Chevaleret, octobre 2012
Schedule

More structural insights of chordal graphs

Properties of reduced clique graphs

Interval graphs

Exercises
Playing with the representation

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 \( G \).
6, 5, 4, 3, 2, 1  LexBFS ordering
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 $\pi$ of the vertices of $G$, such that any induced subgraph is optimally colored by the greedy algorithm using the subsequence of $\pi$ 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.
A bad LexBFS
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
Some questions about geometric representations

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)
Graph classes visited so far

- Interval graphs: intersection graph of intervals of the real line
- Unit interval graphs (no $K_{1,3}$)
  all interval have the same length (no proper inclusion)
- Path graphs (resp. directed path graphs): intersection graph of paths in a tree (resp. directed paths in a rooted directed tree)
  (Directed path graph appear in some polynomial CSP class).
- Chordal graphs intersection graphs of subtrees in a tree.
Research problems

- A linear time algorithm for Path graphs recognition.
- Using LBFS as for interval graphs and chordal graphs?
- Meta Conjecture:
  Most graph classes can be recognized using some graph search, or using a series of consecutive graph searches.
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 $C \cap C'$. 
Let $G = (V, E)$ be a chordal graph.

$G$ admits at most $|V| - 1$ maximal cliques and therefore the tree is also bounded by $|V| - 1$ (vertices).

But some vertices of the original graph can be repeated in the cliques. If we consider a simplicial elimination ordering the size of a given maximal clique is bounded by the neighbourhood of the first vertex of the maximal clique.

Therefore any maximal clique tree is bounded by $|V| + |E|$. If we label the edges of the maximal clique tree with minimal separators the size remains in $O(|V| + |E|)$.
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 $|V| - 2$ different minimal separators.
**Theorem**
Every minimal separator belongs to every maximal clique tree.

**Lemma**
Every minimal separator is the intersection of at least 2 maximal cliques of $G$. 
Proof of the lemma

Démonstration.

We know that $S$ is a clique. Let us consider $G_1$ a connected component of $G - S$. Let $x_1, \ldots, x_k$ be the vertices of $G_1$ having a maximal neighbourhood in $S$.

If $k = 1$ then $x_1$ must be universal to $S$, since $S$ is a minimal separator.

Else, consider a shortest path $\mu$ in $G_1$ from $x_1$ to $x_k$. Necessarily $x_1$ (resp. $x_k$) has a private neighbour $z$ (resp. $t$) in $S$.

Then the cycle $[x_1, \mu, x_k, t, z]$ has no chord, a contradiction.

Therefore $x_1 \cup S$ is a clique, and is contained in some maximal clique $C$ in $G_1$. We finish the proof by taking another connected component of $G - S$. 
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 $\mu$ in $T$ joigning $C'$ to $C''$.

All the internal maximal cliques in $\mu$ must contain $S$. Suppose that all the edges of $\mu$ 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 $\mu$ is labelled with $S$. 
Clique graph

the *clique graph* $C(G)$ of $G = \text{intersection graph of maximal cliques of } G$
Reduced clique graph

the *reduced clique graph* $C_r(G)$ of $G = \text{graph on maximal cliques of } G$ where $CC'$ is an edge of $C_r(G)$ $\iff$ $C \cap C'$ is a minimal separator.
More structural insights of chordal graphs
Combinatorial structure of $C_r(G)$

**Lemma 1 : M.H and C. Paul 95**

If $C_1, C_2, C_3$ is a cycle in $C_r(G)$, with $S_{12}, S_{23}$ and $S_{23}$ be the associated minimal separators then two of these three separators are equal and included in the third.

**Lemma 2 : M.H. and C. Paul 95**

Let $C_1, C_2, C_3$ be 3 maximal cliques, if $C_1 \cap C_2 = S_{12} \subset S_{23} = C_2 \cap C_3$ then it yields a triangle in $C_r(G)$.
Lemma 3: Equality case

Let $C_1, C_2, C_3$ be 3 maximal cliques, if $S_{12} = S_{23}$ then:

- either the $C_1 \cap C_3 = S_{13}$ is a minimal separator
- or the edges $C_1 C_2$ and $C_2 C_3$ cannot belong together to a maximal clique tree of $G$. 
Properties of reduced clique graphs
Theorem (Gavril 87, Shibata 1988, Blayr and Payton 93)

The clique trees of $G$ are precisely the maximum weight spanning trees of $\mathcal{C}(G)$ where the weight of an edge $CC'$ is defined as $|C \cap C'|$.

Theorem (Galinier, Habib, Paul 1995)

The clique trees of $G$ are precisely the maximum weight spanning trees of $\mathcal{C}_r(G)$ where the weight of an edge $CC'$ is defined as $|C \cap C'|$.

Moreover, $\mathcal{C}_r(G)$ is the union of all clique trees of $G$. 
Applications

- All clique trees have exactly the same labels, including repetitions.
Maximal Cardinality Search : MCS

**Data:** a graph $G = (V, E)$ and a start vertex $s$

**Result:** an ordering $\sigma$ of $V$

Assign the label 0 to all vertices

$\text{label}(s) \leftarrow 1$

**for** $i \leftarrow n$ **to** 1 **do**

Pick an unnumbered vertex $v$ with largest label

$\sigma(i) \leftarrow v$

**foreach** unnumbered vertex $w$ adjacent to $v$ **do**

$\text{label}(w) \leftarrow \text{label}(w) + 1$

end

end
Maximum spanning trees

Maximal Cardinality Search can be seen as Prim algorithm for computing a maximal spanning tree of $C_r(G)$. 
How to compute a clique tree?
How to generate all simplicial elimination schemes?
LexBFS, MCS or MNS visit maximal cliques “consecutively” (i.e. when the search explores a vertex $x$ of a maximal clique $C$ that does not belong to any of the previously visited maximal cliques then all the unvisited vertices of $C$ will appear consecutively just after $x$).

Therefore when applying a search (LexBFS, MCS or MNS) one can compute a clique tree, by considering the strictly increasing sequences of labels.
Properties of reduced clique graphs

\[ \begin{align*}
    b, a, c, e, d, h, f, g & \text{ is a LexBFS ordering.} \\
    \text{we can find the maximal cliques } & b, a, c \text{ then } b, c, e \text{ then } b, e, d \text{ then } c, h \text{ then } c, e, f \text{ and } e, g.
\end{align*} \]
Simplicial elimination schemes

1. Choose a maximal clique tree $T$
2. While $T$ is not empty do
   Select a vertex $x \in F - S$ in a leaf $F$ of $T$;
   $F \leftarrow F - x$
   If $F = S$ delete $F$;
Canonical simplicial elimination scheme

1. Choose a maximal clique tree $T$
2. While $T$ is not empty do
   Choose a leaf $F$ of $T$
   Select successively all vertices in $F - S$
   delete $F$
Remark

Does there exist other simplicial elimination scheme?
Size of $\mathcal{CS}(G)$

Considering a star on $n$ vertices, shows $|\mathcal{CS}(G)| \in O(n^2)$

Not linear in the size of $G$
\( \mathcal{C}S(G) \) is not chordal!
$\mathcal{C}(G)$ is not chordal!
In fact $\mathcal{C}S(G)$ is dually chordal (almost chordal) and $\mathcal{C}S(\mathcal{C}S(G))$ is chordal.
Cannonical representation

- For an interval graph, its PQ-tree represents all its possible models and can be taken as a cannonical representation of the graph (for example for graph isomorphism).
- But even path graphs are isomorphism complete. Therefore a canonical tree representation is not obvious for chordal graphs.
- $C_r(G)$ is a Pretty Structure to study chordal graphs.
  To prove structural properties of all maximal clique trees of a given chordal graph.
Characterization Theorem for interval graphs

(i) $G = (V, E)$ is interval graph.

(ii) It exists a total ordering $\tau$ of the vertices of $V$ s.t.
     $\forall x, y, z \in G$ with $x \leq_{\tau} y \leq_{\tau} z$ and $xz \in E$ then
     $xy \in E$.

(iii) $G$ has a maximal clique path. (A maximal clique path is just a maximal clique tree $T$, reduced to a path).

(iv) $G$ is the intersection graph of a family of intervals of the real line.
To recognize an interval graph, we just have to compute a maximal clique tree and check if it is a path?

Difficulty: an interval graph has many clique trees among them some are paths
Interval graphs
Many linear time algorithms already proposed for interval graph recognition .... using nice algorithmic tools: graph searches, modular decomposition, partition refinement, PQ-trees ...
Linear time recognition algorithms for interval graphs

- Booth and Lueker 1976, using PQ-trees.
- Korte and Mohring 1981 using LexBFS and Modified PQ-trees.
- Hsu and Ma 1995, using modular decomposition and a variation on Maximal Cardinality Search.
- M.H, McConnell, Paul and Viennot 2000, using LexBFS and partition refinement on maximal cliques.
A partition refinement algorithm working on maximal cliques

1. Compute a tree $T$ using LexBFS
   If $T$ is not a maximal clique tree; then $G$ is not chordal, neither interval.

2. Start from the last maximal clique
   Refine the cliques with the minimal separator.

3. Refine until each part is a singleton

4. If a part is not a singleton start recursively from the last clique of this part according to LexBFS.
Lex BFS: \[ cab \mid d \mid e \mid f \mid g \]

Tree of maximal cliques:

- \( a, b, d \)
- \( a, b, e \)
- \( a, f \)
- \( b, g \)

\( a \) n’appartient rien

- \( af \mid \text{abc} \mid abd \mid \text{abe} \mid bg \)
- \( b \) n’appartient rien

- \( af \mid \text{abc} \mid abd \mid \text{abe} \mid bg \)
Exercise 1

Ends of a LexBFS

Many properties can be expressed on the last vertex of a LexBFS. Example: if $G$ is a chordal graph, the last vertex is simplicial.

1. Show that the last maximal clique visited can be taken as the end of some chain of cliques if $G$ is an interval graph.

2. Complexity of the following decision problem:

**Data:** a graph $G = (V, E)$ and a given vertex $x \in V$

**Result:** Does there exist a LexBFS of $G$ ending at $x$?
Exercise 2

If we consider the edges of the clique tree labelled with the size of the minimal separators, show that:
for every maximal clique tree $T$
$\text{weight}(T) = \sum_{1 \leq i \leq k} |C_i| - n$, where $C_1, \ldots, C_k$ are the maximal cliques of $G$. 