Fast ADMM for Semidefinite Programs (SDPs) with Chordal Sparsity

Yang Zheng

Department of Engineering Science,
University of Oxford

Joint work with Giovanni Fantuzzi, Antonis Papachristodoulou,
Paul Goulart and Andrew Wynn

UNIVERSITY OF OXFORD

Seminar talk, University of Birmingham, February, 2017

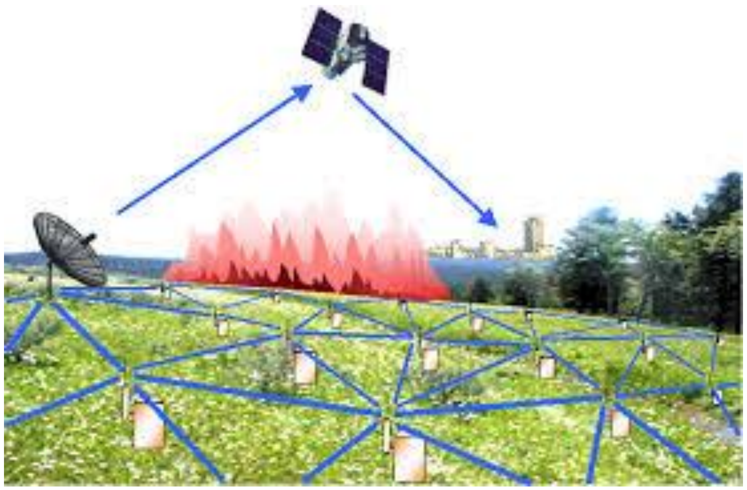
OUTLINE

1. SDPs with Chordal Sparsity

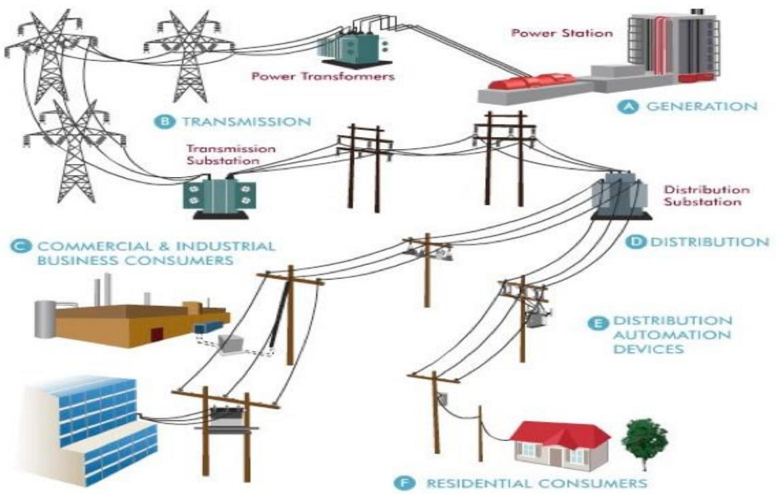
Standard Primal-dual Semidefinite Programs (SDPs)

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

- **Applications:** control theory, power systems, polynomial optimization, combinatorics, operations research, etc.



Control of a networked system
(e.g., via Lyapunov theory)



Optimal power flow problem
(*e.g.*, by dropping a rank constraint)

Picture sources: http://scholar.princeton.edu/ghall/home

1. SDPs with Chordal Sparsity

Standard Primal-dual Semidefinite Programs (SDPs)

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

- **Interior-point solvers:** SeDuMi, SDPA, SDPT3 (suitable for small and medium-sized problems); Modelling package: YALMIP, CVX;
- **Nonlinear SDPs**: using penalty methods; PENNON (PENLAB), Michal Kocvara and Michael Stingl, 2003;
- **Large-scale cases:** it is important to exploit the inherent structure of the instances (De Klerk, 2010):
  - ➢ Low Rank
  - ➢ 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, Andersen, and Vandenberghe, 2014.

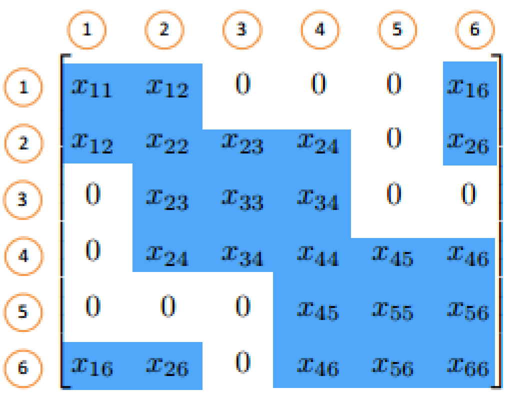
1. SDPs with Chordal Sparsity

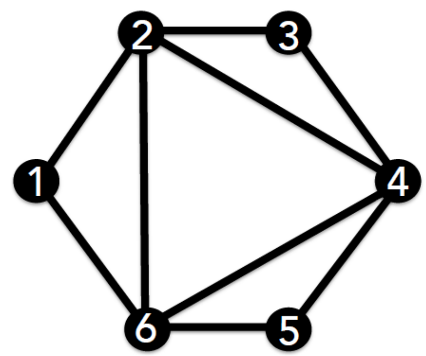
Sparsity Pattern of Matrices

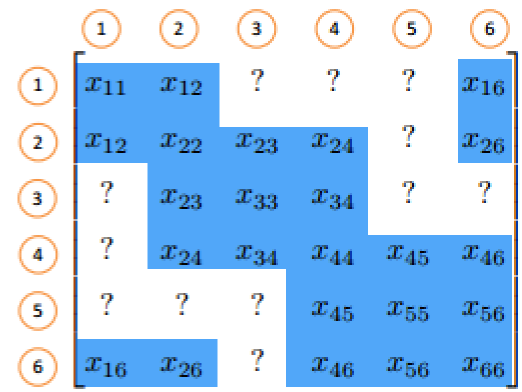
$$\min_{X} \ \langle C, X\rangle \quad \text{subject to} \ \mathcal{A}(X) = b, \ X \in \mathbb{S}^n_+, \qquad \xleftrightarrow{\text{Dual}} \qquad \max_{y,Z} \ \langle b, y\rangle \quad \text{subject to} \ \mathcal{A}^*(y) + Z = C, \ Z \in \mathbb{S}^n_+.$$

- **Sparse matrices**

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







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

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

$\mathbb{S}^n(\mathcal{E}, ?)$ = the set of $n \times n$ partial symmetric matrices with elements defined on $\mathcal{E}$.

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

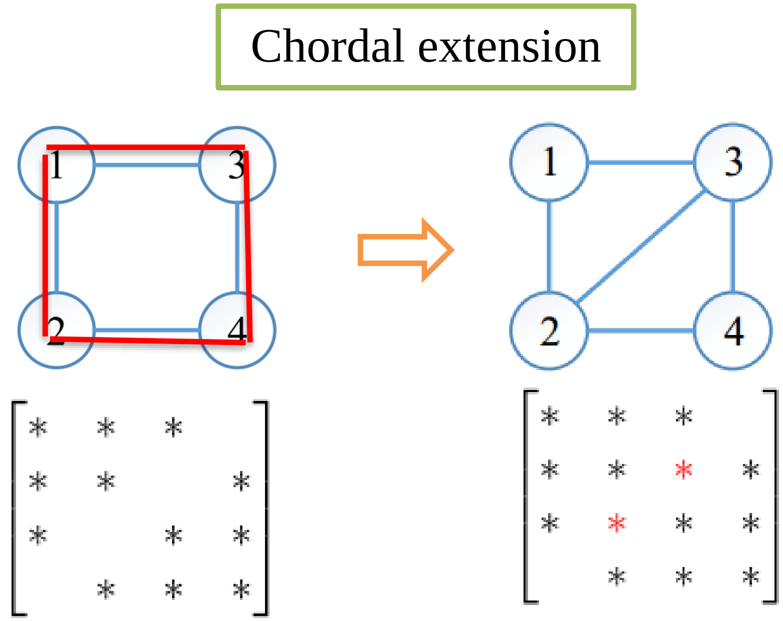
$\mathbb{S}^n_+(\mathcal{E}, ?)$ and $\mathbb{S}^n_+(\mathcal{E}, 0)$ are dual cones of each other.

1. SDPs with Chordal Sparsity

Chordal Graph

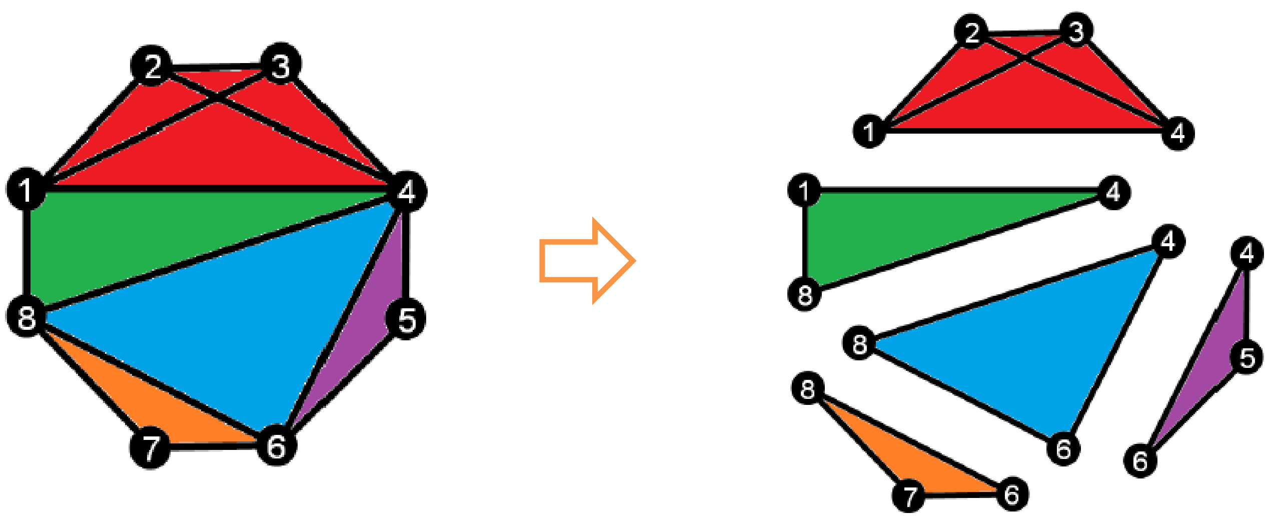
A graph G is *chordal* if every cycle of length at least four has a chord.

- Any non-chordal graph can be chordal extended;



A chordal graph can be decomposed into its maximal cliques $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_p\}$.

- Cliques in a graph are maximal complete subgraphs



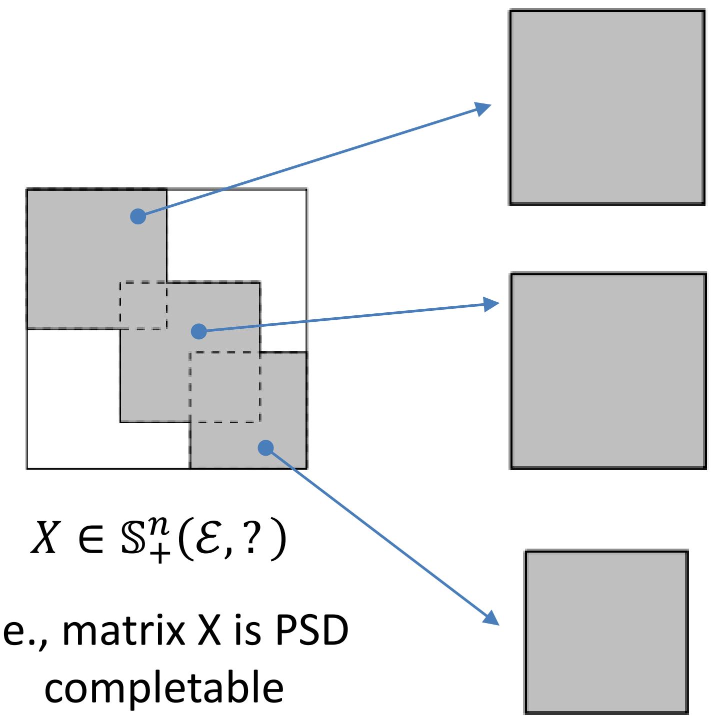
1. SDPs with Chordal Sparsity

Clique Decomposition

Given a choral graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a set of maximal cliques $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_p$

Grone's Theorem:

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



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

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

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

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

i.e., matrix X is PSD completable

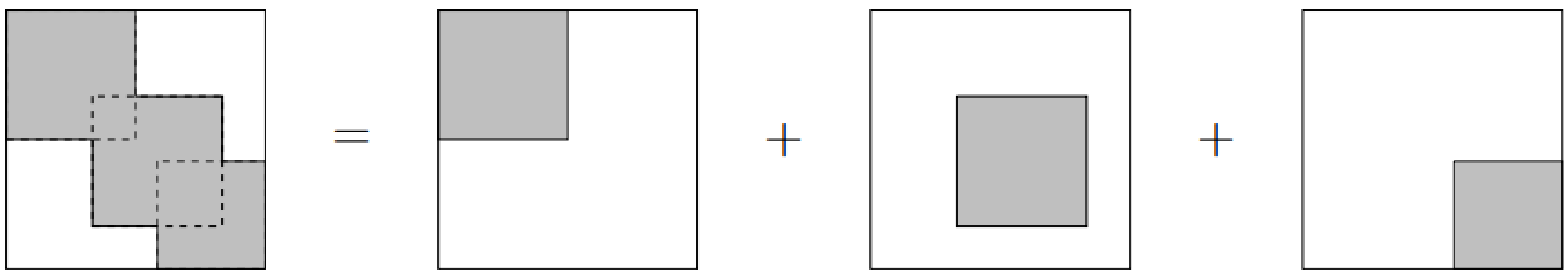
1. SDPs with Chordal Sparsity

Clique Decomposition

Given a choral graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a set of maximal cliques $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_p$

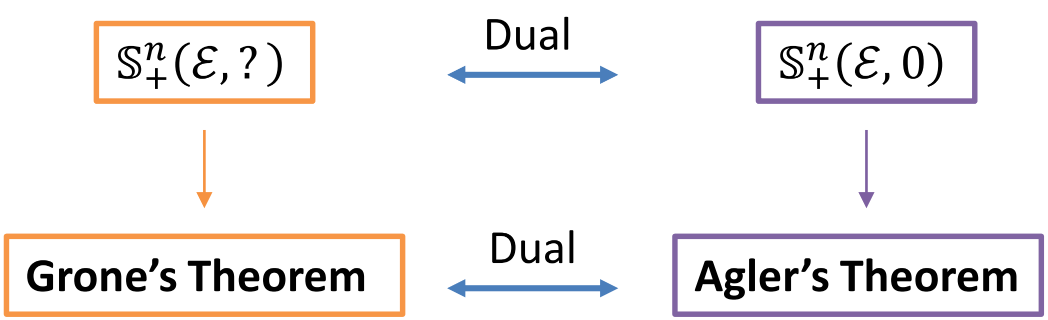
Agler's Theorem:

$$X \in \mathbb{S}^n_+(\mathcal{E}, 0) \text{ if and only if there exists } M_k \in \mathbb{S}^n_+(\mathcal{C}_k) \text{ such that } X = \sum_{k=1}^{p} M_k\,.$$



$X \in \mathbb{S}^n_+(\mathcal{E}, 0)$ $\quad$ $M_1 \in \mathbb{S}^n_+(\mathcal{C}_1)$ $\quad$ $M_2 \in \mathbb{S}^n_+(\mathcal{C}_2)$ $\quad$ $M_3 \in \mathbb{S}^n_+(\mathcal{C}_3)$

Sparse Cone Decomposition (chordal)



Topics in this talk

- ✓ ADMM for primal and dual SDPs;
- ✓ ADMM for the homogeneous self-dual embedding;
- ✓ CDCS: Cone Decomposition Conic Solver.

1. SDPs with Chordal Sparsity

Alternating Direction Method of Multipliers (ADMM)

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

Augmented Lagrangian

$$L_\rho(x, y, z) = f(x) + g(y) + \frac{\rho}{2} \left\| Ax + By - c + \frac{1}{\rho} z \right\|^2$$

ADMM steps

Iterations of ADMM:

$$x^{(n+1)} = \arg\min_x L_\rho(x, y^{(n)}, z^{(n)}),$$
→ a) An x-minimization step

$$y^{(n+1)} = \arg\min_y L_\rho(x^{(n+1)}, y, z^{(n)}),$$
→ b) A y-minimization step

$$z^{(n+1)} = z^{(n)} + \rho(Ax^{(n+1)} + By^{(n+1)} - c).$$
→ c) A dual variable update

Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, *3*(1), 1-122.

OUTLINE

1. SDPs with Chordal Sparsity
2. ADMM for Primal and Dual Sparse SDPs
3. ADMM for the Homogeneous Self-dual Embedding
4. CDCS: Cone Decomposition Conic Solver
5. Conclusion

2. ADMM for Primal and Dual Sparse SDPs

Aggregate sparsity pattern of matrices

A union of patterns of C, A_1, A_2

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

$$\begin{bmatrix} * & * & 0 \\ * & * & * \\ 0 & * & * \end{bmatrix}$$

Primal

$$\begin{aligned} \min_{X} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_1, X \rangle = b_1, \\ & \langle A_2, X \rangle = b_2, \\ & X \in \mathbb{S}^3_+. \end{aligned}$$

Dual

$$\begin{aligned} \max_{y_1, y_2} \quad & b_1 y_1 + b_2 y_2 \\ \text{subject to} \quad & y_1 A_1 + y_2 A_2 + Z = C, \\ & Z \in \mathbb{S}^3_+. \end{aligned}$$

$$X \in \begin{bmatrix} * & * & ? \\ * & * & * \\ ? & * & * \end{bmatrix}$$

Patterns of feasible solutions

$$Z \in \begin{bmatrix} * & * & 0 \\ * & * & * \\ 0 & * & * \end{bmatrix}$$

$$X \in \mathbb{S}^3_+(\mathcal{E}, ?)$$

Cone replacement

$$Z \in \mathbb{S}^3_+(\mathcal{E}, 0)$$

Chordal Decomposition

2. ADMM for Primal and Dual Sparse SDPs

Cone Decomposition of Sparse SDPs

Primal

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

Dual ⟷

Dual

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

Aggregate sparsity pattern $\mathcal{E}$ is union of patterns of C, $A_1, \ldots, A_\mathrm{m}$

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

Cone (chordal) Decomposition

$$\min_{X} \quad \langle C, X\rangle \quad \text{subject to} \quad \mathcal{A}(X) = b, \quad E_{\mathcal{C}_k} X E_{\mathcal{C}_k}^T \in \mathbb{S}^{|\mathcal{C}_k|}_+, \ k = 1, \ldots, p.$$

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

- ✓ A big sparse PSD cone is equivalently replaced by a set of coupled small PSD cones;
- ✓ Our idea: introduce additional variables to decouple the coupling constraints.

2. ADMM for Primal and Dual Sparse SDPs

ADMM for primal SDPs

$$\begin{aligned}\min_{x} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & \mathrm{mat}(H_k x) \in \mathbb{S}_+^{|\mathcal{C}_k|}, k = 1, \ldots, p.,\end{aligned}$$

$$\begin{aligned}\min_{x, x_1, \ldots, x_p} \quad & \langle c, x \rangle \\ \text{subject to} \quad & Ax = b, \\ & x_k = H_k x \\ & x_k \in \mathcal{S}_k, \qquad k = 1, \ldots, p.\end{aligned}$$

Consensus

- Reformulate using indicator functions

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

Function: $g(z)$

Function: $f(x)$

- Augmented Lagrangian

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

- Regroup the variables

$$\begin{aligned}\mathcal{X} &:= \{x\}, \\ \mathcal{Y} &:= \{x_1, \ldots, x_p\}, \\ \mathcal{Z} &:= \{\lambda_1, \ldots, \lambda_p\}.\end{aligned}$$

2. ADMM for Primal and Dual Sparse SDPs

ADMM for primal SDPs

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & \text{mat}(H_k x) \in \mathbb{S}_+^{|\mathcal{C}_k|}, k = 1, \ldots, p., \end{aligned}$$

⇨

$$\begin{aligned} \min_{x, x_1, \ldots, x_p} \quad & \langle c, x \rangle \\ \text{subject to} \quad & Ax = b, \\ & \boxed{x_k = H_k x} \qquad \text{Consensus} \\ & x_k \in \mathcal{S}_k, \qquad k = 1, \ldots, p. \end{aligned}$$

- 1) Minimization over block X

$$\begin{aligned} \min_{x} \quad & \langle c, x \rangle + \frac{\rho}{2} \sum_{k=1}^{p} \left\| x_k^{(n)} - H_k x + \frac{1}{\rho} \lambda_k^{(n)} \right\|^2 \\ \text{subject to} \quad & Ax = b. \end{aligned}$$

QP with linear constraint (Projections onto a linear subspace)

- 2) Minimization over block Y

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

Projections onto small PSD cones; Can be computed in parallel.

- 3) Update multipliers

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

2. ADMM for Primal and Dual Sparse SDPs

ADMM for dual SDPs

$$\begin{aligned} \min_{y, z_k} \quad & -\langle b, y\rangle \\ \text{subject to} \quad & A^T y + \sum_{k=1}^{p} H_k^T z_k = c, \\ & z_k \in \mathcal{S}_k, \qquad k = 1, \ldots, p. \end{aligned}$$

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

Consensus

- Reformulate using indicator functions

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

- Augmented Lagrangian

$$\begin{aligned} \mathcal{L} := & -\langle b, y\rangle + \delta_0\left(c - A^T y - \sum_{k=1}^{p} H_k^T v_k\right) \\ & + \sum_{k=1}^{p}\left[\delta_{\mathcal{S}_k}(z_k) + \frac{\rho}{2}\left\| z_k - v_k + \frac{1}{\rho}\lambda_k \right\|^2\right], \end{aligned}$$

$$\begin{aligned} \mathcal{X} &:= \{y, v_1, \ldots, v_p\}, \\ \mathcal{Y} &:= \{z_1, \ldots, z_p\}, \\ \mathcal{Z} &:= \{\lambda_1, \ldots, \lambda_p\}. \end{aligned}$$

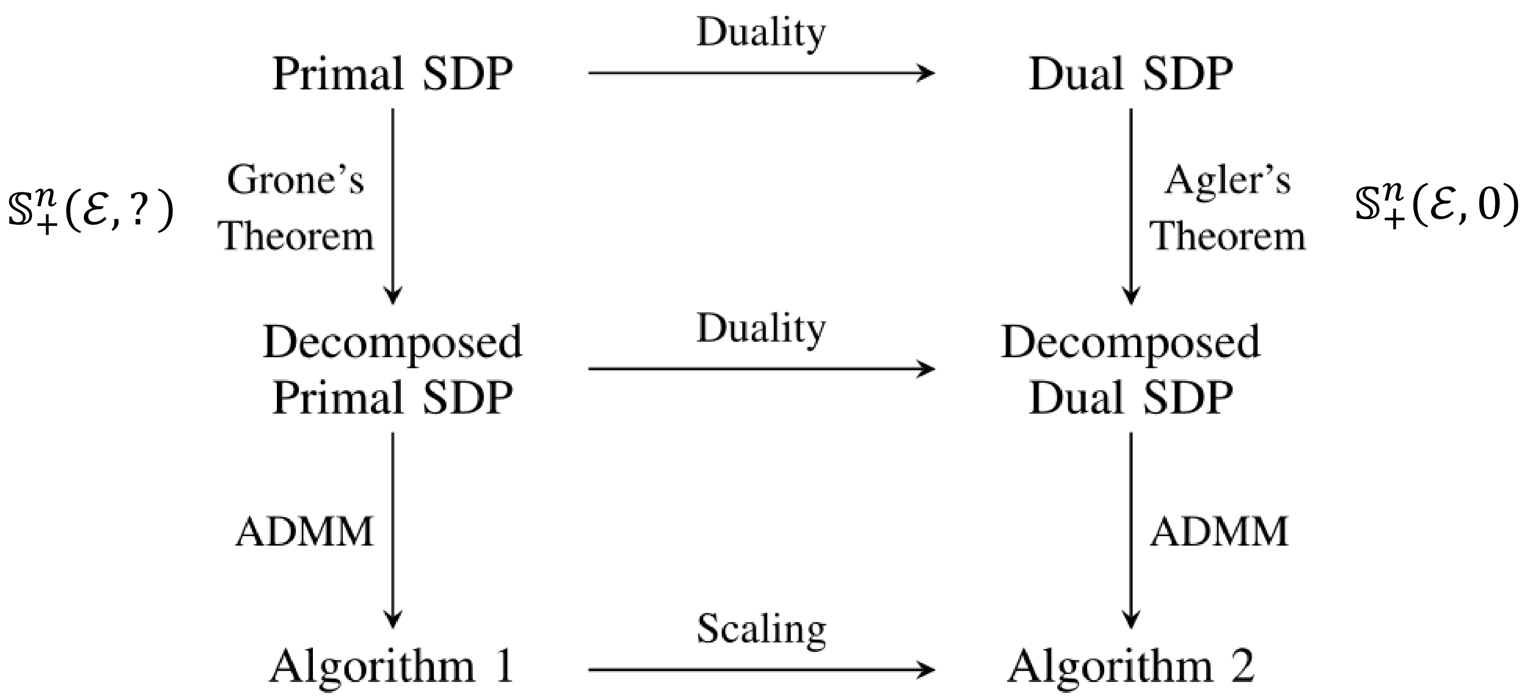
✓ **QP with linear constraints**

✓ **Projections in parallel**

ADMM steps in the dual form are scaled versions of those in the primal form !

2. ADMM for Primal and Dual Sparse SDPs

The Big Picture



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.

OUTLINE

1 SDPs with Chordal Sparsity

2 ADMM for Primal and Dual Sparse SDPs

3 **ADMM for the Homogeneous Self-dual Embedding**

4 CDCS: Cone Decomposition Conic Solver

5 Conclusion

3. ADMM for the Homogenous Self-dual Embedding

KKT condition

Primal

$$\begin{aligned} \min_{x, x_1, \dots, x_p} \quad & \langle c, x \rangle \\ \text{subject to} \quad & Ax = b, \\ & x_k = H_k x \\ & x_k \in \mathcal{S}_k, \qquad k = 1, \dots, p. \end{aligned}$$

Dual

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

- Notational simplicity

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

- KKT conditions
  - Primal feasible
  $$\begin{aligned} Ax^* - r^* = b, \qquad & r^* = 0, \\ s^* + w^* = Hx^*, \quad & w^* = 0, \quad s^* \in \mathcal{S}. \end{aligned}$$
  - Dual feasible
  $$\begin{aligned} A^T y^* + H^T v^* + h^* = c, \quad & h^* = 0, \\ z^* - v^* = 0, \quad & z^* \in \mathcal{S}. \end{aligned}$$
  - Zero-duality gap
  $$c^T x^* - b^T y^* = 0.$$

3. ADMM for the Homogenous Self-dual Embedding

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: two non-negative and complementary variables

- Notational simplicity

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

$$\mathcal{K} := \mathbb{R}^{n^2} \times \mathcal{S} \times \mathbb{R}^m \times \mathbb{R}^{n_d} \times \mathbb{R}_+ \qquad \mathcal{S} := \mathcal{S}_1 \times \cdots \times \mathcal{S}_p$$

- Feasibility problem

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

✓ The big sparse PSD cone has already been equivalently replaced by a set of coupled small PSD cones;

3. ADMM for the Homogenous Self-dual Embedding

ADMM algorithm

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

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

$$\begin{aligned} \hat{u}^{k+1} &= (I + Q)^{-1}(u^k + v^k), \\ u^{k+1} &= \Pi_{\mathcal{K}}(\hat{u}^{k+1} - v^k), \\ v^{k+1} &= v^k - \hat{u}^{k+1} + u^{k+1}. \end{aligned}$$

→ Projection onto a subspace

→ Projection onto cones (smaller dimension)

$$\mathcal{K} := \mathbb{R}^{n^2} \times \mathcal{S} \times \mathbb{R}^m \times \mathbb{R}^{n_d} \times \mathbb{R}_+$$

$$\mathcal{S} := \mathcal{S}_1 \times \cdots \times \mathcal{S}_p$$

Q is highly structured and sparse

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

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

✓ 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., and Parikh, Nealand Boyd, S. (2016b). Conic optimization via operator splitting and homogeneous self-dual embedding. Journal of Optimization Theory and Applications, 169(3), 1042– 1068

OUTLINE

4. CDCS: Cone Decomposition Conic Solver

CDCS

- 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 in accordance with SeDuMi;
- Works with latest Yalmip release.

Syntax:

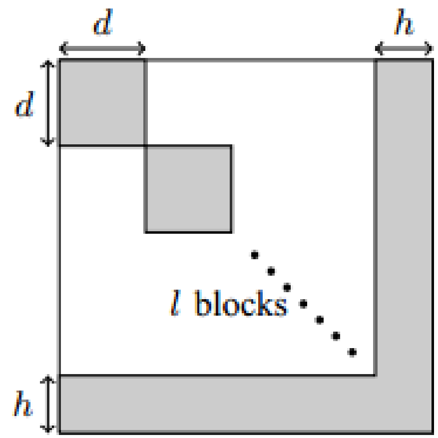
```
[x,y,z,info] = cdcs(At,b,c,K,opts);
```

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

4. 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 (interior-point solver)
- SCS (first-order solver)
- sparseCoLO (preprocessor) +SeDuMi

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

Numerical Results

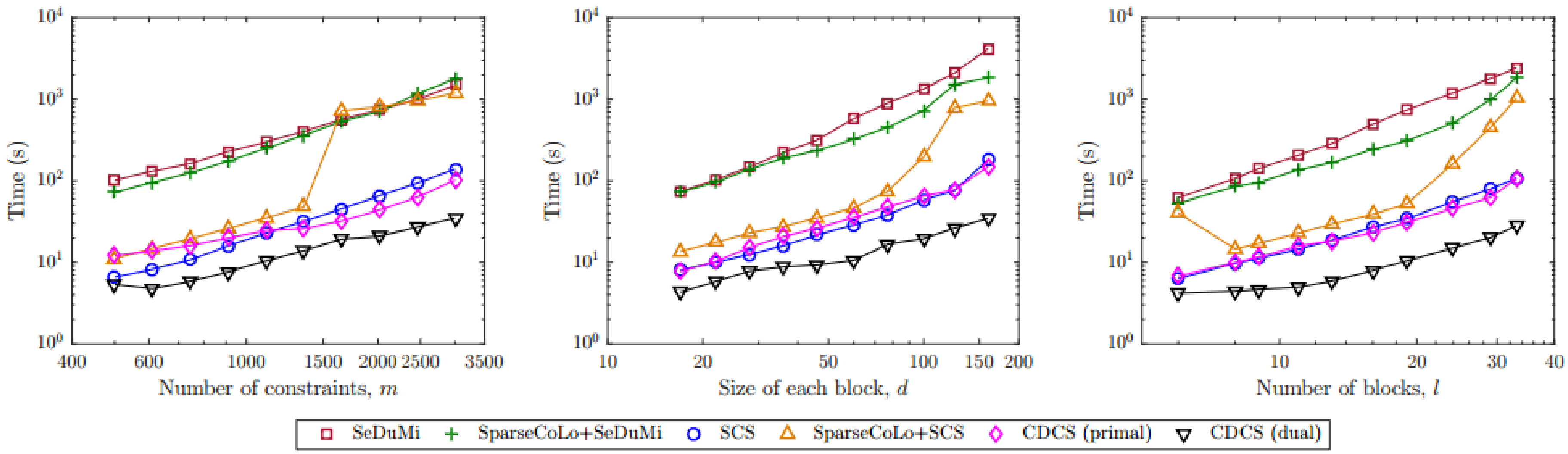


Fig. 3. CPU time for SDPs with block-arrow patterns. Left to right: varying number of constraints; varying number of blocks; varying block size.

4. CDCS: Cone Decomposition Conic Solver

Benchmark problems in SDPLIB [2]

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

[2] Borchers, Brian. "SDPLIB 1.2, a library of semidefinite programming test problems." *Optimization Methods and Software* 11.1-4 (1999): 683-690.

4. 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

4. 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

- **maxG32**: original cone size **2000**; after chordal decomposition, maximal size **60**;
- **qpG11**: original cone size **1600**; after chordal decomposition, maximal size **24**;

4. CDCS: Cone Decomposition Conic Solver

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

4. 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
small and medium size	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}
large-scale and sparse	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

✓ Work with smaller semidefinite cones for large-scale sparse problems

- Our codes are currently written in MATLAB
- SCS is implemented in C.

Outline

1. SDPs with Chordal Sparsity
2. ADMM for Primal and Dual Sparse SDPs
3. ADMM for the Homogeneous Self-dual Embedding
4. CDCS: Cone Decomposition Conic Solver
5. Conclusion

5. Conclusion

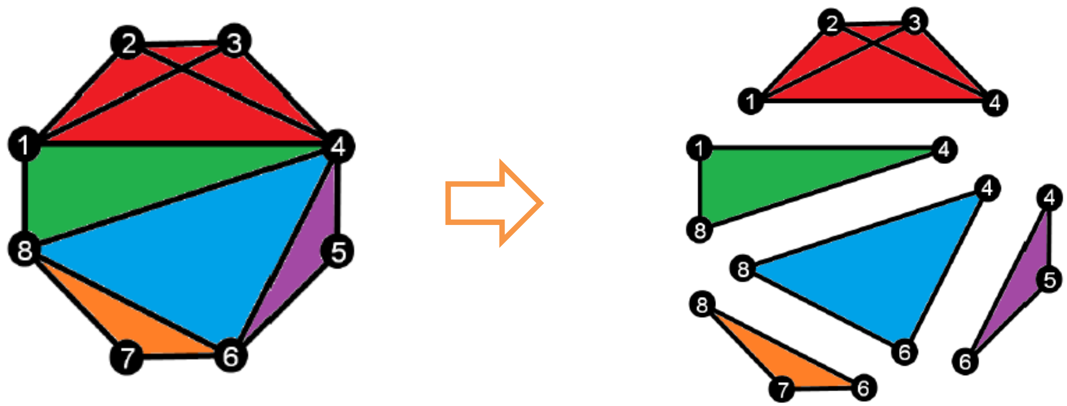
Summary

$$\min_X \ \langle C, X\rangle \quad \text{subject to} \quad \mathcal{A}(X) = b, \ X \in \mathbb{S}^n_+, \qquad \xleftrightarrow{\text{Dual}} \qquad \max_{y,Z} \ \langle b, y\rangle \quad \text{subject to} \quad \mathcal{A}^*(y) + Z = C, \ Z \in \mathbb{S}^n_+.$$

- Large-scale and sparse instances

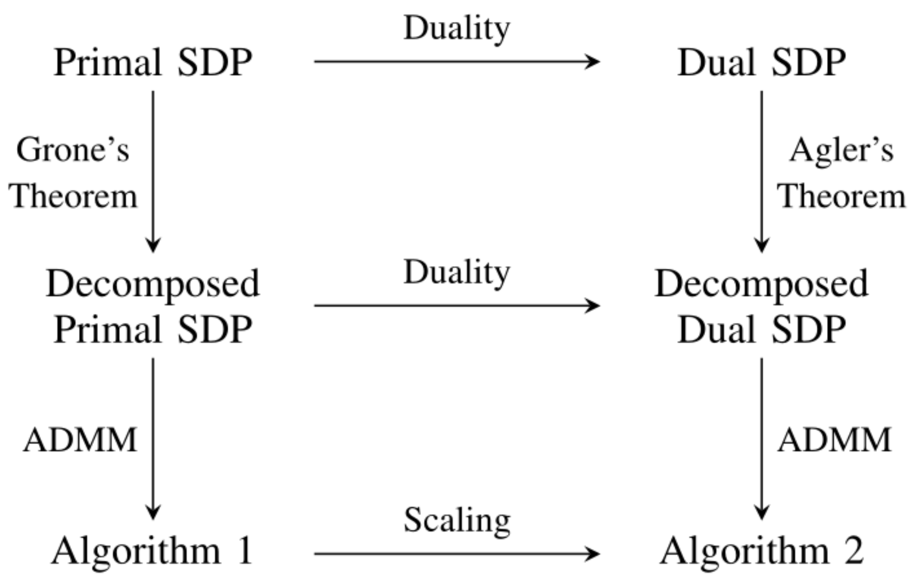
Modelling Sparsity — **Chordal graphs**, leading to sparse PSD cone decompositions (Grone's and Agler's theorems);

Fast Computation — **ADMM**, dealing with the small and coupled cone constraints (Alternating projections onto a linear subspace and small cones)



5. Conclusion

Summary



- Introduced a conversion framework for sparse SDPs suitable for first-order methods;
- Developed efficient ADMM algorithms
  - ✓ Primal and dual standard form;
  - ✓ The homogeneous self-dual embedding;
- CDCS: Download from https://github.com/OxfordControl/CDCS

Ongoing work

- Develop ADMM algorithms for sparse SDPs arising in Sum-of-Squares (SOS).
- Applications in networked systems and power systems.

Thank you for your attention!
Q & A

1. Zheng, Yang, Giovanni Fantuzzi, Antonis Papachristodoulou, Paul Goulart, and Andrew Wynn. "Fast ADMM for Semidefinite Programs with Chordal Sparsity." *arXiv preprint arXiv:1609.06068* (2016).
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.