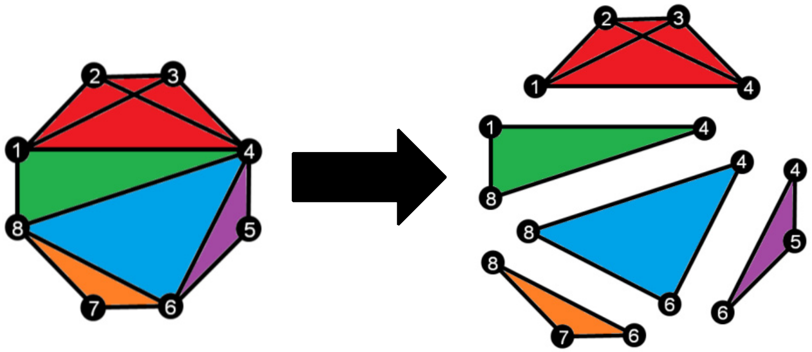
Exploiting Structure in SDPs with Chordal Sparsity

Antonis Papachristodoulou

Department of Engineering Science, University of Oxford

Joint work with Yang Zheng, Giovanni Fantuzzi, Paul Goulart and Andrew Wynn



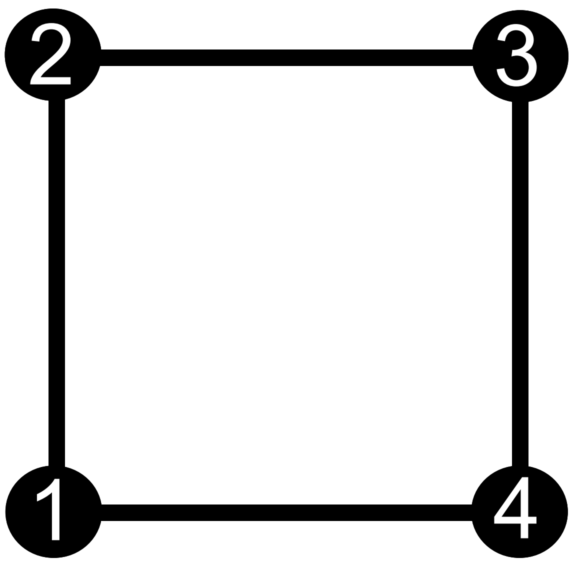
CDC 2016 Pre-conference Workshop

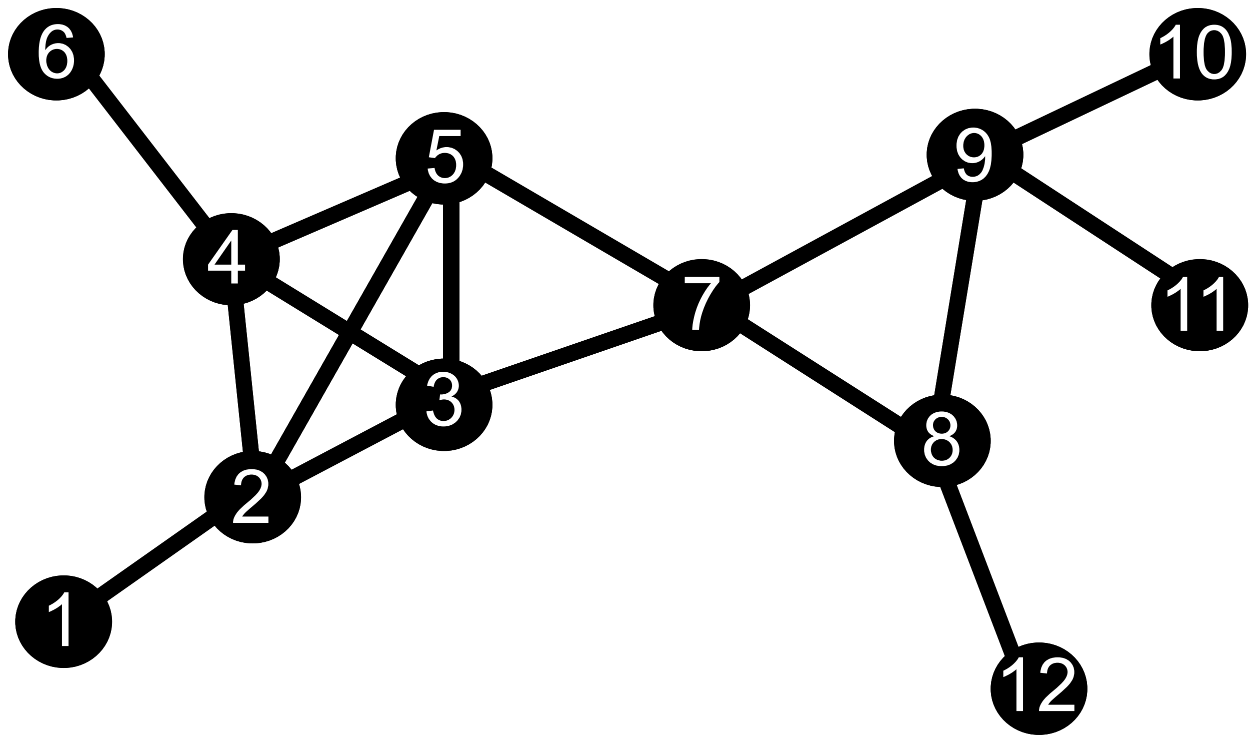
UNIVERSITY OF OXFORD

OUTLINE

Chordal Graphs

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is chordal if every cycle of length $\geq$ 4 has a chord.

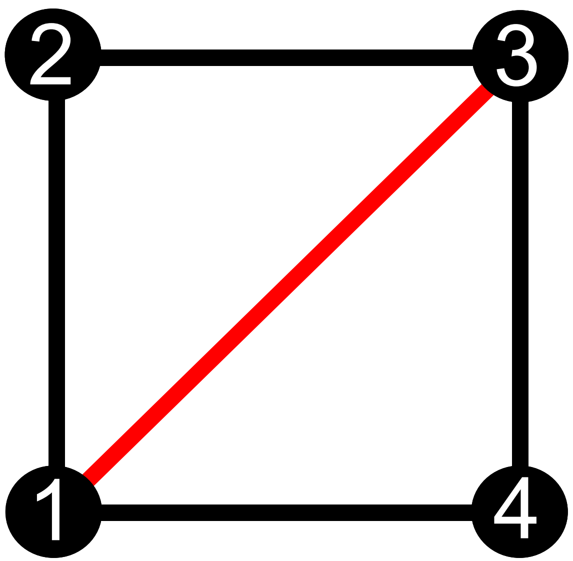


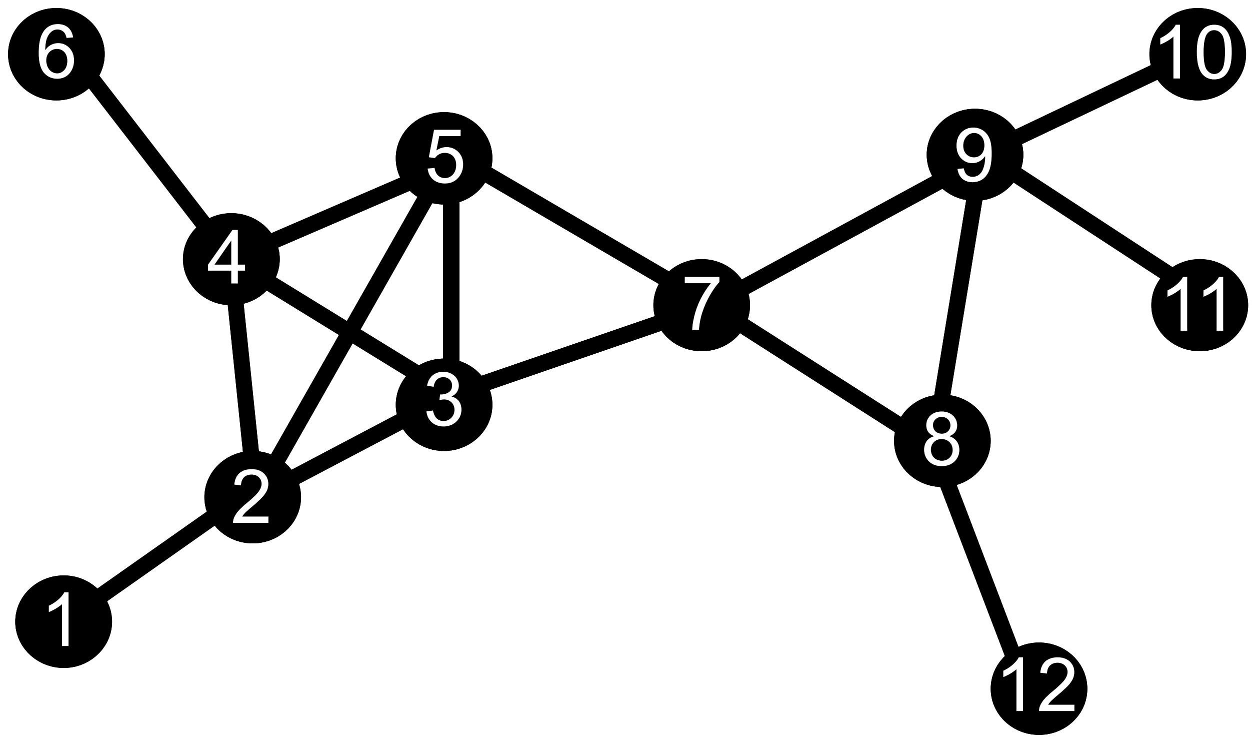


Can recognise chordal graphs in $O(|\mathcal{V}| + |\mathcal{E}|)$ time

Chordal Graphs

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is chordal if every cycle of length $\geq$ 4 has a chord.





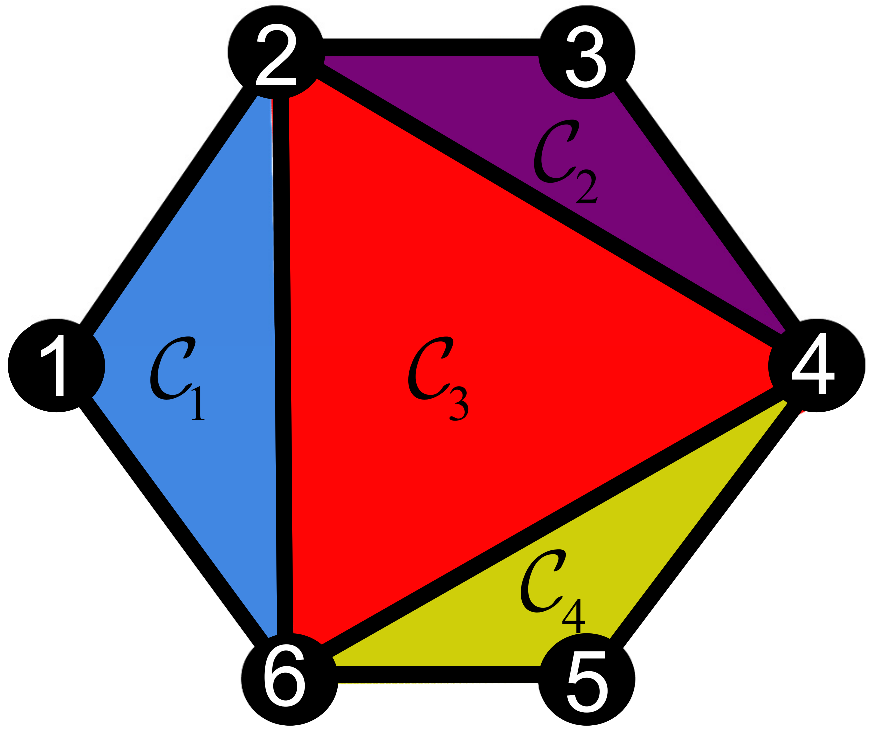
Can recognise chordal graphs in $O(|\mathcal{V}| + |\mathcal{E}|)$ time

Maximal Cliques

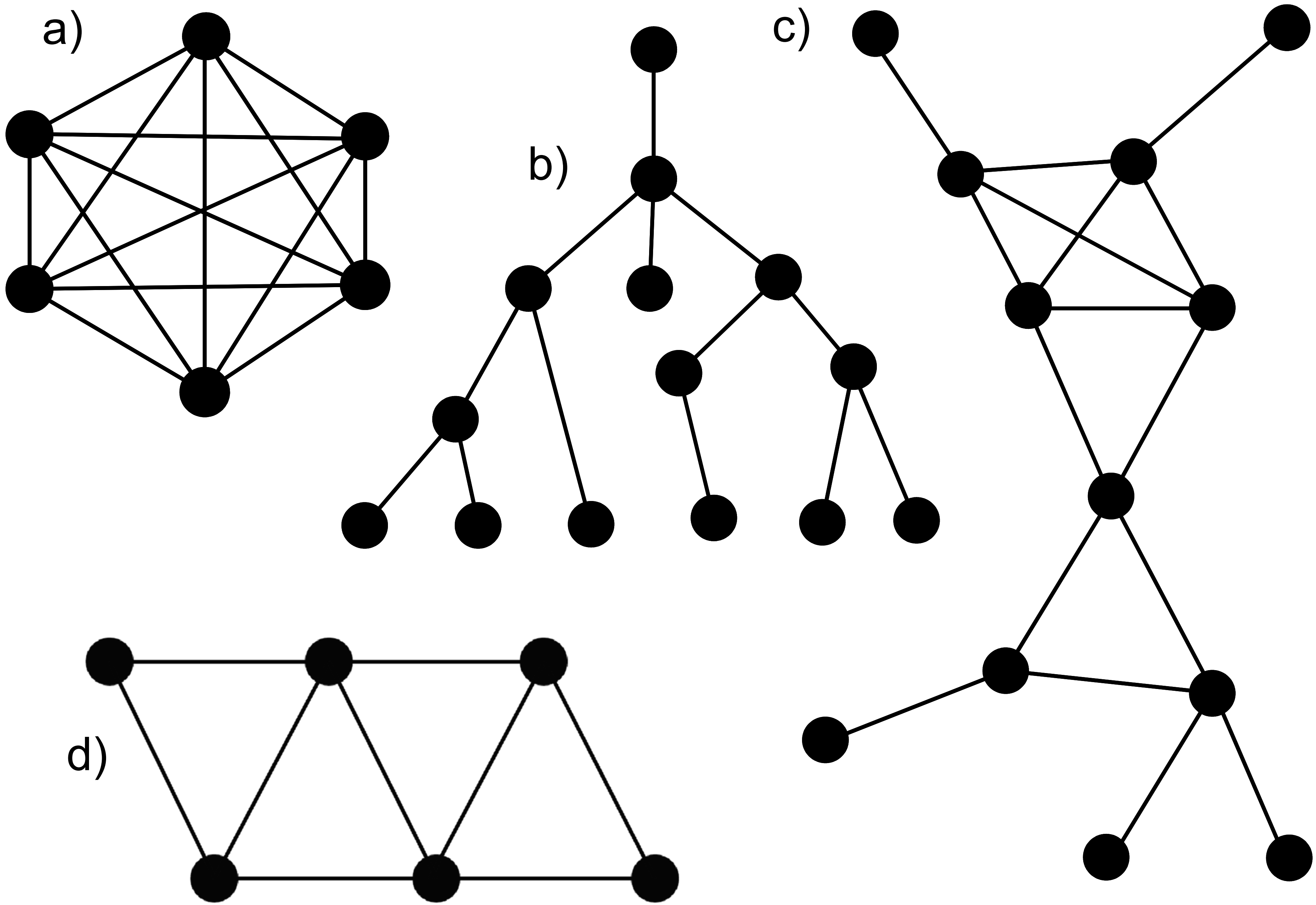
A maximal clique is a clique that is not a subset of another clique.

e.g. $\mathcal{C}_1 = \{1, 2, 6\}$.

Can find the maximal cliques of a chordal graph in $O(|\mathcal{V}| + |\mathcal{E}|)$ time.



Examples of Chordal Graphs

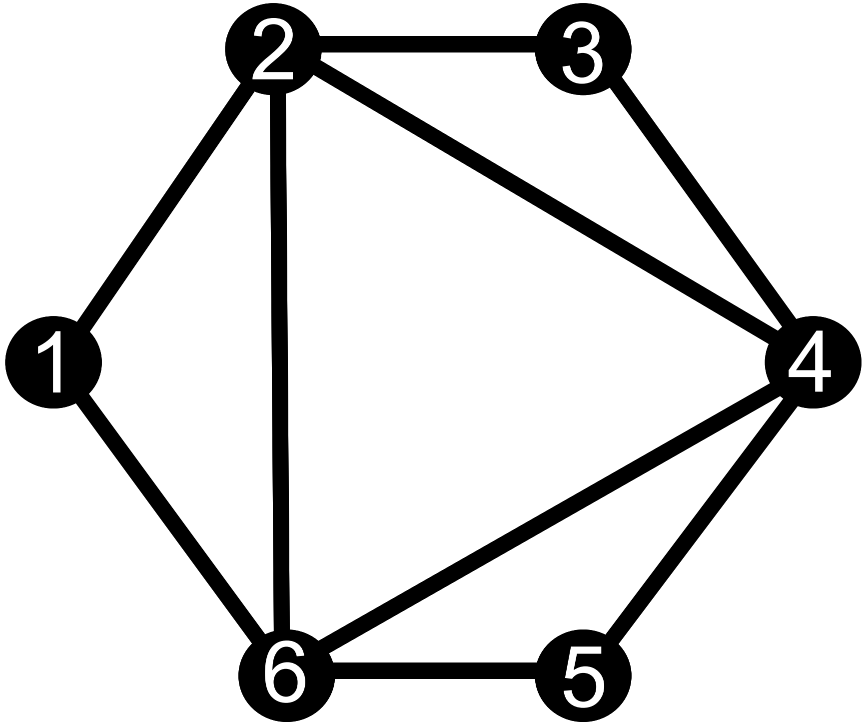


Sparse Matrices

$$Z$$

$$\begin{bmatrix} Z_{11} & Z_{12} & 0 & 0 & 0 & Z_{16} \\ Z_{12} & Z_{22} & Z_{23} & Z_{24} & 0 & Z_{26} \\ 0 & Z_{23} & Z_{33} & Z_{34} & 0 & 0 \\ 0 & Z_{24} & Z_{34} & Z_{44} & Z_{45} & Z_{46} \\ 0 & 0 & 0 & Z_{45} & Z_{55} & Z_{56} \\ Z_{16} & Z_{26} & 0 & Z_{46} & Z_{56} & Z_{66} \end{bmatrix}$$

$$\mathcal{G}(Z) = (\mathcal{V}, \mathcal{E})$$



$$Z \in \mathbb{S}^n(\mathcal{E},0) = \left\{ Z \in \mathbb{S}^n \mid Z_{ij} = 0, \forall (i,j) \notin \mathcal{E} \right\}$$

$$\mathbb{S}^n_+(\mathcal{E},0) = \{ Z \in \mathbb{S}^n(\mathcal{E},0) \mid Z \succeq 0 \}$$

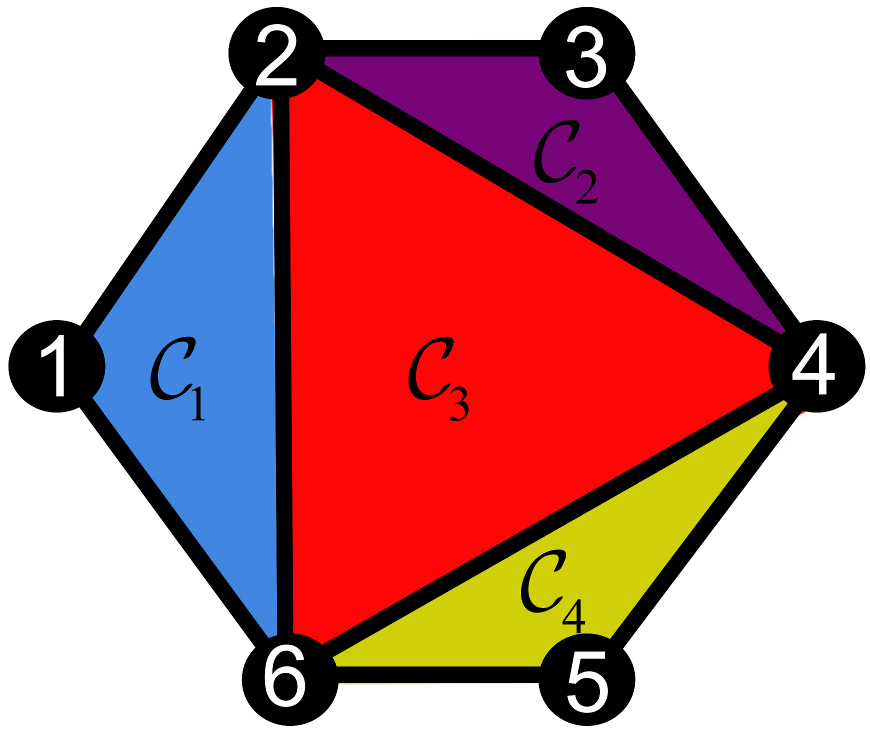
Agler's Theorem

Theorem: Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a chordal graph with set of maximal cliques $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_p\}$. Suppose that $Z \in \mathbb{S}^n(\mathcal{E}, 0)$. Then $Z \in \mathbb{S}^n_+(\mathcal{E}, 0)$ if and only if there exists a set of matrices $\{Z^1, Z^2, \ldots, Z^p\}$ such that $Z = \sum_{k=1}^{p} Z^k$, $Z^k \succeq 0$, $Z^k_{ij} = 0$ if $(i, j) \notin \mathcal{E}_k \times \mathcal{E}_k$.

Example: Applying Agler's Theorem

$$Z$$

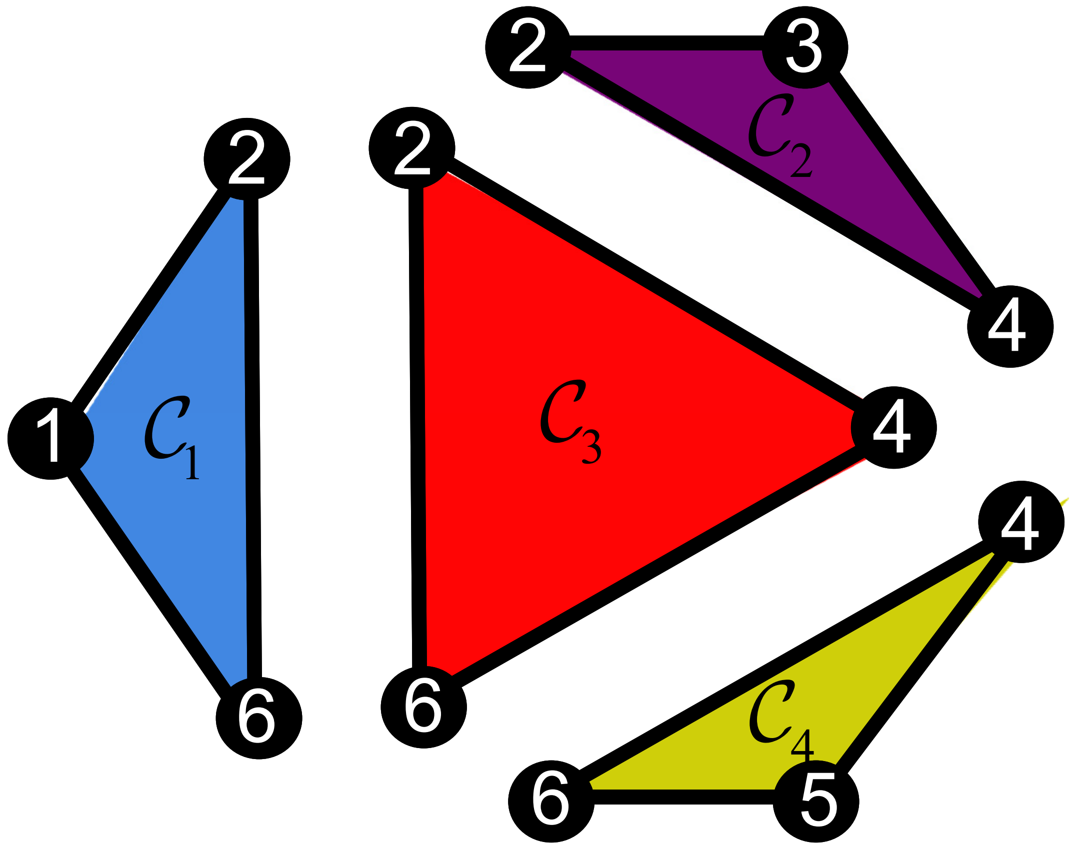
$$\begin{bmatrix} z_{11} & z_{12} & 0 & 0 & 0 & z_{16} \\ z_{12} & z_{22} & z_{23} & z_{24} & 0 & z_{26} \\ 0 & z_{23} & z_{33} & z_{34} & 0 & 0 \\ 0 & z_{24} & z_{34} & z_{44} & z_{45} & z_{46} \\ 0 & 0 & 0 & z_{45} & z_{55} & z_{56} \\ z_{16} & z_{26} & 0 & z_{46} & z_{56} & z_{66} \end{bmatrix}$$



Example: Applying Agler's Theorem

$$Z$$

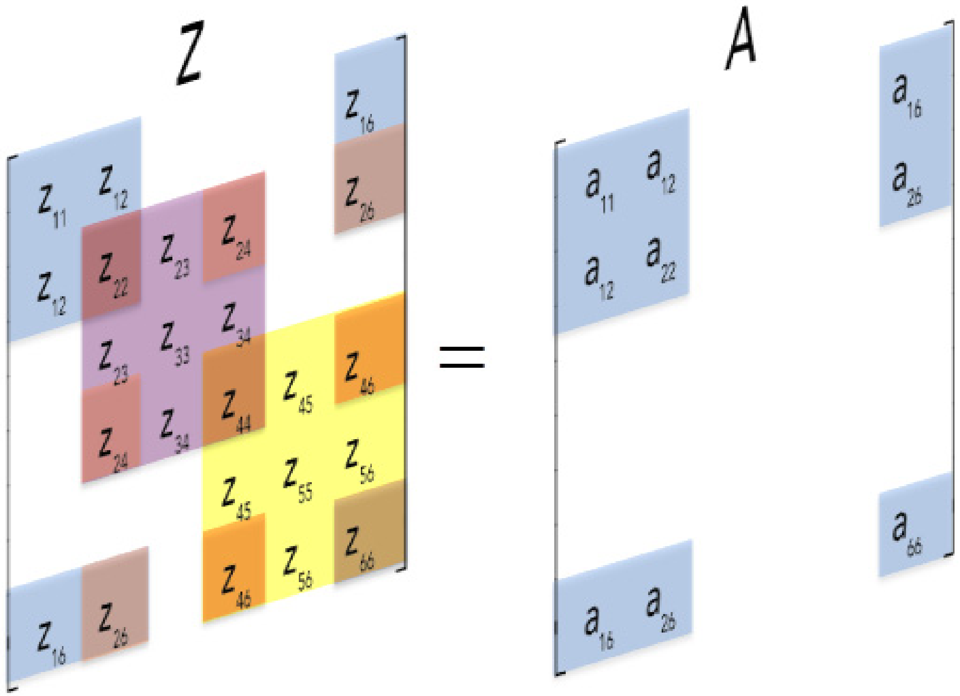
$$\begin{bmatrix} z_{11} & z_{12} & 0 & 0 & 0 & z_{16} \\ z_{12} & z_{22} & z_{23} & z_{24} & 0 & z_{26} \\ 0 & z_{23} & z_{33} & z_{34} & 0 & 0 \\ 0 & z_{24} & z_{34} & z_{44} & z_{45} & z_{46} \\ 0 & 0 & 0 & z_{45} & z_{55} & z_{56} \\ z_{16} & z_{26} & 0 & z_{46} & z_{56} & z_{66} \end{bmatrix}$$



Example: Applying Agler's Theorem

$$Z \succeq 0 \iff \exists\, A,\ B,\ C,\ D \text{ such that}$$

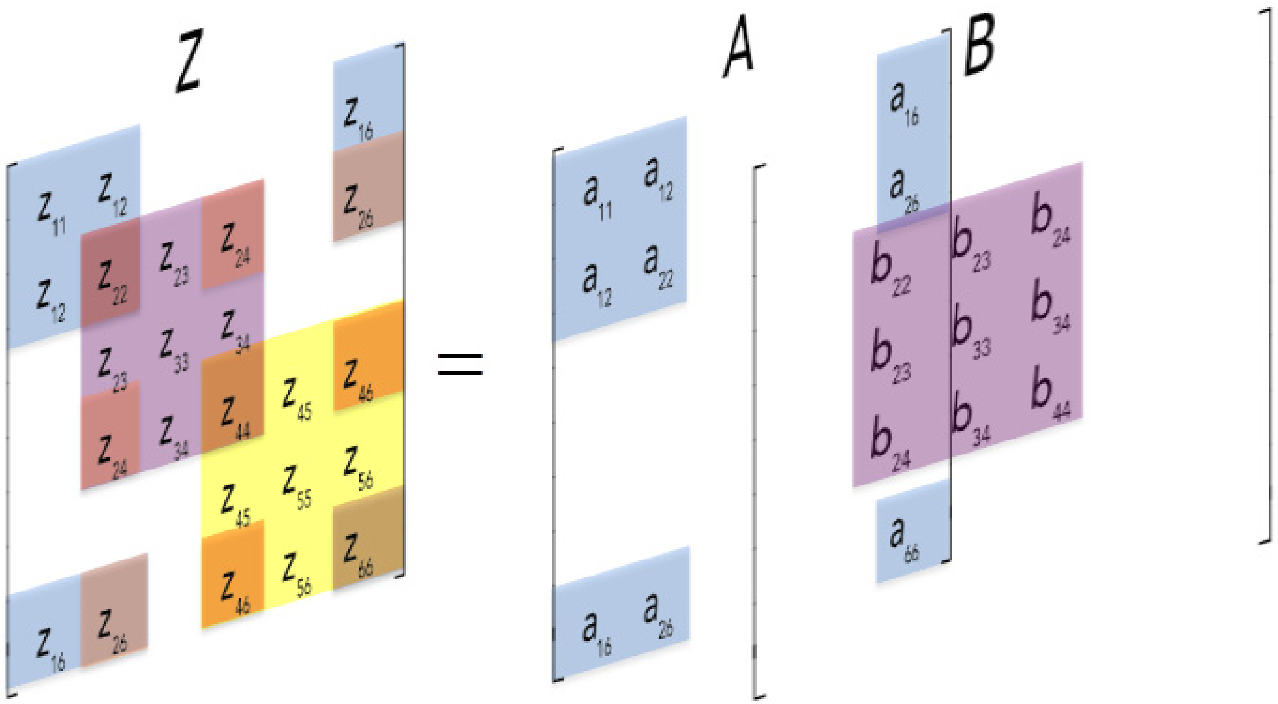
$$A + B + C + D = Z, \quad A,\ B,\ C,\ D \succeq 0$$



Example: Applying Agler's Theorem

$$Z \succeq 0 \;\Leftrightarrow\; \exists\, A,\ B,\ C,\ D \text{ such that}$$

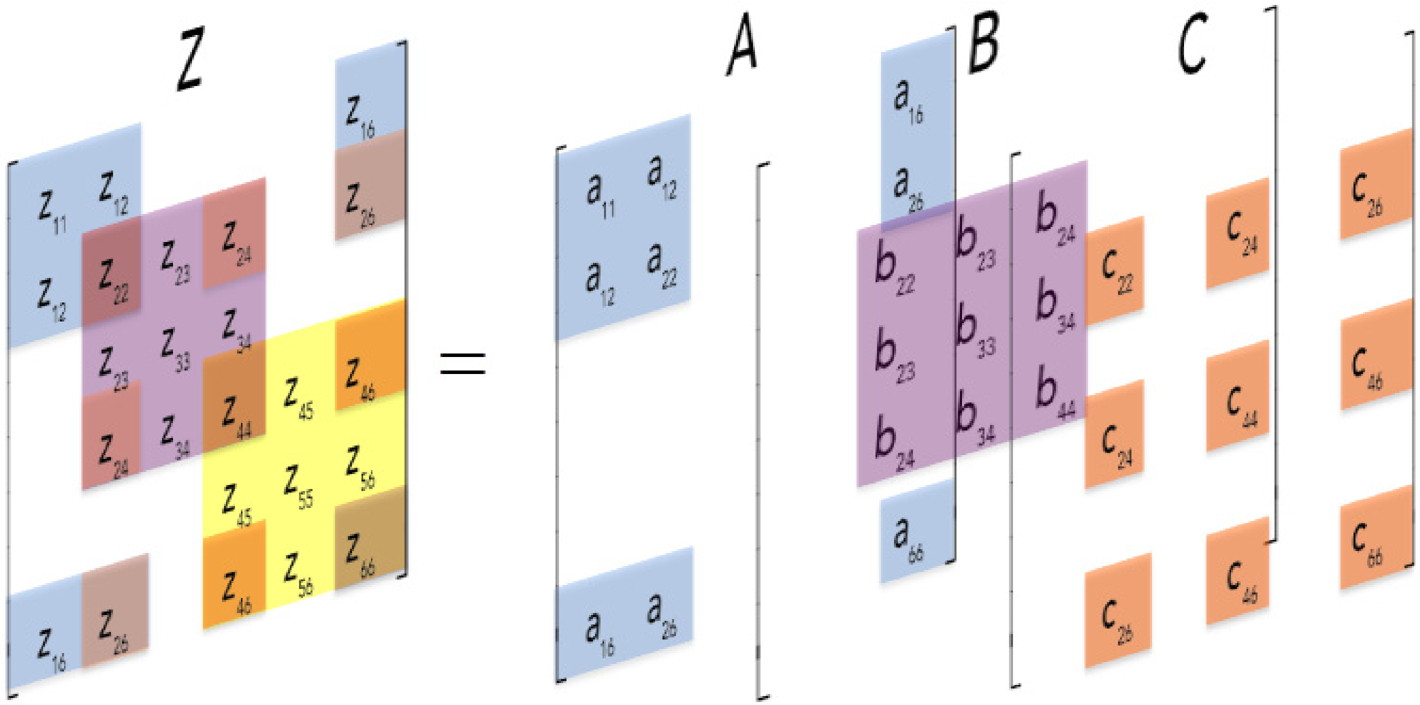
$$A + B + C + D = Z, \qquad A,\ B,\ C,\ D \succeq 0$$



Example: Applying Agler's Theorem

$$Z \succeq 0 \quad \Leftrightarrow \quad \exists\, A,\ B,\ C,\ D \text{ such that}$$

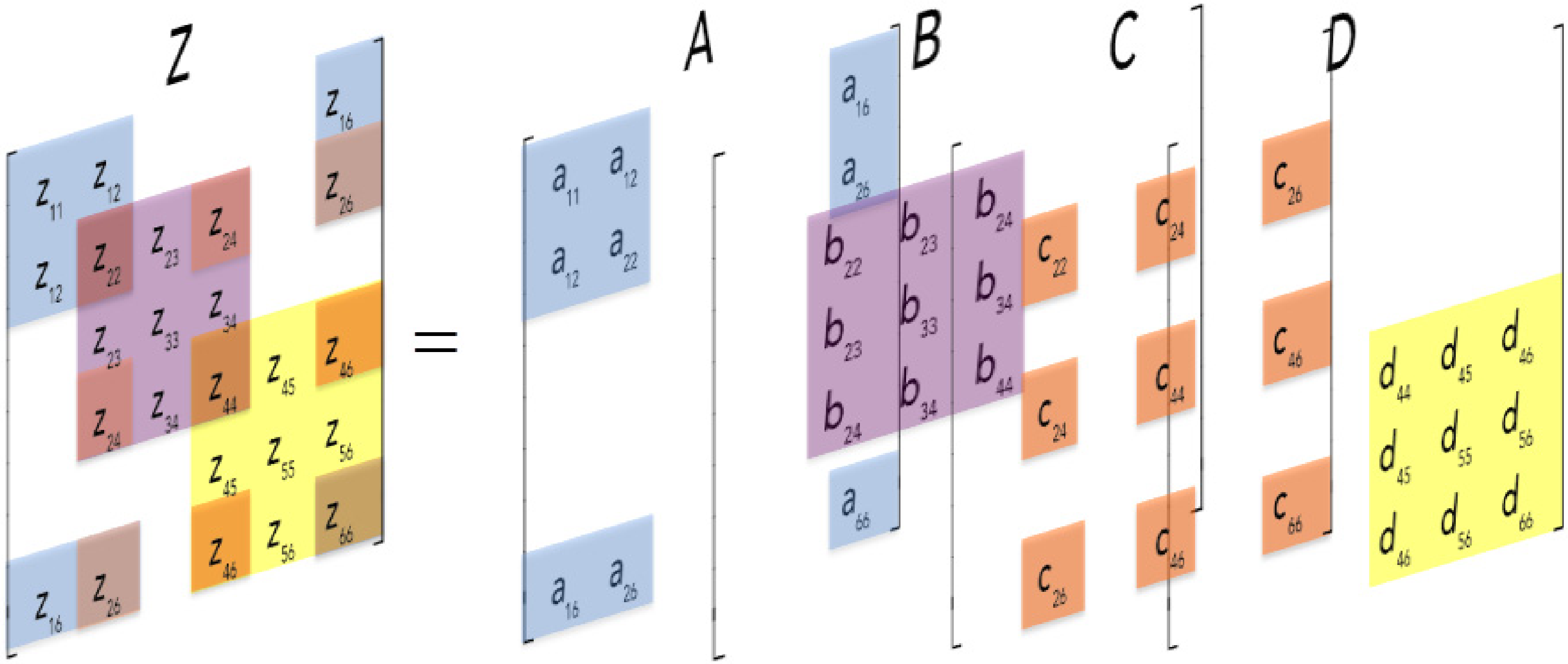
$$A + B + C + D = Z, \qquad A,\ B,\ C,\ D \succeq 0$$



Example: Applying Agler's Theorem

$$Z \succeq 0 \;\Leftrightarrow\; \exists\, A,\ B,\ C,\ D \text{ such that}$$

$$A + B + C + D = Z, \quad A,\ B,\ C,\ D \succeq 0$$

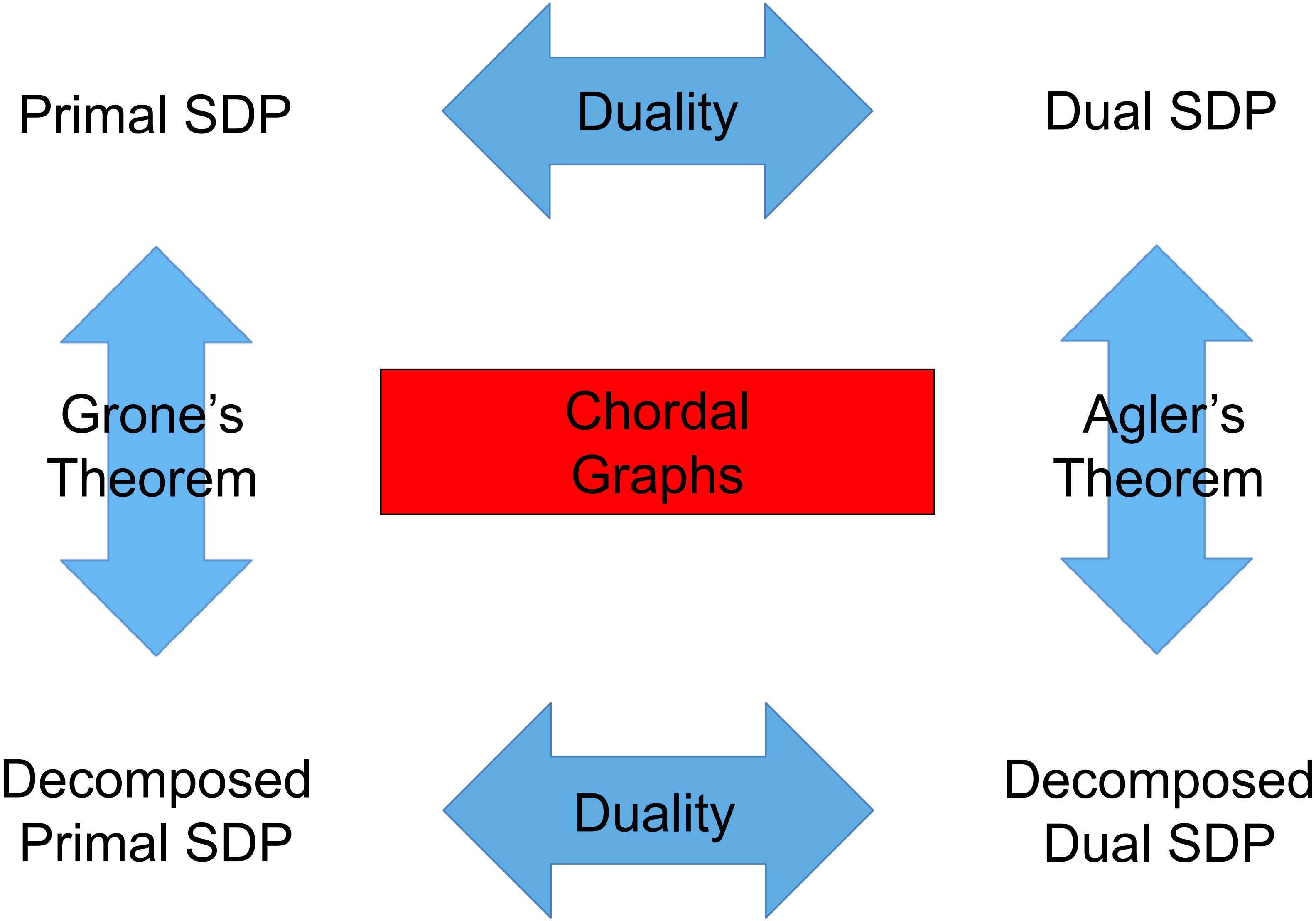


SDPs with Chordal Sparsity

$$\begin{aligned} \min_{X} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \mathcal{A}(X) = b \\ & X \in \mathbb{S}_{+}^{n} \end{aligned} \quad \xleftrightarrow{\text{Dual}} \quad \begin{aligned} \max_{y,Z} \quad & \langle b, y \rangle \\ \text{subject to} \quad & \mathcal{A}^{*}(y) + Z = C \\ & Z \in \mathbb{S}_{+}^{n} \end{aligned}$$

- Applications: control theory, fluid mechanics, machine learning, polynomial optimization, combinatorics, operations research, finance, etc.
- Second-order solvers : SeDuMi, SDPA, SDPT3
- Large-scale cases: exploit the inherent structure of the instances (De Klerk, 2010):
  - ➢ Low Rank and Algebraic Symmetry.
  - ➢ **Chordal Sparsity:**
    - ✓ Second-order methods: Fukuda *et al.*, 2001; Nakata *et al.*, 2003; Andersen *et al.*, 2010;
    - ✓ **First-order methods:** Madani *et al.,* 2015; Sun *et al.,* 2014.

Decomposing Sparse SDPs



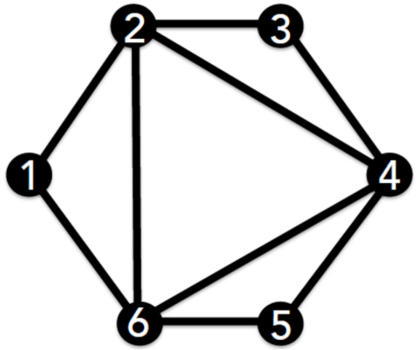
SDPs with Chordal Sparsity

$$\mathbb{S}^n(\mathcal{E},0) = \left\{ X \in \mathbb{S}^n \mid X_{ij} = 0, \forall (i,j) \notin \mathcal{E} \right\}$$

$$\mathbb{S}^n_+(\mathcal{E},0) = \{ X \in \mathbb{S}^n(\mathcal{E},0) \mid X \succeq 0 \}$$

$$\begin{array}{c|c} & \begin{matrix} 1 & \;\;2 & \;\;3 & \;\;4 & \;\;5 & \;\;6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{bmatrix} x_{11} & x_{12} & 0 & 0 & 0 & x_{16} \\ x_{12} & x_{22} & x_{23} & x_{24} & 0 & x_{26} \\ 0 & x_{23} & x_{33} & x_{34} & 0 & 0 \\ 0 & x_{24} & x_{34} & x_{44} & x_{45} & x_{46} \\ 0 & 0 & 0 & x_{45} & x_{55} & x_{56} \\ x_{16} & x_{26} & 0 & x_{46} & x_{56} & x_{66} \end{bmatrix} \end{array}$$

$$\mathcal{G} = (\mathcal{V}, \mathcal{E})$$



$$\begin{array}{c|c} & \begin{matrix} 1 & \;\;2 & \;\;3 & \;\;4 & \;\;5 & \;\;6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{bmatrix} x_{11} & x_{12} & ? & ? & ? & x_{16} \\ x_{12} & x_{22} & x_{23} & x_{24} & ? & x_{26} \\ ? & x_{23} & x_{33} & x_{34} & ? & ? \\ ? & x_{24} & x_{34} & x_{44} & x_{45} & x_{46} \\ ? & ? & ? & x_{45} & x_{55} & x_{56} \\ x_{16} & x_{26} & ? & x_{46} & x_{56} & x_{66} \end{bmatrix} \end{array}$$

$\mathbb{S}^n(\mathcal{E},?) = n \times n$ partial symmetric matrices with entries defined on $\mathcal{E}$

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

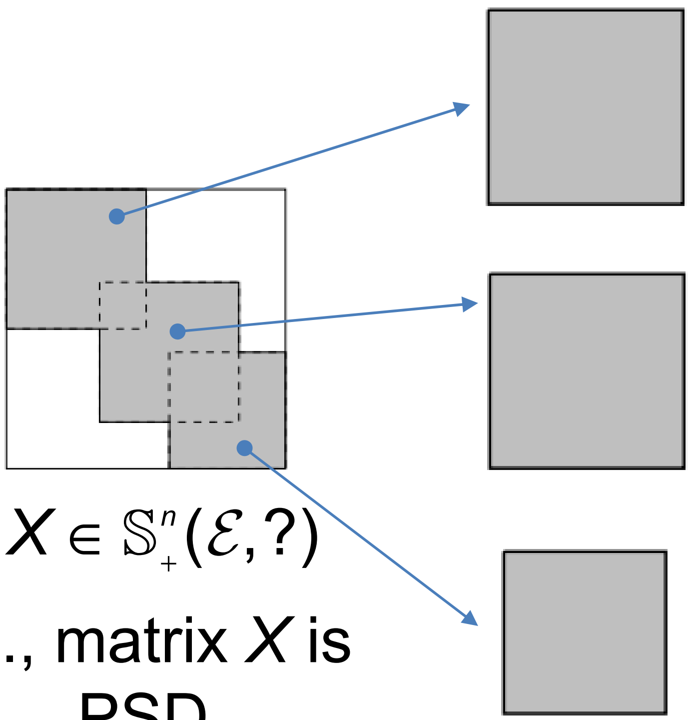
$\boxed{\mathbb{S}^n_+(\mathcal{E},?) \text{ and } \mathbb{S}^n_+(\mathcal{E},0) \text{ are dual cones of each other}}$

Grone's theorem

Consider a choral graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a set of maximal cliques $\mathcal{C}_1, \ldots, \mathcal{C}_p$.

Grone's Theorem:

$$X \in \mathbb{S}^n_+(\mathcal{E}, ?) \text{ if and only if } X(\mathcal{C}_k) \succeq 0, \; k = 1, \ldots, p.$$



$X(\mathcal{C}_1) \succeq 0$

$X(\mathcal{C}_2) \succeq 0$

$X(\mathcal{C}_3) \succeq 0$

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

i.e., matrix X is PSD completable

Cone Decomposition of Primal and Dual SDPs

Primal

$$\min_X \quad \langle C, X\rangle$$
$$\text{subject to} \quad \mathcal{A}(X) = b$$
$$\boxed{X \in \mathbb{S}_+^n}$$

$$\Downarrow$$

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

$$\Downarrow$$

$$\min_X \quad \langle C, X\rangle$$
$$\text{s.t.} \quad \mathcal{A}(X) = b$$
$$\boxed{E_{\mathcal{C}_k} X E_{\mathcal{C}_k}^T \in \mathbb{S}_+^{|\mathcal{C}_k|}, \ k = 1, \dots, p}$$

Dual

$$\max_{y,Z} \quad \langle b, y\rangle$$
$$\text{subject to} \quad \mathcal{A}^*(y) + Z = C$$
$$\boxed{Z \in \mathbb{S}_+^n}$$

$$\Downarrow$$

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

$$\Downarrow$$

$$\max_{y,Z} \quad \langle b, y\rangle$$
$$\text{s.t} \quad \mathcal{A}^*(y) + \sum_{k=1}^{p} E_{\mathcal{C}_k}^T Z_k E_{\mathcal{C}_k} = C$$
$$\boxed{Z \in \mathbb{S}_+^{|\mathcal{C}_k|}, \quad k = 1, \dots, p}$$

OUTLINE

1. Chordal Graphs and Positive Semidefinite Matrices
2. ADMM for Primal and Dual Sparse SDPs
3. CDCS: Cone Decomposition Conic Solver
4. Conclusion

Alternating Direction Method of Multipliers

An operator splitting method for a problem of the form

$$\begin{aligned} \min \quad & f(x)+g(z) \\ \text{s.t.} \quad & x=z \end{aligned}$$

f, g may be nonsmooth.

Lagrangian:

$$\mathcal{L}=f(x)+g(z)+\frac{\rho}{2}\left\| x-z+\frac{1}{\rho}\lambda \right\|_2^2$$

ADMM:

$$x^{n+1}=\operatorname*{argmin}_x \left(f(x)+\frac{\rho}{2}\left\| x-z^n+\frac{1}{\rho}\lambda^n \right\|_2^2 \right)$$

$$z^{n+1}=\operatorname*{argmin}_z \left(g(z)+\frac{\rho}{2}\left\| x^{n+1}-z+\frac{1}{\rho}\lambda^n \right\|_2^2 \right)$$

$$\lambda^{n+1}=\lambda^n+\rho(x^{n+1}-z^{n+1})$$

Reformulation and decomposition of the PSD constraint

$$\begin{aligned}\min_{X}\quad & \langle C, X\rangle \\ \text{s.t.}\quad & \mathcal{A}(X) = b \\ & X_k = E_{\mathcal{C}_k} X E_{\mathcal{C}_k}^T,\ k = 1,\dots,p \\ & X_k \in \mathbb{S}_+^{|\mathcal{C}_k|},\ k = 1,\dots,p\end{aligned} \quad \Rightarrow \quad \begin{aligned}\min_{x,x_1,\dots,x_p}\quad & c^T x \\ \text{s.t.}\quad & Ax = b \\ & x_k = H_k x,\ k = 1,\dots,p \\ & x_k \in \mathcal{S}_k,\ k = 1,\dots,p\end{aligned}$$

- Using indicator functions

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

- Augmented Lagrangian

$$\mathcal{L} = c^T x + \delta_0(Ax - b) + \sum_{k=1}^{p} \left[\delta_{\mathcal{S}_k}(x_k) + \frac{\rho}{2} \left\| x_k - H_k x + \frac{1}{\rho}\lambda_k \right\|^2 \right]$$

- Regroup the variables

$$\mathcal{X} \triangleq \{x\};\ \ \mathcal{Y} \triangleq \{x_1,\dots x_p\};\ \ \mathcal{Z} \triangleq \{\lambda_1,\dots \lambda_p\}$$

ADMM for Primal SDPs

$$\mathcal{L} = c^T x + \delta_0(Ax - b) + \sum_{k=1}^{p}\left[\delta_{\mathcal{S}_k}(x_k) + \frac{\rho}{2}\left\| x_k - H_k x + \frac{1}{\rho}\lambda_k \right\|^2\right]$$

$$\mathcal{X} \triangleq \{x\};\quad \mathcal{Y} \triangleq \{x_1, \ldots x_p\};\quad \mathcal{Z} \triangleq \{\lambda_1, \ldots \lambda_p\}$$

1) Minimization over $\mathcal{X}$

$$\min_x \quad c^T x + \frac{\rho}{2}\sum_{k=1}^{p}\left\| x_k^{(n)} - H_k x + \frac{1}{\rho}\lambda_k^{(n)} \right\|^2$$

$$\text{s.t.} \quad Ax = b$$

QP with linear constraint

✓ the KKT system matrix only depends on the problem data.

2) Minimization over $\mathcal{Y}$

$$\min_{x_k} \quad \left\| x_k - H_k x^{(n+1)} + \frac{1}{\rho}\lambda_k^{(n)} \right\|^2$$

$$\text{s.t.} \quad x_k \in \mathcal{S}_k$$

Projections in parallel

3) Update multipliers

$$\lambda_k^{(n+1)} = \lambda_k^{(n)} + \rho\left(x_k^{(n+1)} - H_k x^{(n+1)}\right)$$

ADMM for Primal and Dual SDPs

$$\begin{tikzcd}[column sep=huge, row sep=large]
\text{Primal SDP} \arrow[r, "\text{Duality}"] \arrow[d, "\text{Grone's Theorem}"'] & \text{Dual SDP} \arrow[d, "\text{Agler's Theorem}"] \\
\text{Decomposed Primal SDP} \arrow[r, "\text{Duality}"] \arrow[d, "\text{ADMM}"'] & \text{Decomposed Dual SDP} \arrow[d, "\text{ADMM}"] \\
\text{Algorithm 1} \arrow[r, "\text{Scaling}"] & \text{Algorithm 2}
\end{tikzcd}$$

The duality between the primal and dual SDP is inherited by the decomposed problems by virtue of the duality between Grone's and Agler's theorems.

ADMM for the Homogeneous self-dual embedding

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

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

- Notational simplicity

$$s \triangleq \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix}, \ z \triangleq \begin{bmatrix} z_1 \\ \vdots \\ z_p \end{bmatrix}, \ v \triangleq \begin{bmatrix} v_1 \\ \vdots \\ v_p \end{bmatrix}, \ H \triangleq \begin{bmatrix} H_1 \\ \vdots \\ H_p \end{bmatrix}$$

- KKT conditions
  - Primal feasible $\quad Ax^* - r^* = b, \ \ r^* = 0,$

    $s^* + w^* = Hx^*, \ w^* = 0, \ s^* \in \mathcal{S}$
  - Dual feasible $\quad A^T y^* + H^T v^* + h^* = c, \ \ h^* = 0,$

    $z^* - v^* = 0, \ z^* \in \mathcal{S}$
  - Zero-duality gap $\quad c^T x^* - b^T y^* = 0$

ADMM for the Homogeneous self-dual embedding

$$\begin{bmatrix} h \\ z \\ r \\ w \\ \kappa \end{bmatrix} = \begin{bmatrix} 0 & 0 & -A^T & -H^T & c \\ 0 & 0 & 0 & I & 0 \\ A & 0 & 0 & 0 & -b \\ H & -I & 0 & 0 & 0 \\ -c^T & 0 & b^T & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ s \\ y \\ v \\ \tau \end{bmatrix}$$

τ, κ are two non-negative and complementary variables

- Notational simplicity

$$v \triangleq \begin{bmatrix} h \\ z \\ r \\ w \\ \kappa \end{bmatrix}, \ u \triangleq \begin{bmatrix} x \\ s \\ y \\ v \\ \tau \end{bmatrix}, \ Q \triangleq \begin{bmatrix} 0 & 0 & -A^T & -H^T & c \\ 0 & 0 & 0 & I & 0 \\ A & 0 & 0 & 0 & -b \\ H & -I & 0 & 0 & 0 \\ -c^T & 0 & b^T & 0 & 0 \end{bmatrix}, \ \mathcal{K} = \mathbb{R}^{n^2} \times \mathcal{S} \times \mathbb{R}^m \times \mathbb{R}^{n_d} \times \mathbb{R}_+$$

- Feasibility problem

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

ADMM for the Homogeneous self-dual embedding

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

- ADMM steps (similar to solver SCS [1])

$$\hat{u}^{k+1} = (I+Q)^{-1}(u^k + v^k) \longrightarrow \text{Projection to a subspace}$$

$$u^{k+1} = \mathrm{P}_{\mathcal{K}}(\hat{u}^{k+1} - v^k) \longrightarrow \text{Projection to cones}$$

$$v^{k+1} = v^k - \hat{u}^{k+1} + u^{k+1}$$

$\boxed{Q \text{ is highly structured and sparse}}$

$$Q \triangleq \begin{bmatrix} 0 & 0 & -A^T & -H^T & c \\ 0 & 0 & 0 & I & 0 \\ A & 0 & 0 & 0 & -b \\ H & -I & 0 & 0 & 0 \\ -c^T & 0 & b^T & 0 & 0 \end{bmatrix}$$

✓ Block elimination can be applied here to speed up the projection;

✓ Then, the per-iteration cost is the same as applying a splitting method to the primal or dual alone.

[1] O'Donoghue, B., Chu, E., Parikh, N. and Boyd, S. (2016). Conic optimization via operator splitting and homogeneous self-dual embedding. Journal of Optimization Theory and Applications, 169(3), 1042– 1068

Outline

1. Chordal Graphs and Positive Semidefinite Matrices
2. ADMM for Primal and Dual Sparse SDPs
3. CDCS: Cone Decomposition Conic Solver
4. Conclusion

CDCS: Cone Decomposition Conic Solver

- An open source MATLAB solver for partially decomposable conic programs;
- CDCS supports constraints on the following cones:
  - ✓ Free variables
  - ✓ non-negative orthant
  - ✓ second-order cone
  - ✓ the positive semidefinite cone.
- Input-output format is aligned with SeDuMi;
- Works with latest YALMIP release.

Syntax:

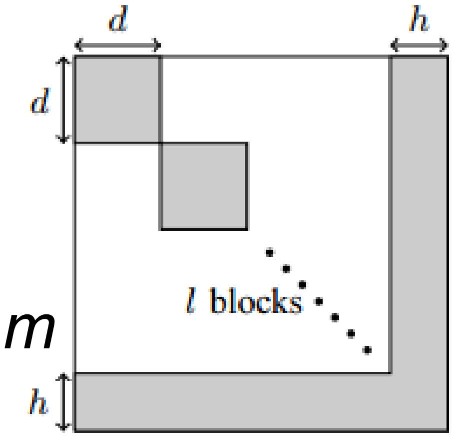
```
[x,y,z,info] = cdcs(At,b,c,K,opts);
```

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

CDCS: Cone Decomposition Conic Solver

Random SDPs with block-arrow pattern

- Block size: *d*
- Number of Blocks: *l*
- Arrow head: *h*
- Number of constraints: *m*

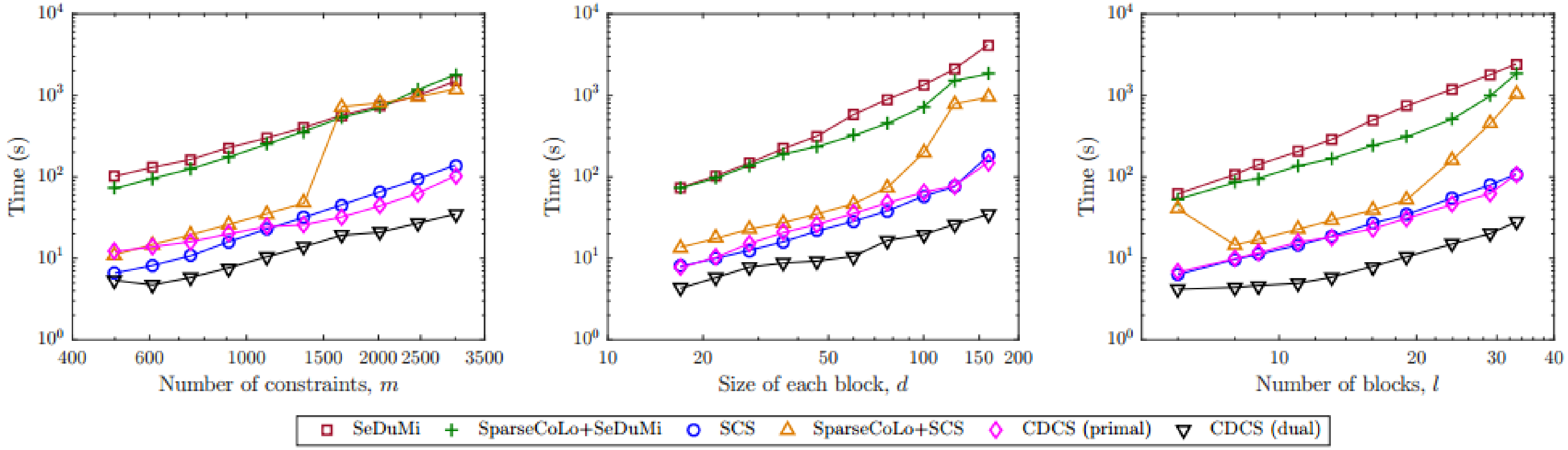


Numerical Comparison

- SeDuMi
- sparseCoLO+SeDuMi
- SCS
- sparseCoLO+SCS

CDCS and SCS $\epsilon_{\text{tol}} = 10^{-3}$

Numerical Result



CPU time for SDPs with block-arrow patterns. Left to right: varying number of constraints; varying number of blocks; varying block size.

Benchmark problems in SDPLIB

Three sets of benchmark problems in SDPLIB (Borchers, 1999):

1) Four small and medium-sized SDPs (theta1, theta2, qap5 and qap9);
2) Four large-scale sparse SDPs (maxG11, maxG32, qpG11 and qpG51);
3) Two infeasible SDPs (infp1 and infd1).

Table 1. Details of the SDPLIB problems considered in this work.

	Small and medium-size ($n \leq 100$)				Large-scale and sparse ($n \geq 800$)				Infeasible	
	theta1	theta2	qap5	qap9	maxG11	maxG32	qpG11	qpG51	infp1	infd1
Original cone size, n	50	100	26	82	800	2000	1600	2000	30	30
Affine constraints, m	104	498	136	748	800	2000	800	1000	10	10
Number of cliques, p	1	1	1	1	598	1499	1405	1675	1	1
Maximum clique size	50	100	26	82	24	60	24	304	30	30
Minimum clique size	50	100	26	82	5	5	1	1	30	30

CDCS: Cone Decomposition Conic Solver

Result: small and medium-sized instances

Table 2. Results for some small and medium-sized SDPs in SDPLIB.

		SeDuMi	SparseCoLO+ SeDuMi	SCS	CDCS (primal)	CDCS (dual)	Self-dual
theta1	Total time (s)	0.262	0.279	0.145	0.751	0.707	0.534
	Pre- time (s)	0	0.005	0.011	0.013	0.010	0.012
	Iterations	14	14	240	317	320	230
	Objective	2.300×10^1	2.300×10^1	2.300×10^1	2.299×10^1	2.299×10^1	2.303×10^1
theta2	Total time (s)	1.45	1.55	0.92	1.45	1.30	0.60
	Pre- time (s)	0	0.014	0.018	0.046	0.036	0.031
	Iterations	15	15	500	287	277	110
	Objective	3.288×10^1	3.288×10^1	3.288×10^1	3.288×10^1	3.288×10^1	3.287×10^1
qap5	Total time (s)	0.365	0.386	0.412	0.879	0.748	1.465
	Pre- time (s)	0	0.006	0.026	0.011	0.009	0.009
	Iterations	12	12	320	334	332	783
	Objective	-4.360×10^2	-4.360×10^2	-4.359×10^2	-4.360×10^2	-4.364×10^2	-4.362×10^2
qap9	Total time (s)	6.291	6.751	3.261	7.520	7.397	1.173
	Pre- time (s)	0	0.012	0.010	0.064	0.036	0.032
	Iterations	25	25	2000	2000	2000	261
	Objective	-1.410×10^3	-1.410×10^3	-1.409×10^3	-1.407×10^3	-1.409×10^3	-1.410×10^3

CDCS: Cone Decomposition Conic Solver

Result: large-sparse instances

Table 3. Results for some large-scale sparse SDPs in SDPLIB.

		SeDuMi	SparseCoLO+ SeDuMi	SCS	CDCS (primal)	CDCS (dual)	Self-dual
maxG11	Total time (s)	92.0	9.83	160.5	126.6	114.1	23.9
	Pre- time (s)	0	2.39	0.07	3.33	4.28	2.45
	Iterations	13	15	1860	1317	1306	279
	Objective	6.292×10^2	6.292×10^2	6.292×10^2	6.292×10^2	6.292×10^2	6.295×10^2
maxG32	Total time (s)	1.385×10^3	577.4	2.487×10^3	520.0	273.8	87.4
	Pre- time (s)	0	7.63	0.589	53.9	55.6	30.5
	Iterations	14	15	2000	1796	943	272
	Objective	1.568×10^3	1.568×10^3	1.568×10^3	1.568×10^3	1.568×10^3	1.568×10^3
qpG11	Total time (s)	675.3	27.3	1.115×10^3	273.6	92.5	32.1
	Pre- time (s)	0	11.2	0.57	6.26	6.26	3.85
	Iterations	14	15	2000	1355	656	304
	Objective	2.449×10^3	2.449×10^3	2.449×10^3	2.449×10^3	2.449×10^3	2.450×10^3
qpG51	Total time (s)	1.984×10^3	–	2.290×10^3	1.627×10^3	1.635×10^3	538.1
	Pre- time (s)	0	–	0.90	10.82	12.77	7.89
	Iterations	22	–	2000	2000	2000	716
	Objective	1.182×10^3	–	1.288×10^3	1.183×10^3	1.186×10^3	1.181×10^3

Result: Infeasible instances

Table 4. Results for two infeasible SDPs in SDPLIB.

		SeDuMi	SparseCoLO+ SeDuMi	SCS	CDCS (primal)	CDCS (dual)	Self-dual
infp1	Total time (s)	0.063	0.083	0.062	*	*	0.18
	Pre- time (s)	0	0.010	0.016	*	*	0.010
	Iterations	2	2	20	*	*	104
	Status	Infeasible	Infeasible	Infeasible	*	*	Infeasible
infd1	Total time (s)	0.125	0.140	0.050	*	*	0.144
	Pre- time (s)	0	0.009	0.013	*	*	0.009
	Iterations	4	4	40	*	*	90
	Status	Infeasible	Infeasible	Infeasible	*	*	Infeasible

CDCS: Cone Decomposition Conic Solver

- **Result: CPU time per iteration**

Table 5. CPU time per iteration (s) for some SDPs in SDPLIB

	SCS	CDCS (primal)	CDCS (dual)	Self-dual
theta1	6×10^{-4}	2.3×10^{-3}	2.2×10^{-3}	2.3×10^{-3}
theta2	1.8×10^{-3}	5.1×10^{-3}	4.7×10^{-3}	5.5×10^{-3}
qap5	1.2×10^{-3}	2.6×10^{-3}	2.2×10^{-3}	1.9×10^{-3}
qap9	1.5×10^{-3}	3.6×10^{-3}	3.7×10^{-3}	4.2×10^{-3}
maxG11	0.086	0.094	0.084	0.077
maxG32	1.243	0.260	0.231	0.209
qpG11	0.557	0.198	0.132	0.093
qpG51	1.144	0.808	0.811	0.741

✓ Our codes are currently written in MATLAB

✓ SCS is implemented in C.

Conclusion

$$\begin{tikzcd}
\text{Primal SDP} \arrow[r, "\text{Duality}"] \arrow[d, "\text{Grone's Theorem}"'] & \text{Dual SDP} \arrow[d, "\text{Agler's Theorem}"] \\
\text{Decomposed Primal SDP} \arrow[r, "\text{Duality}"] \arrow[d, "\text{ADMM}"'] & \text{Decomposed Dual SDP} \arrow[d, "\text{ADMM}"] \\
\text{Algorithm 1} \arrow[r, "\text{Scaling}"] & \text{Algorithm 2}
\end{tikzcd}$$

- Introduced a conversion framework for sparse SDPs
- Developed efficient ADMM algorithms
  - ✓ Primal and dual standard form;
  - ✓ The homogeneous self-dual embedding;

suitable for first-order methods

Ongoing work

- Develop ADMM algorithms for sparse SDPs arising in SOS.
- Applications in networked systems and power systems

Thank you for your attention!

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

1. Zheng, Y., Fantuzzi G., Papachristodoulou A., Goulart, P., and Wynn, A. (2016) Fast ADMM for Semidefinite Programs with Chordal Sparsity. *arXiv preprint arXiv:1609.06068*
2. Zheng, Y., Fantuzzi, G., Papachristodoulou, A., Goulart, P., & Wynn, A. (2016) Fast ADMM for homogeneous self-dual embeddings of sparse SDPs. *arXiv preprint arXiv:1611.01828*