Chordal graphs
MPRI 2017–2018

Michel Habib habib@irif.fr
http://www.irif.fr/~habib

Sophie Germain, septembre 2017

Schedule

Definition

A graph is chordal iff it has no chordless cycle of length ≥ 4 or equivalently it has no induced cycle of length ≥ 4.

- Chordal graphs are hereditary
- Interval graphs are chordal

Applications

- Many NP-complete problems for general graphs are polynomial for chordal graphs.
- Graph theory :
  Treewidth (resp. pathwidth) are very important graph parameters that measure distance from a chordal graph (resp. interval graph).
- Perfect phylogeny [1]

1. In fact chordal graphs were first defined in a biological modelisation perspective !

G.A. Dirac,
On rigid circuit graphs,
Abh. Math. Sem. Univ. Hamburg, 38 (1961), pp. 71–76

About Representations

- Interval graphs are chordal graphs
- How can we represent chordal graphs ?
- As an intersection of some family ?
- This family must generalize intervals on a line

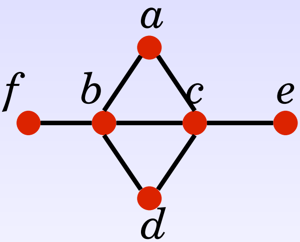
Fundamental objects to play with

- Maximal Cliques under inclusion
- We can have exponentially many maximal cliques :
  A complete graph K_n in which each vertex is replaced by a pair of false twins has
  exactly 2^n maximal cliques.
- Every edge is included in at least one maximal clique

Minimal Separators

A subset of vertices S is a minimal separator if G
if there exist $a, b \in G$ with $ab \notin G$, such that a and b are not connected in $G - S$.
and S is minimal for inclusion with this property .

An example



3 minimal separators $\{b\}$ for f and a, $\{c\}$ for a and e and $\{b, c\}$ for a and d.

If $G = (V, E)$ is connected then for every $a, b \in V$ such that
$ab \notin E$
then there exists at least one minimal separator.
But there could be an exponential number of minimal separators.

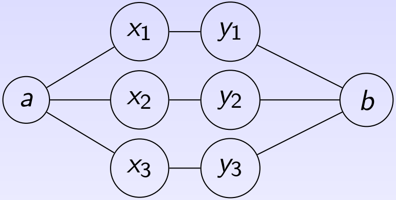


FIGURE: A graph with 2^n minimal ab-separators

VIN : Maximal Clique trees

A maximal clique tree (clique tree for short) is a tree T that satisfies the following three conditions :

- Vertices of T are associated with the maximal cliques of G
- Edges of T correspond to minimal separators.
- For any vertex $x \in G$, the cliques containing x yield a subtree of T.

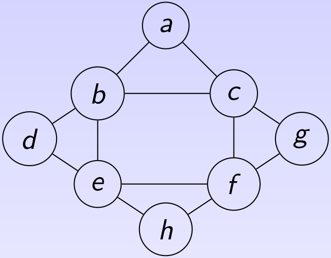
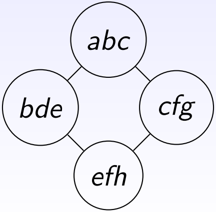


FIGURE: A graph and the intersection graph of its maximal cliques



No tree on these 4 maximal cliques satisfies the third condition of maximal clique trees.

Helly Property

Definition

A subset family $\{T_i\}_{i \in I}$ satisfies Helly property if
$\forall J \subseteq I$ et $\forall i, j \in J$ $T_i \cap T_j \neq \emptyset$ implies $\cap_{i \in J} T_i \neq \emptyset$

Exercise

Subtrees in a tree satisfy Helly property.

Démonstration.

Suppose not. Consider a family of subtrees that pairwise intersect. For each vertex x of the tree T, if x belongs to every subtree of the family, it contradicts the hypothesis. Therefore at least one subtree does not contain x. If the subtrees belongs to two different components of T-x this would contradict the pairwise intersection of the subtrees. Therefore all the subtrees are in exactly one component of T-x (N.B. some subtrees may contain x).
Direct exactly one edge of T from x to this component.
This yields a directed graph G, which has exactly n vertices and n directed edges. Since T is a tree, it contains no cycle, therefore it must exist a pair of symmetric edges in G, which contradicts the pairwise intersection.

□

Main chordal graphs characterization theorem

Using results of Dirac 1961, Fulkerson, Gross 1965, Buneman 1974, Gavril 1974 and Rose, Tarjan and Lueker 1976 :

For a connected graph, the following statements are equivalent and characterize chordal graphs :

(0) G has no induced cycle of length > 3

(i) G admits a simplicial elimination scheme

(ii) Every minimal separator is a clique

(iii) G admits a maximal clique tree.

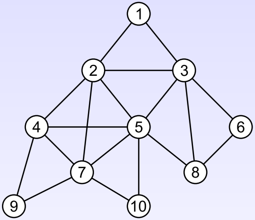
(iv) G is the intersection graph of subtrees in a tree.

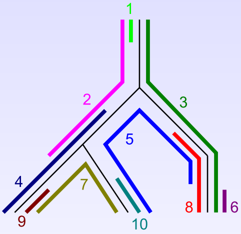
(v) Any MNS (LexBFS, LexDFS, MCS) provides a simplicial elimination scheme.

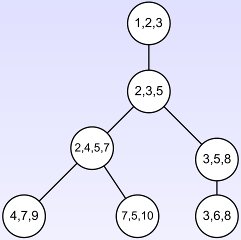
Two subtrees intersect iff they have at least one vertex in common, not necessarily an edge in common.

By no way, these representations can be uniquely defined !

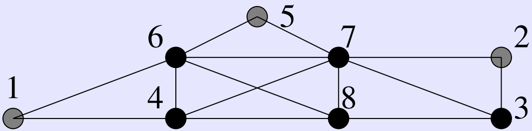
An example







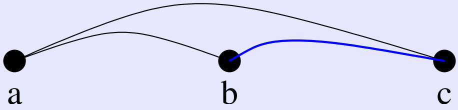
Chordal graph



A vertex is simplicial if its neighbourhood is a clique.

Simplicial elimination scheme

$\sigma = [x_1 \ldots x_i \ldots x_n]$ is a simplicial elimination scheme if x_i is simplicial in the subgraph $G_i = G[\{x_i \ldots x_n\}]$

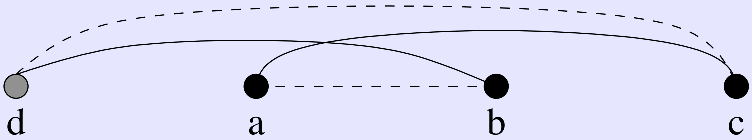


Lexicographic Breadth First Search (LBFS)

Data: a graph $G = (V, E)$ and a start vertex s
Result: an ordering σ of V
Assign the label $\emptyset$ to all vertices
$label(s) \leftarrow \{n\}$
for $i \leftarrow n$ **à** 1 **do**
    Pick an unumbered vertex v with lexicographically largest label
    $\sigma(i) \leftarrow v$
    foreach *unnumbered vertex* w *adjacent to* v **do**
        $label(w) \leftarrow label(w).\{i\}$
    end
end

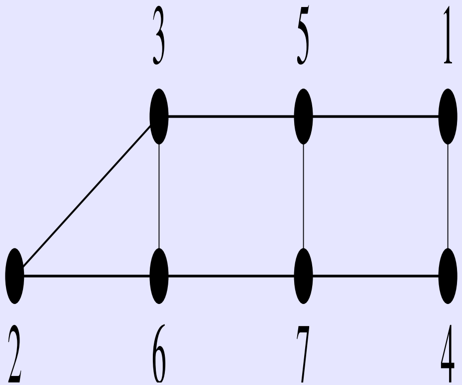
Property (LexB)

For an ordering σ on V, if $a <_\sigma b <_\sigma c$ and $ac \in E$ and $ab \notin E$, then it must exist a vertex d such that $d <_\sigma a$ et $db \in E$ et $dc \notin E$.



Theorem

For a graph $G = (V, E)$, an ordering σ on V is a LBFS of G iff σ satisfies property (LexB).

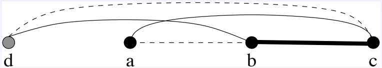

3
5
1
2
6
7
4

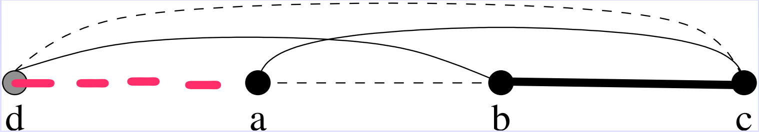
Theorem [Tarjan et Yannakakis, 1984]

G is chordal iff every LexBFS ordering yields a simplicial elimination scheme.

Proof :

Let c be a non simplicial vertex.
There exist $a < b \in N(c)$ avec $ab \notin E$.
Using characterization of LexBFS orderings, it exists $d < a$ with $db \in E$ and $dc \notin E$. Since G is chordal, necessarily $ad \notin E$.





But then from the triple d, a, b, it exists $d' < d$ with $d'a \in E$ and $d'b \notin E$. Furthermore $d'd \notin E$ ...
And using the triple d', d, a , we start an infinite chain .....

Remark

Most of the proofs based on some characteristic ordering of the vertices are like that, with no extra reference to the algorithm itself.

Main chordal graphs characterization theorem

Using results of Dirac 1961, Fulkerson, Gross 1965, Buneman 1974, Gavril 1974 and Rose, Tarjan and Lueker 1976 :

For a connected graph, the following statements are equivalent and characterize chordal graphs :

(0) G has no induced cycle of length > 3

(i) G admits a simplicial elimination scheme

(ii) Every minimal separator is a clique

(iii) G admits a maximal clique tree.

(iv) G is the intersection graph of subtrees in a tree.

(v) Any MNS (LexBFS, LexDFS, MCS) provides a simplicial elimination scheme.

Back to the proof of the main chordal characterization theorem

- Clearly (iii) implies (iv)
- For the converse, each vertex of the tree corresponds to a clique in G.
  But the tree has to be pruned of all its unnecessary nodes, until in each node some subtree starts or ends. Then nodes correspond to maximal cliques.
- We need now to relate these new conditions to chordal graphs.
  (iii) implies (i) since a maximal clique tree yields a simplicial elemination scheme.
  (iv) implies chordal since a cycle without a chord generates a cycle in the tree.
  (iv) implies (ii) since each edge of the tree corresponds to a minimal separator which is a clique as the intersection of two cliques.

from (i) to (iv)

Démonstration.

By induction on the number of vertices. Let x be a simplicial vertex of G.
By induction $G - x$ can be represented with a family of subtrees on a tree T.
$N(x)$ is a clique and using Helly property, the subtrees corresponding to $N(x)$ have a vertex in common α.
To represent G we just add a pending vertex β adjacent to α.
x being represented by a path restricted to the vertex β, and we add to all the subtrees corresponding to vertices in $N(x)$ the edge $\alpha\beta$. □

Exercises

1. Can we use efficiently this representation of chordal graphs as intersection of subtrees ?
2. Same question for path graphs ? (intersection graph of paths in a tree)
3. How to recognize a chordal graph ?

Which kind of representation to look for for special classes of graphs ?

- Easy to manipulate (optimal encoding, easy algorithms for optimisation problems)
- Geometric in a wide meaning (ex : permutation graphs = intersection of segments between two lines)
- Examples : disks in the plane, circular genomes ...

First remark

Proposition

Every undirected graph can be obtained as the intersection of a subset family

Proof

$G = (V, E)$
Let us denote by E_x the set of edges adjacent to x.
$xy \in E$ iff $E_x \cap E_y \neq \emptyset$
We could also have taken the set C_x of all maximal cliques which contains x.
$C_x \cap C_y \neq \emptyset$ iff $\exists$ one maximal clique containing both x and y

Starting from a graph in some application, find its characteristic :

1. 2-intervals on a line (biology), intersection of disks (or hexagons) in the plane (radio frequency), filament graphs, trapezoid graphs . . .
2. A whole book on this subject :
   J. Spinrad, Efficient Graph Representations, Fields Institute Monographs, 2003.
3. A website on graph classes :
   http ://www.graphclasses.org/

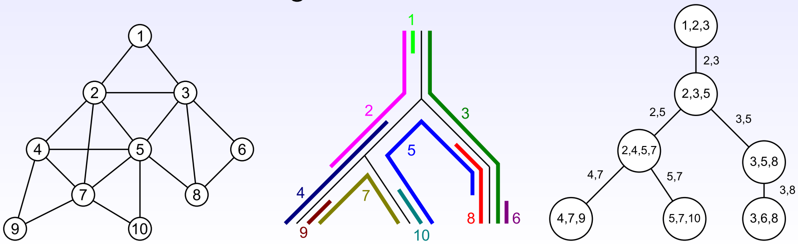
Clique tree

clique tree of G = a minimum size tree model of G

for a clique tree T of G :

- vertices of T correspond precisely to the maximal cliques of G
- for every maximal cliques C, C', each clique on the path in T from C to C' contains $C \cap C'$
- for each edge CC' of T, the set $C \cap C'$ is a *minimal separator* (an inclusion-wise minimal set separating two vertices)

Note : we label each edge CC' of T with the set $C \cap C'$.



Consequences of maximal clique tree

Theorem

Every minimal separator belongs to every maximal clique tree.

Lemma

Every minimal separator is the intersection of at least 2 maximal cliques of G.

Corollary

There are at most n minimal separators.

proof

Since G is chordal, every minimal separator S is a clique. Suppose S is an (x, y) minimal separator. Let us consider G_1 the connected component of $G \setminus S$ containing x.

If G_1 is reduced to x, then x must be universal to S, since S is a minimal separator, and $S + x$ is a maximal clique of G. Similarly if there exists $z \in G_1$ is universal to S then $S + z$ is contained in some maximal clique of G.

Else, suppose there is no vertex in G_1 universal to S. Consider two vertices $x, w \in G_1$ having different maximal neighborhoods in G_1. Such vertices always exist unless S is not minimal.

Therefore both x, w have a private neighbor t, u respectively in S.

proof II

So by assumption $wt, xu \notin E(G)$. Considering a shortest path $\mu \in G_1$ going from x to w. Then the cycle $[x, \mu, w, u, t]$ has no chord, a contradiction. Therefore there must exist some vertex of $w \in G_1$ universal to S, and $S + w$ contained in some maximal clique C of G. We finish the proof by considering the connected component of $G \setminus S$ containing y. This yields another maximal clique C'.
By construction $C \cap C' = S$.

Proof of the theorem

Démonstration.

Therefore $S = C' \cap C''$. These two maximal cliques belong to any maximal clique tree T of G. Let us consider the unique path μ in T joigning C' to C''.

All the internal maximal cliques in μ must contain S. Suppose that all the edges of μ are labelled with minimal separators strictly containing S, then we can construct a path in G from $C' - S$ to $C'' - S$ avoiding S, a contradiction. So at least one edge of μ is labelled with S.

□

Maximal Cardinality Search : MCS

Data: a graph $G = (V, E)$ and a start vertex s
Result: an ordering σ of V
Assign the label 0 to all vertices
$label(s) \leftarrow 1$
for $i \leftarrow n$ **à** 1 **do**
    Pick an unumbered vertex v with largest label
    $\sigma(i) \leftarrow v$
    foreach *unnumbered vertex* w *adjacent to* v **do**
        $label(w) \leftarrow label(w) + 1$
    end
end

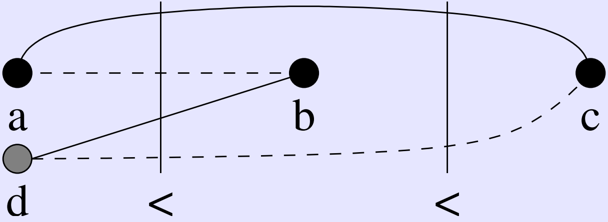
Maximal Neighbourhood Search (MNS)

MNS

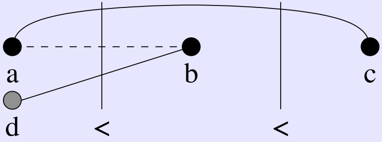
Data: a graph $G = (V, E)$ and a start vertex s
Result: an ordering σ of V
Assign the label $\emptyset$ to all vertices
$label(s) \leftarrow \{0\}$
for $i \leftarrow 1$ **to** n **do**
    Pick an unnumbered vertex v with a maximal under inclusion label
    $\sigma(i) \leftarrow v$
    foreach *unnumbered vertex w adjacent to v* **do**
        $label(w) \leftarrow \{i\} \cup label(w)$
    end
end

MNS property

Let σ be a total ordering $V(G)$, if $a < b < c$ and $ac \in E$ and $ab \notin E$, then it exsits d such that $d < b$, $db \in E$ and $dc \notin E$.



Generic search



MNS

MNS is a kind of completion of Generic search similar to BFS versus LBFS (resp. DFS versus LDFS). This explains why MNS was first named LexGen.

Theorem [Tarjan et Yannakakis, 1984]

G is a chordal graph iff every MNS computes a simplicial ordering.

Proof :

Let c be a non simplicial vertex (to the left). Thus it exists $a < b < c \in N(c)$ with $ab \notin E$. Using MNS property, it exsits $d < b$ with $db \in E$ and $dc \notin E$. Since G is chordal, necessarily $ad \notin E$.

Either $d < a$, considering the triple d, a, b, it exists $d' < a$ such that $d'a \in E$ and $d'b \notin E$. Furthermore $d'd \notin E$.

Or $a < d$, considering the triple a, d, c, it exists $d' < d$ such that $d'd \in E$ and $d'c \notin E$. Furthermore $ad' \notin E$.

In both cases a pattern is propagating to the left, a contradiction.

Corollary

G is a chordal graph iff every MCS, LBFS, LDFS computes a simplicial ordering.

Proof

Maximal for the cardinality, or maximal lexicographically are particular cases of maximality under inclusion.

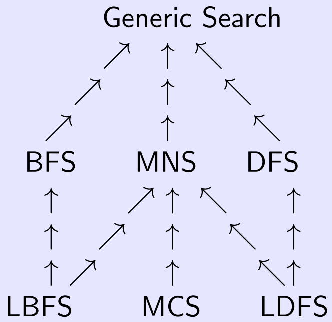
Implementation

MCS, LBFS provide linear time particular implementation sof MNS. But they are many others, less famous.
But in its full generality no linear time implementation is known.

Conclusions

Using the 4-points configurations we can prove the following inclusion ordering between searches

Strict inclusions



Playing with the elimination scheme

Easy Exercises :

1. Find a minimum Coloring (resp. a clique of maximum size) of a chordal graph in $O(|V| + |E|)$.
   Consequences : chordal graphs are perfect.
   At most $|V| - 1$ maximal cliques (best upper bound, since stars have exactly $|V| - 1$ maximal cliques).
2. Find a minimum Coloring (resp. a clique of maximum size) of an interval graph in $O(|V|)$
   using the interval representation.

Greedy colorings

Definitions

Clique number $\omega(G)$ = maximum size of a clique in G
Chromatic number $\chi(G)$ = minimum coloring of G.
$\forall G,\ \chi(G) \geq \omega(G)$

Greedy colorings

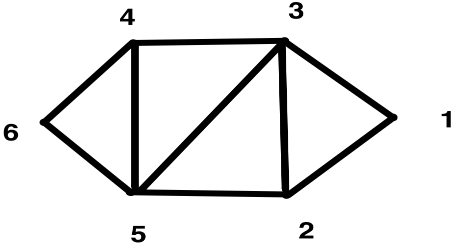
Color with integers from $[1, k]$
Following a vertex ordering, process successively the vertices using the greedy rule :

Take the minimum color not already in the neighbourhood

Chordal graphs

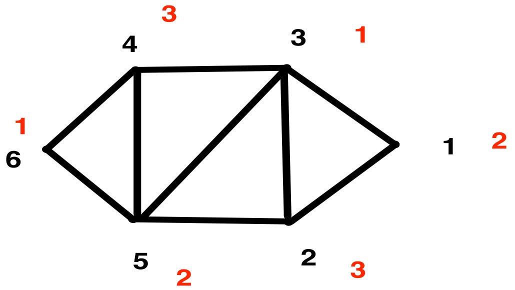
Apply LexBFS from n downto 1
Use the ordering n downto 1 for the greedy coloring.
Let $k = \omega(G)$.
Since every added vertex x is simplicial and $|N(x)| \leq k - 1$, it exists at least one missing color in its neighbourhood of the already colored subgraph.
The value k is reached for the last vertex belonging to each maximum clique of G.

Bad ordering for greedy coloring



6, 5, 4, 3, 2, 1 LexBFS ordering

Good ordering for greedy coloring



6, 5, 4, 3, 2, 1 LexBFS ordering

Perfect Graphs

G such that for every induced subgraphs $H \subseteq G$
$\omega(G) = \chi(G)$

Consequences

Therefore $\omega(G) = \chi(G)$ for chordal graphs.
Since being chordal graphs is an hereditary property, chordal graphs are perfect.

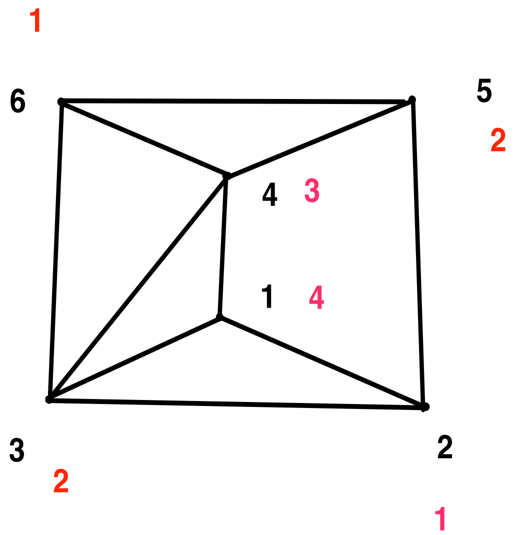
Perfectly orderable graphs

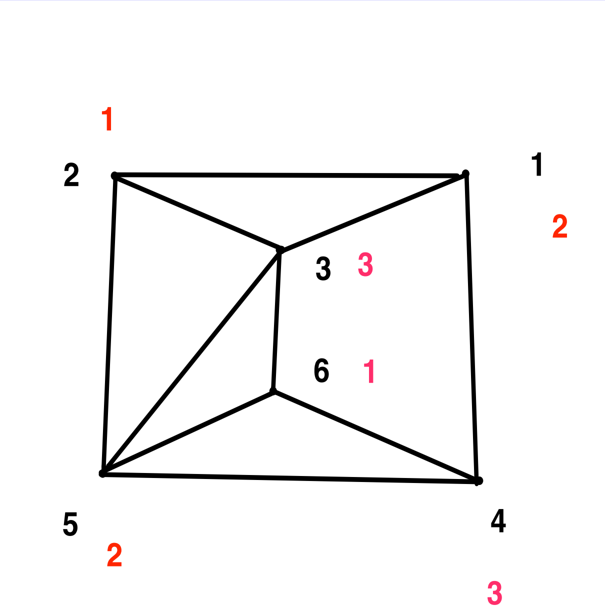
Although $\omega(G)$ and $\chi(G)$ can be computed in polynomial time for perfect graphs using the ellipsoid method, greedy coloring does not work for all perfect graphs.

A graph G is said to be perfectly orderable if there exists an ordering π of the vertices of G, such that any induced subgraph is optimally colored by the greedy algorithm using the subsequence of π induced by the vertices of the subgraph.
Chordal graphs are perfectly orderable.

For which graphs the greedy coloring works ?

Bad news :
NP-complete to recognize perfectly orderable graphs.
Greedy coloring can be far from the optimum, even for subclasses of perfect graphs.


1
6
5
2
4
3
1
4
3
2
2
1


1
2
1
2
3
3
6
1
5
2
4
3

The study of the relationships between $\omega(G)$ and $\chi(G)$ is fundamental for algorithmic graph theory.

1. 1930 Wagner's conjecture and treewidth
2. 1950 Shannon Problem and Perfect graphs and semi-definite programming