FEDOR V. FOMIN

Part II. Computing treewidth

SCHOOL ON PARAMETERIZED ALGORITHMS AND COMPLEXITY

17-22 August 2014

Będlewo, Poland

Treewidth computation

- In previous lectures we were assuming that a tree decomposition of a graph is given.

Treewidth computation

- In previous lectures we were assuming that a tree decomposition of a graph is given.
- Deciding if a **tw**$(G) \leq k$ is NP-complete. End of the story?

How to compute treewidth

Theorem (Bodlaender 96)

There exists an algorithm that, given an n-vertex graph G and integer k, runs in time $k^{\mathcal{O}(k^3)} \cdot n$ and either constructs a tree decomposition of G of width at most k, or concludes that $\mathbf{tw}(G) > k$.

Approximation

- Polynomial time approximation Feige, Hajiaghayi, and Lee; the approximation ratio is $\mathcal{O}(\sqrt{\log OPT})$;

Approximation

- Polynomial time approximation Feige, Hajiaghayi, and Lee; the approximation ratio is $\mathcal{O}(\sqrt{\log OPT})$;
- Some evidences that no constant factor polynomial time approximation;

Approximation

- Polynomial time approximation Feige, Hajiaghayi, and Lee; the approximation ratio is $\mathcal{O}(\sqrt{\log OPT})$;
- Some evidences that no constant factor polynomial time approximation;
- Constant factor approximation in single-exponential FPT time?

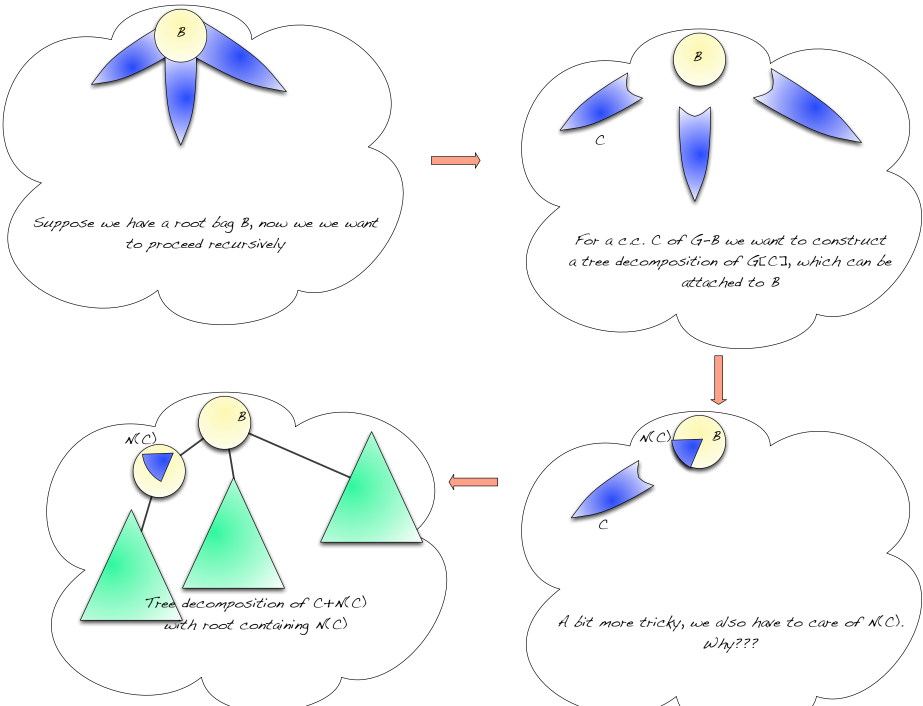
Constant factor parameterized approximation

- Roberston and Seymour 1986: 4-approximation in time $\mathcal{O}(3^{3k} \cdot n^2)$
- $(3+2/3)$-approximation in time $\mathcal{O}(2^{3.6982k} k^3 \cdot n^2)$ [Amir 2010]
- A 5-approximation algorithm running in time $2^{\mathcal{O}(k)} \cdot n$ [Bodlaender, Drange, Dregi, FF, Lokshtanov, and Pilipczuk 2013].

This lecture

Theorem

There exists an algorithm that, given an n-vertex graph G and integer k, runs in time $\mathcal{O}(2^{3k}k^2 \cdot n^2)$ and either constructs a tree decomposition of G of width at most $4k+4$, or concludes that $\mathbf{tw}(G) > k$.


B
Suppose we have a root bag B, now we we want to proceed recursively
B
C
For a c.c. C of G-B we want to construct a tree decomposition of G[C], which can be attached to B
N(C)
B
C
A bit more tricky, we also have to care of N(C). Why???
B
N(C)
Tree decomposition of C+N(C) with root containing N(C)

Basic procedure

- We have a subgraph G' of G and vertex subset $S \subseteq V(G')$;
- S is separating G' from the remaining part of G;
- We want to construct a tree decomposition of G' with root bag containing S.

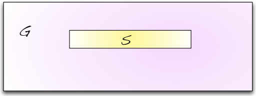
Crucial point

- How to find a bag containing S? Brute force ends up in $n^{\mathcal{O}(k)}$.

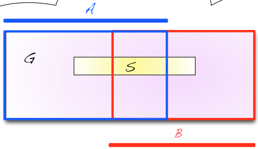
Crucial point

- How to find a bag containing S? Brute force ends up in $n^{\mathcal{O}(k)}$.
- Here balanced separation properties of treewidth become crucial.

Treewidth separation property

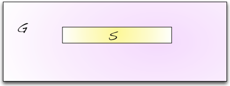


G of treewidth k, S a subset of V(G)

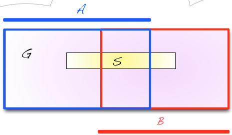


There is a separation (A,B) of G of order at most k+1 such that each of A and B contain at most 2/3 vertices of S

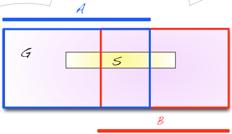
Treewidth separation property: useful point of view



G of treewidth k, S a subset of V(G)



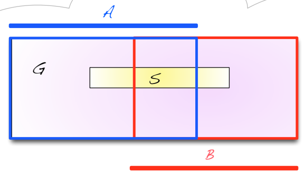
There is a separation (A,B) of G of order at most k+1 such that each of A and B contains at most 2/3 vertices of S



There is a set S of size 3k+1 such that for each separation (A,B) of order k+1, each of A and B contains more than 2k vertices of S, then tw(G)>k.

Question:

For a given set S, how to identify that it is well-linked?



There is a set S of size $3k+1$ such that for each separation (A,B) of order $k+1$, each of A and B contains more than $2k$ vertices of S, then $tw(G)>k$.

Remark

- Takes time $3^{|S|} \cdot n^{\mathcal{O}(1)}$ to try all separations of S.
- In the book we go through all partitions of S, which takes time $2^{|S|} \cdot n^{\mathcal{O}(1)}$, but using partitions is a bit more technical.

Approximating treewidth

- We are given (G, S), $S \subseteq V(G)$ of size $3k+6$. (If G has less than $3k+6$ vertices, construct one-bag decomposition.)
- We want to construct a tree decomposition of width at most $4k+6$ with root bag containing S or to conclude that $\textbf{tw}(G) > k$.

Algorithm

Find a $\frac{2}{3}$-balanced separation (A, B) of S in G of order at most $k + 1$.

If no such (A, B): report that **tw**$(G) > k$

Make $S \cup (A \cap B)$ a root bag, its size does not exceed $3k + 6 + k + 1 = 4k + 7$.

$|A \cap S| \leq \frac{2(3k+6)}{3} = 2k + 4$,
$|(A \cap S) \cup (A \cap B)| \leq 2k + 4 + k + 1 = 3k + 5$.

Create set S_A of size $3k + 6$ by adding to $(A \cap S) \cup (A \cap B)$ some (at least one) vertices of $A \setminus B$. Similarly, create S_B.

Proceed recursively with $(G[A], S_A)$ and $(G[B], S_B)$

Correctness of the algorithm

Running time

Missing detail

- What remains is to prove the Balanced Separation property of tree decompositions

Balanced separation

Lemma (Folklore)

Every n-vertex tree T has a vertex v such that every connected component of $T - v$ has at most $n/2$ vertices.

Proof.

Pick root r and vertex t at the maximum distance from r such that the subtree rooted in t has more than $n/2$ vertices.

□

Balanced separation

- $\mathbf{w}\colon V(G) \to \mathbb{R}_{\geq 0}$: a nonnegative weight function on the vertices of G.
- For $X \subseteq V(G)$, $\mathbf{w}(X) = \sum_{u \in X} \mathbf{w}(u)$.

Definition

For $\alpha \in (0,1)$, set $X \subseteq V(G)$ is an α-*balanced separator* in G if for every connected component D of $G - X$, it holds that $\mathbf{w}(D) \leq \alpha \cdot \mathbf{w}(V(G))$.

Remark: in the approximation algorithm we need only $\alpha \in \{0,1\}$.

Balanced separation

Lemma (Balanced separator)

Assume G is a graph of treewidth at most k, and let $\mathbf{w}\colon V(G) \to \mathbb{R}_{\geq 0}$ be a nonnegative weight function on the vertices of G. Then in G there exists a $\frac{1}{2}$-balanced separator X of size at most $k+1$.

Balanced separation

Lemma (Balanced separator)

Assume G is a graph of treewidth at most k, and let $\mathbf{w}\colon V(G) \to \mathbb{R}_{\geq 0}$ be a nonnegative weight function on the vertices of G. Then in G there exists a $\frac{1}{2}$-balanced separator X of size at most $k+1$.

Proof.

Not much difference with the tree-case. $\square$

Balanced separation

A pair of vertex subsets (A, B) is a *separation* in graph G if $A \cup B = V(G)$ and there is no edge between $A \setminus B$ and $B \setminus A$. The *separator* of (A, B) is $A \cap B$, and the *order* of separation (A, B) is $|A \cap B|$.

A separation (A, B) of G is an α*-balanced separation* if $\mathbf{w}(A \setminus B) \leq \alpha \cdot \mathbf{w}(V(G))$ and $\mathbf{w}(B \setminus A) \leq \alpha \cdot \mathbf{w}(V(G))$.

Balanced separation

Lemma (Balanced separation)

Assume G *is a graph of treewidth at most* k*, and let* $\mathbf{w}\colon V(G) \to \mathbb{R}_{\geq 0}$ *be a nonnegative weight function on the vertices of* G*. Then in* G *there exists a* $\frac{2}{3}$*-balanced separation* (A, B) *of order at most* $k+1$*.*

Proof.

There exists a $\frac{1}{2}$-balanced separator X of size at most $k+1$...

□

Balanced separation

Now the separation property for a set S follows by putting $\mathbf{w}(v) = 1$ if $v \in S$ and $\mathbf{w}(v) = 0$ otherwise.

Balanced separation

Remark: Speed-up from $3^{|}S|$ to $2^{|}S|$ is obtained because of the following

Lemma

Let G be a graph of treewidth at most k and let $S \subseteq V(G)$ be a vertex subset with $|S| = 3k + 4$. Then there exists a partition (S_A, S_B) of S such that $k + 2 \leq |S_A|, |S_B| \leq 2k + 2$ and with a separator of size $\leq k + 1$.