CS 163 & CS 265: Graph Algorithms

Week 7: Perfection and width

Lecture 7b: Width parameters

David Eppstein
University of California, Irvine

Winter Quarter, 2025

Path decompositions and coloring

The main idea

Instead of describing interval graphs geometrically, by coordinates of interval endpoints, describe them combinatorially, as a sequence of sets of vertices

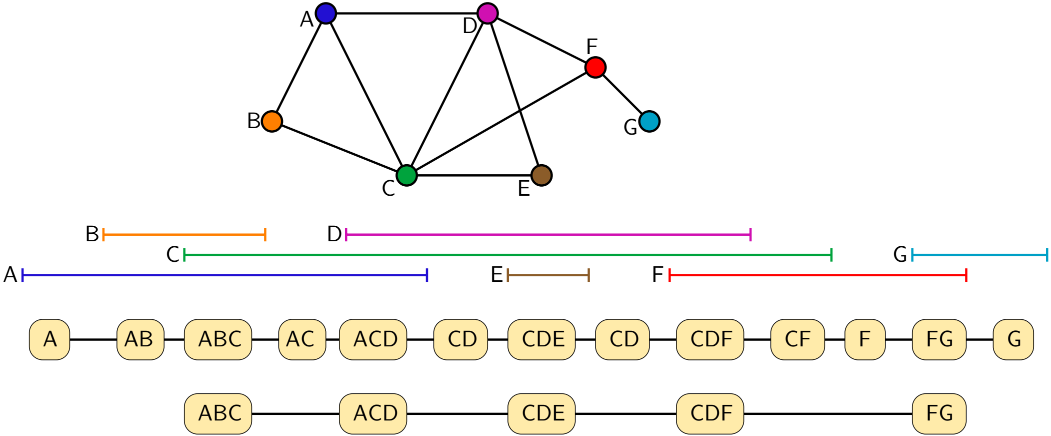
The same combinatorial description can be generalized to subgraphs of interval graphs

It leads to fast algorithms for graphs that are subgraphs of interval graphs whose cliques and chromatic number are small

Path decomposition of an interval graph

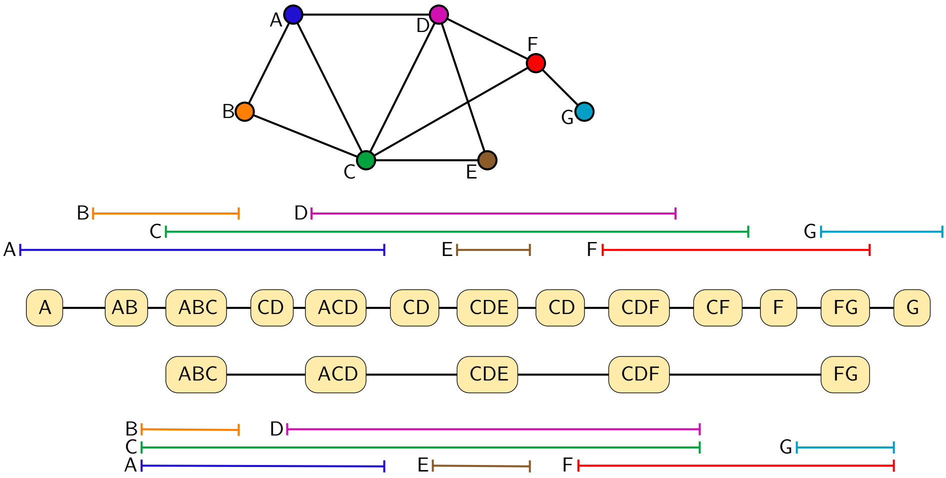
Write down a sequence of sets (called bags) naming the intervals above each point of the line

Simplify by keeping only bags that are not subsets of larger bags

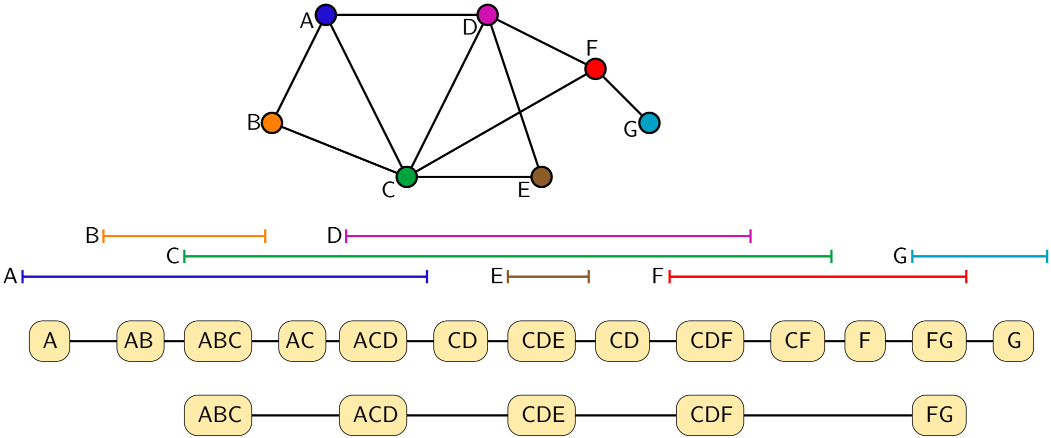


Can recover interval representation from path decomposition

Start each interval before first bag containing its vertex, and end each interval after last bag containing its vertex



Properties of interval graph path decomposition

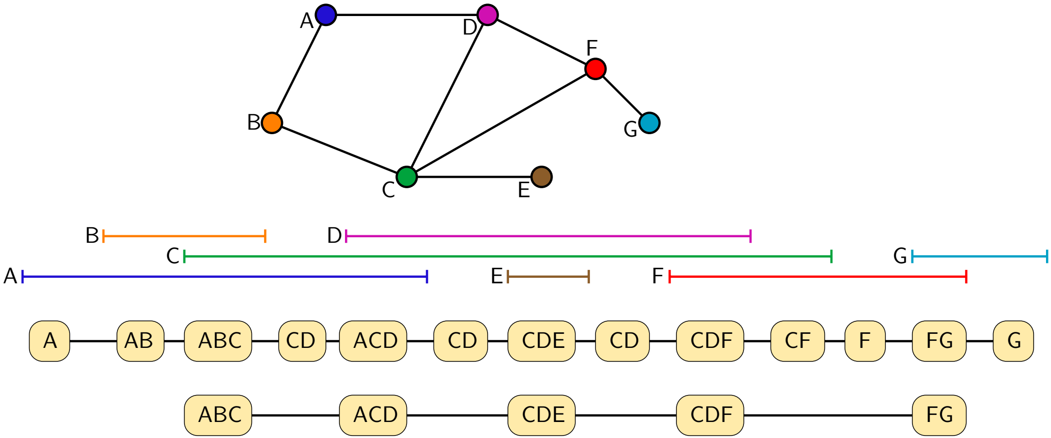


Every vertex belongs to a consecutive sequence of bags

Edges = pairs of vertices that both appear in the same bag

Largest clique = biggest bag

Path decomposition for subgraphs



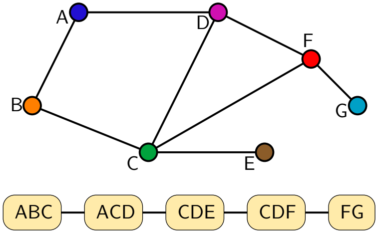
Every vertex belongs to a consecutive sequence of bags

Every edge has both its endpoints in some bag
(but not all pairs of vertices in bags form edges)

Path decomposition in general

Throw away the intervals, just use a sequence of bags with the same two properties:

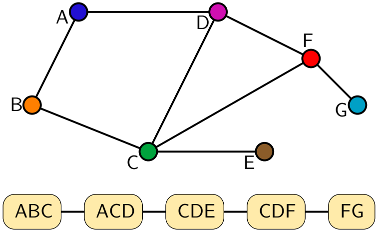
- Every vertex appears in a consecutive subsequence of bags
- Every edge has ≥ 1 bag containing both endpoints



Pathwidth

The same graph may have many different path decompositions (e.g. put all vertices into one big bag)

Width of a decomposition = size of biggest bag, minus one *

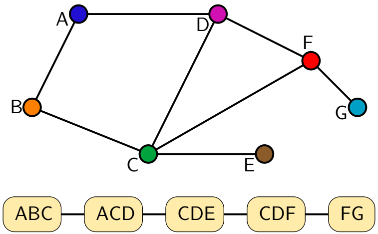


width = 2

Pathwidth of a graph is smallest width of any path decomposition

* we subtract one in order to make path graphs have pathwidth one

Pathwidth and cliques



Vertices of a clique have overlapping subsequences of bags

$\Rightarrow$ left endpoints of subsequences $\leq$ right endpoints

$\Rightarrow$ some bag contains the whole clique

Pathwidth(G) = smallest possible max clique size (minus one) of an interval graph that contains G

Small pathwidth ⇒ fast clique-finding

If you have a low-width path decomposition of G, then:

for each bag in the path decomposition:
    for each subset of the bag:
        if it's a clique, bigger than best found so far
            remember it as max clique

Simplified decomposition has $\leq n$ bags (each one adds a vertex)

So time is $O(n\,2^w)$ even though clique-finding is NP-complete

Times like this: polynomial in n, with the same exponent, whenever w is constant, are called fixed-parameter tractable

Small pathwidth $\Rightarrow$ fast coloring

If B is a bag, and C is a coloring of vertices in B with $\leq k$ colors, define Good(B, C) = true when C can be extended to a valid k-coloring of all vertices to the left of B

For each bag B in left-to-right order, and each coloring C:

Let B' be the bag just before B

Good(B,C) = true if C compatible with a good coloring of B'

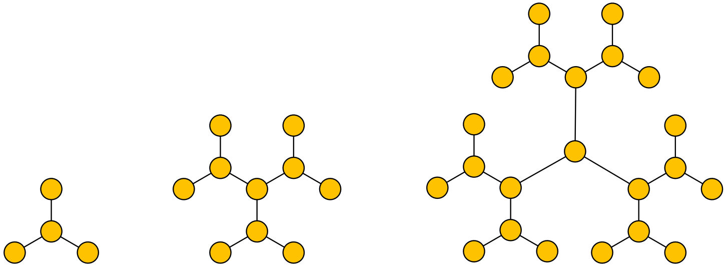
G has a k-coloring if and only if its last bag has a good coloring

Time $O(nk^{w+1})$, again fixed-parameter tractable

Trees and treewidth

The pathwidth of trees

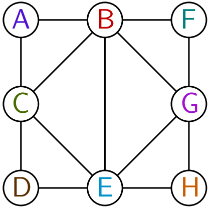
Trees are easy to find cliques in and easy to color, but they can still have large pathwidth

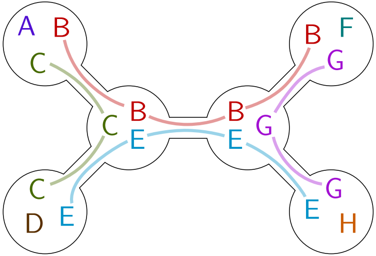


width = 1 width = 2 width = 3

(Almost the same as Strahler number, but for unrooted trees rather than rooted trees)

Tree decomposition



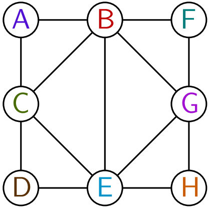


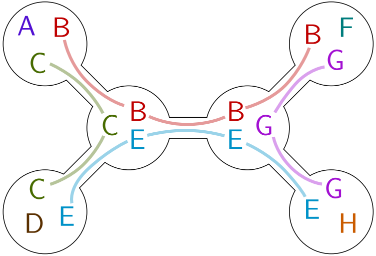
Make a tree of bags instead of a sequence of bags

Every vertex must belong to a connected subtree of bags

Every edge must have both endpoints in at least one bag

Treewidth





Width = max bag size minus one

Treewidth = smallest width of any decomposition

= smallest possible max clique size (minus one) of a chordal graph that contains G

Treewidth $\Rightarrow$ fast algorithms

Clique algorithm: No change

Coloring algorithm: Good colorings are colorings of bags that extend to all vertices in their subtree

Compute bottom-up looking for colorings compatible with good colorings of all children

Both algorithms have same runtime as for pathwidth, but work for more graphs (because more graphs have small treewidth)

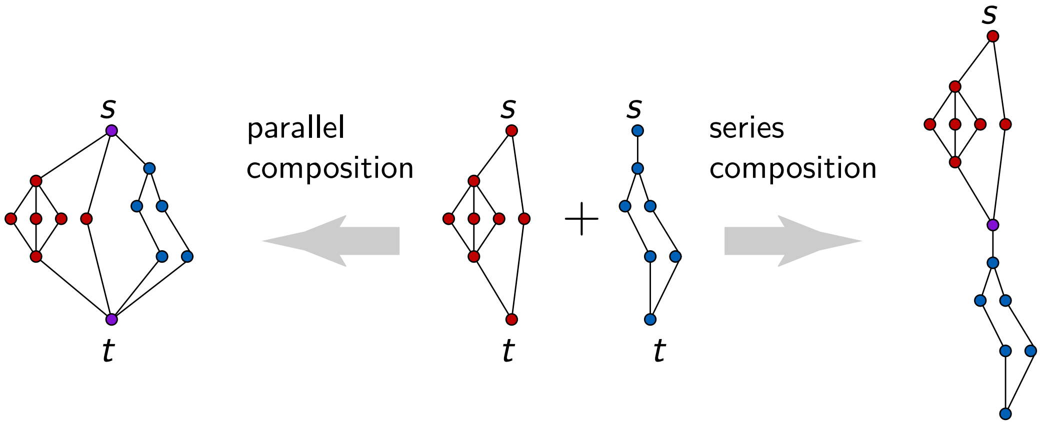
Tricker part: Finding path decompositions or tree decompositions

Also fixed-parameter tractable but much more complicated

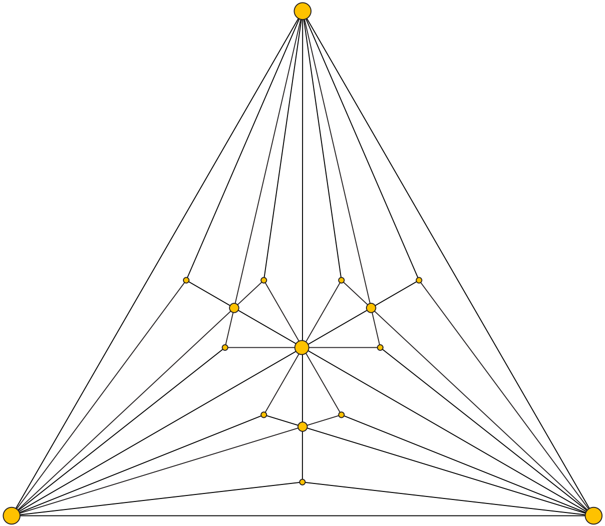
Which graphs have small treewidth?

The connected graphs with treewidth = 1 are exactly the trees

Connected graphs with treewidth = 2 are subgraphs of series-parallel graphs, constructed recursively:



A graph of treewidth three



The bigger picture

A maze of width parameters

Researchers have studied many other related width parameters:

bandwidth, cutwidth, carving width, branchwidth, clique-width, tree-depth, linear width, mim-width, twin-width, flip-width, . . .

Tradeoff between parameters that remain small on larger classes of graphs, vs parameters with fast algorithms for small parameters

Still an active subject, especially in extensions to directed graphs

https://commons.wikimedia.org/wiki/File:The_maze,_Longleat_safari_park_-_geograph.org.uk_-_938546.jpg

Which problems have FPT algorithms?

Courcelle's theorem: FPT algorithm for any existence or minimum-weight or maximum-weight optimization problem on classes of graphs that can be described as a logical formula involving vertices, edges, sets of vertices and edges, and adjacency

Example: Hamiltonian cycle can be described as a set C of edges such that

- Each vertex touches exactly two edges in C:

$$\forall v \exists e \exists f \big((v.e) \wedge (v.f) \wedge (e \neq f) \wedge (e \in C) \wedge (f \in C) \wedge \\ \forall g \ (v.g \wedge g \in C) \rightarrow (e = g \vee f = g) \big)$$

- Every nontrivial cut is crossed by an edge in C:

$$\forall K \big((\exists v \ v \in K) \wedge (\exists v \ v \notin K) \big) \rightarrow \\ \exists u \exists w \exists e \ \big(u \in K \wedge u.e \wedge w \notin K \wedge w.e \wedge (e \in C) \big)$$

Morals of the story

Pathwidth and treewidth can be defined from clique size of interval or chordal completions

Equivalent and more useful definition involves paths or trees with nodes decorated by sets of vertices called bags

When the width is small, we can find cliques and optimal colorings quickly and solve many other graph problems

"Fixed parameter tractable": time is polynomial in n with fixed exponent, but may depend badly on width

Active research on defining and classifying width parameters