Treewidth I

Michel Habib

Cours MPRI, 2009–2010

Schedule

Is it worth to work on graph decompositions ?

- Apply divide and conquer techniques to solve hard problems on difficult instances.
- Find efficient encoding to store graphs
- Find structural properties (comparability graphs, tournaments . . .)
- Byproduct. Many efficient algorithms are based on a decomposition technique.
  Example : ear decomposition for planarity testing.

Measure the distance from a given graph to a tree

Easy cases

Some graphes are like trees : chordal graphs, cographs and some variations of P_4-free graphs . . .

General case

One can associated a tree to a graph : modular decomposition tree, split decomposition tree . . .

Bad news

Most graphs are prime.

Using Google Scholar on 28th october 2008

- Graph Decomposition 372 000 answers
- Treewidth 33 000 answers
- NP-complete 42 000 answers

Sparse graphs

In fact most of the grah classes we are going to consider are made wiht sparse graphs.

The most general search for a tree structure inside a graph.
Many times rediscovered form many areas.

k-trees

Recursive Definition

A clique with k+1 vertices is a k-tree. If G is a k-tree, then we obtain another k-tree G' by adding a new vertex and joining this vertex to a k-clique of G.

Clearly a k-tree is a chordal graph.
1-trees =ordinary trees
2-trees = series-parallel networks

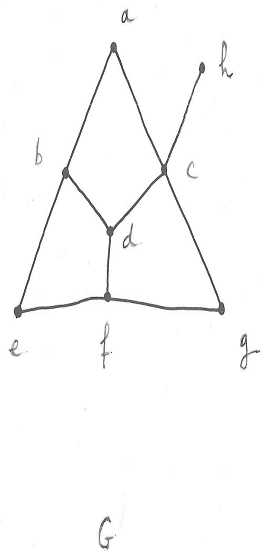
For fixed k, a k-tree on n vertices admits $O(n)$ edges.

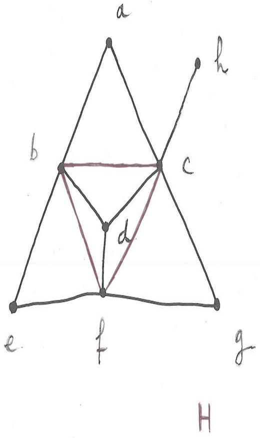
Partial k-trees

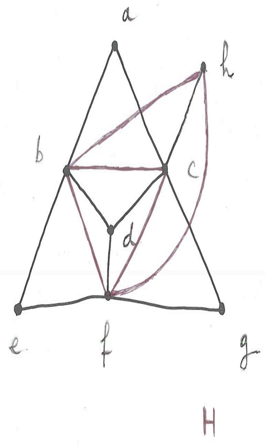
A partial k-tree is a partial subgraph of a k-tree (deleting edges and vertices)

Definition

Treewith(G)= least integer k s.t. G is a partial k-tree.


a
h
b
c
d
e
f
g
G


a
h
b
c
d
e
f
g
H


a
h
b
c
d
e
f
g
H

Tree decomposition

$G = (V, E)$ has a tree decomposition $D = (S, T)$
S is a collection of subsets of V, T a tree whose vertices are elements of S such that :

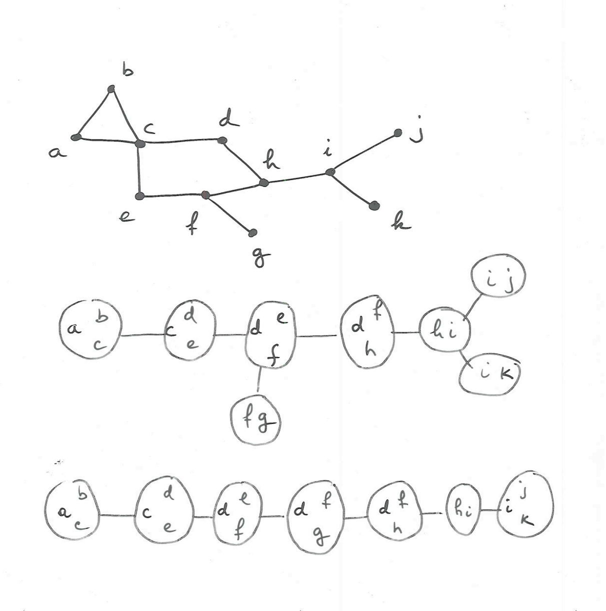
(0) The union of elements in S is V

(i) $\forall e \in E, \exists i \in I$ with $e \in G(S_i)$.

(ii) $\forall x \in V$, the elements of S containing x form a subtree of T.

Definition

$Decomp(G) = Min_D(Max_{S_i \in S}\{|S_i| - 1\})$


b
a
c
d
h
i
j
k
e
f
g
a b c
c d e
d e f
f g
d f h
h i
i j
i k
a b c
c d e
d e f
d f g
d f h
h i
i j k

Equivalences

1. $\forall G,\ Treewidth(G) = Decomp(G)$
2. $Treewidth(G) = Min_{H \text{ triangulation of } G}\{\omega(H) - 1\}$
3. Computing treewidth is NP-hard.

Some examples

1. G is a tree iff $treewidth(G) = 1$
2. $treewidth(K_n) = n - 1$
3. If G is a cycle then $treewidth(G) = 2$. (It can be seen as two chains in parallel, i.e. a series-parallel graph)
4. $treewidth(K_{n,m}) = min(n, m)$
5. $treewidth(G_{n,m}) = min(n, m)$, the lower bound is hard to obtain !
6. $treewidth(G)$ (resp. pathwidth) measures the distance from G to a tree (resp. to a chain)

If G is outerplanar then $treewidth(G) \leq 2$
Hint : compute a triangulation of G

Easy properties

treewidth$(G) = k$ iff G can be decomposed using only separators of size less than k.

Fundamental lemma

Let ab an edge of T some tree decomposition of G and T_1, T_2 be the two connected components of $T - ab$, then $V_a \cap V_b$ is a separator between $V_1 - V_2$ and $V_2 - V_1$, where $V_1 = \cup_{i \in T_1} V_i$ and $V_2 = \cup_{j \in T_2} V_j$.

Computations of treewidth

- There exists polynomial approximation algorithms
- $\forall k$, it exists a linear algorithm to check wether $Treewidth(G) \leq k$ Boedlander 1992.
- Find an efficient algorithm for small values $3, 4, 5 \ldots$ is still a research problem

Real applications of treewidth

1. Graphs associated with programs have bounded treewidth 4,5, 6 depending on the programming language, Thorup 1997.
2. Constraint Satisfaction Problem, Feuder's Theorem
3. Can also be defined on hypergraphs which yields applications for Data bases via Acyclic Hypergraphs
4. Good Heuristic for the Traveling Salesman Problem, Chvatal

Other definitions of trewidth in terms of cop-robber games, using graph grammars ....
But it turns out that this parameter is a fundamental parameter for graph theory and has many theoretical applications.

1. $Treewidth(G) = Min_{H \text{ triangulation of } G}\{\omega(H) - 1\}$
2. $MinFillin(G) = Min_{H=(V,F) \text{ triangulation of } G}\{|F|\}$
3. $Pathwidth(G) = Min_{H \text{ intervalcompletion of } G}\{\omega(H) - 1\}$
4. $MinIntervalcompletion(G) =$
   $Min_{H=(V,F) \text{ intervalcompletion of } G}\{|F|\}$
5. Theorem Arnborg, Corneil, Proskurowski, 1987 : These 4 problems are NP-hard.

Divide and Conquer Approach

Proskurowski

when he introduced partial k-trees
as a generalisation of trees preserving dynamic programming
he aimed at polynomial algorithms for bounded tree-graphs (i.e. the size of the bags is bounded).

Generic Divide and Conquer Algorithm

Solve the problem for a leaf and then recurse

The study of the interplay between logics and combinatorial structures
yields knowledge on complexity theories

B. Courcelle studying graph rewriting systems or graph grammars obtained :

Theorem

Any graph problem that can be expressed with a formula of the Monadic Second Order Logic (MSOL),
if G has bounded treewidth then it exists a linear algorithm to solve this problem on G.

Monadic Second Order Logic

For graphs :
$x_1, \dots x_n$ variables
X_i subset of vertices
Atoms :
$E(x, y)$ true iff xy is an edge of G
$X(x)$ true if $x \in X$
Classical logical connectors (equality, implication, negation ...) to make formulas
Quantification over variables and subsets of vertices are allowed.

Examples :

- $\phi : \exists x_1 \ldots \exists x_k \forall y \bigwedge_{1 \leq i \leq k}((x_i = y) \bigvee (E(x_i, y))$
  $\phi =$ G admet un ensemble dominant de taille k.
- Idem $\psi_k =$ G admet une k-coloration
- The number of quantifiers is an indication of the complexity of the formula (i.e. the problem)
  We do not need to use quantifiers on subets of edges (via le biparti d'incidence)

Not all graph problems can be expressed in MSOL.

Linear ?

Linear but with a giant constant :
$(2^{2^{2^{\cdots}}})^h$.
h the size of this exponential tower which depends on the MSOL formula.

Grohe, Frick 2005

h is unbounded unless $P = NP$.

Graph Minors

- A graph H is a minor of a graph G if H is isomorphic to a graph obtained from G by contracting edges, deleting edges, and deleting isolated nodes
- The minor ordering of graphs is that defined by $H \leq G$ if H is a minor of G
- A set S of graphs is downwardly closed with respect to the minor ordering if, whenever $G \in S$ and H is a minor of G, it holds $H \in S$.

Equivalent definition :
remove vertices and edges and then contract some connected subgraphs to vertices
d-shallow minors : contract only path between vertices at most at distance d.

P. Seymour and Roberston introduced treewith and branchwidth as parameters for an induction proof of Wagner's conjecture.

Wagner's Conjecture

Wagner conjectured in the 1930s (although this conjecture was not published until later) that in any infinite set of graphs, one graph is isomorphic to a minor of another. The truth of this conjecture implies that any family of graphs closed under the operation of taking minors (as planar graphs are) can automatically be characterized by finitely many forbidden minors analogously to Wagner's theorem characterizing the planar graphs.

Obstructions

An obstruction H to a family F of graphs, is a graph which does not belong to F, but every minor of H belongs ot F.
For example a triangle is an obstruction for the family of forests.

Theorem 1

For every class of graphs $\mathcal{G}$ closed under minors then there exist a set of finite obstructions $Ob(\mathcal{G})$ such that :
$G \in \mathcal{G}$ iff $\nexists H \in Ob(\mathcal{G})$ with $H \leq G$.

Example Kuratowski's theorem

G is planar iff G does not contain any subgraph homeomorphic to K_5 or $K_{3,3}$.

Theorem 2

For every graph H, there exists an algorithm in $O(n^3)$ to test wether $H \leq G$ for a given graph G.

Remark

B. Reed claims $O(nlogn)$ 2007.

Other graph classes

- G is a forest iff G does not contain any subgraph homeomorphic to K_3 .
- G is series-parallel iff G does not contain any subgraph homeomorphic to K_4 and $K_{3,3}$.
- G is planar iff G does not contain any subgraph homeomorphic to K_5 or $K_{3,3}$.
- G is an outerplanar graph iff G does not contain any subgraph homeomorphic to K_4 or $K_{2,3}$
- As a corollary if G does not contain any subgraph homeomorphic to K_4 or $K_{2,3}$ then G is hamiltonian.

Theorem 3

$H \leq G$ implies $treewidth(H) \leq treewidth(G)$
and for every k, graphs with treewidth bounded by k are well quasi-ordered for $\leq$.

No infinite antichain or no infinite strictly decreasing chain or finite number of obstructions.

1. $treewidth(G) = 2$ iff G has no K_4 as minor.
2. $treewidth(G) = 3$ iff G has no K_5, Petersen, 8-wheel, XX as minor.
3. $treewidth(G) = 4$ iff G has no $H \in \mathcal{H}$ as minor, and $\mathcal{H}$ contains 80 graphs.
4. ...

Theorem 4

Finite graphs are well ordered with $\leq$

Consequences

Any class of graphs closed under minors has a polynomial recognition algorithm

Another meta theorem

2006 Dawer, Kreutzer,

Every optimisation problem expressible in first order logic on class of graphs defined with minor exclusion, has a polynomial approximation algorithm.

How to prove such a theorem ?

Theorem

Gaifman 1981
Every FOL formula can be expressed using local formulas

Theorem

Roberston Seymour 1999
For any graph class defined using minor exclusion there exists a decomposition using graphs almost embeddable on some surface

Theorem

Grohe, Kawabashi 2008
This decomposition can be computed in $O(n^c)$, and c does not depend on the size of the minor.

Applications

- Change our knowledge about P versus NP
- Linear algorithms but with extremely big constants
- Improve our knowledge on NP-complete problems with Fixed parameter Tractablility (FPT) a theory proposed by M. Fellows
- Non constructive algorithms