# Important separators and parameterized algorithms

Dániel Marx[1]

[1]Institute for Computer Science and Control,
Hungarian Academy of Sciences (MTA SZTAKI)
Budapest, Hungary

School on Parameterized Algorithms and Complexity
Będlewo, Poland
August 19, 2014

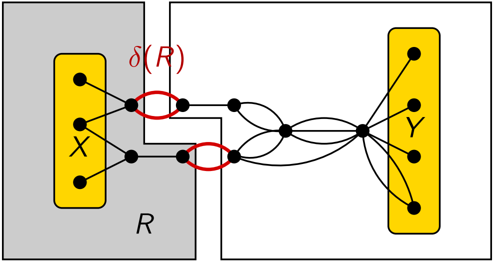
## Main message

Small separators in graphs have interesting extremal properties that can be exploited in combinatorial and algorithmic results.

- Bounding the number of “important” cuts.
- Edge/vertex versions, directed/undirected versions.
- Algorithmic applications: FPT algorithm for
  - MULTIWAY CUT,
  - DIRECTED FEEDBACK VERTEX SET, and
  - $(p, q)$-CLUSTERING.
- Random selection of important separators: a new tool with many applications.

**Definition:** $\delta(R)$ is the set of edges with exactly one endpoint in $R$.
**Definition:** A set $S$ of edges is a **minimal** $(X, Y)$-**cut** if there is no $X - Y$ path in $G \setminus S$ and no proper subset of $S$ breaks every $X - Y$ path.

**Observation:** Every minimal $(X, Y)$-cut $S$ can be expressed as $S = \delta(R)$ for some $X \subseteq R$ and $R \cap Y = \emptyset$.

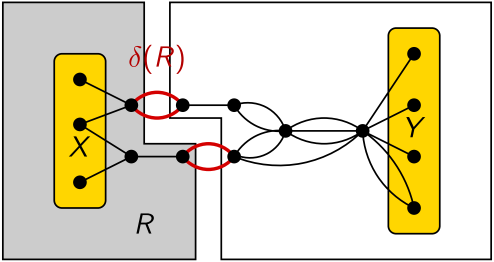


**Theorem**

A minimum $(X, Y)$-cut can be found in polynomial time.

**Theorem**

The size of a minimum $(X, Y)$-cut equals the maximum size of a pairwise edge-disjoint collection of $X - Y$ paths.



There is a long list of algorithms for finding disjoint paths and minimum cuts.

- Edmonds-Karp: $O(|V(G)| \cdot |E(G)|^2)$
- Dinitz: $O(|V(G)|^2 \cdot |E(G)|)$
- Push-relabel: $O(|V(G)|^3)$
- Orlin-King-Rao-Tarjan: $O(|V(G)| \cdot |E(G)|)$
- $\ldots$

But we need only the following result:

**Theorem**

An $(X, Y)$-cut of size at most $k$ (if exists) can be found in time $O(k \cdot (|V(G)| + |E(G)|))$.

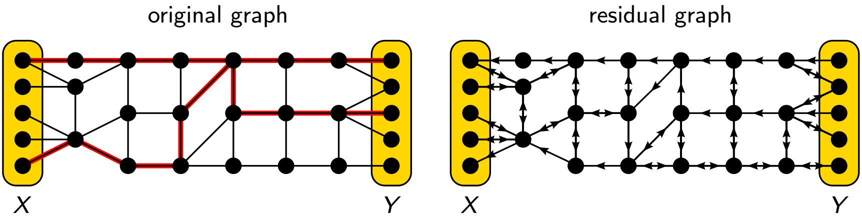
**Theorem**

An $(X,Y)$-cut of size at most $k$ (if exists) can be found in time $O(k \cdot (|V(G)| + |E(G)|))$.

We try to grow a collection $\mathcal{P}$ of edge-disjoint $X - Y$ paths.

**Residual graph:**

- not used by $\mathcal{P}$: bidirected,
- used by $\mathcal{P}$: directed in the opposite direction.



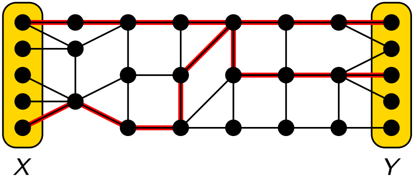
**Theorem**

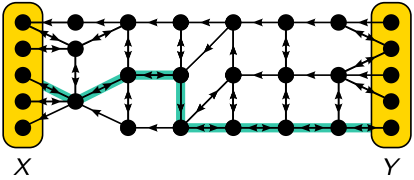
An $(X, Y)$-cut of size at most $k$ (if exists) can be found in time $O(k \cdot (|V(G)| + |E(G)|))$.

We try to grow a collection $\mathcal{P}$ of edge-disjoint $X - Y$ paths.

**Residual graph:**

- not used by $\mathcal{P}$: bidirected,
- used by $\mathcal{P}$: directed in the opposite direction.


original graph


residual graph

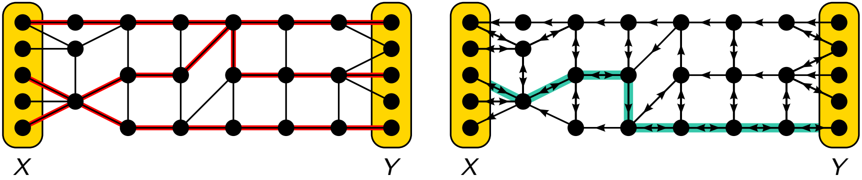
**Theorem**

An $(X, Y)$-cut of size at most $k$ (if exists) can be found in time $O(k \cdot (|V(G)| + |E(G)|))$.

We try to grow a collection $\mathcal{P}$ of edge-disjoint $X - Y$ paths.

**Residual graph:**

- not used by $\mathcal{P}$: bidirected,
- used by $\mathcal{P}$: directed in the opposite direction.


original graph

residual graph

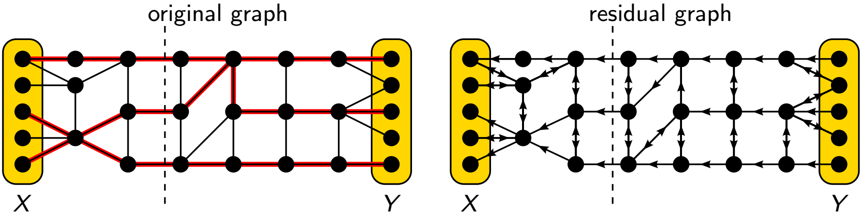
**Theorem**

An $(X, Y)$-cut of size at most $k$ (if exists) can be found in time $O(k \cdot (|V(G)| + |E(G)|))$.

We try to grow a collection $\mathcal{P}$ of edge-disjoint $X - Y$ paths.

**Residual graph:**

- not used by $\mathcal{P}$: bidirected,
- used by $\mathcal{P}$: directed in the opposite direction.



If we cannot find an augmenting path, we can find a (minimum) cut of size $|\mathcal{P}|$.

# Finding minimum cuts

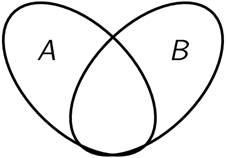
**Fact:** The function $\delta$ is **submodular:** for arbitrary sets $A, B$,

$$|\delta(A)| \quad + \quad |\delta(B)| \quad \geq \quad |\delta(A \cap B)| \quad + \quad |\delta(A \cup B)|$$

**Fact:** The function $\delta$ is **submodular:** for arbitrary sets $A, B$,

$$|\delta(A)| \quad + \quad |\delta(B)| \quad \geq \quad |\delta(A \cap B)| \quad + \quad |\delta(A \cup B)|$$

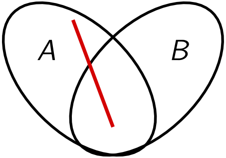
**Proof:** Determine separately the contribution of the different types of edges.



**Fact:** The function $\delta$ is **submodular:** for arbitrary sets $A, B$,

$$\underset{0}{|\delta(A)|} \quad + \quad \underset{1}{|\delta(B)|} \quad \geq \quad \underset{1}{|\delta(A \cap B)|} \quad + \quad \underset{0}{|\delta(A \cup B)|}$$

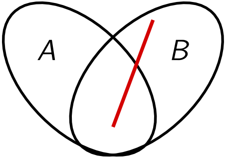
**Proof:** Determine separately the contribution of the different types of edges.



**Fact:** The function $\delta$ is **submodular:** for arbitrary sets $A, B$,

$$\underset{1}{|\delta(A)|} \quad + \quad \underset{0}{|\delta(B)|} \quad \geq \quad \underset{1}{|\delta(A \cap B)|} \quad + \quad \underset{0}{|\delta(A \cup B)|}$$

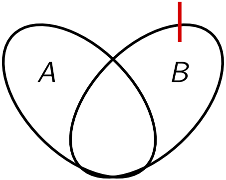
**Proof:** Determine separately the contribution of the different types of edges.



**Fact:** The function $\delta$ is **submodular:** for arbitrary sets $A, B$,

$$\underset{0}{|\delta(A)|} \quad + \quad \underset{1}{|\delta(B)|} \quad \geq \quad \underset{0}{|\delta(A \cap B)|} \quad + \quad \underset{1}{|\delta(A \cup B)|}$$

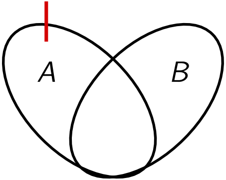
**Proof:** Determine separately the contribution of the different types of edges.



**Fact:** The function $\delta$ is **submodular:** for arbitrary sets $A, B$,

$$\underset{1}{|\delta(A)|} \quad + \quad \underset{0}{|\delta(B)|} \quad \geq \quad \underset{0}{|\delta(A \cap B)|} \quad + \quad \underset{1}{|\delta(A \cup B)|}$$

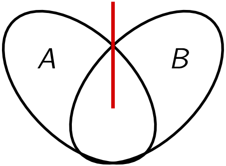
**Proof:** Determine separately the contribution of the different types of edges.



**Fact:** The function $\delta$ is **submodular:** for arbitrary sets $A, B$,

$$\underset{1}{|\delta(A)|} \quad + \quad \underset{1}{|\delta(B)|} \quad \geq \quad \underset{1}{|\delta(A \cap B)|} \quad + \quad \underset{1}{|\delta(A \cup B)|}$$

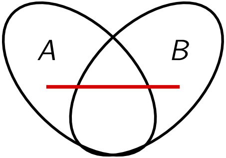
**Proof:** Determine separately the contribution of the different types of edges.



**Fact:** The function $\delta$ is **submodular:** for arbitrary sets $A, B$,

$$\underset{1}{|\delta(A)|} \quad + \quad \underset{1}{|\delta(B)|} \quad \geq \quad \underset{0}{|\delta(A \cap B)|} \quad + \quad \underset{0}{|\delta(A \cup B)|}$$

**Proof:** Determine separately the contribution of the different types of edges.



### Lemma

Let $\lambda$ be the minimum $(X, Y)$-cut size. There is a unique maximal $R_{\text{max}} \supseteq X$ such that $\delta(R_{\text{max}})$ is an $(X, Y)$-cut of size $\lambda$.

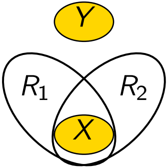
**Lemma**

Let $\lambda$ be the minimum $(X, Y)$-cut size. There is a unique maximal $R_{\max} \supseteq X$ such that $\delta(R_{\max})$ is an $(X, Y)$-cut of size $\lambda$.

**Proof:** Let $R_1, R_2 \supseteq X$ be two sets such that $\delta(R_1), \delta(R_2)$ are $(X, Y)$-cuts of size $\lambda$.

$$\underset{\lambda}{|\delta(R_1)|} + \underset{\lambda}{|\delta(R_2)|} \geq \underset{\geq \lambda}{|\delta(R_1 \cap R_2)|} + |\delta(R_1 \cup R_2)|$$

$$\Rightarrow |\delta(R_1 \cup R_2)| \leq \lambda$$



**Note:** Analogous result holds for a unique minimal $R_{\min}$.

**Lemma**

Given a graph $G$ and sets $X, Y \subseteq V(G)$, the sets $R_{\min}$ and $R_{\max}$ can be found in polynomial time.

**Proof:** Iteratively add vertices to $X$ if they do not increase the minimum $X - Y$ cut size. When the process stops, $X = R_{\max}$. Similar for $R_{\min}$.

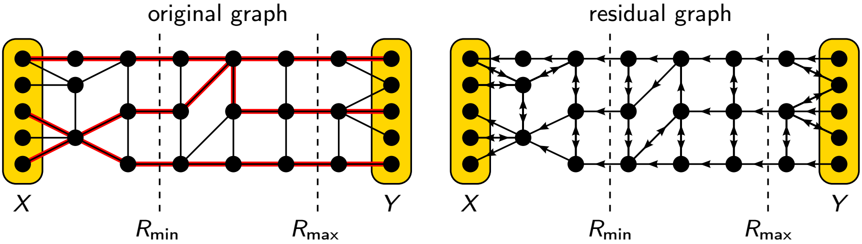
But we can do better!

# Finding $R_{\min}$ and $R_{\max}$

**Lemma**

Given a graph $G$ and sets $X, Y \subseteq V(G)$, the sets $R_{\min}$ and $R_{\max}$ can be found in $O(\lambda \cdot (|V(G)| + |E(G)|))$ time, where $\lambda$ is the minimum $X - Y$ cut size.

**Proof:** Look at the residual graph.



$R_{\min}$: vertices reachable from $X$.

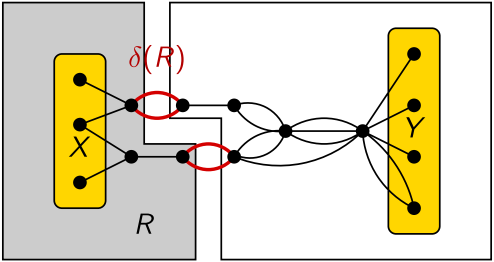
$R_{\max}$: vertices from which $Y$ is **not** reachable.

# Finding $R_{\min}$ and $R_{\max}$

**Definition:** $\delta(R)$ is the set of edges with exactly one endpoint in $R$.

**Definition:** A set $S$ of edges is a **minimal** $(X,Y)$**-cut** if there is no $X-Y$ path in $G\setminus S$ and no proper subset of $S$ breaks every $X-Y$ path.

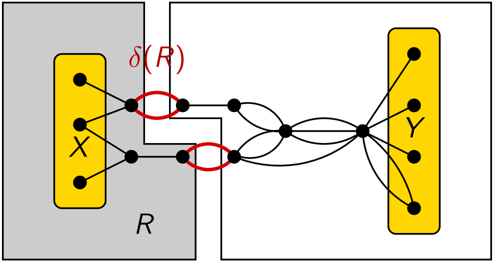
**Observation:** Every minimal $(X,Y)$-cut $S$ can be expressed as $S=\delta(R)$ for some $X\subseteq R$ and $R\cap Y=\emptyset$.



**Definition**

A minimal $(X, Y)$-cut $\delta(R)$ is **important** if there is no $(X, Y)$-cut $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$.

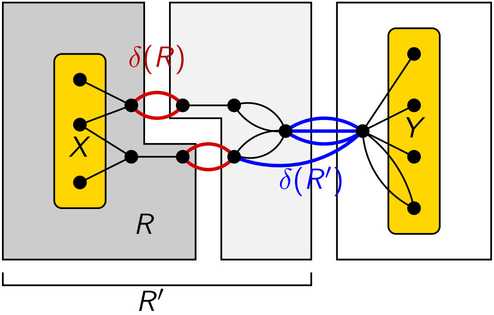
**Note:** Can be checked in polynomial time if a cut is important ($\delta(R)$ is important if $R = R_{\max}$).



**Definition**

A minimal $(X, Y)$-cut $\delta(R)$ is **important** if there is no $(X, Y)$-cut $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$.

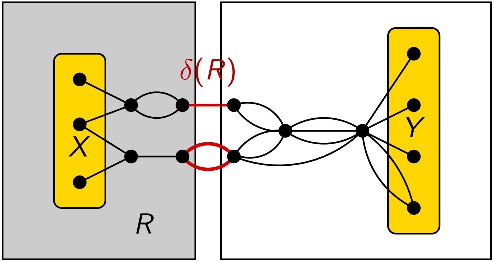
**Note:** Can be checked in polynomial time if a cut is important ($\delta(R)$ is important if $R = R_{\max}$).



**Definition**

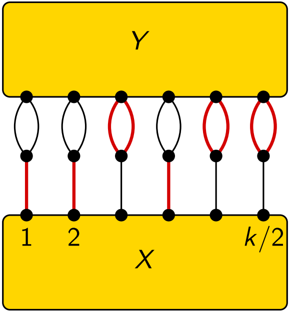
A minimal $(X, Y)$-cut $\delta(R)$ is **important** if there is no $(X, Y)$-cut $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$.

**Note:** Can be checked in polynomial time if a cut is important ($\delta(R)$ is important if $R = R_{\max}$).



The number of important cuts can be exponentially large.

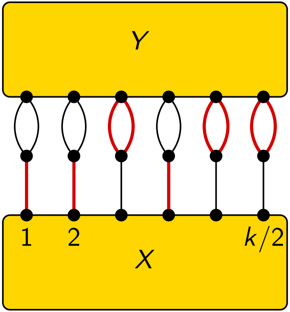
**Example:**



This graph has $2^{k/2}$ important $(X, Y)$-cuts of size at most $k$.

The number of important cuts can be exponentially large.

**Example:**



This graph has $2^{k/2}$ important $(X, Y)$-cuts of size at most $k$.

### Theorem

There are at most $4^k$ important $(X, Y)$-cuts of size at most $k$.

**Theorem**

There are at most $4^k$ important $(X, Y)$-cuts of size at most $k$.

**Proof:** Let $\lambda$ be the minimum $(X, Y)$-cut size and let $\delta(R_{\max})$ be the unique important cut of size $\lambda$ such that $R_{\max}$ is maximal.

(1) We show that $R_{\max} \subseteq R$ for every important cut $\delta(R)$.

**Theorem**

There are at most $4^k$ important $(X, Y)$-cuts of size at most $k$.

**Proof:** Let $\lambda$ be the minimum $(X, Y)$-cut size and let $\delta(R_{\max})$ be the unique important cut of size $\lambda$ such that $R_{\max}$ is maximal.

(1) We show that $R_{\max} \subseteq R$ for every important cut $\delta(R)$.

By the submodularity of $\delta$:

$$\underset{\lambda}{|\delta(R_{\max})|} + |\delta(R)| \geq \underset{\geq \lambda}{|\delta(R_{\max} \cap R)|} + |\delta(R_{\max} \cup R)|$$

$$\Downarrow$$

$$|\delta(R_{\max} \cup R)| \leq |\delta(R)|$$

$$\Downarrow$$

If $R \neq R_{\max} \cup R$, then $\delta(R)$ is not important.

### Theorem

There are at most $4^k$ important $(X, Y)$-cuts of size at most $k$.

**Proof:** Let $\lambda$ be the minimum $(X, Y)$-cut size and let $\delta(R_{\text{max}})$ be the unique important cut of size $\lambda$ such that $R_{\text{max}}$ is maximal.

(1) We show that $R_{\text{max}} \subseteq R$ for every important cut $\delta(R)$.

By the submodularity of $\delta$:

$$\underset{\lambda}{|\delta(R_{\text{max}})|} + |\delta(R)| \geq \underset{\geq \lambda}{|\delta(R_{\text{max}} \cap R)|} + |\delta(R_{\text{max}} \cup R)|$$

$$\Downarrow$$

$$|\delta(R_{\text{max}} \cup R)| \leq |\delta(R)|$$

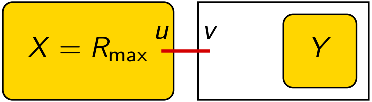
$$\Downarrow$$

If $R \neq R_{\text{max}} \cup R$, then $\delta(R)$ is not important.

Thus the important $(X, Y)$- and $(R_{\text{max}}, Y)$-cuts are the same.
$\Rightarrow$ We can assume $X = R_{\text{max}}$.

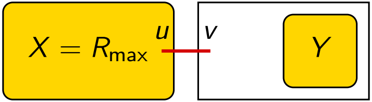
(2) Search tree algorithm for enumerating all these cuts:

An (arbitrary) edge $uv$ leaving $X = R_{\text{max}}$ is either in the cut or not.



(2) Search tree algorithm for enumerating all these cuts:

An (arbitrary) edge $uv$ leaving $X = R_{\max}$ is either in the cut or not.

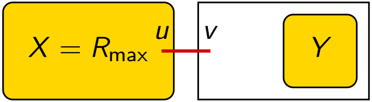


**Branch 1:** If $uv \in S$, then $S \setminus uv$ is an important $(X, Y)$-cut of size at most $k - 1$ in $G \setminus uv$.

**Branch 2:** If $uv \notin S$, then $S$ is an important $(X \cup v, Y)$-cut of size at most $k$ in $G$.

(2) Search tree algorithm for enumerating all these cuts:

An (arbitrary) edge $uv$ leaving $X = R_{\max}$ is either in the cut or not.



**Branch 1:** If $uv \in S$, then $S \setminus uv$ is an important $(X, Y)$-cut of size at most $k - 1$ in $G \setminus uv$.

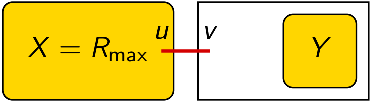
$\Rightarrow$ $k$ decreases by one, $\lambda$ decreases by at most $1$.

**Branch 2:** If $uv \notin S$, then $S$ is an important $(X \cup v, Y)$-cut of size at most $k$ in $G$.

$\Rightarrow$ $k$ remains the same, $\lambda$ increases by $1$.

(2) Search tree algorithm for enumerating all these cuts:

An (arbitrary) edge $uv$ leaving $X = R_{\max}$ is either in the cut or not.



**Branch 1:** If $uv \in S$, then $S \setminus uv$ is an important $(X, Y)$-cut of size at most $k - 1$ in $G \setminus uv$.

$\Rightarrow$ $k$ decreases by one, $\lambda$ decreases by at most $1$.

**Branch 2:** If $uv \notin S$, then $S$ is an important $(X \cup v, Y)$-cut of size at most $k$ in $G$.

$\Rightarrow$ $k$ remains the same, $\lambda$ increases by $1$.

The measure $2k - \lambda$ decreases in each step.

$\Rightarrow$ Height of the search tree $\leq 2k$

$\Rightarrow$ $\leq 2^{2k} = 4^k$ important cuts of size at most $k$.

We are using the following two statements:

**Branch 1:** If $uv \in S$, then

$S$ is an important $(X, Y)$-cut in $G$ ➡ $S \setminus uv$ is an important $(X, Y)$-cut in $G \setminus uv$

**Branch 2:** If $S$ is an $(X \cup v, Y)$-cut, then

$S$ is an important $(X, Y)$-cut in $G$ ➡ $S$ is an important $(X \cup v, Y)$-cut in $G$

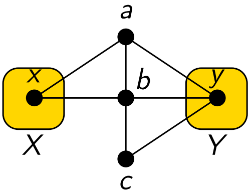
We are using the following two statements:

**Branch 1:** If $uv \in S$, then

$S$ is an important $(X, Y)$-cut in $G$ $\Rightarrow$ $\not\Leftarrow$ $S \setminus uv$ is an important $(X, Y)$-cut in $G \setminus uv$

Converse is not true:

Set $\{ab, ay\}$ is important $(X, Y)$-cut in $G \setminus xb$, but $\{xb, ab, ay\}$ is not an important $(X, Y)$-cut in $G$.



**Branch 2:** If $S$ is an $(X \cup v, Y)$-cut, then

$S$ is an important $(X, Y)$-cut in $G$ $\Leftrightarrow$ $S$ is an important $(X \cup v, Y)$-cut in $G$

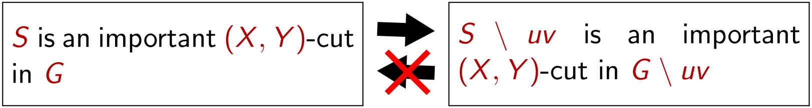
Converse is true!

# Important cuts — some details

# Important cuts — some details

We are using the following two statements:

**Branch 1:** If $uv \in S$, then



$S$ is an important $(X, Y)$-cut in $G$

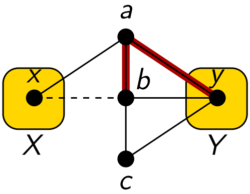
$\Rightarrow$

$\cancel{\Leftarrow}$

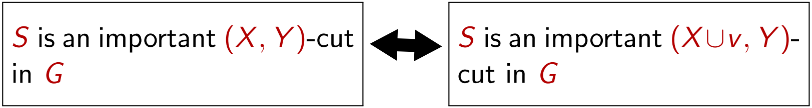
$S \setminus uv$ is an important $(X, Y)$-cut in $G \setminus uv$

Converse is not true:

Set $\{ab, ay\}$ is important $(X, Y)$-cut in $G \setminus xb$, but $\{xb, ab, ay\}$ is not an important $(X, Y)$-cut in $G$.



**Branch 2:** If $S$ is an $(X \cup v, Y)$-cut, then



$S$ is an important $(X, Y)$-cut in $G$

$\Leftrightarrow$

$S$ is an important $(X \cup v, Y)$-cut in $G$

Converse is true!

**Theorem**

There are at most $4^k$ important $(X, Y)$-cuts of size at most $k$ and they can be enumerated in time $O(4^k \cdot k \cdot (|V(G)| + |E(G)|))$.

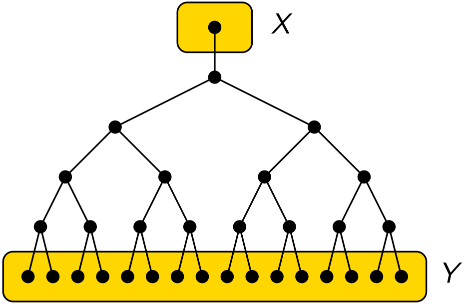
Algorithm for enumerating important cuts:

1. Handle trivial cases ($k = 0$, $\lambda = 0$, $k < \lambda$)
2. Find $R_{\max}$.
3. Choose an edge $uv$ of $\delta(R_{\max})$.
    - Recurse on $(G - uv, R_{\max}, Y, k - 1)$.
    - Recurse on $(G, R_{\max} \cup v, Y, k)$.
4. Check if the returned cuts are important and throw away those that are not.

### Theorem

There are at most $4^k$ important $(X, Y)$-cuts of size at most $k$.

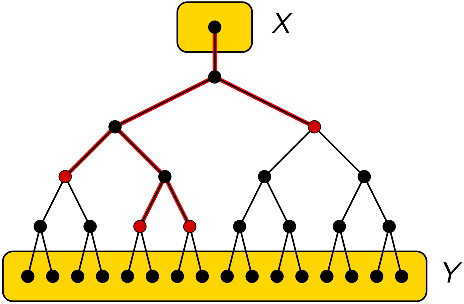
**Example:** The bound $4^k$ is essentially tight.



**Theorem**

There are at most $4^k$ important $(X,Y)$-cuts of size at most $k$.

**Example:** The bound $4^k$ is essentially tight.

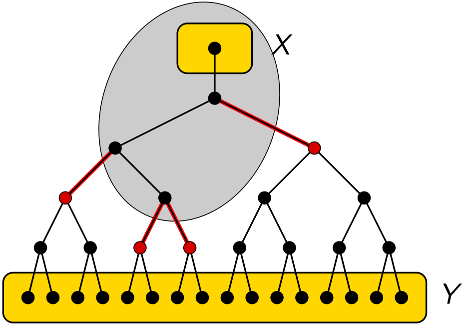


Any subtree with $k$ leaves gives an important $(X,Y)$-cut of size $k$.

**Theorem**

There are at most $4^k$ important $(X, Y)$-cuts of size at most $k$.

**Example:** The bound $4^k$ is essentially tight.

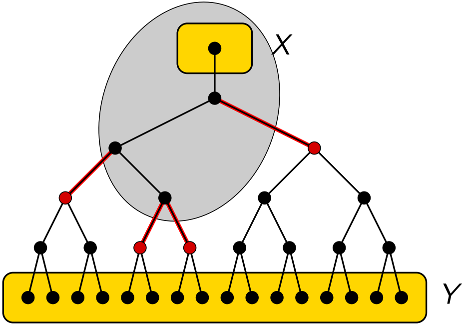


Any subtree with $k$ leaves gives an important $(X, Y)$-cut of size $k$.

**Theorem**

There are at most $4^k$ important $(X, Y)$-cuts of size at most $k$.

**Example:** The bound $4^k$ is essentially tight.



Any subtree with $k$ leaves gives an important $(X, Y)$-cut of size $k$.
The number of subtrees with $k$ leaves is the Catalan number

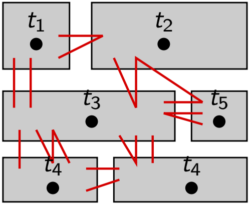
$$C_{k-1} = \frac{1}{k}\binom{2k-2}{k-1} \geq 4^k/\text{poly}(k).$$

**Definition:** A **multiway cut** of a set of terminals $T$ is a set $S$ of edges such that each component of $G \setminus S$ contains at most one vertex of $T$.

MULTIWAY CUT

**Input:** Graph $G$, set $T$ of vertices, integer $k$

**Find:** A **multiway cut** $S$ of at most $k$ edges.

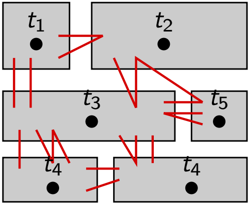


Polynomial for $|T| = 2$, but NP-hard for any fixed $|T| \geq 3$ [Dalhaus et al. 1994].

**Definition:** A **multiway cut** of a set of terminals $T$ is a set $S$ of edges such that each component of $G \setminus S$ contains at most one vertex of $T$.

MULTIWAY CUT

| | |
|---|---|
| **Input:** | Graph $G$, set $T$ of vertices, integer $k$ |
| **Find:** | A **multiway cut** $S$ of at most $k$ edges. |

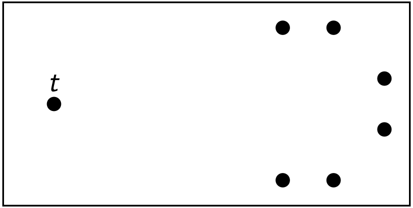


Trivial to solve in polynomial time for fixed $k$ (in time $n^{O(k)}$).

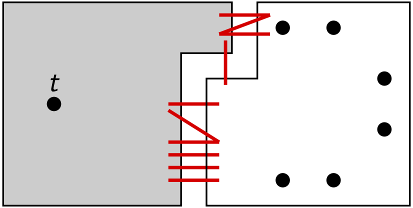
**Theorem**

MULTIWAY CUT can be solved in time $4^k \cdot k^3 \cdot (|V(G)| + |E(G)|)$.

**Intuition:** Consider a $t \in T$. A subset of the solution $S$ is a $(t, T \setminus t)$-cut.

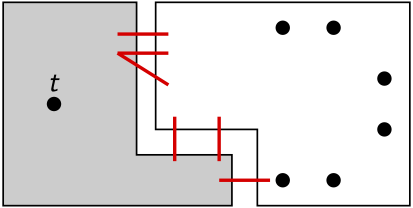


**Intuition:** Consider a $t \in T$. A subset of the solution $S$ is a $(t, T \setminus t)$-cut.



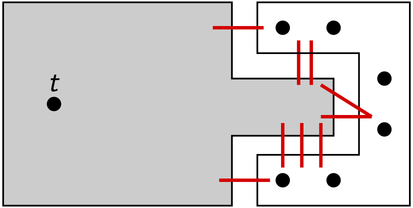
There are many such cuts.

**Intuition:** Consider a $t \in T$. A subset of the solution $S$ is a $(t, T \setminus t)$-cut.



There are many such cuts.

**Intuition:** Consider a $t \in T$. A subset of the solution $S$ is a $(t, T \setminus t)$-cut.



There are many such cuts.

But a cut farther from $t$ and closer to $T \setminus t$ seems to be more useful.

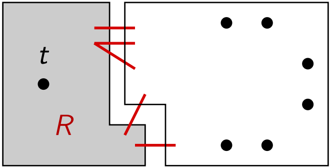
**Pushing Lemma**

Let $t \in T$. The MULTIWAY CUT problem has a solution $S$ that contains an important $(t, T \setminus t)$-cut.

**Pushing Lemma**

Let $t \in T$. The MULTIWAY CUT problem has a solution $S$ that contains an important $(t, T \setminus t)$-cut.

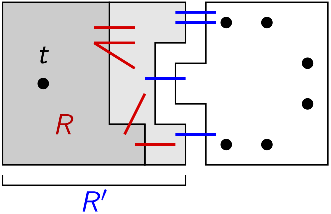
**Proof:** Let $R$ be the vertices reachable from $t$ in $G \setminus S$ for a solution $S$.



## Pushing Lemma

Let $t \in T$. The MULTIWAY CUT problem has a solution $S$ that contains an important $(t, T \setminus t)$-cut.

**Proof:** Let $R$ be the vertices reachable from $t$ in $G \setminus S$ for a solution $S$.

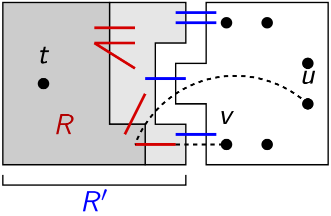


$\delta(R)$ is not important, then there is an important cut $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$. Replace $S$ with $S' := (S \setminus \delta(R)) \cup \delta(R') \Rightarrow |S'| \leq |S|$

### Pushing Lemma

Let $t \in T$. The MULTIWAY CUT problem has a solution $S$ that contains an important $(t, T \setminus t)$-cut.

**Proof:** Let $R$ be the vertices reachable from $t$ in $G \setminus S$ for a solution $S$.



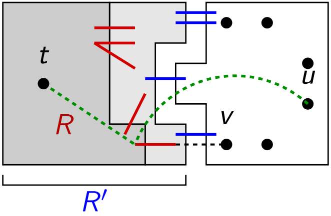
$\delta(R)$ is not important, then there is an important cut $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$. Replace $S$ with
$S' := (S \setminus \delta(R)) \cup \delta(R') \Rightarrow |S'| \leq |S|$

$S'$ is a multiway cut: (1) There is no $t$-$u$ path in $G \setminus S'$ and (2) a $u$-$v$ path in $G \setminus S'$ implies a $t$-$u$ path, a contradiction.

**Pushing Lemma**

Let $t \in T$. The Multiway Cut problem has a solution $S$ that contains an important $(t, T \setminus t)$-cut.

**Proof:** Let $R$ be the vertices reachable from $t$ in $G \setminus S$ for a solution $S$.



$\delta(R)$ is not important, then there is an important cut $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$. Replace $S$ with
$S' := (S \setminus \delta(R)) \cup \delta(R') \Rightarrow |S'| \leq |S|$

$S'$ is a multiway cut: (1) There is no $t$-$u$ path in $G \setminus S'$ and (2) a $u$-$v$ path in $G \setminus S'$ implies a $t$-$u$ path, a contradiction.

1. If every vertex of $T$ is in a different component, then we are done.
2. Let $t \in T$ be a vertex that is not separated from every $T \setminus t$.
3. Branch on a choice of an important $(t, T \setminus t)$ cut $S$ of size at most $k$.
4. Set $G := G \setminus S$ and $k := k - |S|$.
5. Go to step 1.

We branch into at most $4^k$ directions at most $k$ times: $4^{k^2} \cdot n^{O(1)}$ running time.

**Next:** Better analysis gives $4^k$ bound on the size of the search tree.

We have seen: at most $4^k$ important cut of size at most $k$.

Better bound:

**Lemma**

If $\mathcal{S}$ is the set of all important $(X, Y)$-cuts, then $\sum_{S \in \mathcal{S}} 4^{-|S|} \leq 1$ holds.

# A refined bound

**Lemma**

If $\mathcal{S}$ is the set of all important $(X, Y)$-cuts, then $\sum_{S \in \mathcal{S}} 4^{-|S|} \leq 1$ holds.

**Proof:** We show the stronger statement $\sum_{S \in \mathcal{S}} 4^{-|S|} \leq 2^{-\lambda}$, where $\lambda$ is the minimum $(X, Y)$-cut size.

**Branch 1: removing $uv$.**

$\lambda$ increases by at most one and we add the edge $uv$ to each separator, increasing the cut by one. Thus the total contribution is

$$\sum_{S \in \mathcal{S}_1} 4^{-(|S|+1)} = \sum_{S \in \mathcal{S}_1} 4^{-|S|}/4 \leq 2^{-(\lambda-1)}/4 = 2^{-\lambda}/2.$$

**Branch 2: replacing $X$ with $X \cup v$.**

$\lambda$ increases by at least one. Thus the total contribution is

$$\sum_{S \in \mathcal{S}_2} 4^{-|S|} \leq 2^{-(\lambda+1)} = 2^{-\lambda}/2.$$

**Lemma**

The search tree for the Multiway Cut algorithm has $4^k$ leaves.

**Proof:** Let $L_k$ be the maximum number of leaves with parameter $k$. We prove $L_k \leq 4^k$ by induction. After enumerating the set $\mathcal{S}_k$ of important separators of size $\leq k$, we branch into $|S_k|$ directions.

$$\sum_{S \in \mathcal{S}_k} 4^{k-|S|} = 4^k \cdot \sum_{S \in \mathcal{S}_k} 4^{-|S|} \leq 4^k$$

**Still need:** bound the work at each node.

We have seen:

**Lemma**

We can enumerate every important $(X, Y)$-cut of size at most $k$ in time $O(4^k \cdot k \cdot (|V(G)| + |E(G)|))$.

**Problem:** running time at a node of the recursion tree is not linear in the number children.

We have seen:

**Lemma**

We can enumerate every important $(X, Y)$-cut of size at most $k$ in time $O(4^k \cdot k \cdot (|V(G)| + |E(G)|))$.

**Problem:** running time at a node of the recursion tree is not linear in the number children.

Easily follows:

**Lemma**

We can enumerate a superset $\mathcal{S}'_k$ of every important $(X, Y)$-cut of size at most $k$ in time $O(|\mathcal{S}'_k| \cdot k^2 \cdot (|V(G)| + |E(G)|))$ such that $\sum_{S \in \mathcal{S}'_k} 4^{-|S|} \leq 1$ holds.

We have seen:

**Lemma**

We can enumerate every important $(X, Y)$-cut of size at most $k$ in time $O(4^k \cdot k \cdot (|V(G)| + |E(G)|))$.

**Problem:** running time at a node of the recursion tree is not linear in the number children.

Needs more work:

**Lemma**

We can enumerate the set $\mathcal{S}_k$ of every important $(X, Y)$-cut of size at most $k$ in time $O(|\mathcal{S}_k| \cdot k^2 \cdot (|V(G)| + |E(G)|))$.

**Theorem**

MULTIWAY CUT can be solved in time $O(4^k \cdot k^3 \cdot (|V(G)| + |E(G)|))$.

1. If every vertex of $T$ is in a different component, then we are done.
2. Let $t \in T$ be a vertex that is not separated from every $T \setminus t$.
3. Branch on a choice of an important $(t, T \setminus t)$ cut $S$ of size at most $k$.
4. Set $G := G \setminus S$ and $k := k - |S|$.
5. Go to step 1.

# Algorithm for MULTIWAY CUT

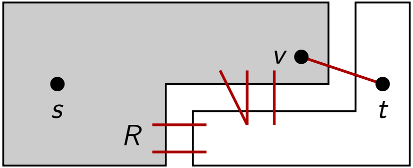
**Lemma:**

At most $k \cdot 4^k$ edges incident to $t$ can be part of an inclusionwise minimal $s - t$ cut of size at most $k$.

**Lemma:**

At most $k \cdot 4^k$ edges incident to $t$ can be part of an inclusionwise minimal $s - t$ cut of size at most $k$.

**Proof:** We show that every such edge is contained in an important $(s, t)$-cut of size at most $k$.

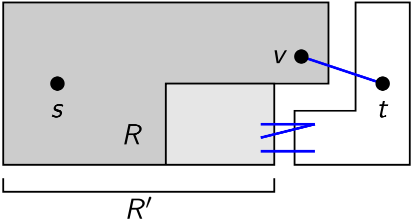


Suppose that $vt \in \delta(R)$ and $|\delta(R)| = k$.

**Lemma:**

At most $k \cdot 4^k$ edges incident to $t$ can be part of an inclusionwise minimal $s - t$ cut of size at most $k$.

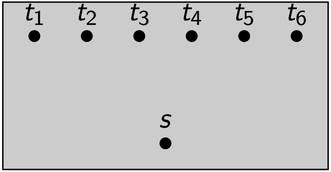
**Proof:** We show that every such edge is contained in an important $(s,t)$-cut of size at most $k$.



Suppose that $vt \in \delta(R)$ and $|\delta(R)| = k$.
There is an important $(s,t)$-cut $\delta(R')$ with $R \subseteq R'$ and $|\delta(R')| \leq k$.
Clearly, $vt \in \delta(R')$: $v \in R$, hence $v \in R'$.

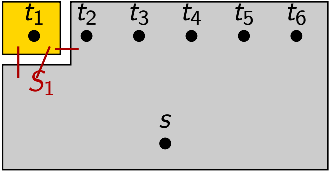
Let $s, t_1, \ldots, t_n$ be vertices and $S_1, \ldots, S_n$ be sets of at most $k$ edges such that $S_i$ separates $t_i$ from $s$, but $S_i$ **does not** separate $t_j$ from $s$ for any $j \neq i$.

It is possible that $n$ is "large" even if $k$ is "small."



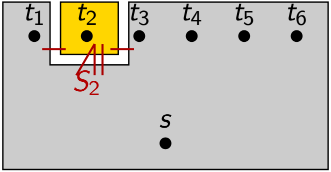
Let $s, t_1, \dots, t_n$ be vertices and $S_1, \dots, S_n$ be sets of at most $k$ edges such that $S_i$ separates $t_i$ from $s$, but $S_i$ **does not** separate $t_j$ from $s$ for any $j \neq i$.

It is possible that $n$ is "large" even if $k$ is "small."



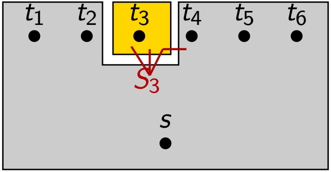
Let $s, t_1, \ldots, t_n$ be vertices and $S_1, \ldots, S_n$ be sets of at most $k$ edges such that $S_i$ separates $t_i$ from $s$, but $S_i$ **does not** separate $t_j$ from $s$ for any $j \neq i$.

It is possible that $n$ is "large" even if $k$ is "small."



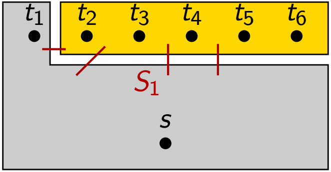
Let $s, t_1, \ldots, t_n$ be vertices and $S_1, \ldots, S_n$ be sets of at most $k$ edges such that $S_i$ separates $t_i$ from $s$, but $S_i$ **does not** separate $t_j$ from $s$ for any $j \neq i$.

It is possible that $n$ is "large" even if $k$ is "small."



Let $s, t_1, \ldots, t_n$ be vertices and $S_1, \ldots, S_n$ be sets of at most $k$ edges such that $S_i$ separates $t_i$ from $s$, but $S_i$ **does not** separate $t_j$ from $s$ for any $j \neq i$.

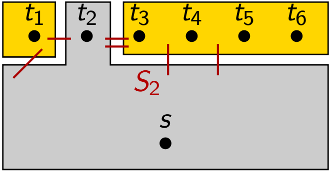
It is possible that $n$ is "large" even if $k$ is "small."



Is the opposite possible, i.e., $S_i$ separates every $t_j$ **except** $t_i$?

Let $s, t_1, \dots, t_n$ be vertices and $S_1, \dots, S_n$ be sets of at most $k$ edges such that $S_i$ separates $t_i$ from $s$, but $S_i$ **does not** separate $t_j$ from $s$ for any $j \neq i$.

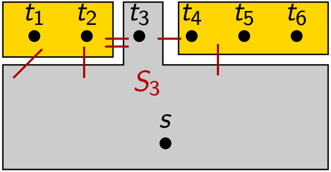
It is possible that $n$ is "large" even if $k$ is "small."



Is the opposite possible, i.e., $S_i$ separates every $t_j$ **except** $t_i$?

Let $s, t_1, \ldots, t_n$ be vertices and $S_1, \ldots, S_n$ be sets of at most $k$ edges such that $S_i$ separates $t_i$ from $s$, but $S_i$ **does not** separate $t_j$ from $s$ for any $j \neq i$.

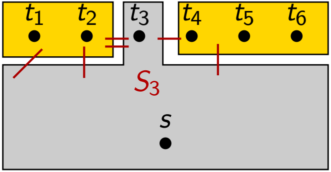
It is possible that $n$ is "large" even if $k$ is "small."



Is the opposite possible, i.e., $S_i$ separates every $t_j$ **except** $t_i$?

Let $s, t_1, \dots, t_n$ be vertices and $S_1, \dots, S_n$ be sets of at most $k$ edges such that $S_i$ separates $t_i$ from $s$, but $S_i$ **does not** separate $t_j$ from $s$ for any $j \neq i$.

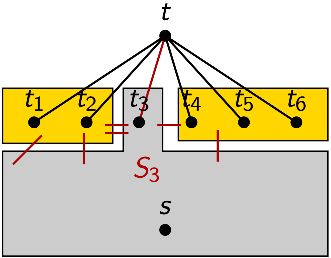
It is possible that $n$ is "large" even if $k$ is "small."



Is the opposite possible, i.e., $S_i$ separates every $t_j$ **except** $t_i$?

## Lemma

If $S_i$ separates $t_j$ from $s$ if and only $j \neq i$ and every $S_i$ has size at most $k$, then $n \leq (k+1) \cdot 4^{k+1}$.

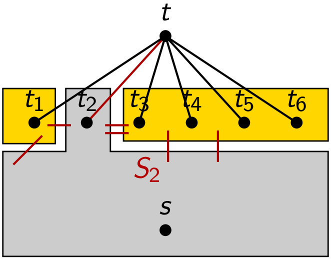


Is the opposite possible, i.e., $S_i$ separates every $t_j$ **except** $t_i$?

### Lemma

If $S_i$ separates $t_j$ from $s$ if and only $j \neq i$ and every $S_i$ has size at most $k$, then $n \leq (k+1) \cdot 4^{k+1}$.

**Proof:** Add a new vertex $t$. Every edge $tt_i$ is part of an (inclusionwise minimal) $(s,t)$-cut of size at most $k+1$. Use the previous lemma.

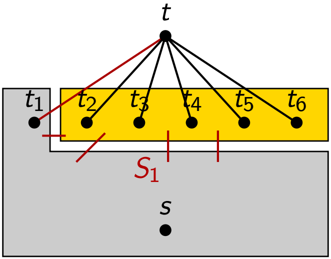


Is the opposite possible, i.e., $S_i$ separates every $t_j$ **except** $t_i$?

**Lemma**

If $S_i$ separates $t_j$ from $s$ if and only $j \neq i$ and every $S_i$ has size at most $k$, then $n \leq (k+1) \cdot 4^{k+1}$.

**Proof:** Add a new vertex $t$. Every edge $tt_i$ is part of an (inclusionwise minimal) $(s,t)$-cut of size at most $k+1$. Use the previous lemma.



Is the opposite possible, i.e., $S_i$ separates every $t_j$ **except** $t_i$?

**Lemma**

If $S_i$ separates $t_j$ from $s$ if and only $j \neq i$ and every $S_i$ has size at most $k$, then $n \leq (k+1) \cdot 4^{k+1}$.

**Proof:** Add a new vertex $t$. Every edge $tt_i$ is part of an (inclusionwise minimal) $(s,t)$-cut of size at most $k+1$. Use the previous lemma.