Chordal Decomposition in Semidefinite Programming: Trading Stability for Scalability

Arvind U. Raghunathan
Mitsubishi Electric Research Laboratories

Lorenz T. Biegler
Carnegie Mellon University

2019 SIAM - Computational Science & Engineering
26 February, 2019

Outline

Problem Description

Conversion Approach

Primal Degeneracy

Nondegenerate Formulation

Semidefinite Program (SDP)

$$\begin{aligned} \min_{\boldsymbol{X} \in \mathbb{S}^n} \quad & \boldsymbol{A}_0 \bullet \boldsymbol{X} \\ \text{s.t.} \quad & \boldsymbol{A}_p \bullet X = \boldsymbol{b}_p \; \forall \; p = 1, \ldots, m \\ & \boldsymbol{X} \succeq 0 \end{aligned} \qquad \text{(SDP)}$$

- $\mathbb{S}^n$ - set of symmetric $n \times n$ matrices
- $\boldsymbol{A}_p \in \mathbb{S}^n$
- $\boldsymbol{X} \succeq 0$ - $\boldsymbol{X}$ is positive semidefinite
- $\bullet$ - trace inner product

Semidefinite Program (SDP)

$$
\begin{aligned}
\min_{\boldsymbol{X} \in \mathbb{S}^n} \quad & \boldsymbol{A}_0 \bullet \boldsymbol{X} \\
\text{s.t.} \quad & \boldsymbol{A}_p \bullet X = \boldsymbol{b}_p \; \forall \, p = 1, \ldots, m \\
& \boldsymbol{X} \succeq 0
\end{aligned}
\qquad \text{(SDP)}
$$

Applications

- Combinatorial Optimization (relaxations)
- Controls Design - LMIs
- Polynomial Optimization
- Optimal Power Flow relaxations

Solution of SDPs

- Convex program
  - Intersection of semidefinite cone and affine space
- Interior Point Methods (IPMs)
- Implementations: SDPA, SDPT3, SeDuMi, Mosek

Complexity of Step Computation - $O(mn^3 + m^2n^2)$

Computes a matrix $\boldsymbol{M}$ where,

$\boldsymbol{M}_{[ij]} = \boldsymbol{A}_i G \bullet \boldsymbol{A}_j G$

G is an SDP direction-specific, iteration dependent matrix

Sparsity in Problem Data

- Define graph - G(N, E)

$$\mathsf{N} := \{1, \ldots, n\}$$
$$\mathsf{E} := \{(i,j) \mid (i,j) - \text{th entry of some data matrix is non-zero}\}$$

- $\boldsymbol{A}_p$ - sparse $\implies |\mathsf{E}| << n^2$
- Trace inner product has few terms

$$\boldsymbol{A}_p \bullet \boldsymbol{X} = \sum_{i,j} \boldsymbol{A}_{p[ij]} \boldsymbol{X}_{[ij]} = \sum_{i,j \in \mathsf{E}} \boldsymbol{A}_{p[ij]} \boldsymbol{X}_{[ij]}$$

- $\boldsymbol{X}_{[ij]}$ for $(i,j) \in \mathsf{E}$ are the **relevant entries**

Can (SDP) computations be restricted to $\boldsymbol{X}_{[ij]}$ for $(i,j) \in \mathsf{E}$?

Outline

Problem Description

Conversion Approach

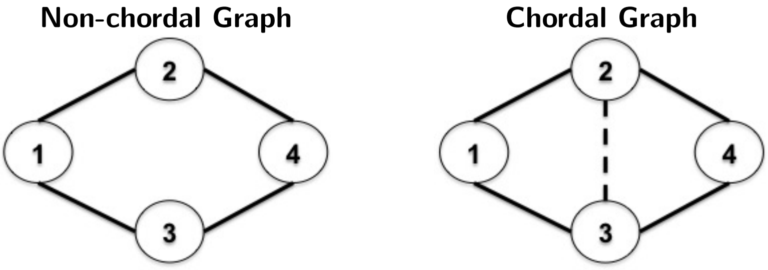
Primal Degeneracy

Nondegenerate Formulation

Chordal Graphs

Chordal Graph

$G(N, E)$ - no cycles of length ≥ 4.



$$F = E \cup \{(2, 3)\}$$

$G'(N, F)$ - **Chordal Extension of** $G(N, E)$**.**

Chordal Graphs (contd.)

Clique - $\mathsf{C} \subset \mathsf{N}$

C is a **clique** if $(i, j) \in \mathsf{E}$ for all $i, j \in \mathsf{C}$.

Maximal Clique

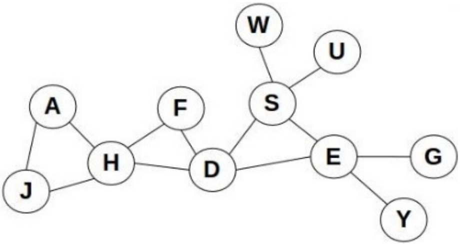
C is **maximal** if there does not exist clique $\mathsf{C}' \supset \mathsf{C}$ in $\mathsf{G}(\mathsf{N}, \mathsf{E})$.

Clique Tree $\mathcal{T}(\mathcal{N}, \mathcal{E})$

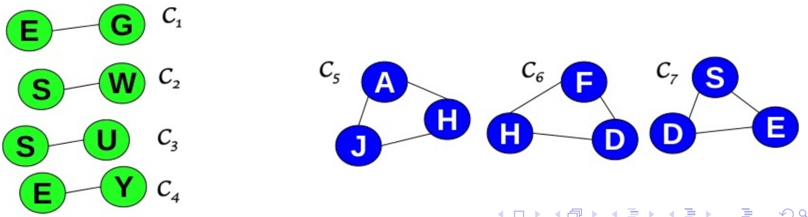
For a **chordal graph**, the maximal cliques can be arranged as a tree, called **clique tree**,
$\mathcal{T}(\mathcal{N}, \mathcal{E})$ in which $\mathcal{N} = \{\mathsf{C}_1, \ldots, \mathsf{C}_\ell\}$ and $(\mathsf{C}_s, \mathsf{C}_t) \in \mathcal{E}$ are edges between the cliques.

Chordal Graphs (contd.)

Chordal Graph

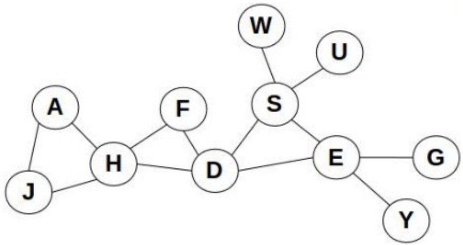


Maximal Cliques

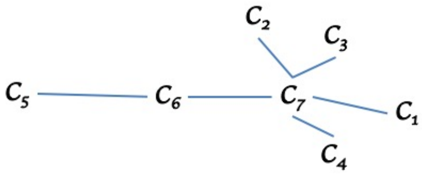


Chordal Graphs (contd.)

Chordal Graph



Clique Tree

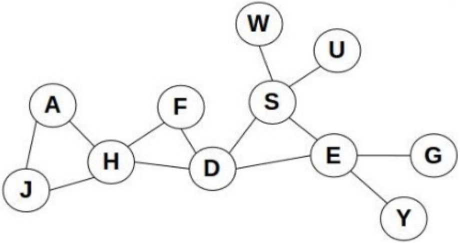


Maximal Clique Decomposition in SDPs

Grone, Johnson, Sá, Wolkowicz (1984)

Chordal Graph

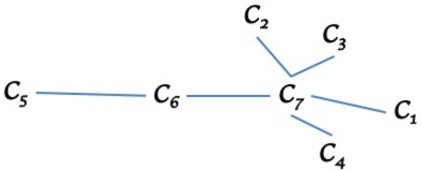
$$X \succeq 0$$



$$\Updownarrow$$

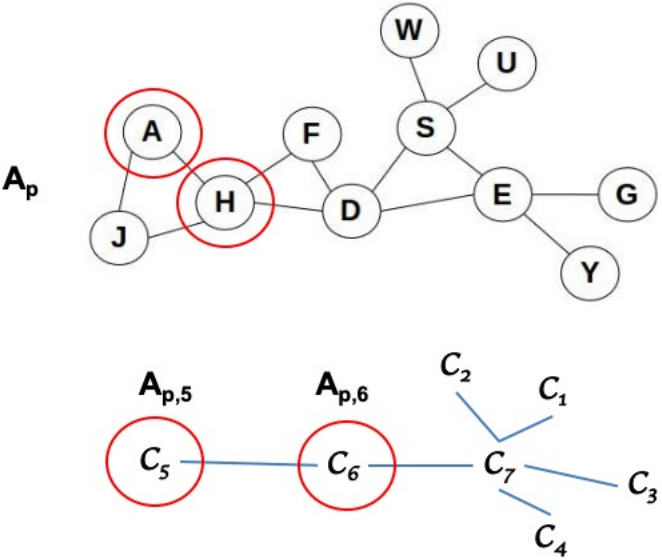
Clique Tree

$$X_{C_l C_l} \succeq 0$$



Exploiting Sparsity in SDPs

Fukuda, Kojima, Murota and Nakata (2000)



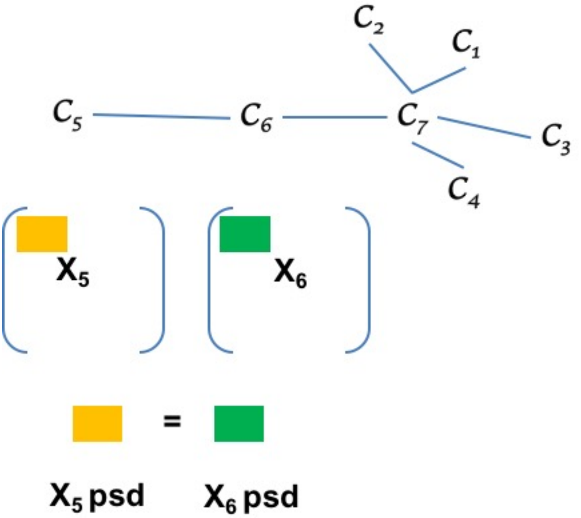
Exploiting Sparsity in SDPs

Fukuda, Kojima, Murota and Nakata (2000)

$$
\begin{aligned}
\min_{\boldsymbol{X}_s \in \mathbb{S}^{|\mathsf{C}_s|}} \quad & \sum_{s=1}^{\ell} \boldsymbol{A}_{s,0} \bullet \boldsymbol{X}_s \\
\text{s.t.} \quad & \sum_{s=1}^{\ell} \boldsymbol{A}_{s,p} \bullet \boldsymbol{X}_s = \boldsymbol{b}_p \qquad \forall\, p = 1, \ldots, m \\
& E_{s,ij} \bullet \boldsymbol{X}_s = E_{t,ij} \bullet \boldsymbol{X}_t \quad \forall\, i \le j, i, j \in \mathsf{C}_{st}, \\
& \qquad\qquad (s,t) \in \mathcal{E} \\
& \boldsymbol{X}_s \succeq 0 \qquad \forall\, s = 1, \ldots, \ell.
\end{aligned} \qquad \text{(SDPconv)}
$$

- Smaller semidefinite matrices
- Additional equality constraints - equate duplicated entries in cliques
- Constraints are sparse

Conversion Approach



Outline

Problem Description

Conversion Approach

Primal Degeneracy

Nondegenerate Formulation

Analogy with Linear Programming

Original

$$\min_{x \in \mathbb{R}^3} c^T x$$
s.t. $x \geq 0$

$\rightarrow$

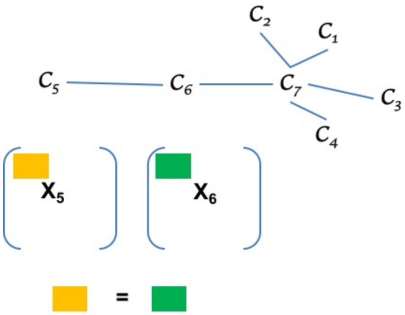
"Conversion"

$$\min_{x_i} c_1^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + c_2^T \begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$$
s.t. $x_2 = x_3$
$$x_i \geq 0$$

- Suppose at "Conversion" optimum, $x_2^\star = x_3^\star = 0$
- Loss of Linear Independence of constraint gradients
- Multiplicity of dual multipliers
- Schur complement matrix severely ill-conditioned

Intuition for Degeneracy

- $X_5, X_6 \succeq 0 \implies$ **psd of overlapping sub-matrix**
- Semidefinite constraint **imposed on both minors**
- rank$(X^*) <$ (size of overlap) then, **submatrices lose rank**
- v is 0-eigenvector subblock $\implies w = \begin{bmatrix} v \\ 0 \end{bmatrix}$ is 0-eigenvector of X_5, X_6
- ww^T ies in range of coupling constraints



Loss of Linear Independence*

- Suppose (SDP) has solution $\boldsymbol{X}^*$
- Then, $\boldsymbol{X}_s^*$ solves (SDPconv) with $\boldsymbol{X}_s^* = \boldsymbol{X}^*_{\mathsf{C}_s\mathsf{C}_s}$
- Assume, $\text{rank}(X^*) < |\mathsf{C}_s \cap \mathsf{C}_t|$ for some s, t

Theorem

(SDPconv) **fails to satisfy Linear Independence Contraint Qualification (LICQ)** at the solution.

Remarks

- Typically interested in rank-1 solns of (SDP)
- Cliques sharing ≥ 1 edge $\implies$ LICQ fails for (SDPConv)

* - A.U.R and A. Knyazev, Degeneracy in maximal clique decomposition for Semidefinite Programs, IEEE American Control Conference, 5605-5611 (2016).

Dual Multiplicity*

- Suppose (SDP) has solution $\boldsymbol{X}^*$
- Then, $\boldsymbol{X}_s^* = \boldsymbol{X}_{\mathsf{C}_s\mathsf{C}_s}^*$ solves (SDPconv)
- Assume, $\text{rank}(X^*) < |\mathsf{C}_s \cap \mathsf{C}_t|$ for some s, t

Theorem

(SDPconv) **has multiple dual solutions** and possibly one that fails **strict complementarity**.

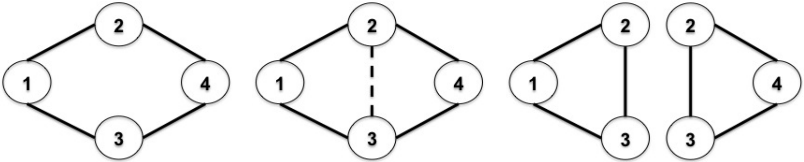
Remarks

- Loss of LICQ $\implies$ Multiple duals
- Loss of strict complementarity can lead to ill-conditioning.

* - A.U.R and A. Knyazev, Degeneracy in maximal clique decomposition for Semidefinite Programs, IEEE American Control Conference, 5605-5611 (2016).

MAXCUT Example

$$A_0 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}, A_p = e_p e_p^T, b_p = 1 \; \forall \; p = 1, \ldots, 4.$$



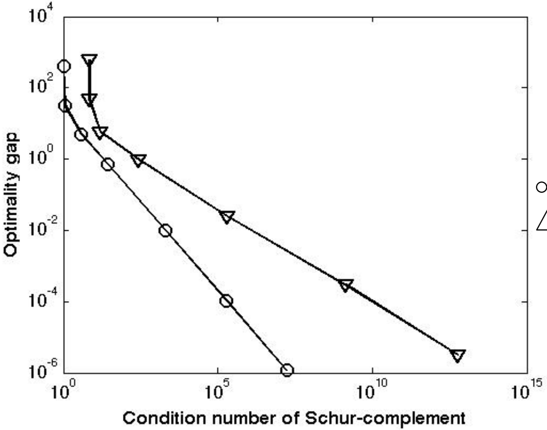
(a) G(N, E)

(b) G(N, F)

(c) $C_1 = \{2, 3, 1\}, C_2 = \{2, 3, 4\}$

MAXCUT Example (contd.)

- (SDP) has **rank-1 solution, non-degenerate**
- (SDPconv) - **fails LICQ, multiple dual solutions**
- **Cond#(SDPconv)** $\approx$ **Cond#(SDP)**^{2}



○ - (SDP),
△ - (SDPConv)

SDPLIB - MaxCut

SparseCoLO
http://www.is.titech.ac.jp/ kojima/SparseCoLO/SparseCoLO.htm

SeDuMi

	(SDP)		(SDPConv)	
	Status	Cond. #	Status	Cond. #
mcp100	Solved	2.08e+07	NumErr	Inf
mcp124-1	Solved	1.50e+07	Solved	Inf
mcp124-2	Solved	2.50e+07	NumErr	Inf
mcp124-3	Solved	3.97e+06	NumErr	Inf
mcp124-4	Solved	1.76e+07	NumErr	Inf
mcp250-1	Solved	4.50e+07	NumErr	Inf

SDPT3

	(SDP)		(SDPConv)	
	Status	Cond. #	Status	Cond. #
mcp100	Solved	2.19e+08	Solved	4.34e+15
mcp124-1	Solved	2.00e+08	Solved	2.48e+15
mcp124-2	Solved	5.37e+08	NumErr	6.59e+15
mcp124-3	Solved	2.59e+07	Solved	7.10e+15
mcp124-4	Solved	2.59e+08	Solved	1.20e+13
mcp250-1	Solved	1.01e+09	Solved	1.45e+17

SDPLIB - arch*

SparseCoLO

http://www.is.titech.ac.jp/ kojima/SparseCoLO/SparseCoLO.htm

SeDuMi

	(SDP)		(SDPConv)	
	Status	Cond. #	Status	Cond. #
arch0	Solved	5.81e+08	NumErr	Inf
arch2	Solved	1.46e+09	NumErr	Inf
arch4	Solved	3.63e+08	Solved	Inf
arch8	Solved	4.25e+09	NumErr	Inf

SDPT3

	(SDP)		(SDPConv)	
	Status	Cond. #	Status	Cond. #
arch0	Solved	2.16e+10	NumErr	1.02e+25
arch2	Solved	2.08e+10	NumErr	1.05e+27
arch4	Solved	2.15e+10	Solved	4.22e+26
arch8	Solved	3.38e+10	NumErr	1.67e+25

Polynomial Optimization

- J. S. Campos and P. Parpasa, Multigrid Approach to SDP Ralaxations of Sparse Polynomial Optimization Problems, SIAM J Optimization, 28(1): 1-29 (2018).

TABLE 4

Condition number of the Schur-complement matrix for the last iteration at the fine level usi SDPT3 and $Multi_{L\geq 2}$ for the nonlinear differential equations.

Differential equation	1	2	3	4	5	6	7	8	9
# $CN_{SDPT3} > CN_{Multi_{L\geq 2}}$	95	99	89	97	94	97	76	85	42
$\text{mean} CN_{SDPT3}/CN_{Multi_{L\geq 2}}$	5e+13	7e+13	6e+07	4e+14	7e+06	3e+12	7e+15	3e+02	7e+01
$\text{min} CN_{SDPT3}/CN_{Multi_{L\geq 2}}$	8e-02	4e+00	1e-06	5e-02	5e-02	9e-02	5e-37	6e-02	6e-04
$\text{max} CN_{SDPT3}/CN_{Multi_{L\geq 2}}$	4e+15	5e+15	5e+09	4e+16	1e+08	9e+13	7e+17	1e+04	6e+03

- IPMs solver fewer problems compared to their approach

Easy to Fix for LP

$$\min_{x_i} c_1^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + c_2^T \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} \quad \text{s.t. } x_2 = x_3, \quad x_i \geq 0$$

$$\rightarrow$$

$$\min_{x_i} c_1^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + c_2^T \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} \quad \text{s.t. } x_2 = x_3, \quad x_1, x_2, x_4 \geq 0$$

Can we do the same for SDP?

Yes

A. U. R and L. T. Biegler, LDL^T Direction Interior Point Method for Semidefinite Programming, SIAM J. Optim., 28(1), 693–734 (2018).

Outline

LDL^T Formulation

$$\begin{aligned} \min_{\boldsymbol{X} \in \mathbb{S}^n} \ & \boldsymbol{A}_0 \bullet \boldsymbol{X} \\ \text{s.t. } & \boldsymbol{A}_p \bullet X = \boldsymbol{b}_p \ \forall \, p = 1, \ldots, m \\ & d_{[i]}(X) \geq 0 \end{aligned} \qquad \text{(SDP-LDL)}$$

R. Fletcher, Semidefinite matrix constraints in optimization, SIAM J. Control. Opt., 23: 493-513 (1985).
H. Y. Benson and R. J. Vanderbei, MPB, 95:. 279–302 (2003).

For any $X \succ 0$, there exists

- **unique** $L(X), D(X)$ such that $X = L(X)D(X)L(X)^T$
- $L(X)$ - unit lower triangular
- $D(X) = \text{Diag}(d_{[1]}(X), \ldots, d_{[n]}(X))$ with $d_{[i]}(X) > 0$

LDL^T Formulation

$$\begin{aligned} \min_{\boldsymbol{X} \in \mathbb{S}^n} \quad & \boldsymbol{A}_0 \bullet \boldsymbol{X} \\ \text{s.t.} \quad & \boldsymbol{A}_p \bullet X = \boldsymbol{b}_p \; \forall \; p = 1, \ldots, m \\ & d_{[i]}(X) \geq 0 \end{aligned} \qquad \text{(SDP-LDL)}$$

- $X \succeq 0$ - linear matrix inequality
- Convex matrix inequality.
- Strictly convex for $X \in \mathbb{S}^n_{++}$

- $d_{[i]}(X) \geq 0$ - nonlinear inequality
- Concave nonlinear inequality.
- Strictly concave for $X \in \mathbb{S}^n_{++}$

Derivatives for $d_{[i]}(X)$?

LDL^T Factorization

$$X = \begin{bmatrix} X_{i-1} & x_i & * \\ x_i^T & X_{[ii]} & * \\ * & * & * \end{bmatrix}$$

$$L = \begin{bmatrix} L_{i-1} & \mathbf{0} & * \\ l_i^T & 1 & * \\ * & * & * \end{bmatrix}$$

$$D = \begin{bmatrix} D_{i-1} & \mathbf{0} & * \\ \mathbf{0} & d_{[i]} & * \\ * & * & * \end{bmatrix}$$

where $X_i, L_i, D_i \in \mathbb{R}^{i \times i}$ are the ith principal minor of X, L, D, respectively and $x_i, l_i \in \mathbb{R}^{i-1}$.

Factorization

- Set $L_{[11]} = 1$, $d_{[1]} = X_{[11]}$
- For all $i > 1$,
  - $l_i = D_{i-1}^{-1} L_{i-1}^{-1} x_i$
  - $d_{[i]} = X_{[ii]} - l_i^T D_{i-1} l_i$
    $= X_{[ii]} - x_i^T X_{i-1}^{-1} x_i$

$d_{[i]}(X)$ is Schur-complement of block X_{i-1} in matrix X_i.

Derivatives for $d_{[i]}(X)$

Note,

- $X_i = L_i D_i L_i^T \implies \det(X_i) = \det(D_i) = \prod_{j=1}^{i} d_{[j]}(X)$
- $d_{[i]}(X) = \frac{\det(D_i)}{\det(D_{i-1})} = \frac{\det(X_i)}{\det(X_{i-1})}$
- $\ln(d_{[i]}(X)) = \ln(\det(X_i)) - \ln(det(X_{i-1}))$
- (SDP-LDL) Barrier: $-\sum_{i=1}^{n} \ln(d_{[i]}) = -\ln(\det(X))$:(SDP) Barrier

$$\nabla_X d_{[i]}(X) = L^{-T} e_i e_i^T L^{-1}$$

$$\nabla_X \ln(\det(X)) = \sum_{i=1}^{n} \nabla_X \ln(d_{[i]}(X)) = \sum_{i=1}^{n} \frac{1}{d_{[i]}} L^{-T} e_i e_i^T L^{-1} = X^{-1}$$

Easily derive higher-order derivatives as well

Barrier Formulation

Barrier Form:

$$\min A_0 \bullet X - \mu \sum_{i=1}^{n} \ln(d_{[i]}(X))$$
$$\text{s.t. } \mathcal{A}(X) = b$$

with $\mathcal{A}(X) = \left[A_1 \bullet X, \ldots, A_m \bullet X\right]^T$.

Stationary Conditions:

$$\begin{aligned} C + \mathcal{A}^*(\lambda) - \sum_{i=1}^{n} z_{[i]} \nabla d_{[i]}(X) &= 0 \\ \mathcal{A}(X) &= b \\ d_{[i]}(X) z_{[i]} &= \mu \; \forall\, i = 1, \ldots, n. \end{aligned}$$

LDL^T Direction

- Newton step on stationary conditions
- Eliminating Δz yand some transformations

$$\begin{aligned} K \circ \widetilde{\Delta X} \quad + \quad \widetilde{\mathcal{A}}^*(\Delta\lambda) &= \quad \widetilde{r}_d \\ \widetilde{\mathcal{A}}(\widetilde{\Delta X}) \quad\quad\quad\quad\quad &= \quad r_p \end{aligned}$$

where

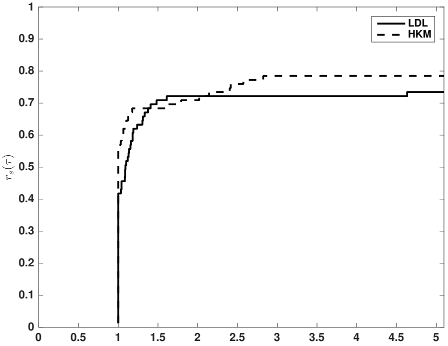
- $\circ$ - element-wise product
- $\widetilde{\Delta X} = L^{-1} \Delta X L^{-T}$
- $\widetilde{\mathcal{A}}(X) = \left[(L^T A_1 L) \bullet X, \ldots, (L^T A_m L) \bullet X\right]^T$
- $K = \begin{bmatrix} z_{[1]} & z_{[2]} & \cdots & z_{[n]} \\ z_{[2]} & z_{[2]} & \cdots & z_{[n]} \\ \vdots & \vdots & \ddots & \vdots \\ z_{[n]} & z_{[n]} & \cdots & z_{[n]} \end{bmatrix} \circ^{-1} \begin{bmatrix} d_{[1]} & d_{[1]} & \cdots & d_{[1]} \\ d_{[1]} & d_{[2]} & \cdots & d_{[2]} \\ \vdots & \vdots & \ddots & \vdots \\ d_{[1]} & d_{[2]} & \cdots & d_{[n]} \end{bmatrix}$

Comparison on SDPLIB

solved

ϵ	Barrier	LDL^T	HKM	HKMPC	SeDuMi	SDPT3
10^{-6}	46	58	62	64	45	65
10^{-5}	54	67	66	78	59	71
10^{-4}	56	74	73	78	70	74

iterations

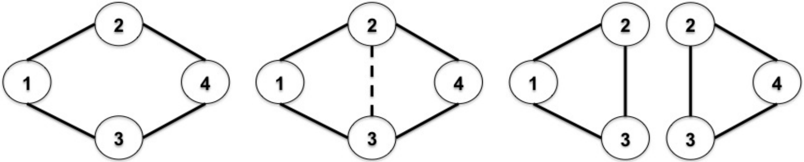


Nondegenerate Formulation

$$\min_{X_1, X_2} C_1 \bullet X_1 + C_2 \bullet X_2$$

$$\text{s.t.} \begin{bmatrix} * & * & * \\ * & \circ & \triangle \\ * & \triangle & \diamond \end{bmatrix} = \begin{bmatrix} \circ & \triangle & * \\ \triangle & \diamond & * \\ * & * & * \end{bmatrix}$$

$$X_1, X_2 \succeq 0$$



(d) $G(N, E)$ (e) $G(N, F)$ (f) $C_1 = \{1, 2, 3\}, C_2 = \{2, 3, 4\}$

Nondegenerate Formulation

$$\min_{X_1, X_2} C_1 \bullet X_1 + C_2 \bullet X_2$$

$$\text{s.t.} \begin{bmatrix} * & * & * \\ * & \circ & \triangle \\ * & \triangle & \diamond \end{bmatrix} = \begin{bmatrix} \circ & \triangle & * \\ \triangle & \diamond & * \\ * & * & * \end{bmatrix}$$

$$d_{[i]}(X_1) \geq 0 \text{ for } i = 1, 2, 3$$

$$d_{[3]}(X_2) \geq 0$$

- $d_{[i]}$ - Schur complement of X_{i-1} in X_i
- Ensure initial point satisfies overlapping constraints
- Common stepsize for all blocks ensures constraints hold

Sufficient to ensure X_1, X_2 are positive definite

Running Intersection Property

Ordering of $\mathcal{N} = (\mathsf{C}_1, \ldots, \mathsf{C}_\ell)$ such that

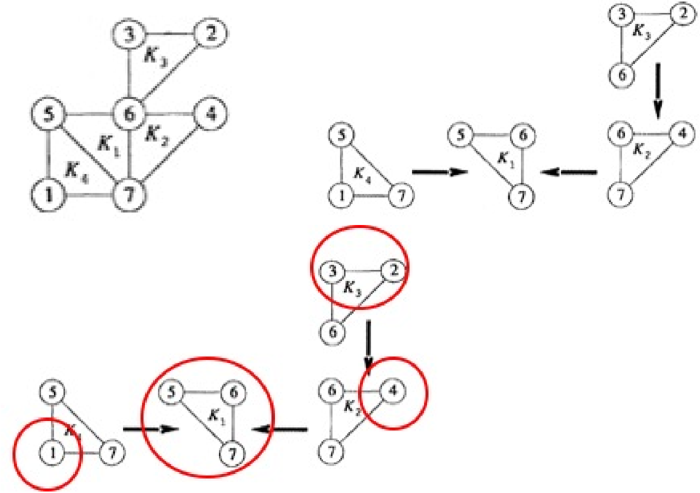
- For each j, $\exists i \leq j-1 : \mathsf{C}_j \cap (\mathsf{C}_1 \cup \cdots \cup \mathsf{C}_{j-1}) \subset \mathsf{C}_i$

Construct $\mathcal{T}$ with $\mathcal{E}$ satisfying

- C_i is parent of C_j

- Utilize ordering to assign the positive definite conditions
- Keep track of edges that have already been considered

Running Intersection Property



General SDPs

- Appropriate definition of constraint, objective matrices
  - "Appropriate" - Zero entries for subblocks whose $\succeq 0$ is ignored
  - the additional constraints and multipliers can be ignored
  - the multipliers for clique linking $= 0$
  - Approach reduces to the Completion Approach of Nakata et al. (2000) (?)

Preprocessing Techniques

- F. N. Permenter and P. A. Parrilo. Partial facial reduction: simplified, equivalent sdps via approximations of the psd cone. Mathematical Programming, (2017)
- Y. Zhu, G. Pataki and Q. Tran-Dinh, Sieve-SDP: a simple facial reduction algorithm to preprocess semidefinite programs, https://arxiv.org/abs/1710.08954
- V. Kungurtsev and J. Marecek, A Two-Step Pre-Processing for Semidefinite Programming, https://arxiv.org/abs/1806.10868