Advanced Graph Algorithms Jan-Apr 2014

Lecture 05 — January 14, 2014

Lecturer: Saket Saurabh *Scribe: Roohani Sharma*

1 Overview

In the last lecture we gave an inductive proof which tells us how to find the tree decomposition of a graph. The aim of this lecture is to give an FPT algorithm to implement the proof of Theorem 9, Lecture 03. The main task is to compute $W-$balanced separators efficiently. The problem arises because there are efficient algorithms (like Ford-Fulkerson Min-Cut Max-Flow algorithm) which find separators of two sets but not multiway separators. Therefore, we agree on a trade-off and find not $W-$balanced separator but $\frac{3}{4} - W-$balanced separator which separate W into two parts (not necessarily connected) that contain not more than $\frac{3}{4}^{th}$ fraction of the vertices in W.

2 Algorithm for finding a tree decomposition of a graph

The problem is to give an algorithm that takes as input a graph G and a positive integer k and outputs correctly one of the following: either G has treewidth $> k$ or outputs a tree decompostion of G of width atmost $5k + 9$.

2.1 Attempt 1

As described earlier, the main task is to find a $k+1$ sized $W-$balanced separator in G. A naive way to do that is to brute force over all $\binom{n}{k+1}$ subsets of $V(G)$, where $n = |V(G)|$. The algorithm NAIVE _COMPUTE _TREE _DECOMPOSITION (G, W, k) solves the mentioned problem naively.

The recurrence for the running time of this algorithm is

$$T(n) \leq n^{t+1} + \sum_{i=1}^{l} T(|G_i| + k + 1)$$

. This solves to $T(n) = O(n^t)$.

2.2 Attempt 2 to give a FPT algorithm

2.2.1 Lemmas and Algorithms to support the main algorithm

Definition 1 ($\alpha - W-$balanced separator)**.** *Let G be a graph and $W \subseteq V(G)$, then for any $\alpha \leq 1$, $S \subseteq V(G)$ is called a α-W-balanced separator if C_X and C_Y are unions of connected components of $G \setminus S$ such that $|C_X \cap W| = X$, $|C_Y \cap W| = Y$ and $0 < |X|, |Y| \leq \alpha|W|$.*

Algorithm 1 COMPUTE_TREE _DECOMPOSITION(G,W,t)

Input:
G: A graph;
k: A positive integer
Output: A tree decomposition of width $3k+4$ or return no concluding that treewidth of G is $> k$

Description:
if $|V| \leq 3k+4$ **then**
return the tree decomposition (T_r, f) such that $V(T) = \{r\}$, $E(G) = \emptyset$ and $f(r) = V(G)$;
else
Take any $W^{'} \supseteq W$ such that $|W^{'}| = 2k+4$;
end if
for all $L \subseteq V(G)$, $|L| = k+1$ {
Check if L is a $W-$balanced separator.
if Yes **then**
$(T^i_{r_i}, f_i) \leftarrow$ COMPUTE_TREE _DECOMPOSITION(G_i,$(W^{'} \cap C_i) \cup S$,k);
return (T_r, f) such that $V(T_r) = \bigcup_{i=1}^{l} V(T^i_{r_i}) \cup \{r\}$, $E(T_r) = \bigcup_{i=1}^{l} E(T^i_{r_i} \cup \bigcup_{i=1}^{l}(r, r_i))$ and $f(b) = S \cup W^{'}$, if $b = r$, and $f(b) = f_i(b)$, if $b \in T^i_{r_i}$;
else
continue ;
end if
}
return NO ;

Lemma 2. *Let G be a graph of treewidth atmost t and $W \subseteq V(G)$ be such that $|W| \geq 2t+3$. Then there exists a $\frac{3}{4} - W-$balanced separator of size atmost $t+1$.*

Proof. Since G has treewidth atmost t, there exists a set S of size $t+1$ such that S is a W-balanced separator (from Theorem , Lecture 04). Let $C_1, \ldots, C_l$ be the connected components of $G \setminus S$ such that $|W \cap C_i \neq 0|$ for all $i \in \{1, \ldots, l\}$. Let $W_i = |W \cap C_i|$. Note that $|W_i| \leq \frac{|W|}{2}$ for all $i \in \{1, \ldots, l\}$ (because S is a W-balanced separator) and $l \geq 2$ (because $W \setminus S = \cup_{i=1}^{l} W_i$, $|W \setminus S| \geq |W| - (t+1) \geq 2t + 3 - t - 1 > \frac{|W|}{2}$. Without loss of generality, assume that $|W_1| \geq \ldots \geq |W_l|$. Let i be minimum such that $\sum_{j=1}^{i} |W_i| > \frac{1}{4}|W \setminus S|$. Let $X = \cup_{j=1}^{i} W_j$ and $Y = \cup_{j=i+1}^{l} W_j$. Clearly, $|X| > \frac{|W \setminus S|}{4} > 0$ and $|Y| = |W \setminus S| - |X| < |W \setminus S| - \frac{|W \setminus S|}{4} = \frac{3}{4}|W \setminus S| \leq \frac{3}{4}|W|$.

Claim: $|X| \leq \frac{3}{4}|W|$.

Proof. If $|W_1| > \frac{1}{4}|W \setminus S|$, then $i = 1$, hence $|X| = |W_1| \leq \frac{|W|}{2} \leq \frac{3}{4}|W|$. Also $|Y| > 0$ because $l \geq 2$.

Otherwise, $|W_1| \leq \frac{|W \setminus S|}{4}$. Therefore, $|W_i| \leq \frac{|W \setminus S}{4}$. Hence, $|X| = \sum_{j=1}^{i} |W_j| = \sum_{j=1}^{i-1} |W_j| + |W_i| \leq \frac{|W \setminus S|}{4} + \frac{|W \setminus S|}{4} \leq \frac{|W \setminus S|}{2} < \frac{3}{4}|W \setminus S| \leq \frac{3}{4}|W|$. Also since $|X| < \frac{3}{4}|W \setminus S|$, $|Y| > |W \setminus S| - \frac{3}{4}|W \setminus S| > \frac{|W \setminus S|}{4} > 0$. Thus S is a $\frac{3}{4} - W-$balanced separator of size atmost $t+1$. □

□

To sum up we make the following remark

Remark 1. *If G has treewidth atmost t then for all $W \subseteq V(G)$, G has a W-balanced separator of size $t+1$ [Theorem , Lecture 04]. If for all $W \subseteq V(G)$, G has a W-balanced separator of size $t+1$ then G has a $\frac{3}{4} - W-$balanced separator of size $t+1$ [Lemma 2]. Hence, if there exists a $W \subseteq V(G)$ such that no $\frac{3}{4} - W-$balanced seperator of size $t+1$ then G has treewidth $> t$.*

Lemma 3. *Given a graph G, a positve integer k and $X, Y \subseteq V(G)$, there exists an algorithm that outputs a set S, if it exists, such that $S \subseteq V(G) \setminus (X \cup Y)$, $|S| \leq k$ and S seperates X from Y.*

Proof. The algorithm FINDSEPERATOR (G, k, X, Y) solves the required problem.

The correctness of the algorithm follows from Menger's Theorem and from the correctness of Ford-Fulkerson algorithm. □

Theorem 4 (Menger's Theorem)**.** *Let G be a graph and $X, Y \subseteq V(G)$. Then the minimum number of vertices separating X from Y in G is equal to the maximum number of disjoint $X - Y$ paths in G.*

Note that the algorithm runs in time $O(n^3) + O(n^2) = O(n^3)$.

Lemma 5. *Given a graph G, a positive integer k and $W \subseteq V(G)$ such that $|W| \geq 2k+3$, there exists an algorirhm that outputs a set S of size atmost $k+1$ such that S is a $\frac{3}{4} - W-$balanced separator, if it exists, otherwise return no concluding that G has treewidth $> k$.*

Algorithm 2 FINDSEPERATOR(G,k,X,Y)

Initialization:

Construct $G^{'}$ as follows.
$V(G^{'}) = (V(G) \setminus (X \cup Y)) \cup \{s, t\}$,
$E(G^{'}) = E(G[V(G) \setminus (X \cup Y)]) \cup \{(s, N_{G\setminus(X\cup Y)}(X))\} \cup \{(t, N_{G\setminus(X\cup Y)}(Y))\}$ and,
$w : E(G^{'}) \rightarrow \mathbb{Z}_{\geq 0}$ such that

$$\begin{aligned} w((u,v)) &= d(v) \text{ if } u = s \text{ or } v = t \\ &= 1 \text{ otherwise} \end{aligned}$$

Description:

Run Ford-Fulkerson Min-Cut Max-Flow algorithm on $G^{'}$. The graph corresponding to the final residual graph will have saturated disjoint paths from s to t. On each of the saturated paths pick a vertex and form the set S. If $S \leq k$ output S else return no.

Proof. From Lemma 2, we know if G has treewidth atmost k, then for all $W \subseteq V(G)$, $|W| \leq 2k+3$, there exists a $\frac{3}{4} - W-$balanced separator of size atmost $k+1$.

If a $\frac{3}{4} - W-$balanced separator of size atmost $k+1$, say S, exists then S should partition W such that $W = X \uplus Y \uplus Z$, $Z \subseteq S$ and for some C_X and C_Y (which are connected components of $G \setminus S$), $X \subseteq C_X$, $Y \subseteq C_Y$.

Let us guess each such partition of W into X, Y, Z and run algorithm FINDSEPERATOR($G \setminus Z$, X, Y, $(t+1) - |Z|$) which returns either $S^{'}$ or no. If for all partitions the algorithm returns no, then return no otherwise, return $S = S^{'} \cup Z$. The correctness of the algorithm follows because, if for all partitions no such S exists that means for the chosen W, there is no $\frac{3}{4} - W-$balanced separator which implies G has treewidth $> k$. □

2.2.2 The main algorithm

Theorem 6. *Given a graph G, a positive integer k and $W \subseteq V(G)$ such that $|W| \leq 4k+8$, there exists an algorithm that correctly outputs one of the following: either G has treewidth $> k$ or returns a tree decomposition of G of width $5k+9$.*

Proof. The algorithm COMPUTE_TREE_DECOMPOSITION(G,W,t) computes the required result.

The correctness of the algorithm follows from lemmas of previous section. The correctness of the recursive step in the algorithm follows from the following argument.

- Note that $|V(G_i)| < |V(G)|$ because S is a $\frac{3}{4} - W-$balanced separator.
- $|(W^{'} \cap C_i)| \cup S =\leq \frac{3}{4}|W^{'}| + k + 1 \leq \frac{3}{4}(4k+8) + k + 1 = 3t + 6 + t + 1 < |W|$.

Algorithm 3 COMPUTE_TREE _DECOMPOSITION(G,W,t)

Input:
G: A graph;
k: A positive integer
Output: A tree decomposition of width $5k+9$ or return no concluding that treewidth of G is $> k$

Description:
if $|V| \leq 5k+9$ **then**
return the tree decomposition (T, f) such that $V(T) = \{r\}$, $E(G) = \emptyset$ and $f(r) = V(G)$;
else
Take any $W^{'} \supseteq W$ such that $|W^{'}| = 4k+8$;
end if
Check if there exists a $\frac{3}{4} - W^{'}-$ balanced separator of size atmost $k+1$ in G using the algorithm described in the proof of Lemma 5.
if the previous step returns Yes **then**
Let S be the $\frac{3}{4} - W^{'}-$balanced separator returned ;
Let $C_1, \ldots, C_l$ be the connected components of $G \setminus S$;
else
return No
end if
for (i=1; $i \leq l; i++$) {
Let $G_i \leftarrow G[C_i \cup S]$;
$(T^i_{r_i}, f_i) \leftarrow$ COMPUTE_TREE _DECOMPOSITION(G_i,$(W^{'} \cap C_i) \cup S$,k);
}
return (T_r, f) such that $V(T_r) = \bigcup_{i=1}^{l} V(T^l_{r_i}) \cup \{r\}$, $E(T_r) = \bigcup_{i=1}^{l} E(T^i_{r_i} \cup \bigcup_{i=1}^{l}(r, r_i))$ and $f(b) = S \cup W^{'}$, if $b = r$ and $f(b) = f_i(b)$, if $b \in T^i_{r_i}$;

□

The recurrence relation for the running time of the algorithm is as follows.

$$T(n) \leq \sum_{i=1}^{n} T(|G_i| + k + 1) + 3^{4k+8} n^3$$

. Thus, $T(n) = 3^{4k+8} O(n^3)$.

3 Other useful properties of tree decomposition

Definition 7 (Simple Tree Decomposition)**.** *A tree decomposition (T_r, f) is simple if for any two nodes $t_1, \quad t_2 \in V(T_r)$, $f(t_1) \not\subseteq f(t_2)$ and $f(t_2) \not\subseteq f(t_1)$.*

Lemma 8. *Let G be a graph of treewidth t then there exists a simple tree decomposition (T_r, f) of width t.*

Proof. Since G has treewidth t, then there exists a tree decomposition of G, say (T_r, f) of width t. For any $t_1, t_2 \in V(T_r)$, such that $f(t_1) \subseteq f(t_2)$, if $(t_1, t_2) \in E(T_r)$ then contract the edge (t_1, t_2). It is easy to see that the tree obtained after this operation is still a valid tree decomposition. If $(t_1, t_2) \notin E(T_r)$, then let $t_1 t_{i_1} \ldots t_{i_x} t_2$ be the unique path from t_1 to t_2 in T_r. Then $f(t_1) \subseteq f(t_{i_1}) \subseteq \ldots \subseteq f(t_{i_x}) \subseteq f(t_2)$. For this case, contract the edges in order on this path. □

Lemma 9. *For any graph G of treewidth t, there exists a tree decomposition (T_r, f) of width t such that $|V(T_r)| \leq |V(G)|$.*

Proof. If G has treewidth t then there exists a simple tree decomposition of width t [Lemma 8]. Let (T_r, f) be the simple tree decomposition of G. Let $\theta : V(G) \to V(T_r)$. We need to prove that θ is a surjective function. For every $v \in V(G)$, let $X_v = \{t | t \in X_v$ and $dist(r, t) = min_{t' \in V(T_r)}\{d(r, t')\}\}$. Clearly θ is a function (and not a map). What remains to show is that for all $t \in V(T_r)$, $\theta^{-1}(t) \neq \{\}$. If there exists a $t \in V(T_r)$ such that $\theta^{-1}(t) = \{\}$ then for all $v \in f(t)$, $\theta(v)$ map v to some ancestor (say t') of t in the tree, which implies $f(t) \subseteq f(t')$. This is a contradiction to the fact that the tree decomposition is simple. If $t = r$, then it implies that $f(r) = \emptyset$, which is again a contradiction beacuse then $f(r) \subseteq f(t'')$, for all $t'' \in V(T_r), t'' \neq r$. □