Chordal graphs and sparse semidefinite optimization

Lieven Vandenberghe

Electrical Engineering Department
University of California, Los Angeles

VI Traditional Summer School for Young Researchers
Control, Information, Optimization

June 22–29, 2014

Semidefinite program (SDP)

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(CX) \\ \text{subject to} & \mathbf{tr}(A_k X) = b_k, \quad k = 1, \ldots, m \\ & X \succeq 0 \end{array}$$

variable X is symmetric matrix; $X \succeq 0$ means X is positive semidefinite

- used in convex modeling systems (CVX, YALMIP, CVXPY, . . .)
- relaxations of nonconvex quadratic and polynomial optimization

Sparse semidefinite optimization

- in many applications the coefficients A_k, C are sparse
- optimal X is typically dense, even for sparse A_k, C

Exploiting sparsity

1. Symmetric primal-dual interior-point methods

exploit sparsity when forming 'Schur complement' equations

2. Non-symmetric interior-point methods (matrix completion methods)

(Fukuda et al. 2000, Burer 2003, Srijuntongsiri et al. 2004, Andersen et al. 2010)

3. Decomposition (combined with interior-point or first-order methods)

(Fukuda et al. 2000, Nakata et al. 2003, Kim et al. 2011, Sun et al. 2014, . . .)

we will discuss approaches 2 and 3

Chordal graphs

chordal graphs have been studied in many fields since the 1960s

- linear algebra (sparse factorization, completion problems)
- combinatorial optimization (a class of 'perfect' graphs)
- machine learning (graphical models, Euclidean distance matrices)
- nonlinear optimization (partial separability)
- computer science (database theory)

first used in semidefinite optimization by Fujisawa, Kojima, Nakata (1997)

Overview

Lecture 1

I. Chordal graphs

II. Sparse matrices

Lecture 2

III. Non-symmetric interior-point methods

IV. Decomposition

V. Semidefinite relaxations

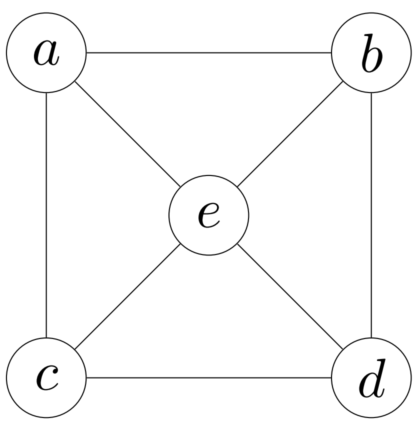
I. Chordal graphs

- definition, fundamental theorems
- perfect elimination
- triangularization

Undirected graph

$$G = (V, E)$$

- V is set of **vertices**
- $E \subseteq \{\{v, w\} \mid v, w \in V\}$ is set of **edges**
- v and w are **adjacent** if $\{v, w\} \in E$
- **neighborhood** $\text{adj}(v)$ of vertex v is set of vertices adjacent to v



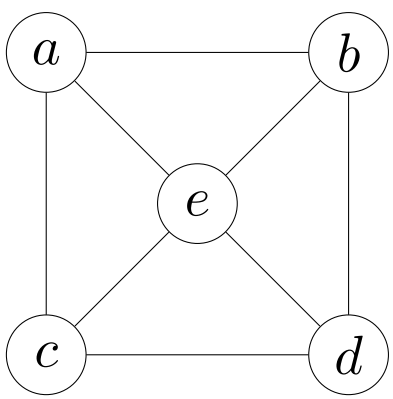
- vertices: $V = \{a, b, c, d, e\}$
- edges: $E = \{\{a, b\}, \{a, c\}, \{a, e\}, \ldots\}$
- $\text{adj}(a) = \{b, c, e\}$

Subgraphs and cliques

Definition: subgraph (induced by) $W \subset V$

$$G(W) = (W, E(W)), \qquad E(W) = \{\{v,w\} \in E \mid v,w \in W\}$$

- subgraph is **complete** if $E(W) = \{\{v,w\} \mid v,w \in W\}$
- a **clique** is a **maximal** complete subgraph



- subgraph (induced by) $W = \{a,b,c,d\}$:

$$E(W) = \{\{a,b\},\{b,d\},\{c,d\},\{a,c\}\}$$

- $W = \{a,b,e\}$ is a clique
- $W = \{a,b\}$ is complete but not a clique

Example: symmetric sparsity pattern

Definition: undirected graph $G = (V, E)$ with $V = \{1, 2, \ldots, n\}$

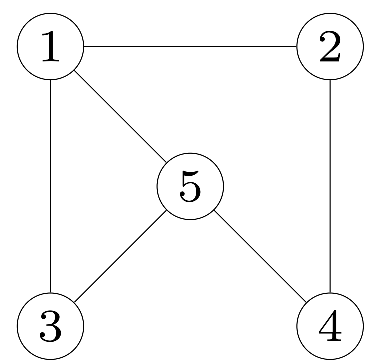
- symmetric matrix A of order n has this sparsity pattern ($A \in \mathbf{S}^n_E$) if

$$i \neq j, \quad \{i, j\} \notin E \qquad \Longrightarrow \qquad A_{ij} = A_{ji} = 0$$

  entries A_{ij} with $i = j$ or $\{i, j\} \in E$ may or may not be zero

- not unique (unless all off-diagonal entries of A are nonzero)
- cliques of G correspond to maximal 'dense' principal submatrices

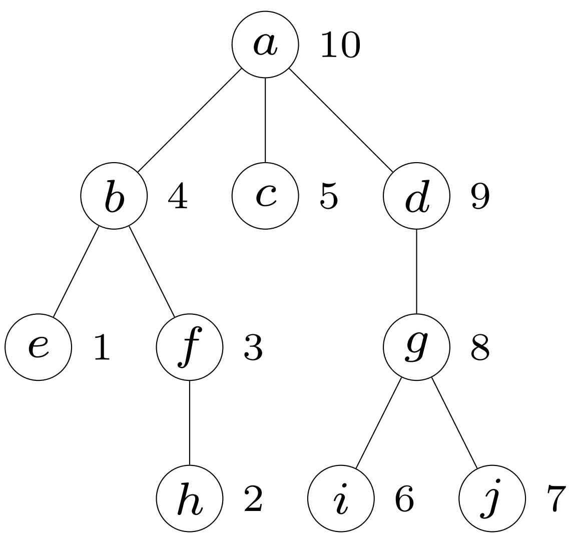
$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & 0 & A_{15} \\ A_{21} & A_{22} & 0 & A_{24} & 0 \\ A_{31} & 0 & A_{33} & 0 & A_{35} \\ 0 & A_{42} & 0 & A_{44} & A_{45} \\ A_{51} & 0 & A_{53} & A_{54} & A_{55} \end{bmatrix}$$



Rooted tree

connected, acyclic graph with one vertex designated as root

- parent of vertex v is denoted $p(v)$
- ancestors of higher degree denoted $p^k(v)$, $k = 1, 2 \ldots$
- topological ordering: visits child before parent
- postordering: descendants of each vertex numbered consecutively

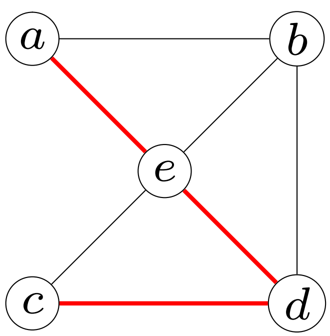


$$p(h) = f, \quad p^2(h) = b, \quad p^3(h) = a, \quad \ldots$$

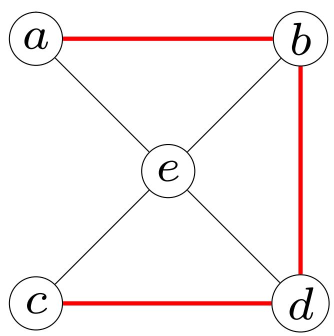
Chorded paths and cycles

a **chord** is an edge between non-consecutive vertices in a path or cycle

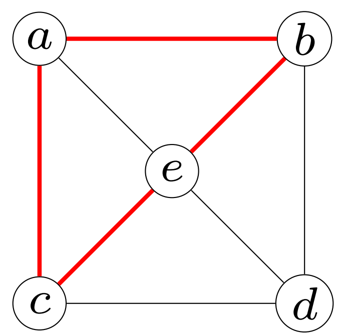
- a one-edge 'shortcut' in a path or cycle
- all shortest paths are chordless



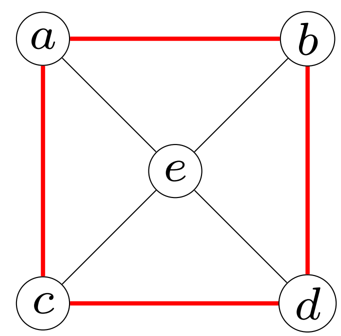
chorded path



chordless path



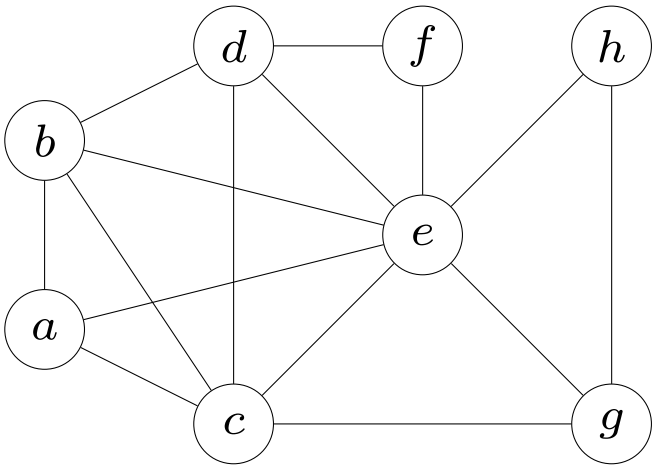
chorded cycle



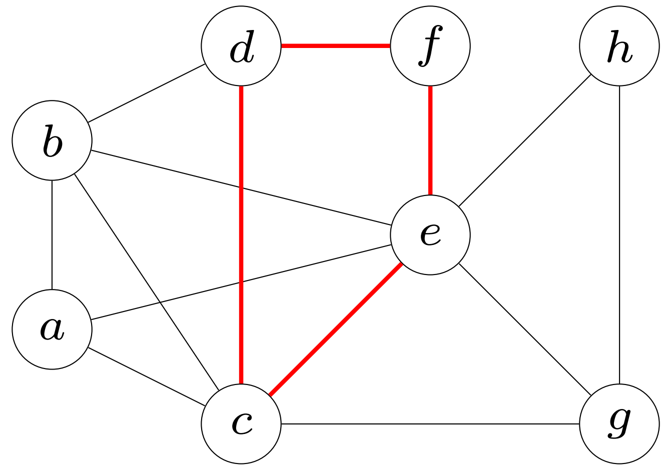
chordless cycle

Chordal graph

Chordal graph: every cycle of length greater than three has a chord



chordal



not chordal

- using chords to take 'shortcuts', all cycles can be reduced to triangles
- subgraphs of chordal graphs are chordal

also known as rigid circuit graphs, triangulated, decomposable graphs, . . .

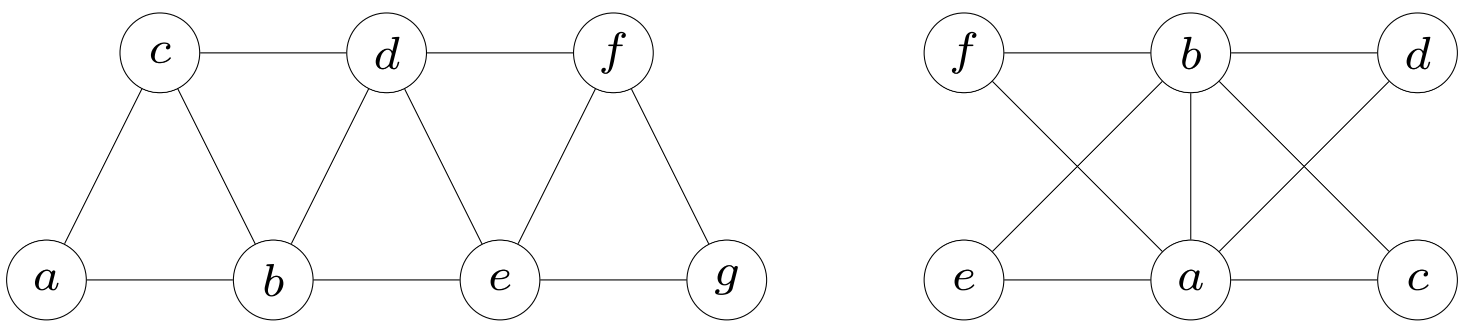
Examples

Trivial: complete graphs, trees, cactus graphs (no cycles of length > 3)

k-**Trees:** constructed recursively

- k-tree with k vertices is complete graph
- to construct k-tree with $n + 1$ vertices from k-tree with n vertices:

  make new vertex adjacent to a complete subgraph of k vertices

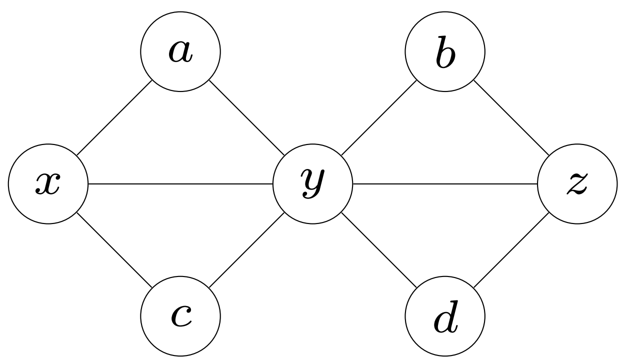


two 2-trees

Minimal vertex separator

Definition: $S \subset V$ is a minimal vw-separator if

- v and w are in different connected components of $G(V \setminus S)$
- no strict subset of S is a vw-separator

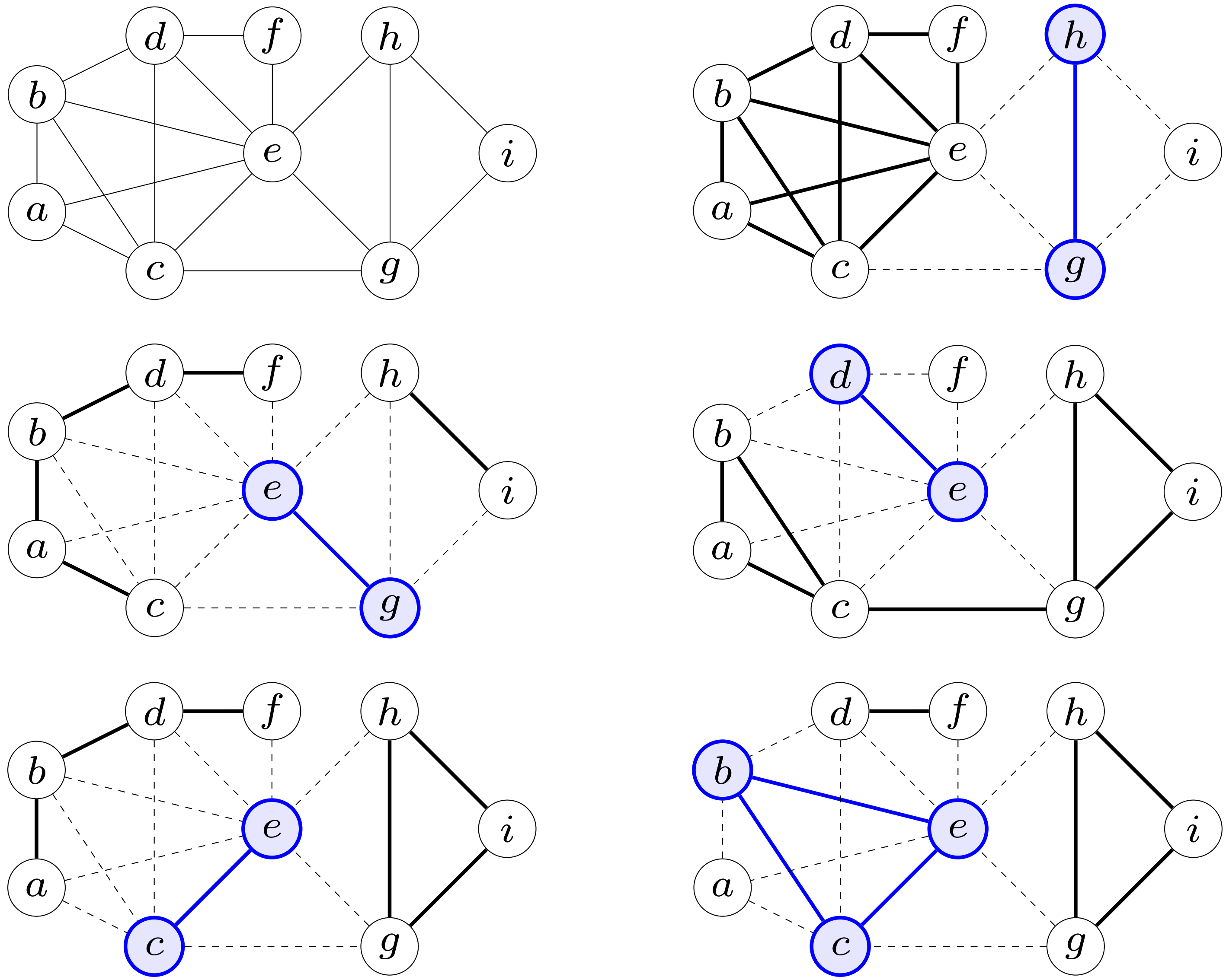


- $\{x, y\}$ is a minimal ac-separator
- $\{y\}$ is a minimal ad-separator

Chordal graphs (Dirac 1961, Buneman 1974)

- graph is chordal if and only if all minimal vertex separators are complete
- every minimal vertex separator is a subset of at least two cliques

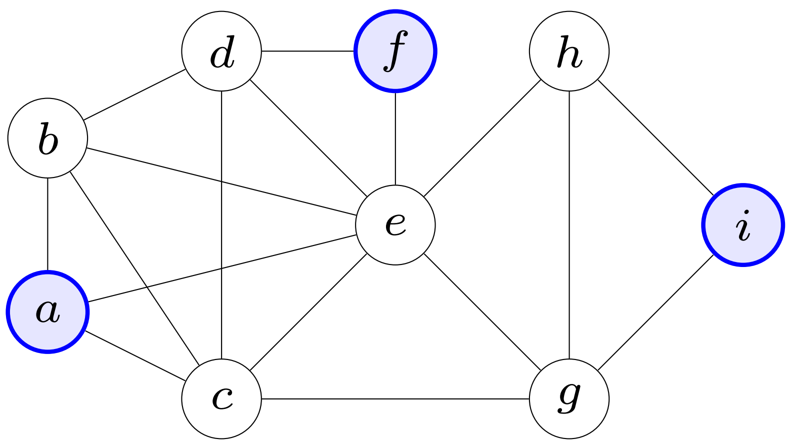
Example



Simplicial vertices

Definition: v is a simplicial vertex if $\mathrm{adj}(v)$ is complete

- closed neighborhood $\{v\} \cup \mathrm{adj}(v)$ is a clique
- $\{v\} \cup \mathrm{adj}(v)$ is the only clique that contains v



three simplicial vertices

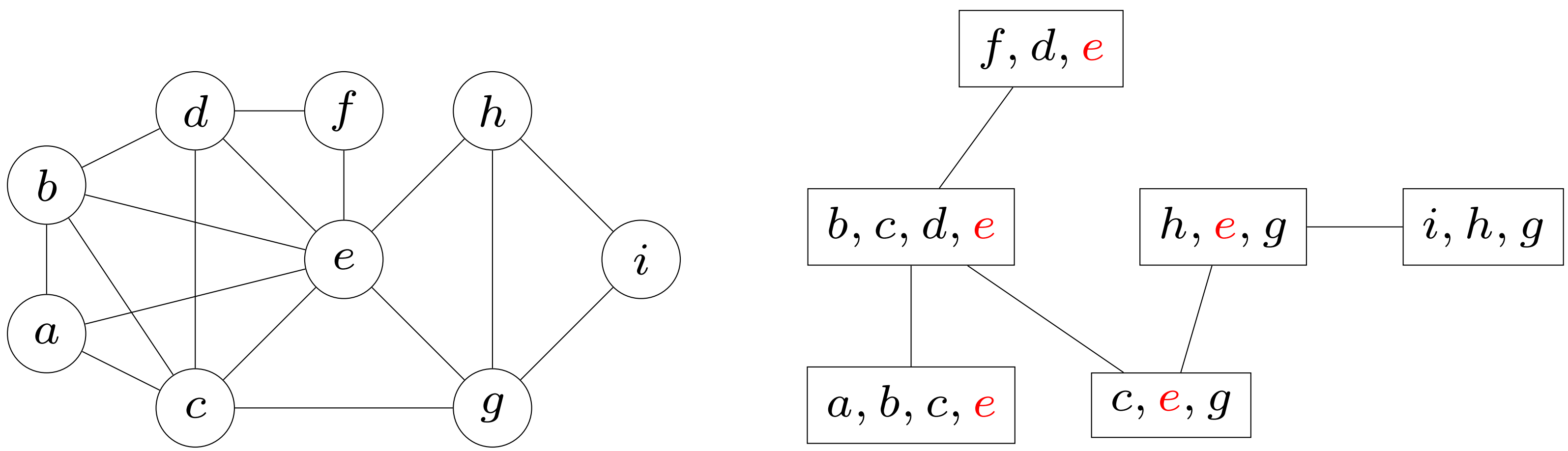
Chordal graphs (Dirac 1961)

a non-complete chordal graph has at least 2 non-adjacent simplicial vertices

Clique tree

Definition: clique tree with the induced subtree property for $G = (V, E)$

- vertices of clique tree are the cliques of G
- for every $v \in V$, cliques that contain v form subtree of clique tree

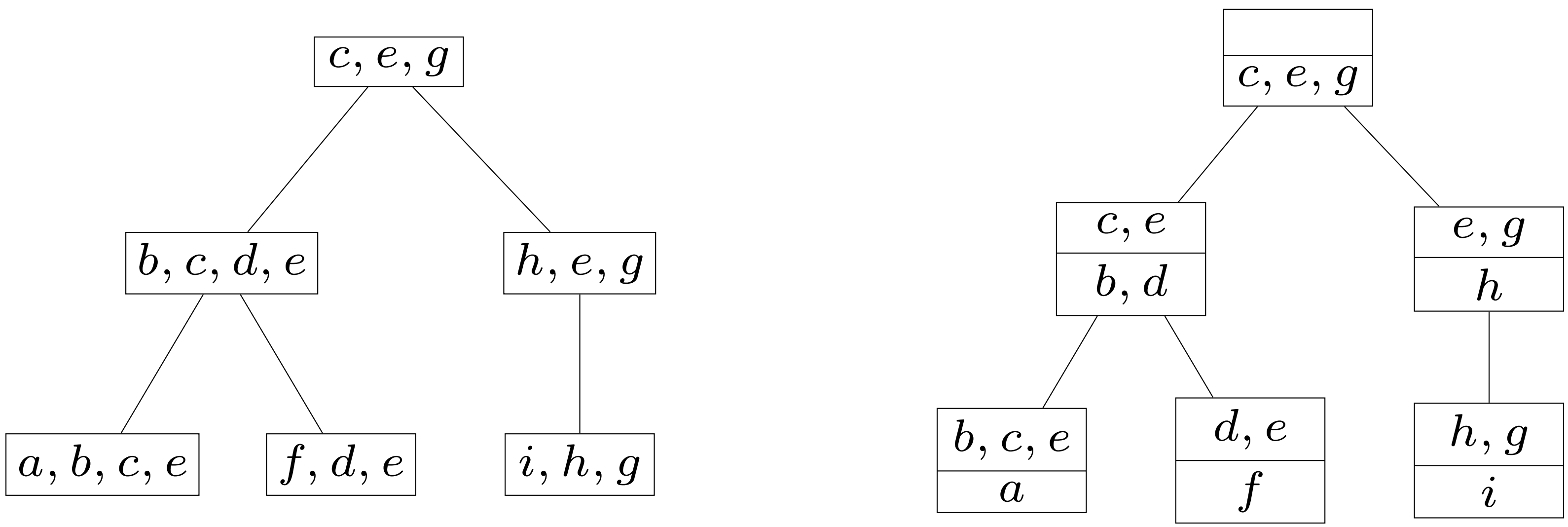


Chordal graphs (Buneman 1974, Gavril 1974)

G is chordal if and only it has a clique tree with induced subtree property

Clique separators and residuals

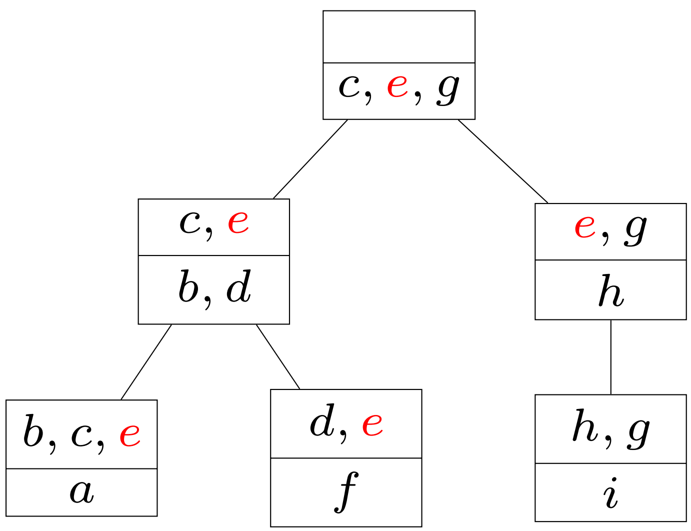
- choose any clique as root; denote parent function as $p(W)$
- clique **separator** of non-root clique W is $\mathrm{sep}(W) = W \cap p(W)$
- clique **residual** is $\mathrm{res}(W) = W \setminus \mathrm{sep}(W)$ (and $\mathrm{res}(W) = W$ for root)

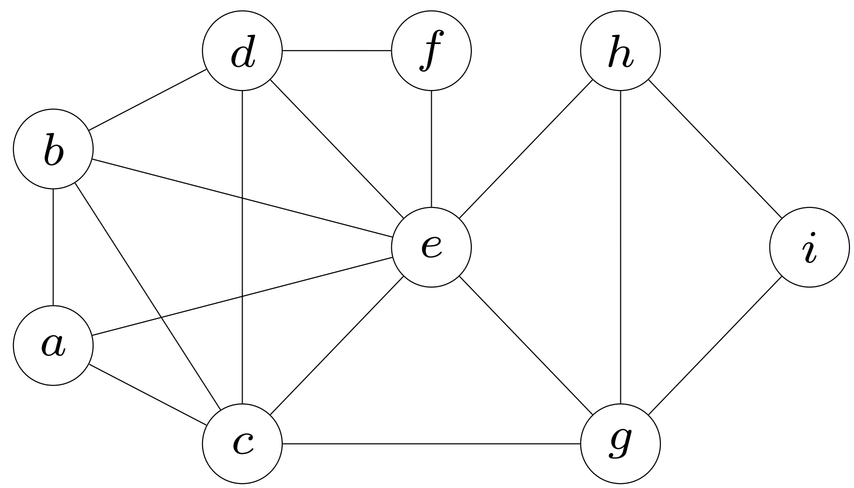


$$W = \{b, c, d, e\}, \qquad \mathrm{res}(W) = \{b, d\}, \qquad \mathrm{sep}(W) = \{c, e\}$$

Graph structure from rooted clique tree

- every vertex v belongs to exactly one clique residual $\mathrm{res}(W)$
- if $v \in \mathrm{res}(W)$ then W is the root of the subtree of cliques that contain v
- the clique separators $\mathrm{sep}(W)$ are the minimal vertex separators
- a vertex is simplicial if it does not belong to any clique separator





chordal graph has at most $n = |V|$ cliques, $n - 1$ minimal vertex separators

Tree intersection graphs

Definition: given a family of subtrees $\{R_v \mid v \in V\}$ of a tree T

- tree intersection graph $G = (V, E)$ has vertex set V
- $\{v, w\} \in E$ if and only if R_v and R_w intersect

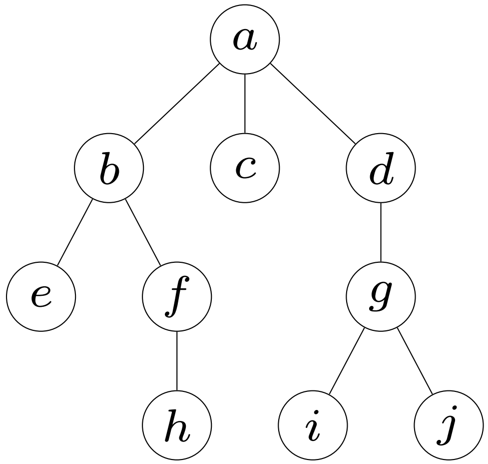
Chordality (Gavril 1974, Buneman 1974)

- a tree intersection graph is chordal
- every chordal graph can be represented as a tree intersection graph

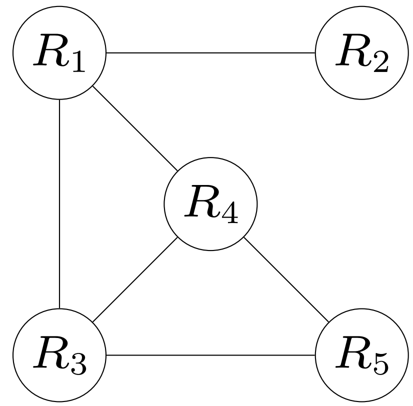
  (for example, T is the clique tree, R_v subtree of cliques that contain v)

Example

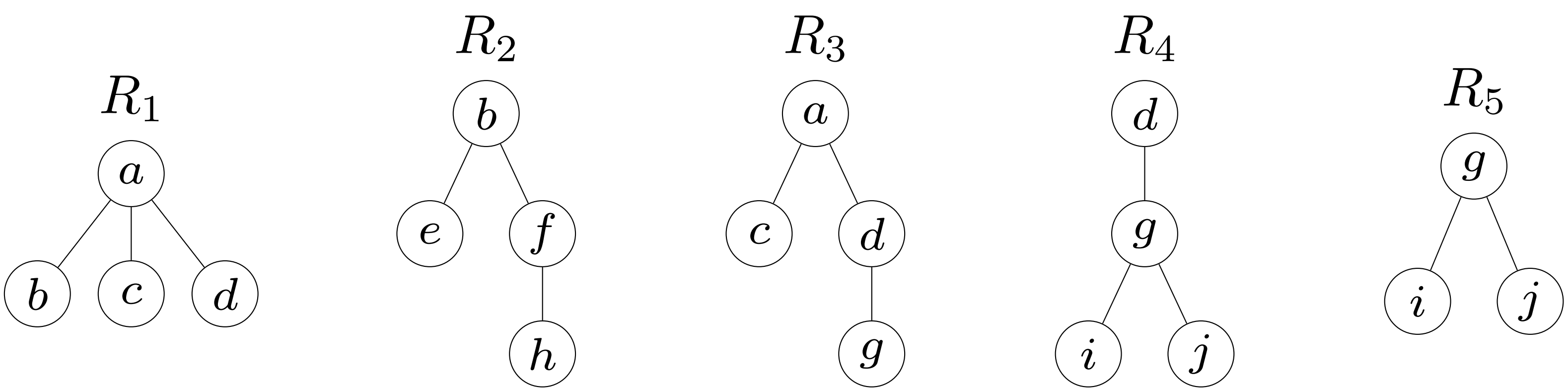
tree T



tree intersection graph



five subtrees of T



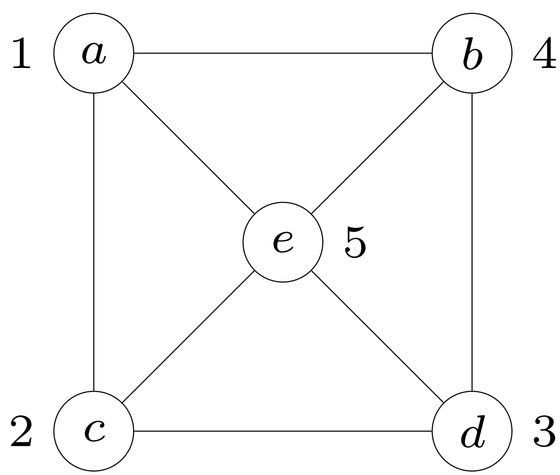
I. Chordal graphs

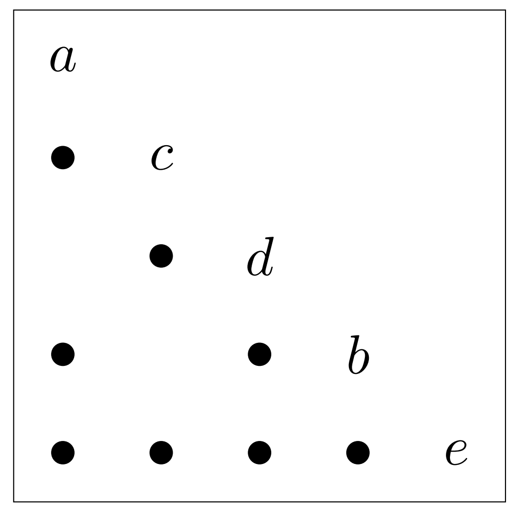
- definition, fundamental theorems
- **perfect elimination**
- triangularization

Ordered undirected graphs

$$G_\sigma = (V, E, \sigma)$$

- σ is a bijection from $\{1, 2, \ldots, |V|\}$ to V
- ordering notation: $v \prec w$ means $\sigma^{-1}(v) < \sigma^{-1}(w)$





can be represented as annotated graph or triangular array

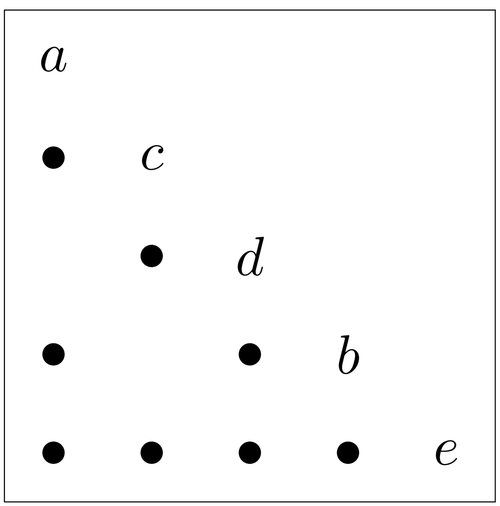
Monotone neighborhoods

- higher and lower (monotone) neighborhoods

$$\mathrm{adj}^+(v) = \mathrm{adj}(v) \cap \{w \mid w \succ v\}, \qquad \mathrm{adj}^-(v) = \mathrm{adj}(v) \cap \{w \mid w \prec v\}$$

- closed higher and lower neighborhoods

$$\mathrm{col}(v) = \{v\} \cup \mathrm{adj}^+(v), \qquad \mathrm{row}(v) = \{v\} \cup \mathrm{adj}^-(v)$$



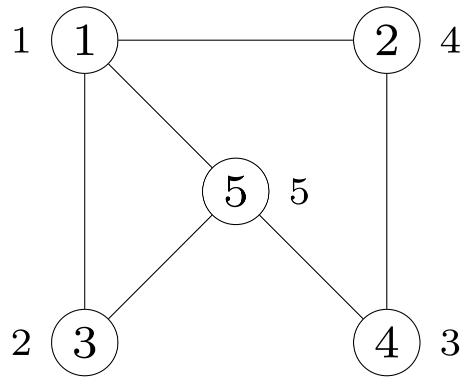
monotone neighborhoods of c:

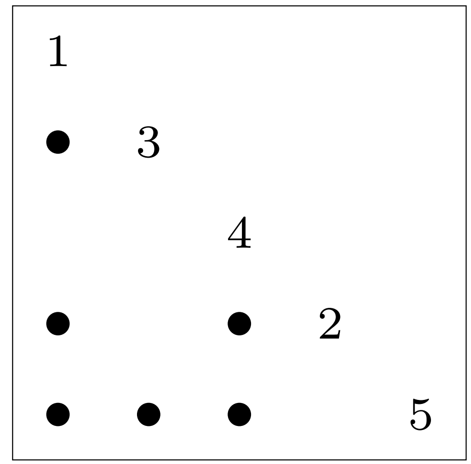
$$\mathrm{adj}^+(c) = \{d, e\}, \quad \mathrm{adj}^-(c) = \{a\}$$

$$\mathrm{col}(c) = \{c, d, e\}, \quad \mathrm{row}(c) = \{c, a\}$$

Example: ordered symmetric sparsity pattern

- ordered sparsity pattern (V, E, σ) of order 5 with $\sigma = (1, 3, 4, 2, 5)$





- represents symmetric reordering (P_σ is permutation matrix defined by σ)

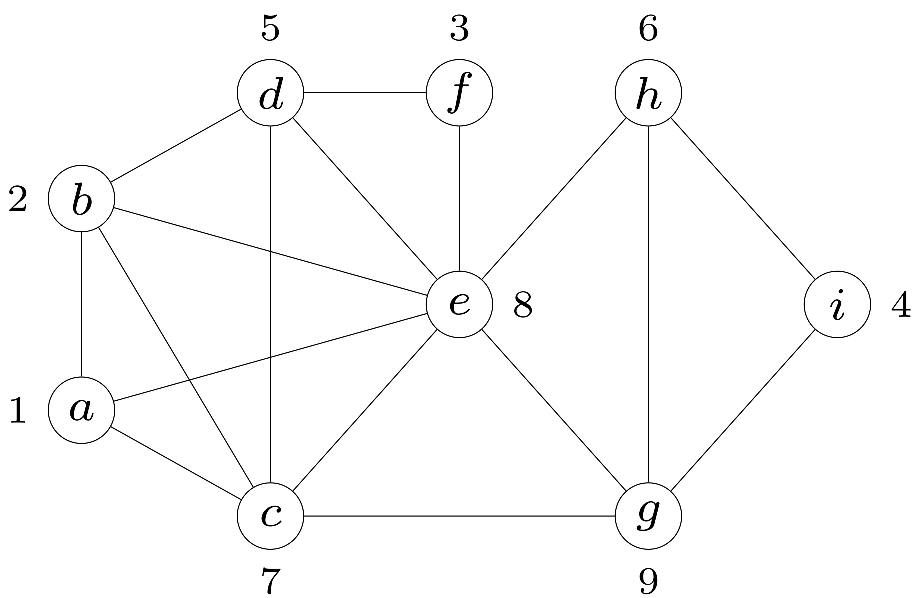
$$P_\sigma A P_\sigma^T = \begin{bmatrix} A_{11} & A_{31} & 0 & A_{21} & A_{51} \\ A_{31} & A_{33} & 0 & 0 & A_{53} \\ 0 & 0 & A_{44} & A_{42} & A_{54} \\ A_{21} & 0 & A_{42} & A_{22} & 0 \\ A_{51} & A_{53} & A_{54} & 0 & A_{55} \end{bmatrix}$$

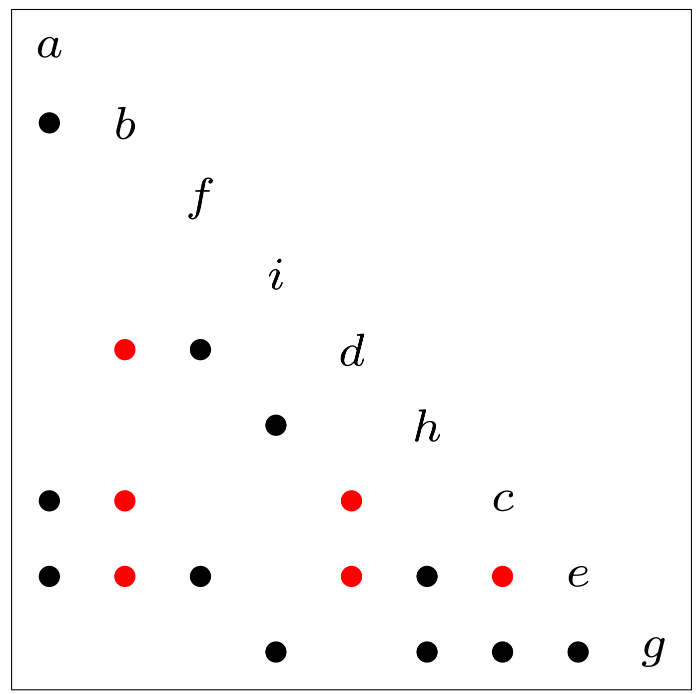
Filled graph

an ordered graph is **filled** or **monotone transitive** if

$$w, z \in \mathrm{adj}^+(v) \quad \Longrightarrow \quad \{w, z\} \in E$$

higher neighborhood of every vertex is complete



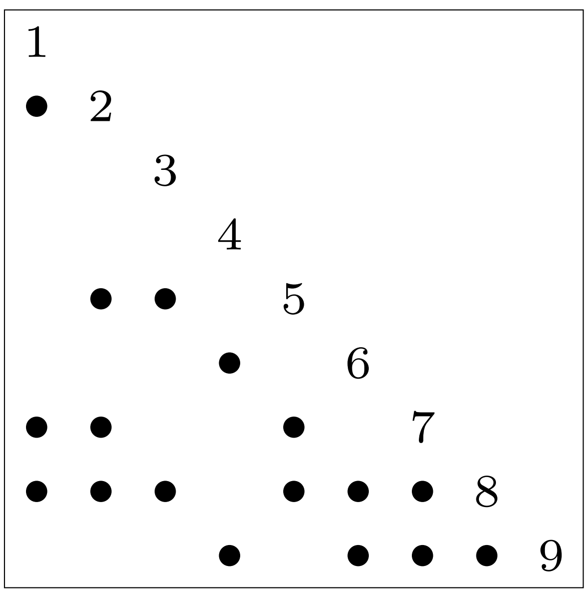


Example: pattern of Cholesky factor

Cholesky factorization of positive define $A \in \mathbf{S}^n$

$$A = LDL^T$$

- L unit lower triangular, D positive diagonal
- $L + L^T$ has a filled sparsity pattern, with $\sigma = (1, 2, \ldots, n)$



$$i > j > k, \qquad \{i, j\} \in E, \qquad \{i, k\} \in E$$

$$\Downarrow$$

$$\{j, k\} \in E$$

Perfect elimination ordering

σ is a **perfect elimination ordering** for (V, E) if (V, E, σ) is filled

Chordal graphs (Fulkerson and Gross 1965)

a graph is chordal if and only if it has a perfect elimination ordering

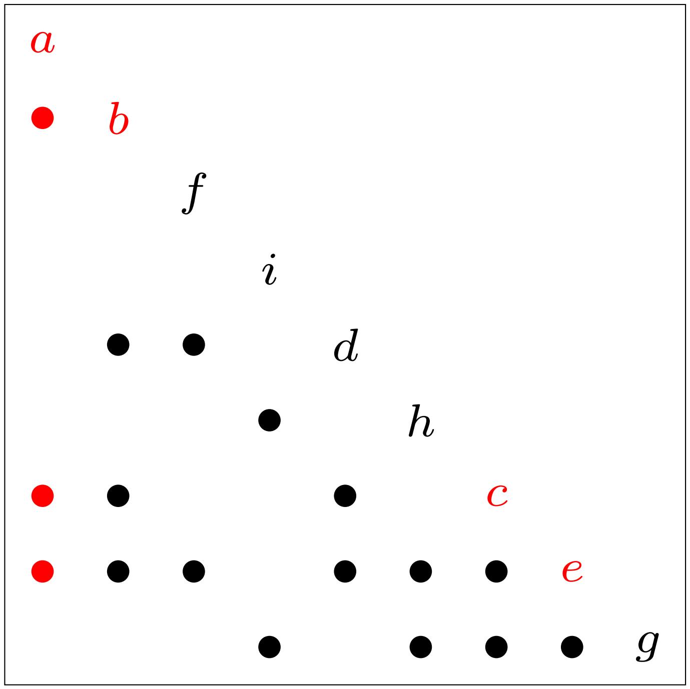
Simplicial elimination

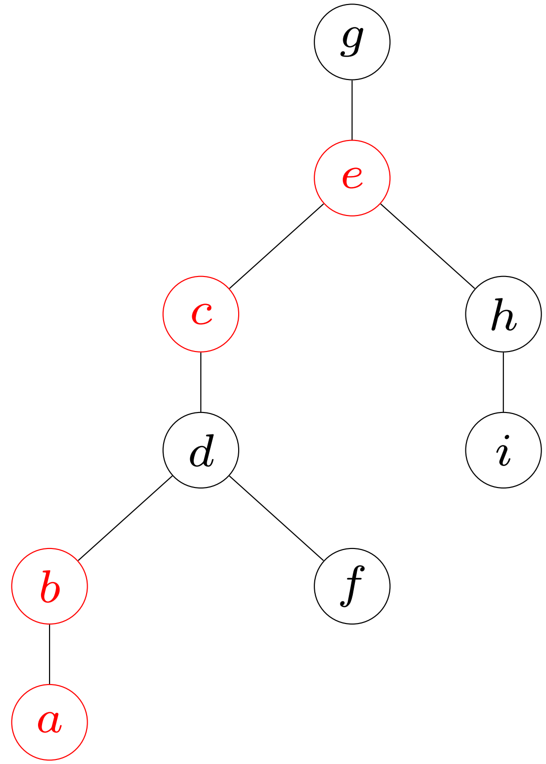
- find a simplicial vertex v and take $\sigma(1) = v$
- let $\sigma(2), \ldots, \sigma(n)$ be a perfect elimination ordering of $G(V \setminus \{v\})$

Practical algorithms

- find perfect elimination ordering in $O(|V| + |E|)$ time
- can be used to test chordality

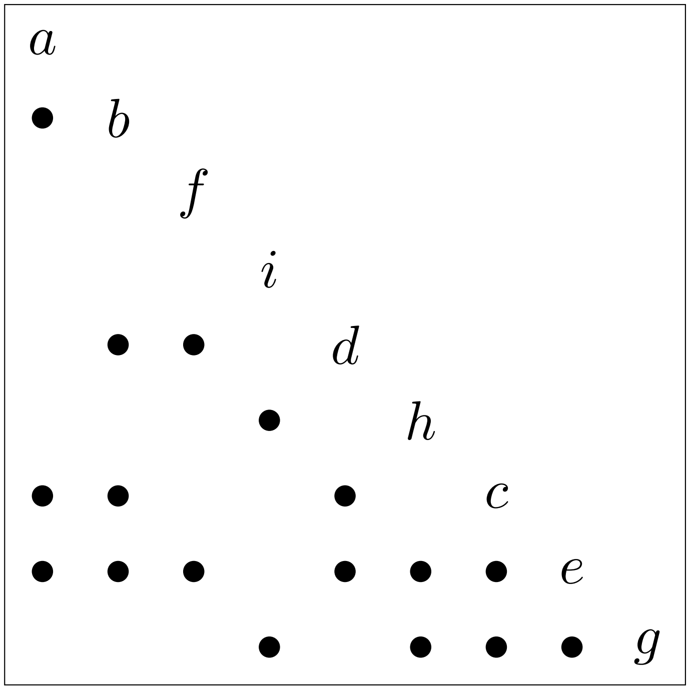
Elimination tree for filled graph

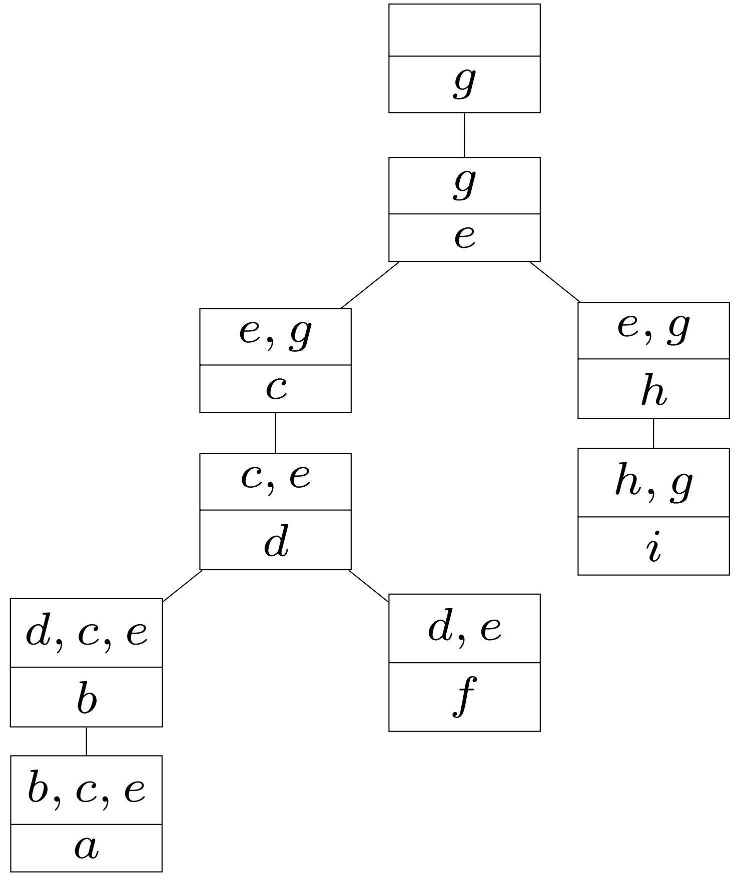




- parent $p(v)$ of vertex v is first vertex in $\mathrm{adj}^+(v)$
- elements of $\mathrm{adj}^+(v)$ are ancestors of v (not necessarily contiguous)

Expanded elimination tree

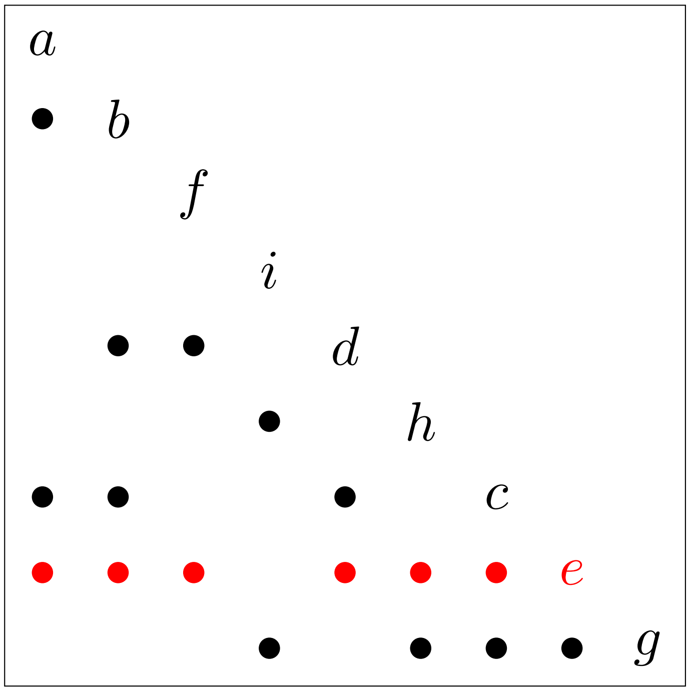


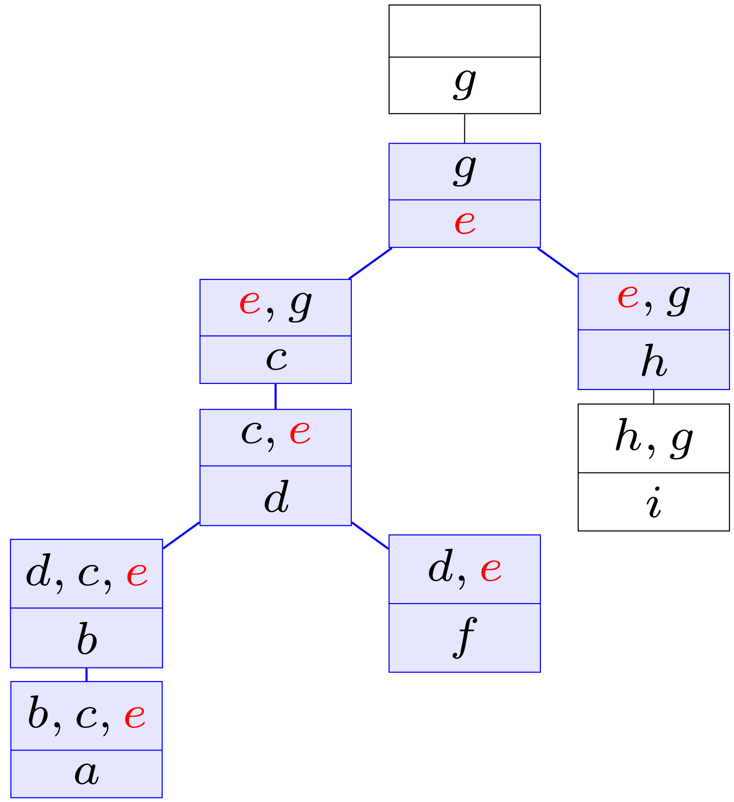


- row above vertex v is $\mathrm{adj}^+(v)$
- monotone transitivity: $\mathrm{adj}^+(v) \subset \mathrm{col}(p(v))$ for every (non-root) v

Induced subtree property

$\mathrm{row}(v) = \{w \mid v \in \mathrm{col}(w)\}$ forms a subtree of elimination tree





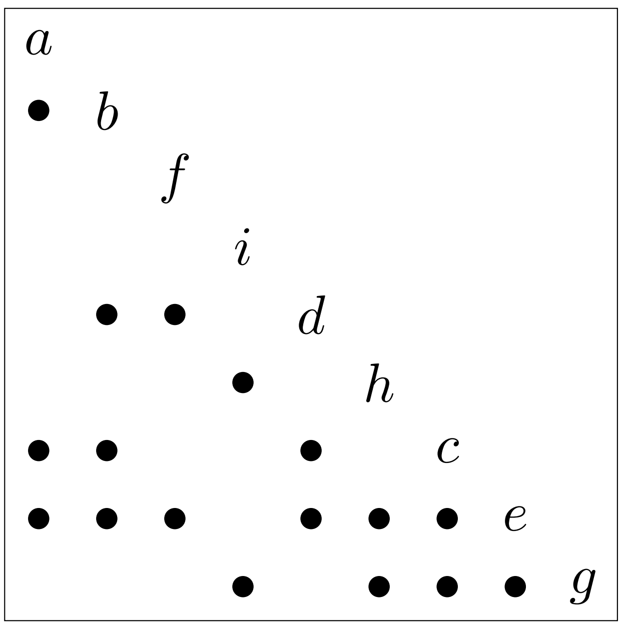
$$\mathrm{row}(e) = \{a, b, f, d, h, c, e\}$$

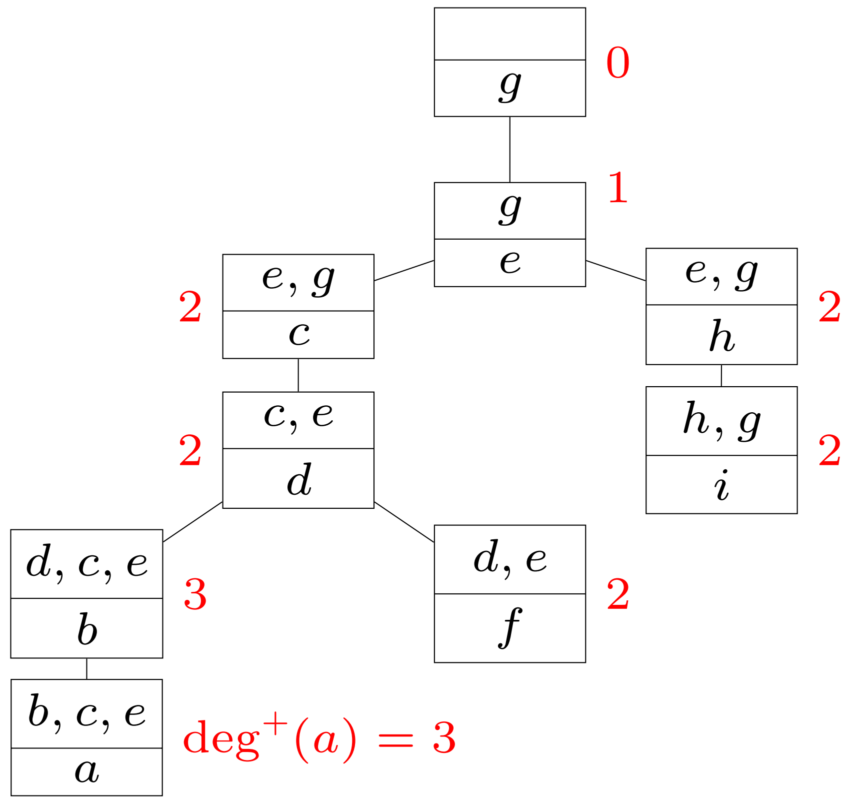
gives another representation of chordal graph as tree intersection graph

Higher degrees

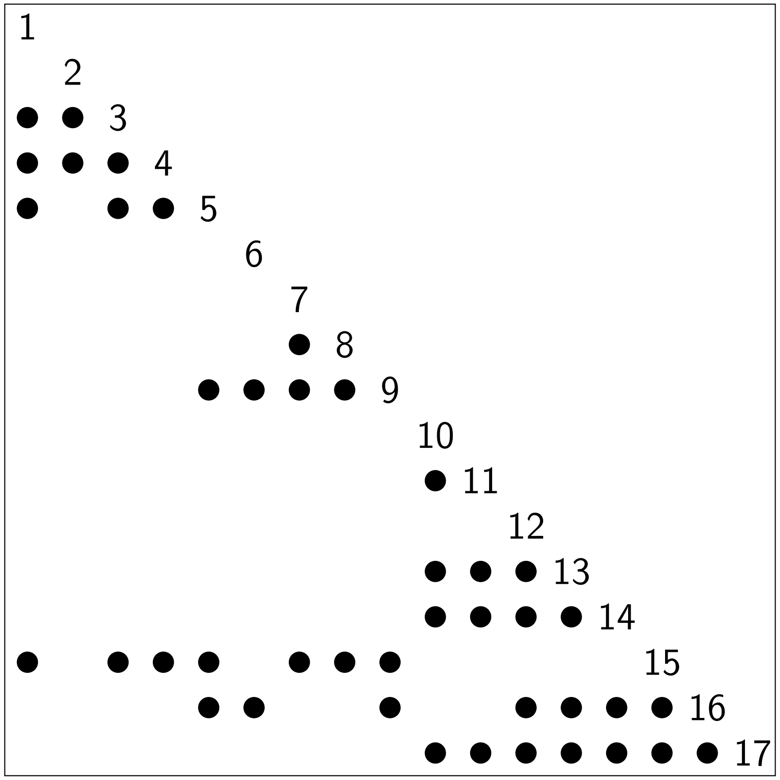
Definition: higher degree of vertex v is $\deg^+(v) = |\text{adj}^+(v)|$

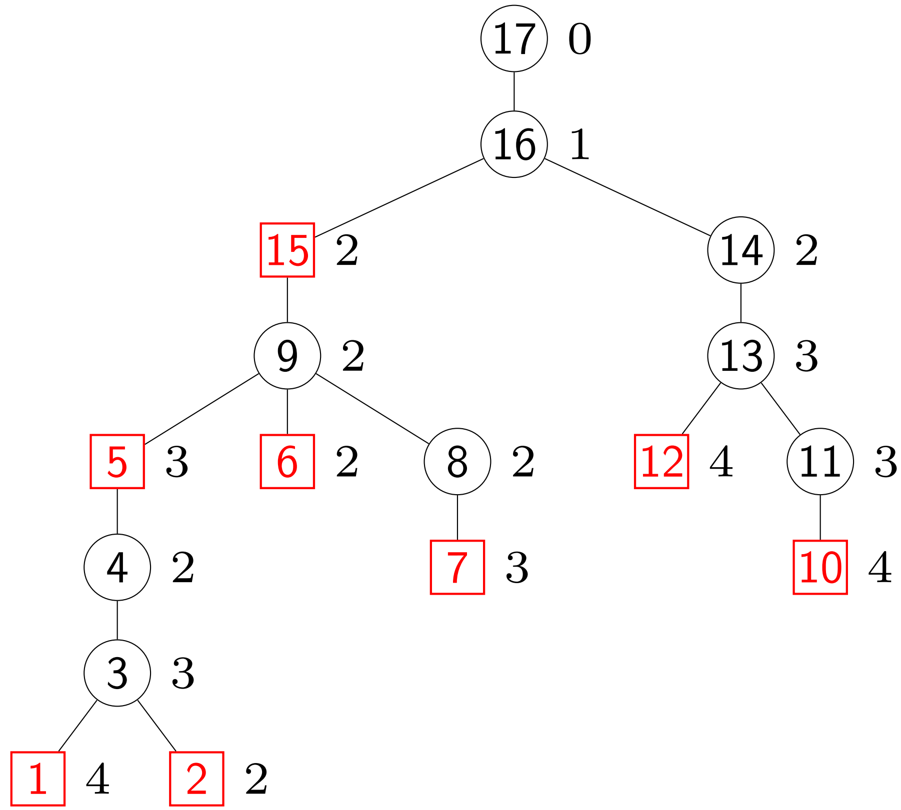
- higher degrees satisfy $\deg^+(v) \leq \deg^+(p(v)) + 1$
- $\deg^+(v) = \deg^+(p(v)) + 1$ holds only if $\text{adj}^+(v) = \text{col}(p(v))$





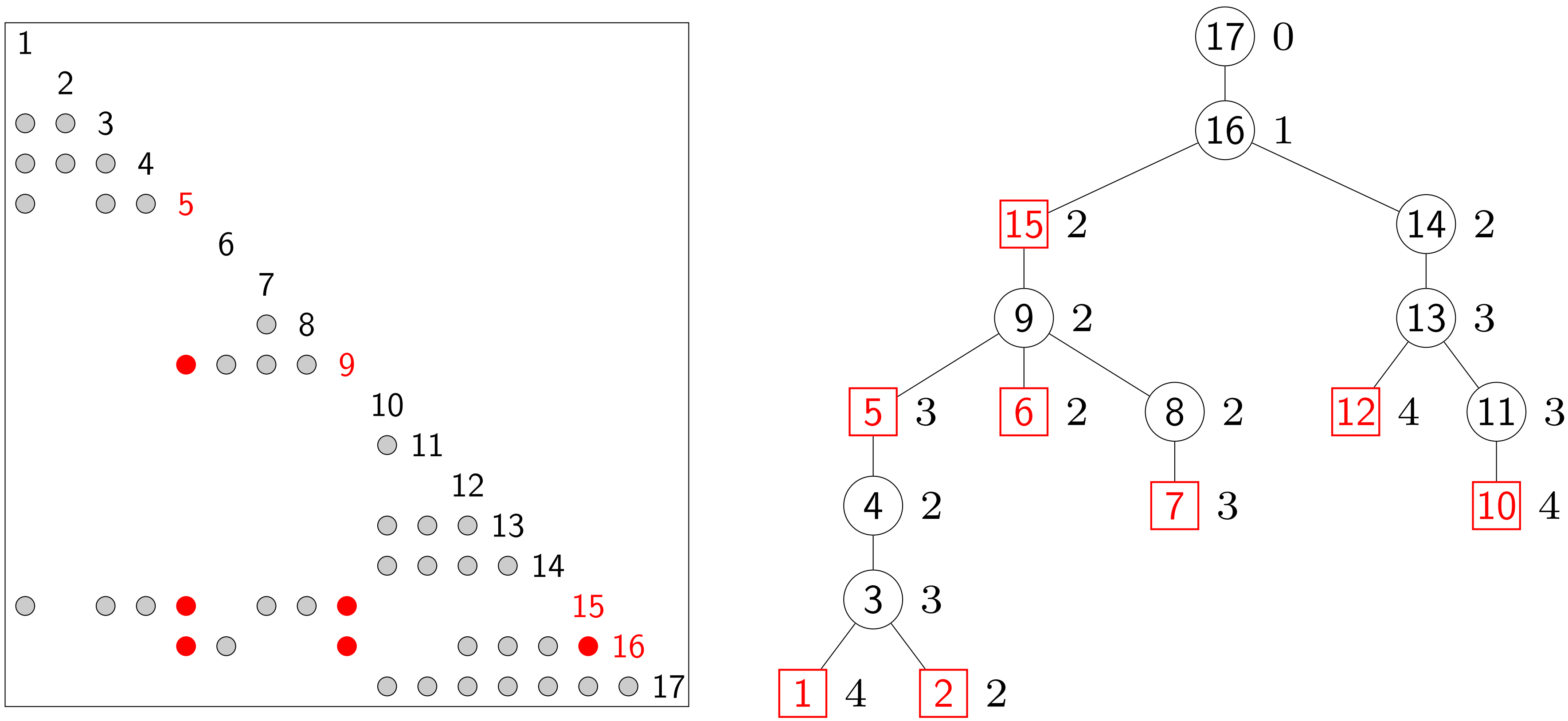
Cliques from elimination tree





- cliques are sets $\mathrm{col}(v)$ with v the **representative vertex** of the clique
- v is clique representative if $\deg^+(w) < \deg^+(v) + 1$ for all its children w

Cliques from elimination tree

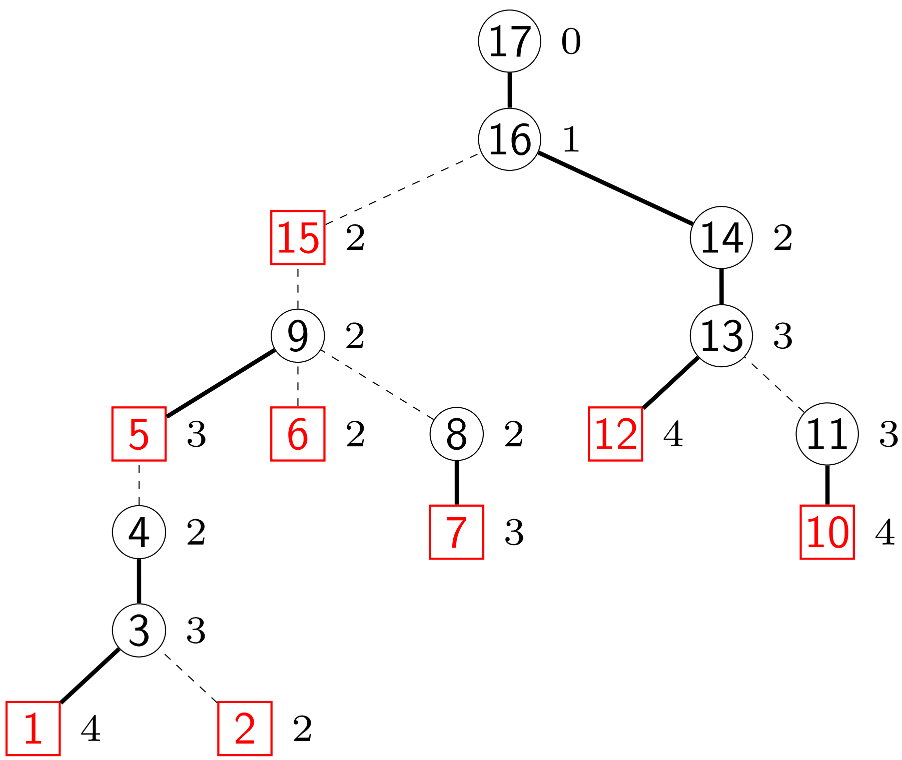


- test only needs elimination tree, higher degrees, not entire graph
- implies that a chordal graph has at most n cliques

Maximal supernodes

Definition: sets $\mathrm{sn}(v) = \{v, p(v), p^2(v), \ldots, p^{n_v}(v)\}$ that partition V

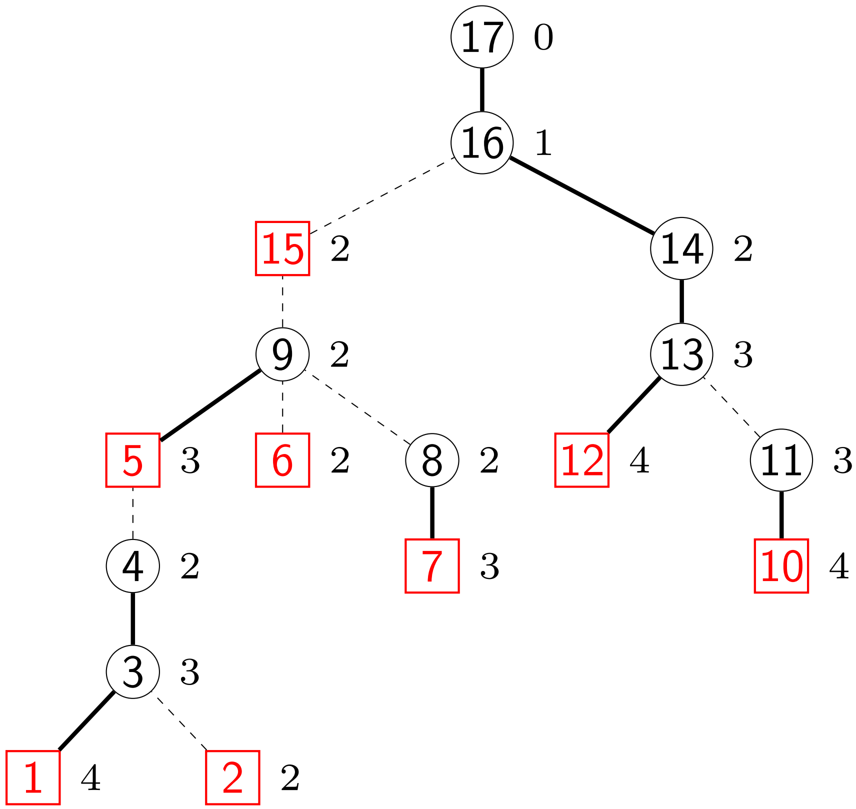
- v is a clique representative vertex
- $\deg^+(p^k(v)) = \deg^+(v) - k$ for $k = 1, \ldots, n_v$

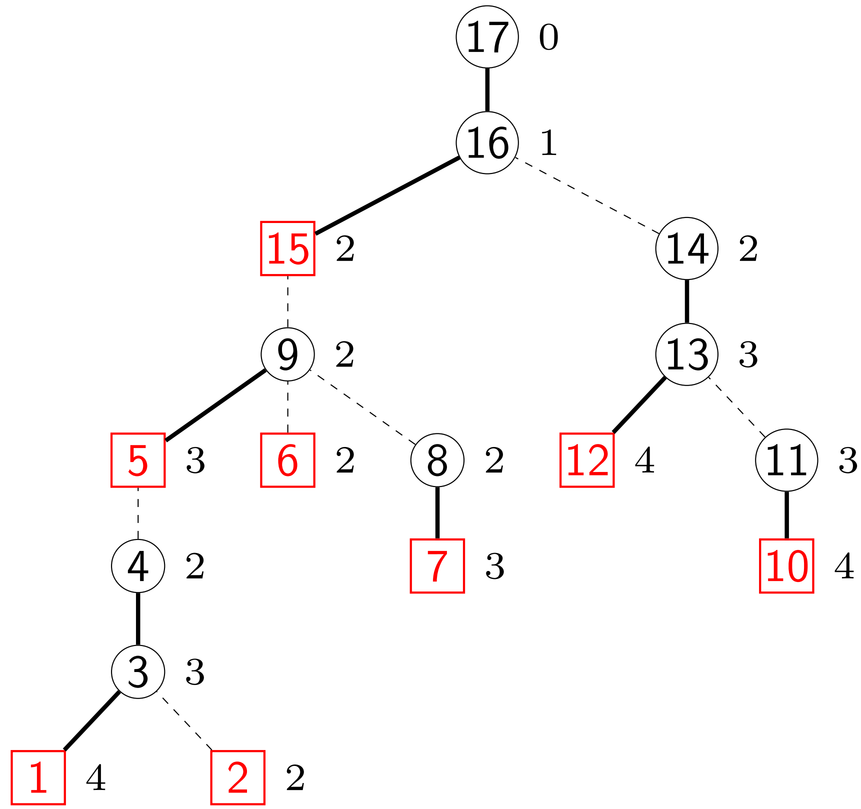


$$
\begin{aligned}
\mathrm{sn}(1) &= \{1, 3, 4\} \\
\mathrm{sn}(2) &= \{2\} \\
\mathrm{sn}(5) &= \{5, 9\} \\
\mathrm{sn}(6) &= \{6\} \\
\mathrm{sn}(7) &= \{7, 8\} \\
\mathrm{sn}(10) &= \{10, 11\} \\
\mathrm{sn}(12) &= \{12, 13, 14, 16, 17\} \\
\mathrm{sn}(15) &= \{15\}
\end{aligned}
$$

(Lewis, Peyton, Pothen 1998, Pothen and Sun 1990)

Nonuniqueness of maximal supernode partition



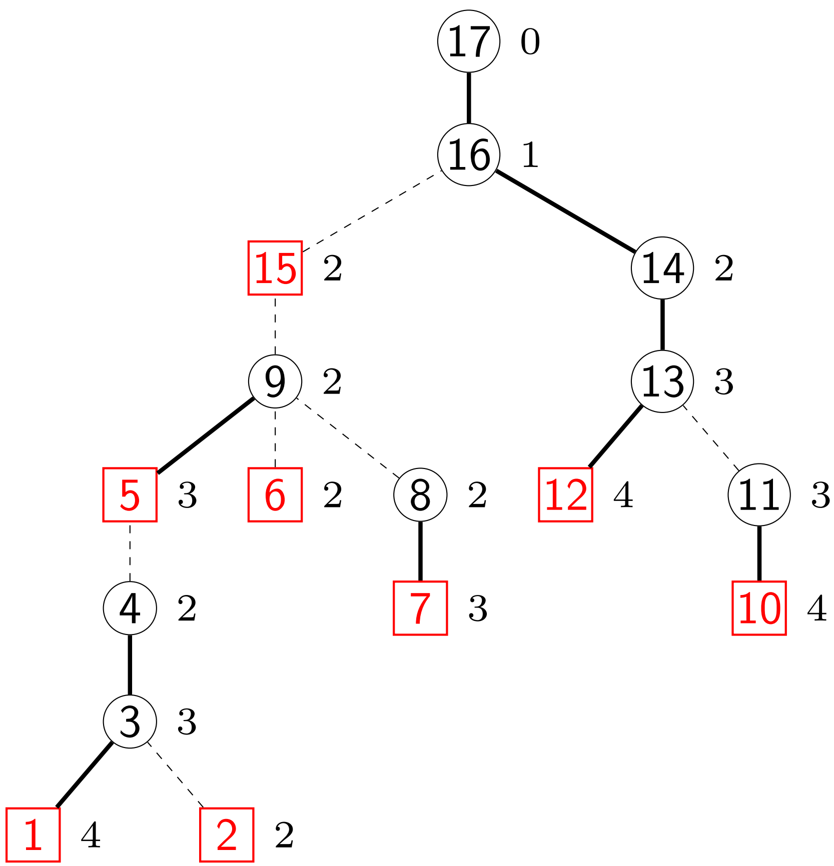


$$\begin{aligned} \mathrm{sn}(15) &= \{15\} \\ \mathrm{sn}(12) &= \{12, 13, 14, 16, 17\} \end{aligned}$$

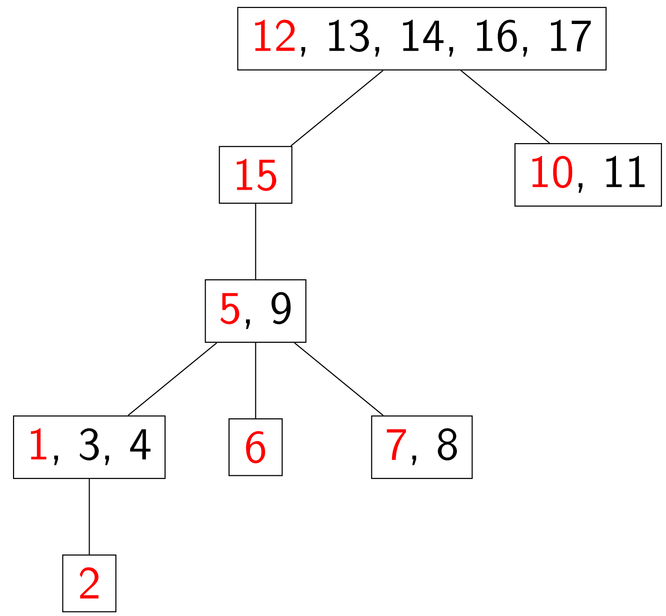
$$\begin{aligned} \mathrm{sn}(15) &= \{15, 16, 17\} \\ \mathrm{sn}(12) &= \{12, 13, 14\} \end{aligned}$$

Supernodal elimination tree

elimination tree

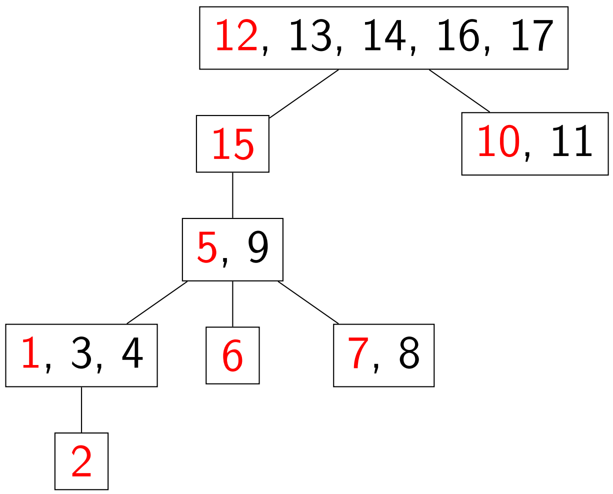


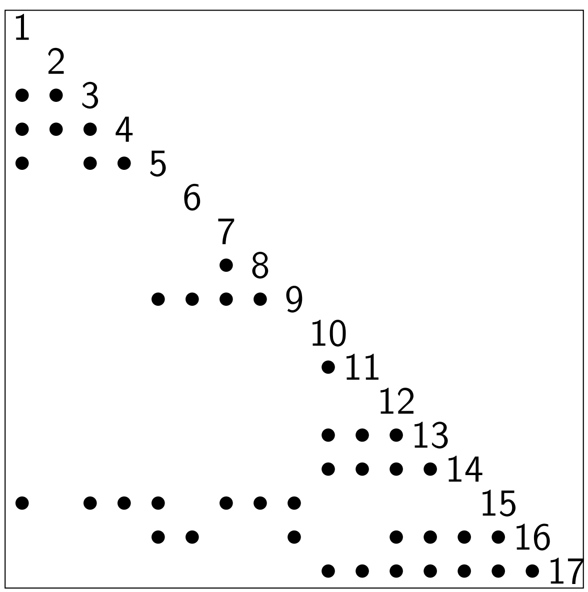
supernodal elimination tree

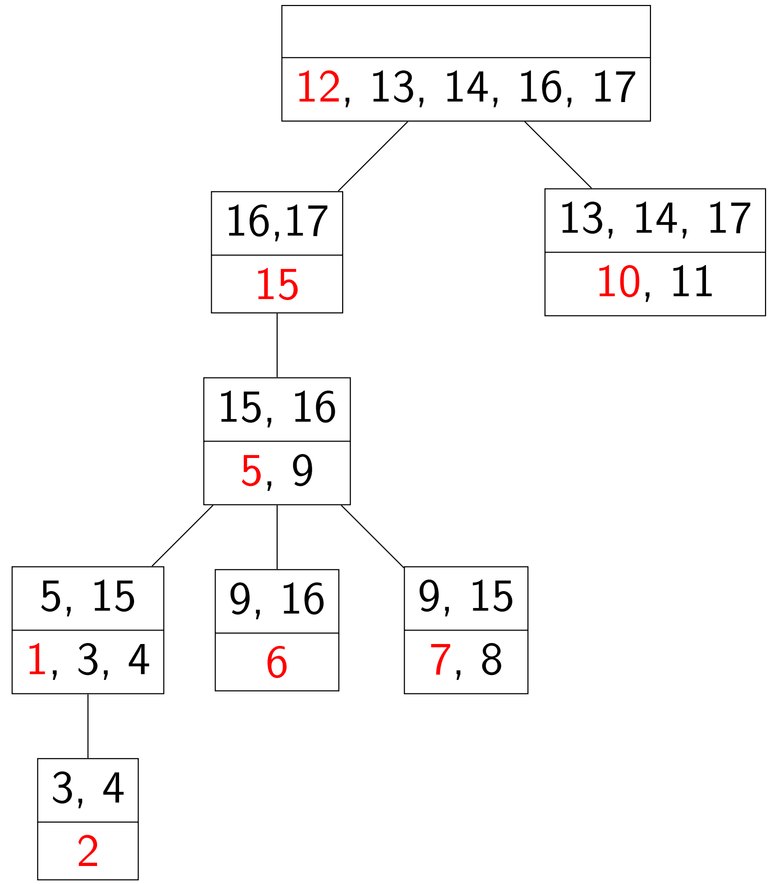


- vertices are supernodes $\mathrm{sn}(v)$
- parent of $\mathrm{sn}(v)$: supernode that contains parent (in etree) of last element of $\mathrm{sn}(v)$

Clique tree from supernodal elimination tree







- $\mathrm{sn}(v)$ is residual of clique $\mathrm{col}(v)$
- separator is $\mathrm{col}(v) \setminus \mathrm{sn}(v)$

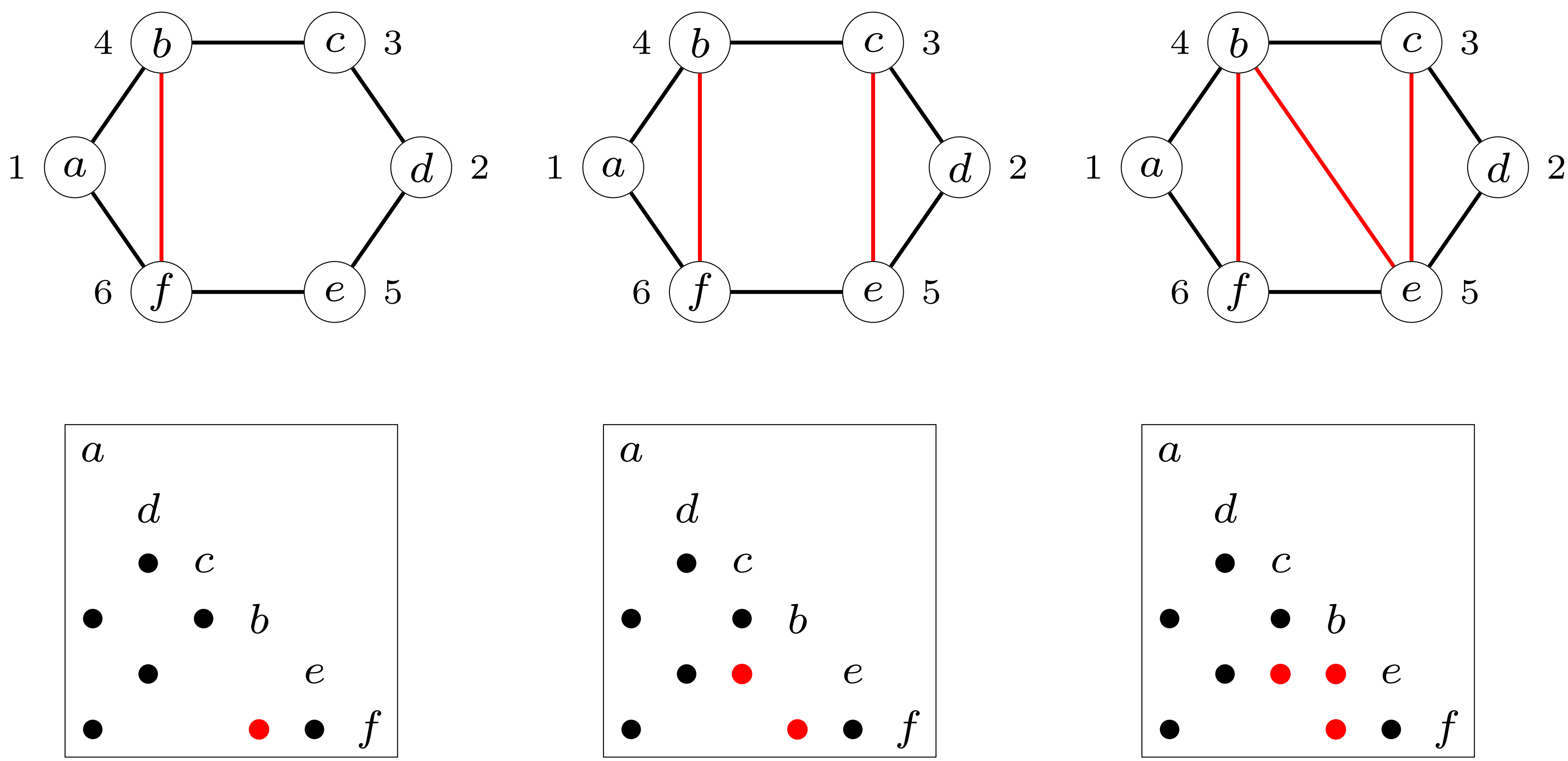
I. Chordal graphs

- chordal graphs
- perfect elimination
- **triangularization**

Triangularization

construct filled graph from ordered graph $G_\sigma = (V, E, \sigma)$:

- enumerate vertices $v = \sigma(i)$ for $i = 1, \ldots, |V|$
- in cycle i, add edges to make higher neighborhood $\mathrm{adj}^+(v)$ complete

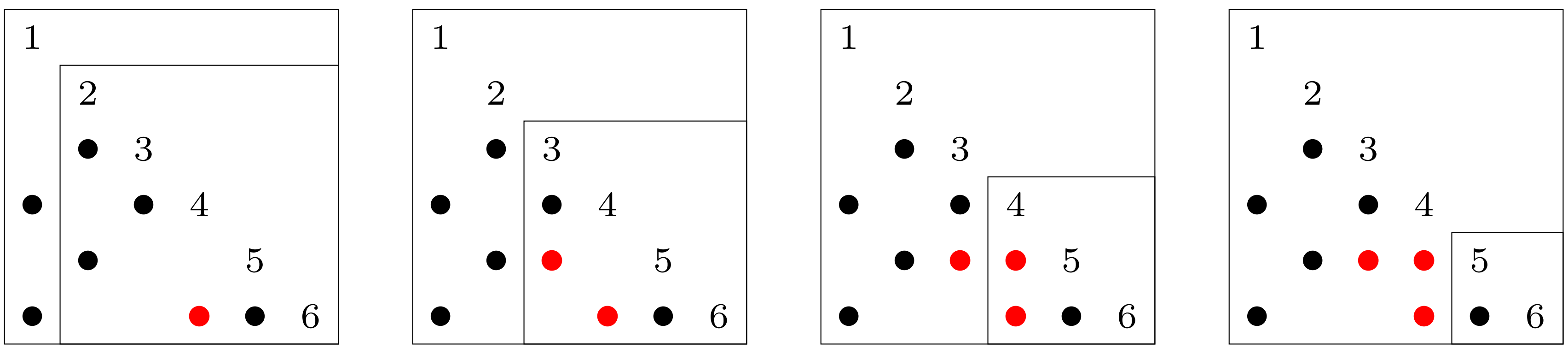


Example: Cholesky factorization

to factor A as $A = LDL^T$, factor A as

$$
\begin{aligned}
A = \begin{bmatrix} a & b^T \\ b & C \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ (1/a)b & I \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & C - (1/a)bb^T \end{bmatrix} \begin{bmatrix} 1 & (1/a)b^T \\ 0 & I \end{bmatrix} \\
&= \hat{L}\hat{D}\hat{L}^T
\end{aligned}
$$

and factor $C - (1/a)bb^T$



pattern of $\hat{L} + \hat{D} + \hat{L}^T$ during iteration

Fill-in

- the added edges during triangularization are called fill-in F_σ
- ordering σ is a perfect elimination ordering for (V, E) if $F_\sigma = \emptyset$
- the graph $(V, E \cup F_\sigma)$ is a chordal extension of (V, E)

Minimum ordering: minimizes fill-in $|F_\sigma|$; NP-complete (Yannakakis 1981)

Minimal ordering: there exists no ordering σ' with $F_{\sigma'} \subset F_\sigma$

- perfect elimination ordering if graph is chordal
- several algorithms with linear complexity (in $|V| + |E|$)
- maximum cardinality search (Tarjan and Yannakakis 1984), several others

Non-minimal heuristics: often give smaller $|F_\sigma|$ than minimal ordering

Analysis of filled graph

algorithms for analyzing filled graph of (V, E, σ) without constructing it:

- constructing elimination tree
- calculating monotone (higher and lower) degrees
- calculating number of filled edges
- finding clique representatives
- finding supernodes, supernodal elimination tree

complexity is linear or nearly linear in $|V| + |E|$ ($i.e.$, size of original graph)

(Liu 1990, Gilbert, Ng, Peyton 1994, Davis 2006)

Overview

Lecture 1

I. Chordal graphs

II. Sparse matrices

Lecture 2

III. Non-symmetric interior-point methods

IV. Decomposition

V. Semidefinite relaxations

II. Sparse matrices

- sparse matrix cones
- chordal decomposition
- logarithmic barriers

Sparse matrix cones

two matrix cones in $\mathbf{S}^n_E$ (symmetric of order n with sparsity pattern E)

- positive semidefinite matrices

$$\mathbf{S}^n_{E,+} = \{X \in \mathbf{S}^n_E \mid X \succeq 0\}$$

- matrices with a positive semidefinite completion

$$\mathbf{S}^n_{E,\mathrm{c}} = \{\Pi_E(X) \mid X \succeq 0\}$$

Π_E is projection on $\mathbf{S}^n_E$:

$$\Pi_E(X)_{ij} = \begin{cases} X_{ij} & \text{if } i = j \text{ or } \{i,j\} \in E \\ 0 & \text{otherwise} \end{cases}$$

Example

for general E, cones are not equal

Example: tridiagonal matrices of order 3 ($\mathbf{S}^3_E = \{A \in \mathbf{S}^3 \mid A_{31} = 0\}$

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

- A is not positive semidefinite ($A \notin \mathbf{S}^n_{E,+}$)
- A has a positive semidefinite completion ($A \notin \mathbf{S}^n_{E,\mathrm{c}}$)

$$A = \Pi_E(\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix})$$

Properties

the two cones are convex, closed, pointed, with nonempty interior

Positive semidefinite cone

$$\mathbf{S}^n_{E,+} = \mathbf{S}^n_+ \cap \mathbf{S}^n_E$$

- intersection of closed convex cone and subspace, hence closed convex
- identity matrix is in the interior
- pointed: $X \in \mathbf{S}^n_{E,+}$ and $-X \in \mathbf{S}^n_{E,+}$ only if $X = 0$

Positive semidefinite completable cone

$$\mathbf{S}^n_{E,\mathrm{c}} = \Pi_E(\mathbf{S}^n_+)$$

- projection on a subspace of a convex cone with nonempty interior
- closed and pointed because $\Pi_E(X) = 0$, $X \succeq 0$ only if $X = 0$

Two cones are dual cones

- dual of positive semidefinite completable cone is p.s.d. cone

$$\begin{aligned}
(\mathbf{S}^n_{E,\mathrm{c}})^* &= \{Y \in \mathbf{S}^n_E \mid \mathbf{tr}(XY) \geq 0 \ \ \forall X \in \mathbf{S}^n_{E,\mathrm{c}}\} \\
&= \{Y \in \mathbf{S}^n_E \mid \mathbf{tr}\left(Y \Pi_E(W)\right) \geq 0 \ \ \forall W \succeq 0\} \\
&= \{Y \in \mathbf{S}^n_E \mid \mathbf{tr}(YW) \geq 0 \ \ \forall W \succeq 0\} \\
&= \mathbf{S}^n_{E,+}
\end{aligned}$$

- dual of p.s.d. cone is positive semidefinite completable cone

$$(\mathbf{S}^n_{E,+})^* = \mathrm{cl}(\mathbf{S}^n_{E,\mathrm{c}}) = \mathbf{S}^n_{E,\mathrm{c}}$$

II. Sparse matrices

- sparse matrix cones
- **chordal decomposition**
- logarithmic barriers

Notation

Index set: ordered list of distinct elements of $V = \{1, 2, \ldots, n\}$

Selection matrix: if β is an index set, P_β is 0-1 matrix of size $|\beta| \times n$

$$(P_\beta)_{ij} = 1 \quad \text{if } j = \beta_i, \qquad (P_\beta)_{ij} = 0 \quad \text{otherwise}$$

a permutation matrix if $|\beta| = n$

- used to select subvectors or principal submatrices:

$$P_\beta x = x_\beta, \qquad P_\beta X P_\beta^T = X_{\beta\beta}$$

- adjoint defines subvector or submatrix in otherwise zero vector or matrix

$$(P_\beta^T y)_i = \begin{cases} y_j & j = \beta_i \\ 0 & j \notin \beta \end{cases} \qquad (P_\beta^T Y P_\beta)_{kl} = \begin{cases} Y_{ij} & i = \beta_k, j \in \beta_l \\ 0 & (i,j) \notin \beta \times \beta \end{cases}$$

Cholesky factorization and chordal sparsity

$$P_\sigma A P_\sigma^T = LDL^T$$

P_σ a permutation, L unit lower triangular, D positive diagonal

- if $A \in \mathbf{S}_E^n$ and σ is a perfect elimination ordering for E, then

$$P_\sigma^T (L + L^T) P_\sigma \in \mathbf{S}_E^n$$

  A has a 'zero fill' Cholesky factorization

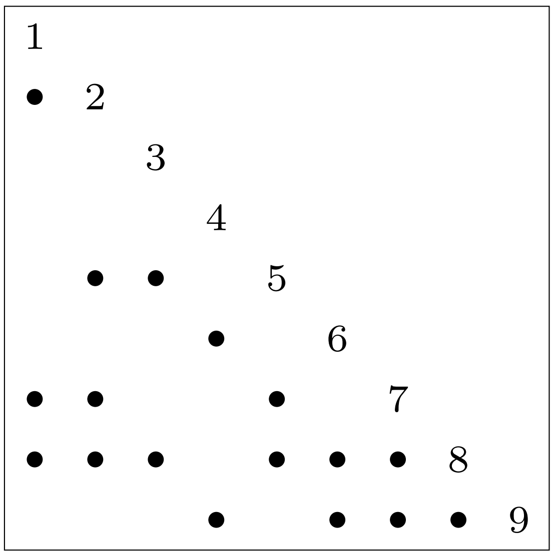
- if E is not chordal, then for every σ there exist p.d. $A \in \mathbf{S}_E^n$ for which

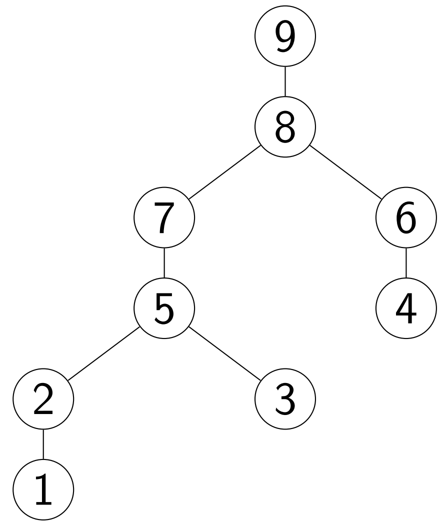
$$P_\sigma^T (L + L^T) P_\sigma \notin \mathbf{S}_E^n$$

(Rose 1070)

Assumptions and notation

E is chordal with perfect elimination ordering $\sigma = (1, 2, \ldots, n)$





- γ_j, η_j: index sets with elements of $\mathrm{col}(j)$, $\mathrm{adj}^+(j)$; for example,

$$\gamma_2 = (2, 5, 7, 8), \qquad \eta_2 = (5, 7, 8)$$

- hence, $\eta_k \subseteq \gamma_j$ if if j is the parent of k in elimination tree
- $V_\mathrm{r} \subseteq V$ is set of clique representative vertices (i for which γ_i is a clique)

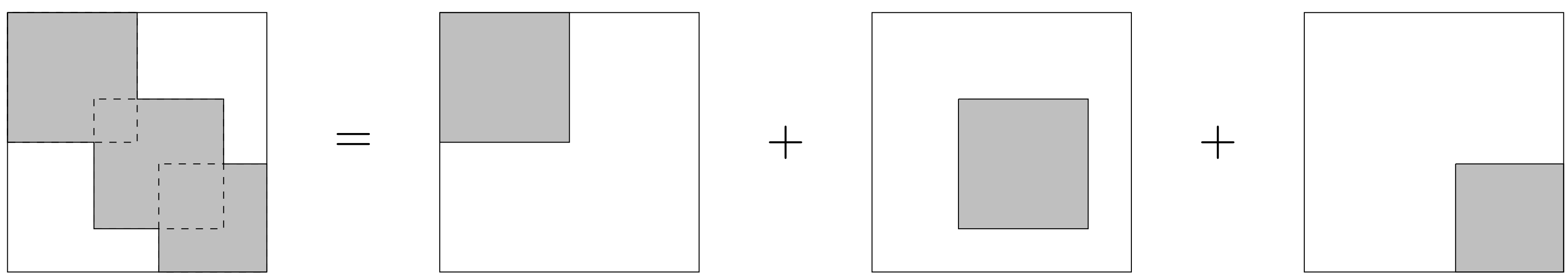
Decomposition of chordal p.s.d. cone

$A \in \mathbf{S}^n_E$ is positive semidefinite if and only if it can be expressed as

$$A = \sum_{i \in V_\mathrm{r}} P_{\gamma_i}^T B_i P_{\gamma_i} \quad \text{with} \quad B_i \succeq 0, \quad i \in V_\mathrm{r}$$

(Griewank and Toint 1984, Agler, Helton, McCullough, Rodman 1988)

Example (three cliques γ_1, γ_j, γ_k)



$A \succeq 0$ $\qquad$ $P_{\gamma_1}^T B_1 P_{\gamma_1} \succeq 0$ $\qquad$ $P_{\gamma_j}^T B_j P_{\gamma_j} \succeq 0$ $\qquad$ $P_{\gamma_k}^T B_k P_{\gamma_k} \succeq 0$

Decomposition from Cholesky factorization

$$A = LDL^T = \sum_{j=1}^{n} D_{jj} L_j L_j^T$$

- use maximal supernode partition $\mathrm{sn}(i)$ to group outer products

$$A = \sum_{i \in V_\mathrm{r}} \sum_{j \in \mathrm{sn}(i)} D_{jj} P_{\gamma_j}^T L_{\gamma_j j} L_{\gamma_j j} P_{\gamma_j}$$

- use property $\gamma_j \subset \gamma_i$ for $j \in \mathrm{sn}(i)$

$$A = \sum_{i \in V_\mathrm{r}} P_{\gamma_i}^T \left(\sum_{j \in \mathrm{sn}(i)} D_{jj} L_{\gamma_i j} L_{\gamma_i j} \right) P_{\gamma_i} = \sum_{i \in V_\mathrm{r}} P_{\gamma_i}^T B_i P_{\gamma_i}$$

where $B_i = \sum_{j \in \mathrm{sn}(i)} D_{jj} L_{\gamma_i j} L_{\gamma_i j}^T$

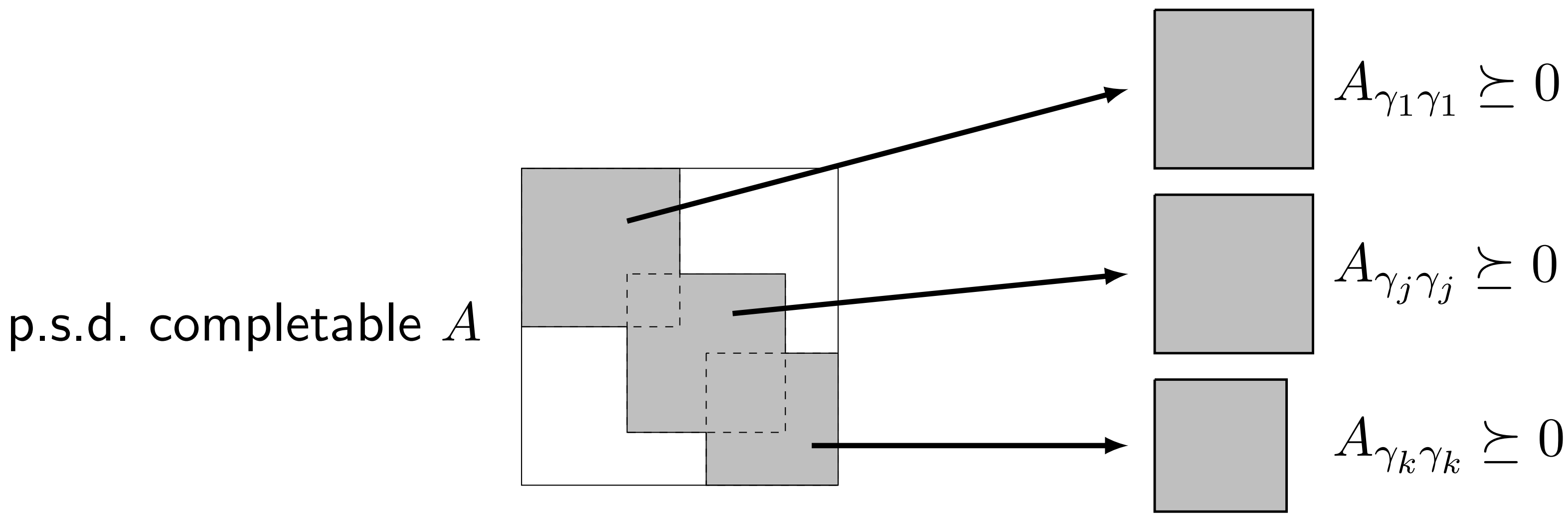
Decomposition of chordal p.s.d. completable cone

$A \in \mathbf{S}^n_E$ has a positive semidefinite completion if and only if

$$A_{\gamma_i\gamma_i} \succeq 0, \quad i \in V_\mathrm{r}$$

(Grone, Johnson, Sá, Wolkowicz, 1984)

Example (three cliques γ_1, γ_j, γ_k)



Proof from duality

- duality of the p.s.d. cone and p.s.d. completable cone

$$A \in \mathbf{S}^n_{E,\mathrm{c}} \qquad \Longleftrightarrow \qquad \mathbf{tr}(AY) \geq 0 \quad \forall Y \in \mathbf{S}^n_{E,+}$$

- use chordal decompositionn of p.s.d. cone

$$\begin{aligned} A \in \mathbf{S}^n_{E,\mathrm{c}} \quad &\Longleftrightarrow \quad \sum_{i \in V_\mathrm{r}} \mathbf{tr}\left(A P_{\gamma_i}^T B_i P_{\gamma_i}\right) \quad \forall B_i \succeq 0 \\ &\Longleftrightarrow \quad \sum_{i \in V_\mathrm{r}} \mathbf{tr}\left(P_{\gamma_i} A P_{\gamma_i}^T B_i\right) \quad \forall B_i \succeq 0 \\ &\Longleftrightarrow \quad P_{\gamma_i} A P_{\gamma_i}^T \succeq 0 \quad \forall i \in V_\mathrm{r} \end{aligned}$$

Positive semidefinite completion

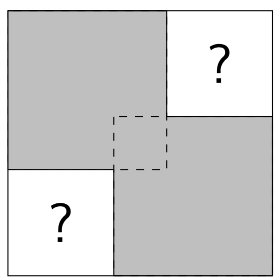
given $A \in \mathbf{S}^n_{E,\mathrm{c}}$ find X such that

$$A = \Pi_E(X), \qquad X \succeq 0$$

(Grone, Johnson, Sá, Wolkowicz, 1984)

Algorithms

- recursion on clique tree or (supernodal) elimination tree
- at each step, consider simple two-block completion problem



Euclidean distance matrices

Definition: $A \in \mathbf{S}^n$ is a Euclidean distance matrix if it can be written as

$$A_{ij} = \|x_i - x_j\|_2^2$$

for a set of vectors $x_1, \ldots, x_n$

Semidefinite parameterization: A is EDM if and only if and

$$A_{ij} = G_{ii} - 2G_{ij} + G_{jj}, \quad G \succeq 0$$

for some G (the Gram matrix with elements $G_{ij} = x_i^T x_j$)

Euclidean distance matrix completion

given $A \in \mathbf{S}^n_E$ find an EDM X such that

$$A = \Pi_E(X)$$

EDM completion with chordal pattern E

$A \in \mathbf{S}^n_E$ has an EDM completion if

$$A_{\gamma_i \gamma_i} \text{ is a Euclidean distance matrix,} \quad \text{for all cliques } \gamma_i$$

and completion can be constructed by a recursion over the elimination tree

(Bakonyi and Johnson 1995)

II. Sparse matrices

- sparse matrix cones
- chordal decomposition
- **logarithmic barriers**

References

- M.S. Andersen, J. Dahl, L. Vandenberghe, "Logarithmic barriers for sparse matrix cones", *Optimization Methods and Software* (2013)
- CHOMPACK Library for chordal matrix computations. `cvxopt.github.io/chompack`

Logarithmic barrier for p.s.d. cone

$$\phi(X) = -\log\det X, \qquad \mathbf{dom}\,\phi = \mathbf{int}\,\mathbf{S}^n_{E,+}$$

Gradient

$$\nabla\phi(X) = -\Pi_E(X^{-1})$$

- Π_E is projection on $\mathbf{S}^n_E$
- evaluation requires entries of X^{-1} on diagonal and for $\{i,j\} \in E$

Hessian: for arbitrary $Y \in \mathbf{S}^n_E$,

$$\nabla^2\phi(X)[Y] = \frac{d}{dt}\phi(x+tY)\bigg|_{t=0} = \Pi_E\left(X^{-1}YX^{-1}\right)$$

Logarithmic barrier for p.s.d. completable cone

Dual of log-det barrier

$$\phi_*(S) = \sup_X \left(-\,\mathbf{tr}(SX) - \phi(X)\right)$$

- optimal $\widehat{X}$ satifies $\Pi_E(\widehat{X}^{-1}) = S$
- $Z = \widehat{X}^{-1}$ solves

$$\begin{array}{ll} \text{maximize} & \log\det Z \\ \text{subject to} & \Pi_E(Z) = S \end{array}$$

$\widehat{X}^{-1}$ is maximum determinant positive definite completion of S

Gradient and Hessian

$$\nabla\phi_*(S) = -\widehat{X}, \qquad \nabla^2\phi_*(S) = \nabla^2\phi(\widehat{X})^{-1}$$

Multifrontal Cholesky factorization

computes factorization $X = LDL^T$ by recursion on elimination tree

- for each vertex i, (temporarily) store a dense **update matrix**

$$U_i = -\sum_{\text{descendants } k \text{ of } i} D_{kk} L_{\eta_i k} L_{\eta_i k}^T$$

(η_i is index set for nonzeros in column i of L below diagonal)

- U_j and column j of factorization (D_{jj}, $L_{\eta_j j}$) computed recursively from

$$\begin{bmatrix} X_{jj} & X_{\eta_j j}^T \\ X_{\eta_j j} & -U_j \end{bmatrix} + P_{\gamma_j} \left(\sum_{i \text{ is child of } j} P_{\eta_i}^T U_i P_{\eta_i} \right) P_{\gamma_j}^T = D_{jj} \begin{bmatrix} 1 \\ L_{\eta_j j} \end{bmatrix} \begin{bmatrix} 1 \\ L_{\eta_j j} \end{bmatrix}^T$$

(Duff and Reid 1983)

Multifrontal algorithm

iterate over etree in topological order (children visited before their parents)

- at vertex j, first form **frontal matrix**

$$\left[\begin{array}{cc} F_{11} & F_{12} \\ F_{21} & F_{22} \end{array}\right] = \left[\begin{array}{cc} X_{jj} & X_{\eta_j j}^T \\ X_{\eta_j j} & 0 \end{array}\right] + P_{\gamma_j} \left(\sum_{\text{children } i \text{ of } j} P_{\eta_i}^T U_i P_{\eta_i} \right) P_{\gamma_j}^T$$

- then execute a **pivoting step**

$$D_{jj} := F_{11}, \qquad L_{\eta_j j} := \frac{1}{D_{jj}} F_{21}, \qquad U_j := F_{22} - D_{jj} L_{\eta_j j} L_{\eta_j j}^T$$

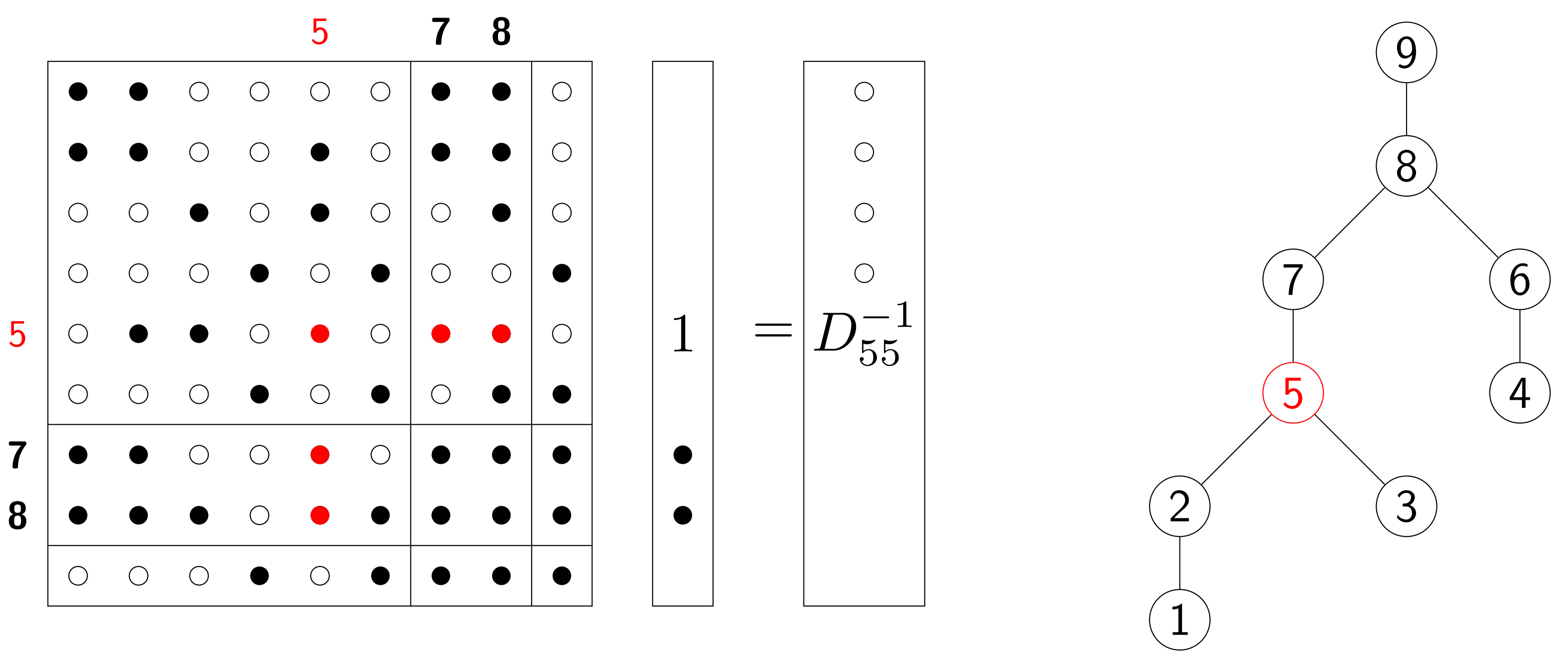
pivoting step is a dense level 2 BLAS operation (matrix-vector operation)

Partial inverse $\Pi_E(X^{-1})$

inverse $S = X^{-1}$ satisfies $SL = L^{-T}D^{-1}$:

$$\left[\begin{array}{cc} S_{jj} & S_{\eta_j j}^T \\ S_{\eta_j j} & S_{\eta_j \eta_j} \end{array}\right] \left[\begin{array}{c} 1 \\ L_{\eta_j j} \end{array}\right] = \left[\begin{array}{c} 1/D_{jj} \\ 0 \end{array}\right], \qquad j = 1, \ldots, n$$

this allows us to compute $S_{\eta_j j}$, S_{jj} from $S_{\eta_j \eta_j}$ (and L, D)



Algorithm for partial inverse

iterate over vertices of elimination tree in reverse topological order

- at vertex j, compute

$$S_{\eta_j j} := -V_j L_{\eta_j j}, \qquad S_{jj} := \frac{1}{D_{jj}} - S_{\eta_j j}^T L_{\eta_j j}$$

  $V_j = S_{\eta_j \eta_j}$ is dense 'update matrix'

- for each child i of j, form

$$V_i := P_{\eta_i} P_{\gamma_j}^T \begin{bmatrix} S_{jj} & S_{\eta_j j}^T \\ S_{\eta_j j} & V_j \end{bmatrix} P_{\gamma_j} P_{\eta_i}^T$$

  (multiplications with $P_{\eta_i} P_{\gamma_j}^T$ extract a principal submatrix)

main operation is dense matrix-vector multiplication $V_j L_{\eta_j j}$

Matrix completion

given $S \in \mathbf{S}^n_E$, solve nonlinear equation

$$\Pi_E(X^{-1}) = S$$

with variable $X \in \mathbf{S}^n_E$

- solution $\widehat{X}$ is inverse of maximum determinant completion of S
- $\widehat{X} = -\nabla \phi_*(S)$ (barrier of p.s.d. completable cone)
- factorization $\widehat{X} = LDL^T$ follows from equation used for partial inverse:

$$\begin{bmatrix} S_{jj} & S_{\eta_j j}^T \\ S_{\eta_j j} & S_{\eta_j \eta_j} \end{bmatrix} \begin{bmatrix} 1 \\ L_{\eta_j j} \end{bmatrix} = \begin{bmatrix} 1/D_{jj} \\ 0 \end{bmatrix}, \qquad j = 1, \ldots, n$$

Algorithm for completion

iterate over vertices of elimination tree in reverse topological order

- at vertex j, compute

$$L_{\eta_j j} := -V_j^{-1} S_{\eta_j j}, \qquad D_{jj} := (S_{jj} - S_{\eta_j j}^T L_{\eta_j j})^{-1}$$

  $V_j = S_{\eta_j \eta_j}$ is dense 'update matrix'

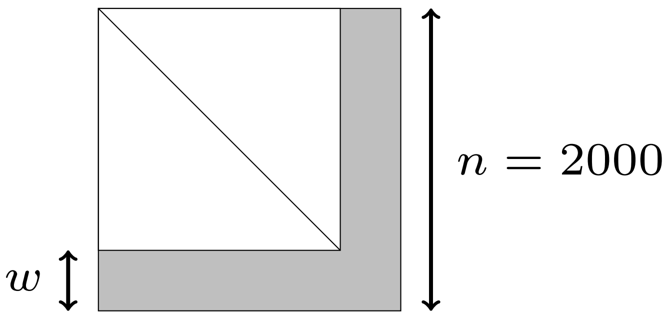
- for each child i of j, form

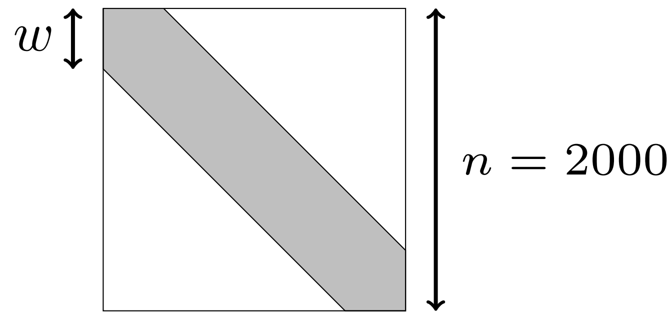
$$V_i := P_{\eta_i}^T P_{\gamma_j}^T \begin{bmatrix} S_{jj} & S_{Jj}^T \\ S_{Jj} & V_j \end{bmatrix} P_{\gamma_j} P_{\eta_i}^T$$

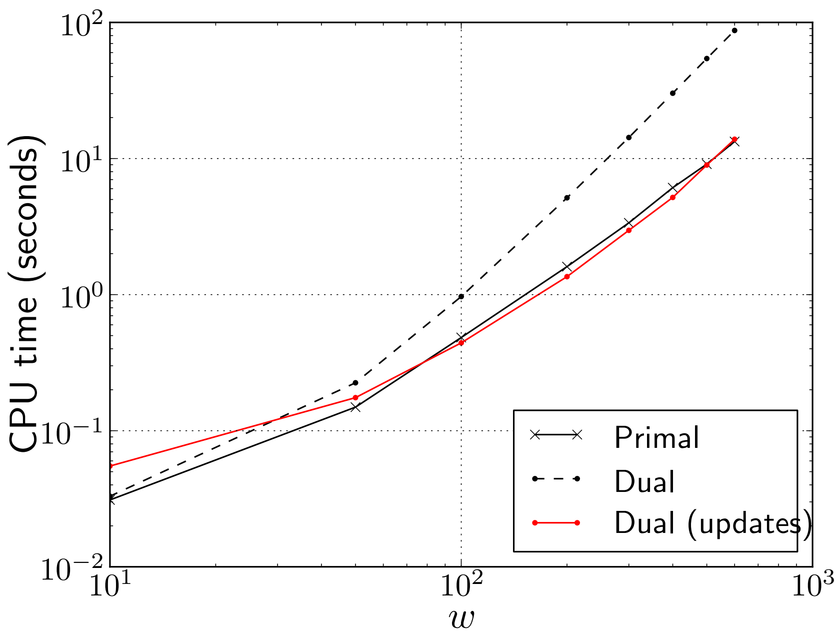
main step is solution of dense system $V_j L_{\eta_j j} = -S_{\eta_j j}$

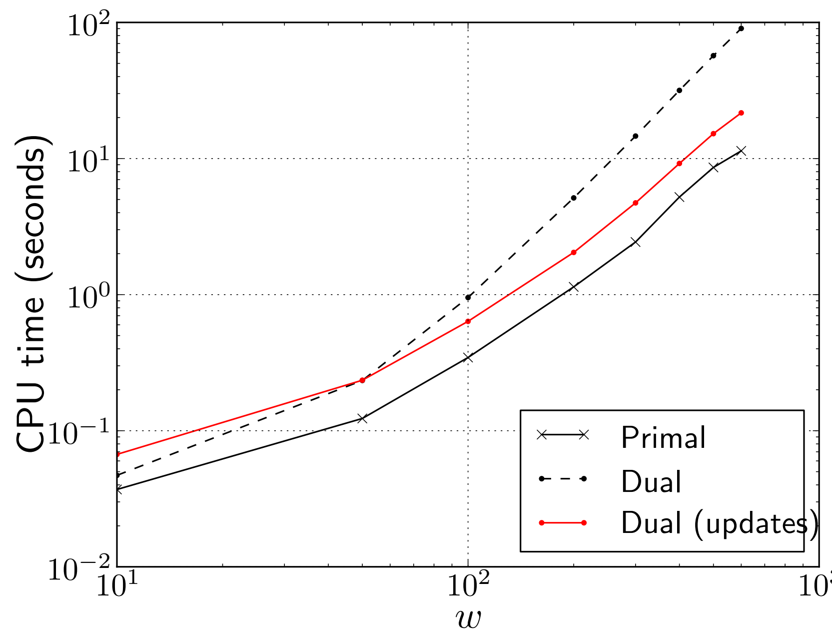
improvement: propagate factorization of V_j, make low-rank updates

Primal/dual gradients for arrow and band patterns







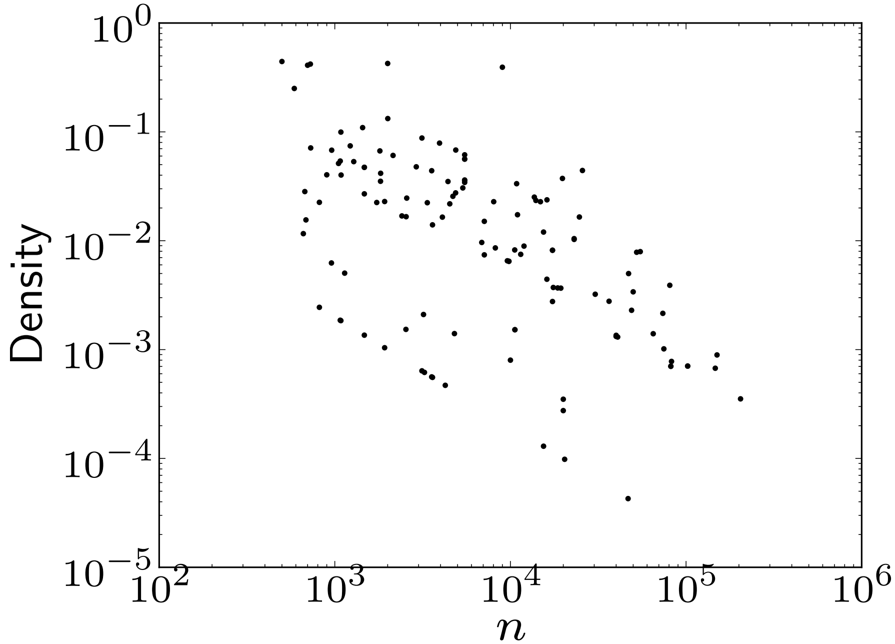


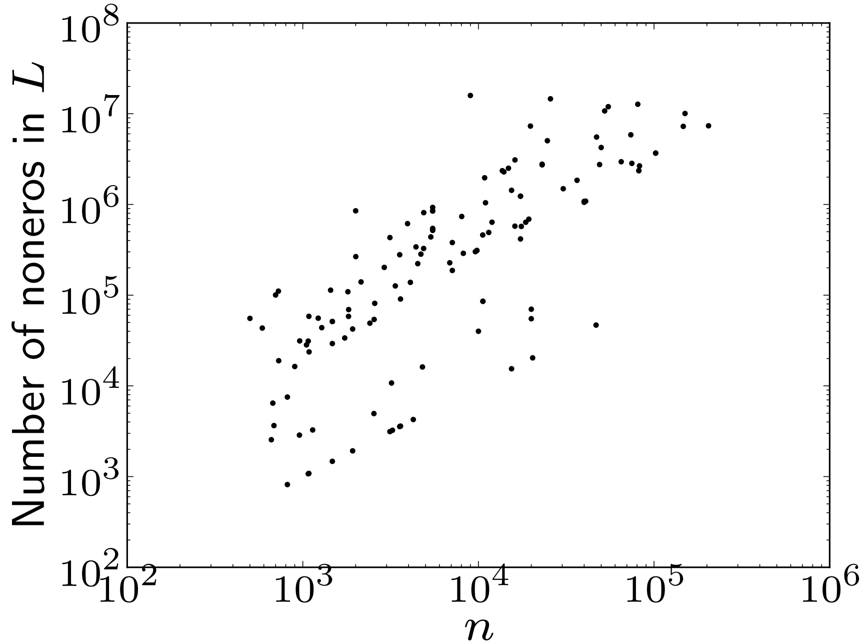
cost is $O(nw^2)$ for primal gradient, dual gradient with low-rank updates

General patterns

128 sparsity patterns from University of Florida sparse matrix collection

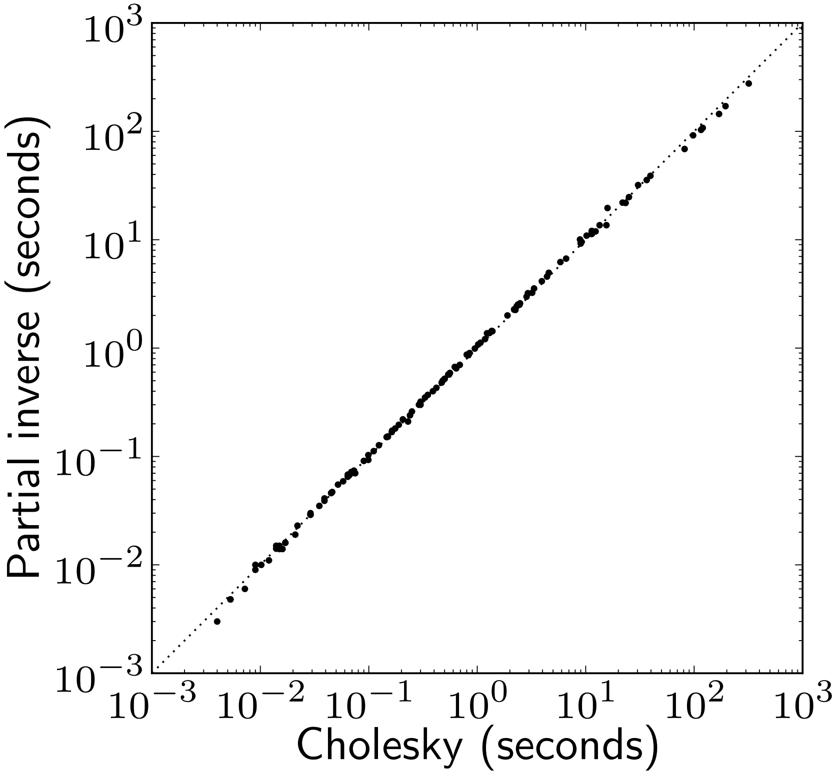
- n ranges between 500 and 204,316
- between 817 and 15,894,180 nonzeros in L



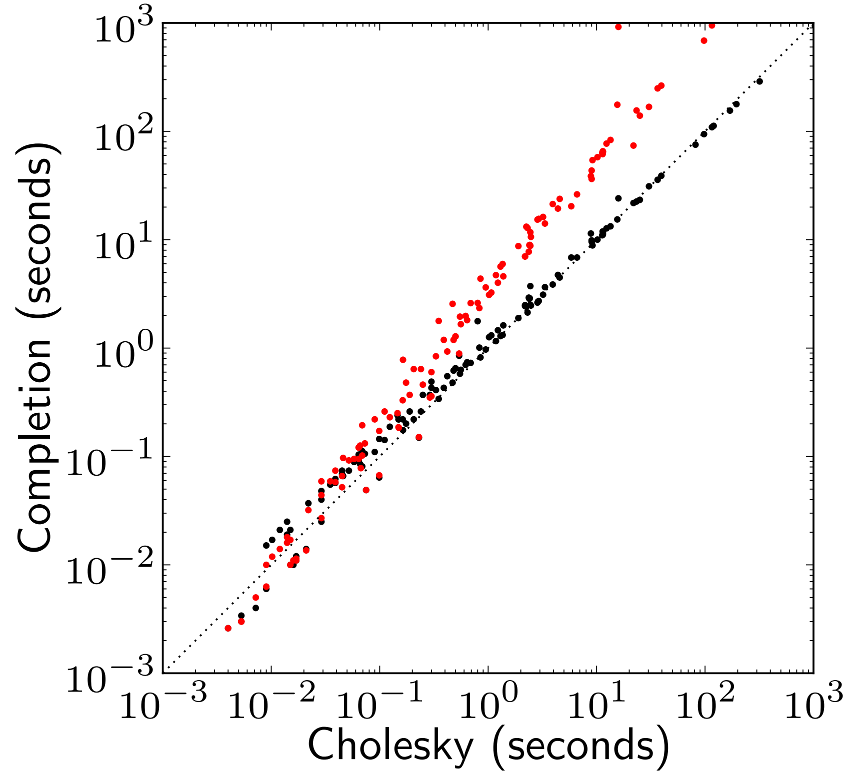


Primal/dual gradient versus Cholesky factorization

primal gradient (partial inverse)



dual gradient (completion)



dual gradient with/without
low-rank updates

Hessian

computing $\nabla\phi(X) = -\Pi_E(X^{-1})$ requires two iterations over etree:

- Cholesky factorization $X = LDL^T$ (iteration in topological order)
- partial inverse from D, L (iteration in reverse topological order)

Hessian evaluation: linearize the two recursions in the gradient algorithm

$$\nabla^2\phi(X)[Y] = \Pi_E(X^{-1}YX^{-1}) = -\frac{d}{dt}\Pi_E(X+tY)^{-1}\bigg|_{t=0}$$

- linearized recursions are adjoints: factorization $\nabla^2\phi(X) = \mathcal{R}_X^{\mathrm{adj}} \circ \mathcal{R}_X$
- factored Hessian useful in interior-point methods to form

$$\mathbf{tr}\left(A_i \nabla^2\phi(X)[A_j]\right) = \mathbf{tr}\left(\mathcal{R}_X(A_i)\mathcal{R}_X(A_j)\right), \quad i,j = 1,\ldots,m$$

Hessian evaluation with sparse arguments

- sparsity patterns from U. Florida sparse matrix collection
- Y dense (w.r.t. E) or sparse (2 randomly chosen nonzero positions)

cost of evaluating $\mathcal{R}_X(Y)$

Sparsity pattern	n	Y dense (seconds)	Y sparse (seconds)	Ratio
HB/plat1919	1919	0.17	0.05	3.7
HB/bcsstk13	2003	1.40	0.36	3.9
HB/lshp3025	3025	0.21	0.05	4.6
Boing/nasa4704	4704	1.05	0.27	4.0
TKK/g3rmt3m3	5357	1.43	0.35	4.1
Schenk_IBMNA/c-36	7479	0.37	0.06	6.1
Wang/swang1	10800	7.05	1.72	4.1
ACUSIM/Pres_Poisson	14822	17.65	4.35	4.1
GHS_psdef/wathen100	30401	6.05	1.38	4.4

Supernodal multifrontal algorithms

- process columns in each supernode $\mathrm{sn}(j)$ as one block column
- iteration over elimination tree replaced by iteration over clique tree or supernodal elimination tree

Advantages

- level 3 BLAS (matrix-matrix) operations replace level 2 BLAS
- fewer frontal matrices to construct

Supernodal factorization and primal gradient

128 sparsity patterns from U. Florida collection