

Extra slides

Alternating Direction Method of Multipliers (ADMM)

The ADMM algorithm solves the optimization problem (Bertsekas and Tsitsiklis, 1989; Boyd, *et al.*, 2011)

$$\begin{aligned} \min_{x,y} \quad & f(x) + g(y) \\ \text{subject to} \quad & Ax + By = c, \end{aligned}$$

where f and g are convex functions.

- **Augmented Lagrangian**

$$\mathcal{L}_\rho(x, y, z) := f(x) + g(y) + z^\mathsf{T}(Ax + By - c) + \frac{\rho}{2}\|Ax + By - c\|^2$$

- **ADMM steps**

$$\begin{aligned} x^{(n+1)} &= \arg\min_x \; \mathcal{L}_\rho(x, y^{(n)}, z^{(n)}), && \rightarrow x\text{-minimization step} \\ y^{(n+1)} &= \arg\min_y \; \mathcal{L}_\rho(x^{(n+1)}, y, z^{(n)}), && \rightarrow y\text{-minimization step} \\ z^{(n+1)} &= z^{(n)} + \rho\,(Ax^{(n+1)} + By^{(n+1)} - c). && \rightarrow \text{dual variable update} \end{aligned}$$

ADMM is particularly suitable when the subproblems have closed-form expressions, or can be solved efficiently.

ADMM for primal decomposed SDPs

$$\begin{aligned} \min_{x, x_k} \quad & \langle c, x \rangle \\ \text{s.t.} \quad & Ax = b, \\ & \boxed{x_k = H_k x}, \quad k = 1, \ldots, p, \\ & x_k \in \mathcal{S}_k, \quad k = 1, \ldots, p, \end{aligned}$$

Reformulation using indicator functions

$$\begin{aligned} \min_{x, x_1, \ldots, x_p} \quad & \langle c, x \rangle + \delta_0 (Ax - b) + \sum_{k=1}^{p} \delta_{\mathcal{S}_k}(x_k) \\ \text{s.t.} \quad & x_k = H_k x, \quad k = 1, \ldots, p. \end{aligned}$$

- *x-minimization step:* QP with linear constraints, KKT condition

$$\begin{bmatrix} D & A^{\mathsf{T}} \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{p} H_k^{\mathsf{T}} \left(x_k^{(n)} + \rho^{-1} \lambda_k^{(n)} \right) - \rho^{-1} c \\ b \end{bmatrix}.$$

- *y-minimization step:* Parallel projections onto **small PSD cones**

$$\begin{aligned} \min_{x_k} \quad & \left\| x_k - H_k x^{(n+1)} + \rho^{-1} \lambda_k^{(n)} \right\|^2 \\ \text{s.t.} \quad & x_k \in \mathcal{S}_k. \end{aligned}$$

- Update multipliers

$$
\begin{aligned}
\max_{y, z_k, v_k} \quad & \langle b, y \rangle \\
\text{s.t.} \quad & A^{\mathsf{T}} y + \sum_{k=1}^{p} H_k^{\mathsf{T}} v_k = c, \\
& \boxed{z_k - v_k = 0}, k = 1, \ldots, p, \\
& z_k \in \mathcal{S}_k, k = 1, \ldots, p.
\end{aligned}
$$

Reformulation using indicator functions

$$
\begin{aligned}
\min \quad & -\langle b, y \rangle + \delta_0 \left(c - A^{\mathsf{T}} y - \sum_{k=1}^{p} H_k^{\mathsf{T}} v_k \right) + \sum_{k=1}^{p} \delta_{\mathcal{S}_k}(z_k) \\
\text{s.t.} \quad & z_k = v_k, \quad k = 1, \ldots, p.
\end{aligned}
$$

- *x-minimization step:* QP with linear constraints, KKT condition

$$
\begin{bmatrix} D & A^{\mathsf{T}} \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} c - \sum_{k=1}^{p} H_k^{\mathsf{T}} \left(z_k^{(n)} + \rho^{-1} \lambda_k^{(n)} \right) \\ -\rho^{-1} b \end{bmatrix},
$$

- *y-minimization step:* Parallel projections onto **small PSD cones**

$$
\begin{aligned}
\min_{z_k} \quad & \left\| z_k - v_k^{(n)} + \rho^{-1} \lambda_k^{(n)} \right\|^2 \\
\text{s.t.} \quad & z_k \in \mathcal{S}_k.
\end{aligned}
$$

- Update multipliers

Homogeneous self-dual embedding of decomposed SDPs

$$\begin{aligned} \min_{x, x_k} \quad & \langle c, x \rangle \\ \text{s.t.} \quad & Ax = b, \\ & x_k = H_k x, \quad k = 1, \ldots, p, \\ & x_k \in \mathcal{S}_k, \quad k = 1, \ldots, p, \end{aligned}$$

$$\begin{aligned} \max_{y, z_k, v_k} \quad & \langle b, y \rangle \\ \text{s.t.} \quad & A^{\mathsf{T}} y + \sum_{k=1}^{p} H_k^{\mathsf{T}} v_k = c, \\ & z_k - v_k = 0, k = 1, \ldots, p, \\ & z_k \in \mathcal{S}_k, k = 1, \ldots, p. \end{aligned}$$

Notional simplicity:

$$s := \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix}, \quad z := \begin{bmatrix} z_1 \\ \vdots \\ z_p \end{bmatrix}, \quad t := \begin{bmatrix} v_1 \\ \vdots \\ v_p \end{bmatrix}, \quad H := \begin{bmatrix} H_1 \\ \vdots \\ H_p \end{bmatrix}, \quad \mathcal{S} := \mathcal{S}_1 \times \cdots \times \mathcal{S}_p$$

KKT conditions

- Primal feasibility

$$Ax^* - r^* = b, \quad s^* + w^* = Hx^*, \quad s^* \in \mathcal{S}, \quad r^* = 0, \quad w^* = 0.$$

- Dual feasibility

$$A^{\mathsf{T}} y^* + H^{\mathsf{T}} t^* + h^* = c, \quad z^* - t^* = 0, \quad z^* \in \mathcal{S}, \quad h^* = 0.$$

- Zero duality gap:

$$c^{\mathsf{T}} x^* - b^{\mathsf{T}} y^* = 0.$$

Homogeneous self-dual embedding of decomposed SDPs

The homogeneous self-dual embedding (HSDE) form (Ye, Todd, Mizuno, 1994)

$$\begin{aligned} \text{find} \quad & (u, v) \\ \text{subject to} \quad & v = Qu, \\ & (u, v) \in \mathcal{K} \times \mathcal{K}^*, \end{aligned}$$

where $\mathcal{K} := \mathbb{R}^{n^2} \times \mathcal{S} \times \mathbb{R}^m \times \mathbb{R}^{n_d} \times \mathbb{R}_+$ is a cone ($\mathcal{S} := \mathcal{S}_1 \times \cdots \times \mathcal{S}_p$) and

$$u := \begin{bmatrix} x \\ s \\ y \\ t \\ \tau \end{bmatrix}, \quad v := \begin{bmatrix} h \\ z \\ r \\ w \\ \kappa \end{bmatrix}, \quad Q := \begin{bmatrix} 0 & 0 & -A^{\mathsf T} & -H^{\mathsf T} & c \\ 0 & 0 & 0 & I & 0 \\ A & 0 & 0 & 0 & -b \\ H & -I & 0 & 0 & 0 \\ -c^{\mathsf T} & 0 & b^{\mathsf T} & 0 & 0 \end{bmatrix}.$$

ADMM steps (similar to the solver SCS, O'Donoghue *et al.*, 2016)

$$\hat{u}^{(n+1)} = (I + Q)^{-1}\left(u^{(n)} + v^{(n)}\right), \quad \longrightarrow \text{Projection onto a linear subspace}$$

$$u^{(n+1)} = \mathbb{P}_{\mathcal{K}}\left(\hat{u}^{(n+1)} - v^{(n)}\right), \quad \longrightarrow \text{Projection onto } \textbf{small PSD cones}$$

$$v^{(n+1)} = v^{(n)} - \hat{u}^{(n+1)} + u^{(n+1)}, \quad \longrightarrow \text{Computationally trivial update}$$

The conic projections in all Algorithms require $\mathcal{O}(\sum_{k=1}^{p} |\mathcal{C}_k|^3)$ flops.

Scaled-diagonally dominant SOS (SDSOS) and DSOS

A new concept of (S)DSOS by Ahmadi and Majumdar, 2017

- *Diagonally dominant (dd) matrix:* a symmetric matrix $A = [a_{ij}]$ is dd if

$$a_{ii} \geq \sum_{j \neq i} |a_{ij}|, \forall i = 1, \ldots, n.$$

- *Scaled-diagonally dominant (sdd) matrix:* a symmetric matrix $A = [a_{ij}]$ is sdd if there exists a PSD diagonal matrix D, such that

$$DAD \text{ is dd.}$$

- *DSOS polynomials:* $p(x) = v_d(x)^\mathsf{T} Q v_d(x)$, where the Gram matrix Q is dd.
- *SDSOS polynomials:* $p(x) = v_d(x)^\mathsf{T} Q v_d(x)$, where the Gram matrix Q is sdd.

LP and SOCP-based optimization (Ahmadi and Majumdar, 2017)

- Optimization over dd matrices or DSOS polynomials is a linear program (LP).
- Optimization over sdd matrices or SDSOS polynomials is a second-order cone program (SOCP).

The gap between DSOS/SDSOS and SOS

A brief summary

- **SOS:** $p(x) = v_d(x)^{\mathsf{T}} Q v_d(x)$, where the Gram matrix Q is PSD $\longrightarrow$ SDP
- **SDSOS:** $p(x) = v_d(x)^{\mathsf{T}} Q v_d(x)$, where the Gram matrix Q is sdd $\longrightarrow$ SOCP
- **DSOS:** $p(x) = v_d(x)^{\mathsf{T}} Q v_d(x)$, where the Gram matrix Q is dd $\longrightarrow$ LP

Another viewpoint

- **SDP** is an optimization problem involving PSD constraints of dimension $N \times N$
- **SOCP** is an optimization problem involving PSD constraints of dimension 2×2
- **LP** is an optimization problem involving PSD constraints of dimension 1×1

What is missing? How about problems that involve PSD constraints of dimension $k \times k$, where $1 \leq k \leq N$

- One approach: factor-width k matrices (Boman, et al. 2005) $\longrightarrow$ Not practical $\binom{n}{k} = \mathcal{O}(n^k)$
- **Chordal decomposition**, considering sparsity and equivalent to sparse factor-width k matrices $\longrightarrow$ the main topic today.

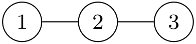
Sparsity in SOS optimization

Sparse polynomial matrix (similar to sparse real matrix)

- Given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, we define a sparse polynomial matrix $P(x)$ where

$$p_{ij}(x) = 0, \text{ if } (i,j) \notin \mathcal{E}^*$$

- For example, for a line graph of three nodes



$$P(x) = \begin{bmatrix} p_{11}(x) & p_{12}(x) & \\ p_{21}(x) & p_{22}(x) & p_{23}(x) \\ & p_{32}(x) & p_{33}(x) \end{bmatrix}.$$

- Define a set of sparse polynomial matrices

$$\mathbb{R}^{r\times r}_{n,2d}(\mathcal{E},0) = \left\{ P(x) \in \mathbb{R}[x]^{r\times r}_{n,2d} \,\middle|\, p_{ij}(x) = p_{ji}(x) = 0, \text{ if } (i,j) \notin \mathcal{E}^* \right\}.$$

- SOS/SDSOS/DSOS matrices with a sparsity pattern $\mathcal{E}$

$$SOS^r_{n,2d}(\mathcal{E},0) = SOS^r_{n,2d} \cap \mathbb{R}^{r\times r}_{n,2d}(\mathcal{E},0),$$
$$SDSOS^r_{n,2d}(\mathcal{E},0) = SDSOS^r_{n,2d} \cap \mathbb{R}^{r\times r}_{n,2d}(\mathcal{E},0),$$
$$DSOS^r_{n,2d}(\mathcal{E},0) = DSOS^r_{n,2d} \cap \mathbb{R}^{r\times r}_{n,2d}(\mathcal{E},0).$$

Sparsity in SOS optimization

Sparsity in $P(x)$ does not necessarily lead to sparsity in the Gram matrix Q !!

For example

$$P(x) = \begin{bmatrix} p_{11}(x) & p_{12}(x) & \\ p_{21}(x) & p_{22}(x) & p_{23}(x) \\ & p_{32}(x) & p_{33}(x) \end{bmatrix} = \begin{bmatrix} v(x)^\mathsf{T} Q_{11} v(x) & v(x)^\mathsf{T} Q_{12} v(x) & v(x)^\mathsf{T} Q_{13} v(x) \\ v(x)^\mathsf{T} Q_{21} v(x) & v(x)^\mathsf{T} Q_{22} v(x) & v(x)^\mathsf{T} Q_{23} v(x) \\ v(x)^\mathsf{T} Q_{31} v(x) & v(x)^\mathsf{T} Q_{32} v(x) & v(x)^\mathsf{T} Q_{33} v(x) \end{bmatrix}$$

$$= (I_3 \otimes v(x))^\mathsf{T} \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} (I_3 \otimes v(x))$$

- If we make **a restriction that** $Q_{ij} = 0$**, if** $p_{ij}(x) = 0$, then the Gram matrix Q has the same pattern with $P(x)$. Now, **chordal decomposition** leads to

$$Q = \underbrace{\begin{bmatrix} * & * & 0 \\ * & * & * \\ 0 & * & * \end{bmatrix}}_{\succeq 0} = \underbrace{\begin{bmatrix} * & * & 0 \\ * & * & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\succeq 0} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & * & * \\ 0 & * & * \end{bmatrix}}_{\succeq 0}$$

- We have the same chordal decomposition for polynomial matrix $P(x)$.

Sparse SOS matrix decomposition

Sparse version of SOS matrices

$$SSOS^r_{n,2d}(\mathcal{E},0) = \left\{ P(x) \in SOS^r_{n,2d}(\mathcal{E},0) \middle| P(x) \text{ admits a Gram matrix } Q \succeq 0, \text{ with } Q_{ij} = 0 \text{ when } p_{ij}(x) = 0 \right\}.$$

Theorem (Sparse SOS matrix decomposition)

If $\mathcal{E}$ is chordal with a set of maximal cliques $\mathcal{C}_1, \ldots, \mathcal{C}_t$, then

$$P(x) \in SSOS^r_{n,2d}(\mathcal{E},0) \Leftrightarrow P(x) = \sum_{k=1}^{\mathsf{T}} E_k^{\mathsf{T}} P_k(x) E_k,$$

where $P_k(x)$ is an SOS matrix of dimension $|\mathcal{C}_k| \times |\mathcal{C}_k|$.

Proof: apply the **Agler's theorem** to the sparse block matrix Q.

$$P(x) = (I_r \otimes v_d(x))^{\mathsf{T}} Q \, (I_r \otimes v_d(x)) = (I_r \otimes v_d(x))^{\mathsf{T}} \left(\sum_{k=1}^{\mathsf{T}} E_{\widetilde{\mathcal{C}}_k}^{\mathsf{T}} Q_k E_{\widetilde{\mathcal{C}}_k} \right) (I_r \otimes v_d(x))$$

$$= \sum_{k=1}^{\mathsf{T}} \left[(I_r \otimes v_d(x))^{\mathsf{T}} E_{\widetilde{\mathcal{C}}_k}^{\mathsf{T}} Q_k E_{\widetilde{\mathcal{C}}_k} (I_r \otimes v_d(x)) \right] = \sum_{k=1}^{\mathsf{T}} E_{\mathcal{C}_k}^{\mathsf{T}} P_k(x) E_{\mathcal{C}_k},$$

We have the following inclusion relationship

$$DSOS^r_{n,2d}(\mathcal{E},0) \subseteq SDSOS^r_{n,2d}(\mathcal{E},0) \subseteq SSOS^r_{n,2d}(\mathcal{E},0) \subseteq SOS^r_{n,2d}(\mathcal{E},0) \subseteq \mathcal{P}^r_{n,2d}(\mathcal{E},0)$$

Key idea: if a matrix Q is (scaled) diagonally dominant, then it is still (scaled) diagonally dominant when replacing any off-diagonal elements with zeros.

- A brief summary (scalability):

$$\begin{aligned}
\mathcal{P}^r_{n,2d}(\mathcal{E},0) &\longrightarrow \text{NP-hard} \\
DSOS^r_{n,2d}(\mathcal{E},0) &\longrightarrow \text{LP (PSD cones: } 1\times 1) \\
SDSOS^r_{n,2d}(\mathcal{E},0) &\longrightarrow \text{SOCP (PSD cones: } 2\times 2) \\
SSOS^r_{n,2d}(\mathcal{E},0) &\longrightarrow \textbf{SDP with smaller PSD cones of } k\times k \\
SOS^r_{n,2d}(\mathcal{E},0) &\longrightarrow \text{SDP with a PSD cone of } N\times N
\end{aligned}$$

Solution quality: $\mathcal{P}_{\text{dsos}}, \mathcal{P}_{\text{sdsos}}$ and $\mathcal{P}_{\text{ssos}}$ are a sequence of inner approximations with increasing accuracy to the SOS problem $\mathcal{P}_{\text{sos}}$, meaning that

$$f^*_{\text{dsos}} \geq f^*_{\text{sdsos}} \geq f^*_{\text{ssos}} \geq f^*_{\text{sos}},$$

- Similar results can be shown for scalar sparse SOS optimization, which rely on the notion of *correlative sparsity pattern* (Waki *et al.*, 2006).

Implementations and numerical comparison

Packages

- SOS optimization: SOSTOOLS, YALMIP
- DSOS/SDSOS optimization: SPOTLESS
- Chordal decomposition: YALMIP (we adapted the option of correlative sparsity technique)
- SDP solver: Mosek

Numerical examples and applications

- Polynomial optimization problems
- Copositive optimization
- Control application: finding Lyapunov functions

Example 1: Polynomial optimization problems

Eigenvalue bounds on matrix polynomials

$$\min_{\gamma} \quad \gamma$$
$$\text{subject to} \quad P(x) + \gamma I \succeq 0,$$

where $n = 2, 2d = 2$, the polynomial is randomly generated. $P(x)$ has an arrow pattern.

Table: CPU time (in seconds) required by Mosek

Dimension r	10	20	30	40	50	60	70	80
SOS	0.30	1.33	6.64	27.3	108.1	308.7	541.3	1 018.6
SSOS	0.34	0.34	0.35	0.35	0.33	0.32	0.32	0.33
SDSOS	0.47	0.63	1.09	1.29	2.67	3.70	4.40	6.02
DSOS	**	**	**	**	**	**	**	**

**: The program is infeasible.

Eigenvalue bounds on matrix polynomials

$$\begin{aligned} \min_{\gamma} \quad & \gamma \\ \text{subject to} \quad & P(x) + \gamma I \succeq 0, \end{aligned}$$

where $n = 2, 2d = 2$, the polynomial is randomly generated. $P(x)$ has an arrow pattern.

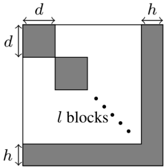
Table: Optimal value γ

Dimension r	10	20	30	40	50	60	70	80
SOS	1.447	4.813	5.917	4.154	21.61	10.09	7.364	10.19
SSOS	1.454	4.878	5.917	4.498	21.64	12.71	7.558	11.39
SDSOS	40.1	279.3	1 254.4	145.5	762.8	1 521.1	1 217.3	598.0
DSOS	**	**	**	**	**	**	**	**

**: The program is infeasible.

Consider the following copositive program

$$\min_{\gamma} \quad \gamma$$
$$\text{subject to} \quad Q + \gamma I \in \mathcal{C}^n,$$



where Q is a random symmetric matrix with a block-arrow sparsity pattern.

Numerical results

In the simulation, the block size is $d = 3$; arrow head is $h = 2$; we vary the number of blocks l

Table: CPU time (in seconds) required by Mosek

l	2	4	6	8	10
SOS	0.45	7.34	248.9	*	*
SSOS	0.39	0.41	0.38	0.49	0.40
SDSOS	0.54	1.22	4.99	11.07	32.18
DSOS	0.59	0.76	2.19	5.72	17.11

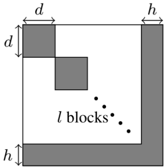
*: Out of memory.

Example 2: Copositive optimization

Consider the following copositive program

$$\begin{aligned} \min_{\gamma} \quad & \gamma \\ \text{subject to} \quad & Q + \gamma I \in \mathcal{C}^n, \end{aligned}$$

where Q is a random symmetric matrix with a block-arrow sparsity pattern.



Numerical results

In the simulation, the block size is $d = 3$; arrow head is $h = 2$; we vary the number of blocks l

Table: Optimal value γ

l	2	4	6	8	10
SOS	1.137	4.197	2.836	*	*
SSOS	1.137	4.197	2.836	4.043	4.718
SDSOS	1.184	4.500	3.282	4.562	5.146
DSOS	2.551	7.775	6.452	12.057	15.203

*: Out of memory.

Example 3: Finding Lyapunov functions

Control application: finding Lyapunov functions

- Consider a dynamical system with a banded pattern

$$\begin{aligned}
\dot{x}_1 &= f_1(x_1, x_2), & g_1(x) &= \gamma - x_1^2 \geq 0 \\
\dot{x}_2 &= f_2(x_1, x_2, x_3), & g_2(x) &= \gamma - x_2^2 \geq 0 \\
&\vdots \\
\dot{x}_n &= f_n(x_{n-1}, x_n), & g_2(x) &= \gamma - x_n^2 \geq 0
\end{aligned}$$

- Generate locally stable systems of degree three;
- Consider a polynomial Lyapunov function of degree two with a banded pattern

$$V(x) = V_1(x_1, x_2) + V_2(x_1, x_2, x_3) + \ldots + V_n(x_{n-1}, x_n)$$

- Then, we consider the following SOS program

$$\begin{aligned}
\text{Find} \quad & V(x), r_i(x) \\
\text{subject to} \quad & V(x) - \epsilon(x^{\mathsf{T}} x) \text{ is SOS} \\
& -\langle \nabla V(x), f(x) \rangle - \sum_{i=1}^{n} r_i(x) g_i(x) \text{ is SOS} \\
& r_i(x) \text{ is SOS}, i = 1, \ldots, n.
\end{aligned}$$

Example 3: Finding Lyapunov functions

Control application: finding Lyapunov functions

Table: CPU time (in seconds) required by Mosek

n	10	15	20	30	40	50
SOS	1.29	18.44	247.84	*	*	*
SSOS	0.55	0.68	0.71	0.83	1.04	1.17
SDSOS	0.71	1.76	4.47	32.21	85.99	257.20
DSOS	0.70	1.42	3.58	35.12	73.64	324.32

*: Out of memory.