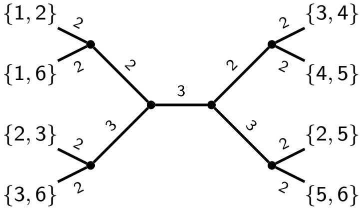
Seminar Combinatorial Optimization:
Graph Decompositions

Niels Lindner



Introduction

October 16, 2019

Main Topics

Idea

- Many combinatorial optimization problems are formulated on graphs.
- Real-world applications typically lead to NP-hard optimization problems on large graphs.
- Global solution algorithms scale badly.
- Idea: Divide and conquer!

Graph Decomposition Toolbox

- Graph Partitioning / Graph Bisection
- Tree Decompositions
- Branch Decompositions

Minimum Cuts

Let G be an undirected graph with edge weights $w : E(G) \to \mathbb{R}_{\geq 0}$.

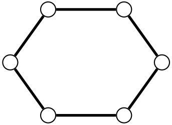
Minimum Cut

Find a non-empty proper subset of vertices $S \subsetneq V(G)$ such that the weight of the edges connecting S with $V(G) \setminus S$ is minimum, i.e., minimize

$$w(\delta(S)) = \sum_{\{u,v\} \in E(G): u \in S, v \notin S} w(uv).$$

A minimum cut $\delta(S)$ can be found in polynomial time.

Example ($w \equiv 1$)



Minimum Cuts

Let G be an undirected graph with edge weights $w : E(G) \to \mathbb{R}_{\geq 0}$.

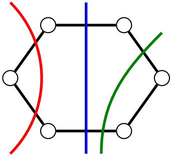
Minimum Cut

Find a non-empty proper subset of vertices $S \subsetneq V(G)$ such that the weight of the edges connecting S with $V(G) \setminus S$ is minimum, i.e., minimize

$$w(\delta(S)) = \sum_{\{u,v\} \in E(G): u \in S, v \notin S} w(uv).$$

A minimum cut $\delta(S)$ can be found in polynomial time.

Example ($w \equiv 1$)



Any minimum cut in the cycle graph C_n with uniform weights has weight 2, but the size of S can be any number between 1 and $n - 1$.

Graph Partitioning

Let G, w be as before, and let $\alpha \geq 0$.

Graph Bisection / Balanced Minimum Cut

Find a non-empty proper subset of vertices $S \subseteq V(G)$ such that $w(\delta(S))$ is minimum **and** $|V(G)|/2 \leq |S| \leq (1+\alpha) \cdot |V(G)|/2$.

Graph k-Partitioning

For $k \in \mathbb{N}$, find a k-partition $(S_1, \ldots, S_k)$ of $V(G)$ such that the weight of the edges between distinct parts is minimum and $|S_i| \leq (1+\alpha) \cdot |V(G)|/k$ for all $i \in \{1, \ldots, k\}$.

Theorem (Garey, Johnson, Stockmeyer, 1976)

The graph bisection problem is NP-hard for $w \equiv 1$ and $\alpha = 0$.

Remark

Terminology: Cut vs. partition $\leftrightarrow$ connected vs. arbitrary. If $\delta(S)$ is an unbalanced minimum (2-)cut in a connected graph, then S is connected.

Treewidth

Idea

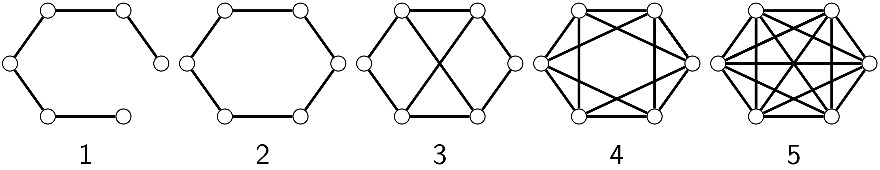
Many NP-hard optimization problems are easy on trees, e.g.:

- longest path
- minimum vertex coloring / chromatic number
- maximum independent set
- periodic timetabling in public transport

Treewidth

Treewidth measures how “similar” a graph is to a tree.

Examples



Tree Decompositions

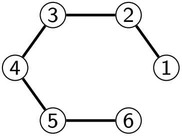
Informal Description

Tree decompositions define a “parsing tree” for vertices of a graph.

Interpretations of Treewidth

- minimum maximum bag size − 1 among all tree decompositions
- minimum clique size − 1 among all chordal completions
- minimum k for which the graph is a partial k-tree

Examples



treewidth 1

$\{1,2\}$ — $\{2,3\}$ — $\{3,4\}$ — $\{4,5\}$ — $\{5,6\}$

Tree Decompositions

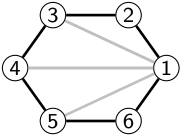
Informal Description

Tree decompositions define a “parsing tree” for vertices of a graph.

Interpretations of Treewidth

- minimum maximum bag size − 1 among all tree decompositions
- minimum clique size − 1 among all chordal completions
- minimum k for which the graph is a partial k-tree

Examples



treewidth 2

$\{1,2,3\}$ — $\{1,3,4\}$ — $\{1,4,5\}$ — $\{1,5,6\}$

Tree Decompositions

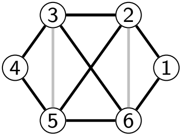
Informal Description

Tree decompositions define a "parsing tree" for vertices of a graph.

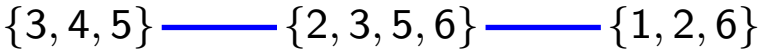
Interpretations of Treewidth

- minimum maximum bag size − 1 among all tree decompositions
- minimum clique size − 1 among all chordal completions
- minimum k for which the graph is a partial k-tree

Examples



treewidth 3



Tree Decompositions

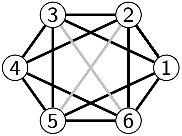
Informal Description

Tree decompositions define a "parsing tree" for vertices of a graph.

Interpretations of Treewidth

- minimum maximum bag size − 1 among all tree decompositions
- minimum clique size − 1 among all chordal completions
- minimum k for which the graph is a partial k-tree

Examples



treewidth 4

$\{1, 2, 3, 5, 6\}$ —— $\{2, 3, 4, 5, 6\}$

Tree Decompositions

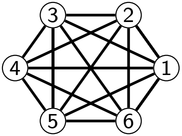
Informal Description

Tree decompositions define a “parsing tree” for vertices of a graph.

Interpretations of Treewidth

- minimum maximum bag size − 1 among all tree decompositions
- minimum clique size − 1 among all chordal completions
- minimum k for which the graph is a partial k-tree

Examples



treewidth 5

$\{1, 2, 3, 4, 5, 6\}$

Branch Decompositions

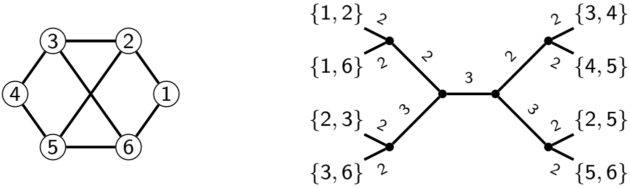
Branch Decomposition

A *branch decomposition* of a graph G is an unrooted binary tree whose leaves are in bijection with $E(G)$.

Branchwidth

Any edge of a branch decomposition tree gives a partition $E(G) = E_1 \dot{\cup} E_2$. The *width* of such an edge is the number of vertices incident to both E_1 and E_2 (cardinality of *vertex separator*). The *branchwidth* of a graph is the minimum maximum width of an edge among all branch decompositions.

Example



Branch Decompositions

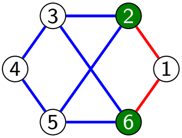
Branch Decomposition

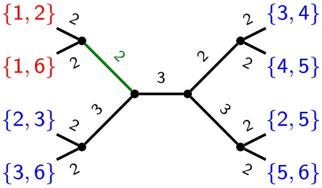
A *branch decomposition* of a graph G is an unrooted binary tree whose leaves are in bijection with $E(G)$.

Branchwidth

Any edge of a branch decomposition tree gives a partition $E(G) = E_1 \dot\cup E_2$. The *width* of such an edge is the number of vertices incident to both E_1 and E_2 (cardinality of *vertex separator*). The *branchwidth* of a graph is the minimum maximum width of an edge among all branch decompositions.

Example





Branch Decompositions

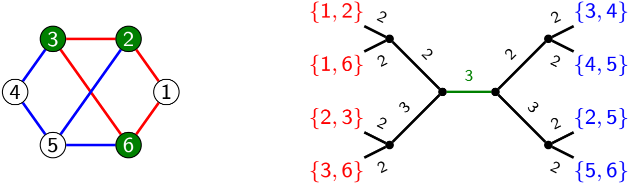
Branch Decomposition

A *branch decomposition* of a graph G is an unrooted binary tree whose leaves are in bijection with $E(G)$.

Branchwidth

Any edge of a branch decomposition tree gives a partition $E(G) = E_1 \dot{\cup} E_2$. The *width* of such an edge is the number of vertices incident to both E_1 and E_2 (cardinality of *vertex separator*). The *branchwidth* of a graph is the minimum maximum width of an edge among all branch decompositions.

Example



More on Tree- and Branchwidth

More Facts

- Computing tree- and branchwidth is in general NP-hard.
- Tree- and branchwidth are within constant factors of each other:

$$bw \leq tw + 1 \leq \lfloor 3/2\, bw \rfloor.$$

Impact on Optimization

Some problems become solvable in polynomial time on graphs with fixed tree- or branchwidth:

- minimum vertex coloring / chromatic number
- maximum independent set
- computing an optimal tree or branch decomposition
- Courcelle's meta theorem on monadic second-order logic

Organization

Outline

- **Papers:** will be sent by e-mail this week.
- **Kick-off:** short (5 minutes) presentation on your topic
  Wednesday, November 20, 10 am, ZIB 2006 (this room)
- **Summary:** 5-8 pages, LaTeX, January
- **Talks:** 45 minutes, February
- Details and reminders will be sent in time by e-mail.

Contact

- Niels Lindner, `lindner@zib.de`

Website

`https://kvv.imp.fu-berlin.de/x/wQ3Qsa` (Whiteboard)