Important separators and parameterized algorithms

Dániel Marx

Humboldt-Universität zu Berlin, Germany

37th International Workshop on Graph-Theoretic Methods in Computer Science

Teplá Monastery, Czech Republic

June 23, 2011

Overview

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

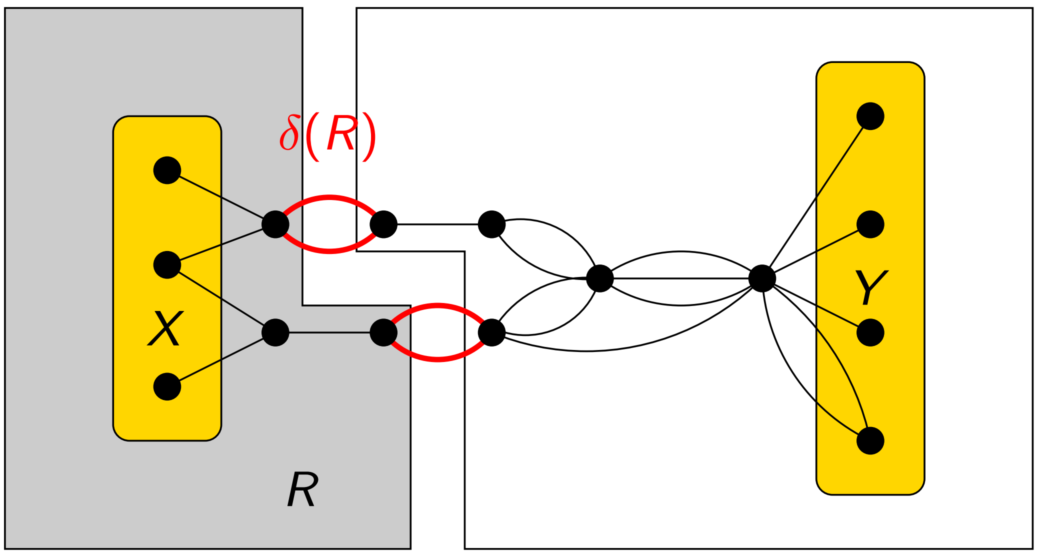
- Bounding the number of “important” separators.
- Some interesting combinatorial consequences.
- Algorithmic applications: FPT algorithm for MULTIWAY CUT and DIRECTED FEEDBACK VERTEX SET.

Important separators

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

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

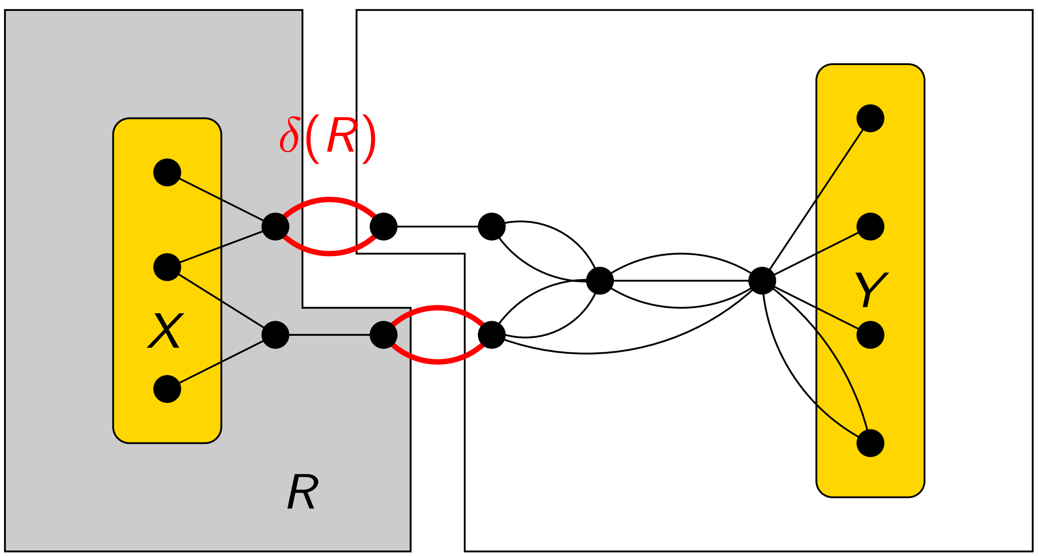
Observation: Every (X, Y)-separator S can be expressed as $S = \delta(R)$ for some $X \subseteq R$ and $R \cap Y = \emptyset$.



Important separators

Definition: An (X, Y)-separator $\delta(R)$ is **important** if there is no (X, Y)-separator $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$.

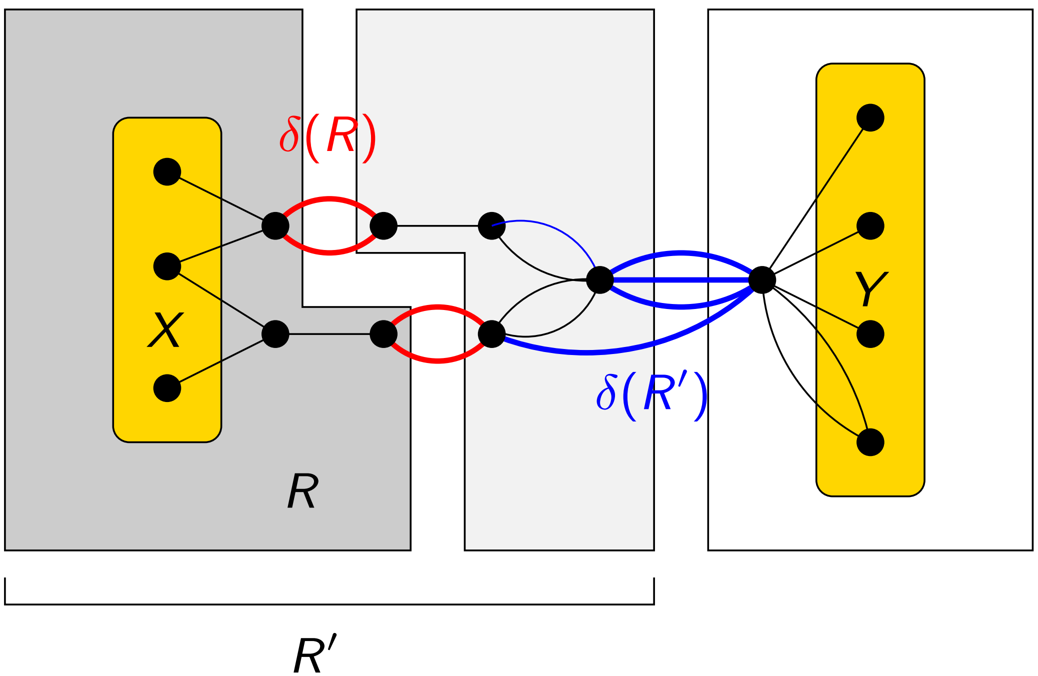
Note: Can be checked in polynomial time if a separator is important.



Important separators

Definition: An (X, Y)-separator $\delta(R)$ is **important** if there is no (X, Y)-separator $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$.

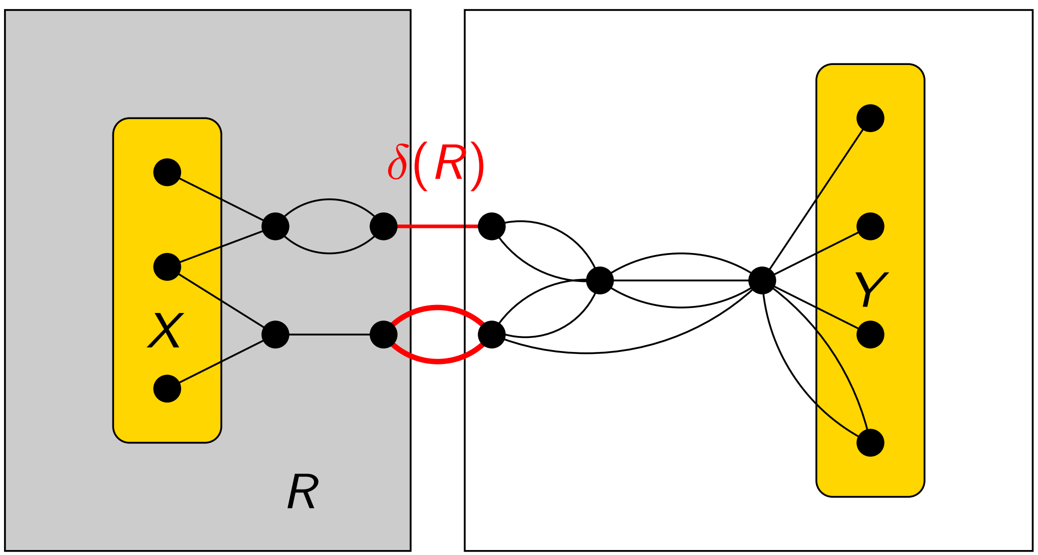
Note: Can be checked in polynomial time if a separator is important.



Important separators

Definition: An (X, Y)-separator $\delta(R)$ is **important** if there is no (X, Y)-separator $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$.

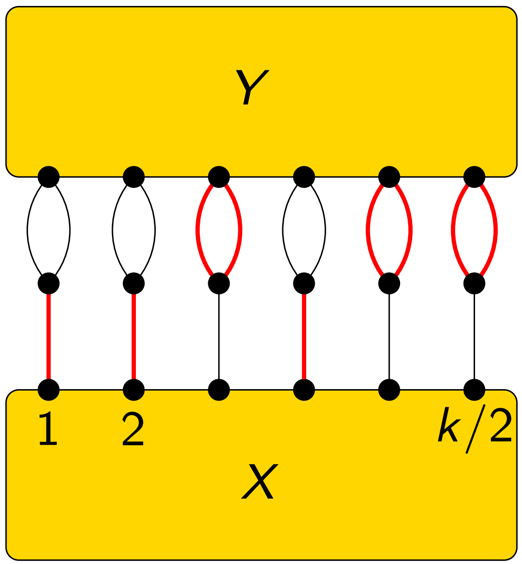
Note: Can be checked in polynomial time if a separator is important.



Important separators

The number of important separators can be exponentially large.

Example:



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

Theorem: There are at most 4^k important (X, Y)-separators of size at most k.
(Proof is implicit in [Chen, Liu, Lu 2007], worse bound in [M. 2004].)

Submodularity

Fact: The function δ is **submodular:** for arbitrary sets A, B,

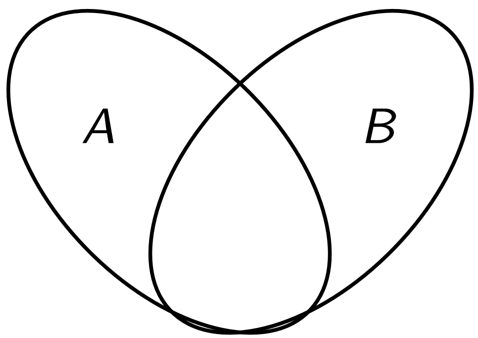
$$|\delta(A)| \quad + \quad |\delta(B)| \quad \geq \quad |\delta(A \cap B)| \quad + \quad |\delta(A \cup B)|$$

Submodularity

Fact: The function δ 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.



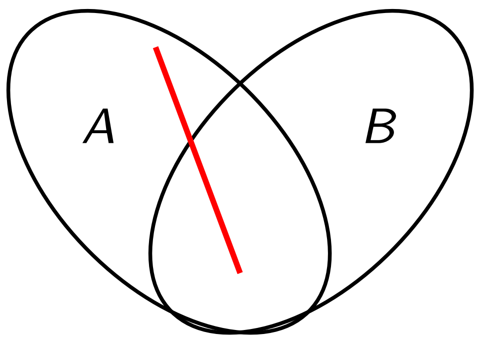
Submodularity

Fact: The function δ is **submodular:** for arbitrary sets A, B,

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

$$0 \qquad\qquad 1 \qquad\qquad 1 \qquad\qquad 0$$

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



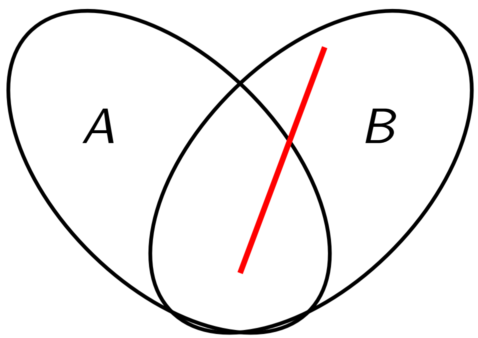
Submodularity

Fact: The function δ is **submodular:** for arbitrary sets A, B,

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

$$1 \qquad\qquad 0 \qquad\qquad\qquad 1 \qquad\qquad\qquad 0$$

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



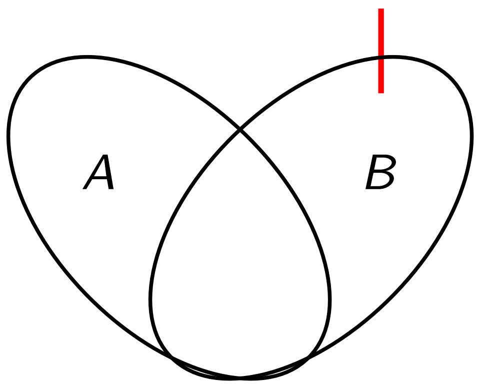
Submodularity

Fact: The function δ is **submodular:** for arbitrary sets A, B,

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

0 1 0 1

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



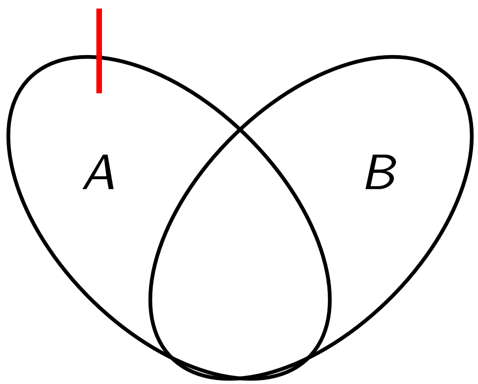
Submodularity

Fact: The function δ is **submodular:** for arbitrary sets A, B,

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

1 0 0 1

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



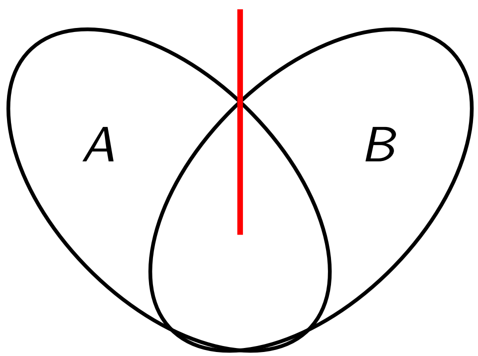
Submodularity

Fact: The function δ is **submodular:** for arbitrary sets A, B,

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

$$1 \qquad\qquad 1 \qquad\qquad\qquad 1 \qquad\qquad\qquad 1$$

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



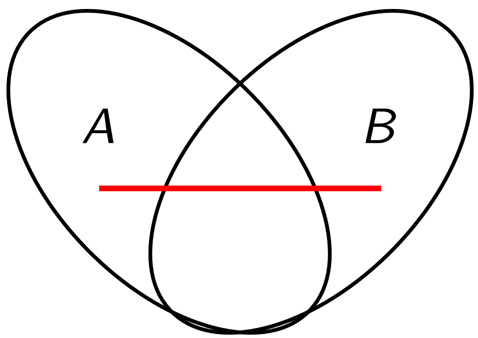
Submodularity

Fact: The function δ is **submodular:** for arbitrary sets A, B,

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

$$1 \qquad\qquad 1 \qquad\qquad\qquad 0 \qquad\qquad\qquad 0$$

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



Submodularity

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

Submodularity

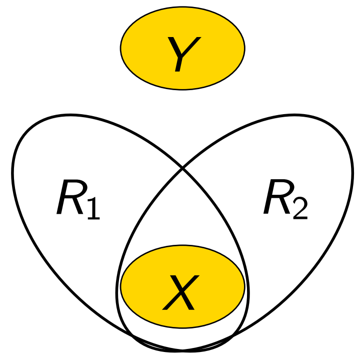
Consequence: Let λ be the minimum (X, Y)-separator size. There is a unique maximal $R_{\max} \supseteq X$ such that $\delta(R_{\max})$ is an (X, Y)-separator of size λ.

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

$$|\delta(R_1)| + |\delta(R_2)| \geq |\delta(R_1 \cap R_2)| + |\delta(R_1 \cup R_2)|$$

$$\lambda \qquad \lambda \qquad \geq \lambda$$

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



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

Important separators

Theorem: There are at most 4^k important (X, Y)-separators of size at most k.

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

First we show that $R_{\text{max}} \subseteq R$ for every important separator $\delta(R)$.

Important separators

Theorem: There are at most 4^k important (X, Y)-separators of size at most k.

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

First we show that $R_{\text{max}} \subseteq R$ for every important separator $\delta(R)$.

By the submodularity of δ:

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

$$\Downarrow$$

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

$$\Downarrow$$

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

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

Important separators

Theorem: There are at most 4^k important (X, Y)-separators of size at most k.

Search tree algorithm for enumerating all these separators:

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

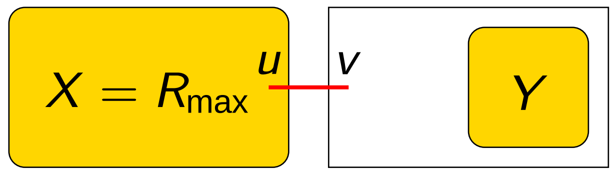
Important separators

Theorem: There are at most 4^k important (X, Y)-separators of size at most k.

Search tree algorithm for enumerating all these separators:

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

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



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

Important separators

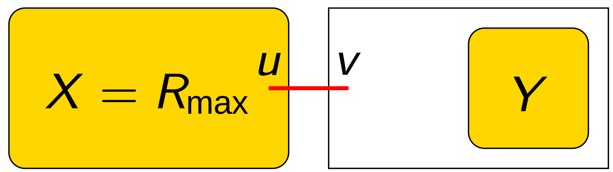
Theorem: There are at most 4^k important (X, Y)-separators of size at most k.

Search tree algorithm for enumerating all these separators:

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

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

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



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

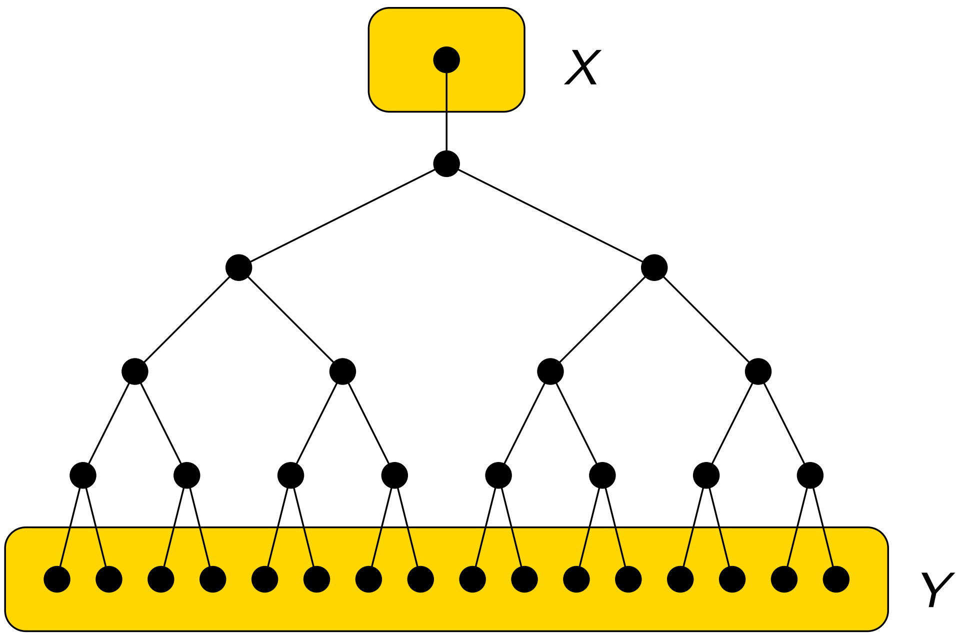
$\Rightarrow$ k remains the same, λ increases by 1.

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

$\Rightarrow$ Height of the search tree $\leq 2k \Rightarrow \leq 2^{2k}$ important separators of size $\leq k$.

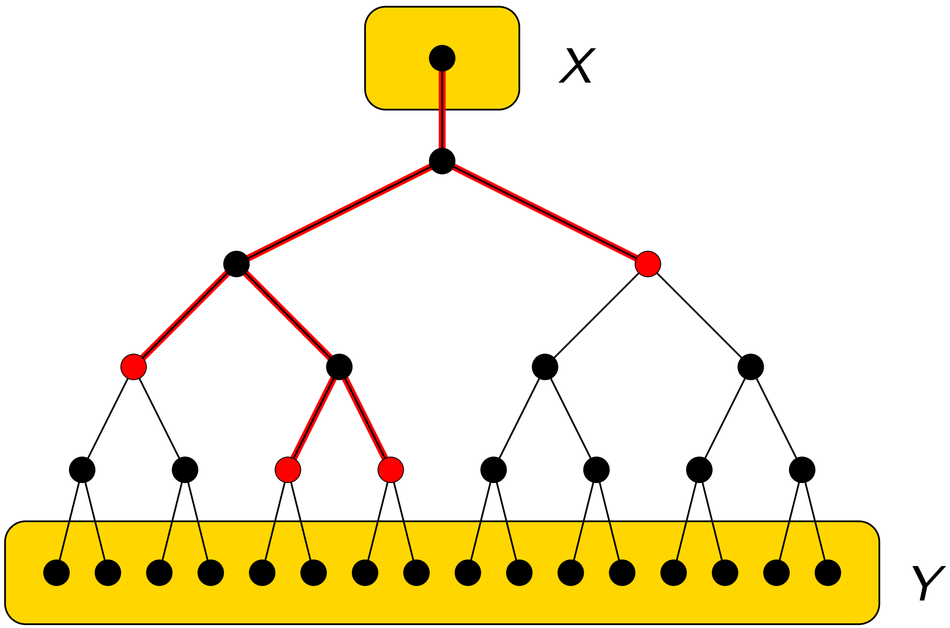
Important separators

Example: The bound 4^k is essentially tight.



Important separators

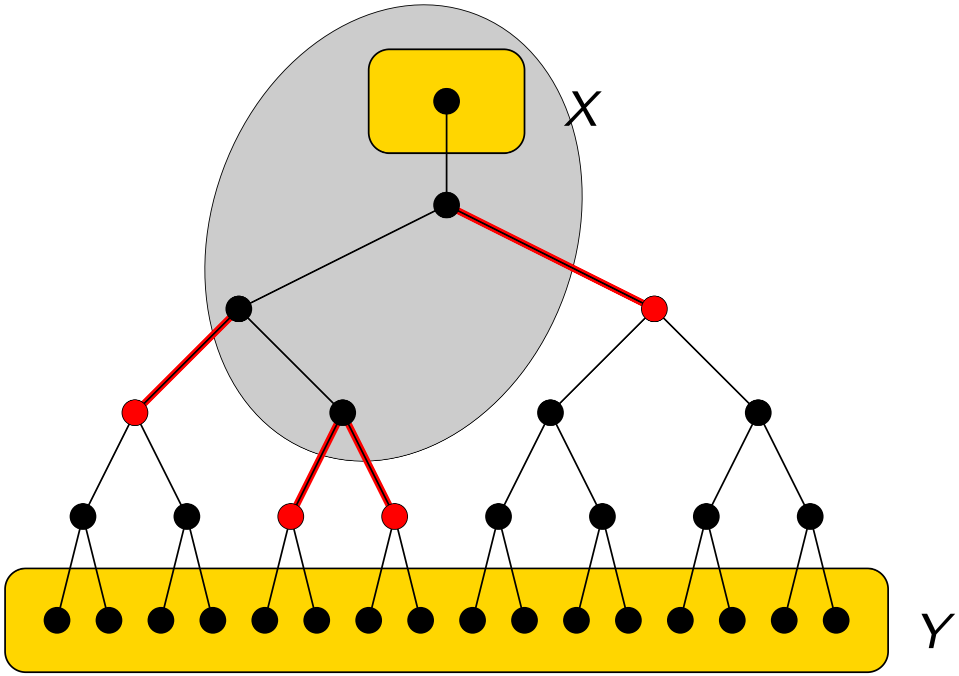
Example: The bound 4^k is essentially tight.



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

Important separators

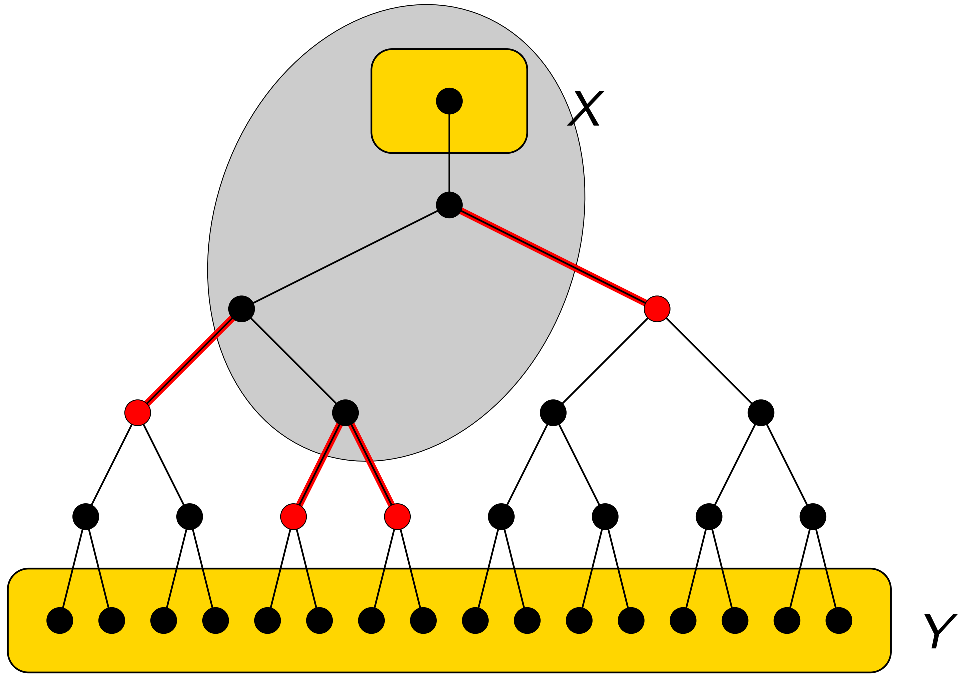
Example: The bound 4^k is essentially tight.



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

Important separators

Example: The bound 4^k is essentially tight.



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

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

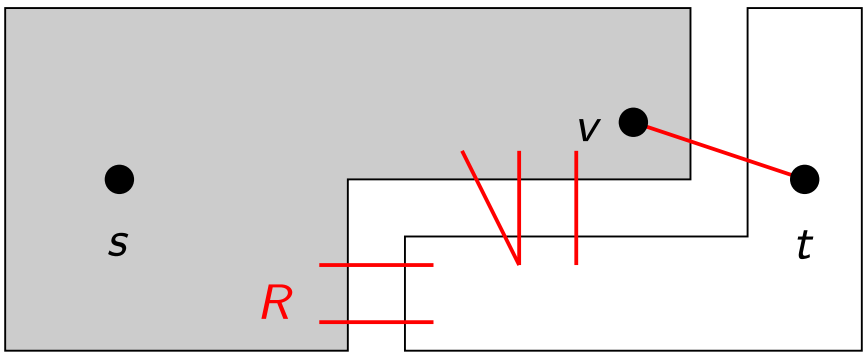
Simple application

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.

Simple application

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)-separator of size at most k.

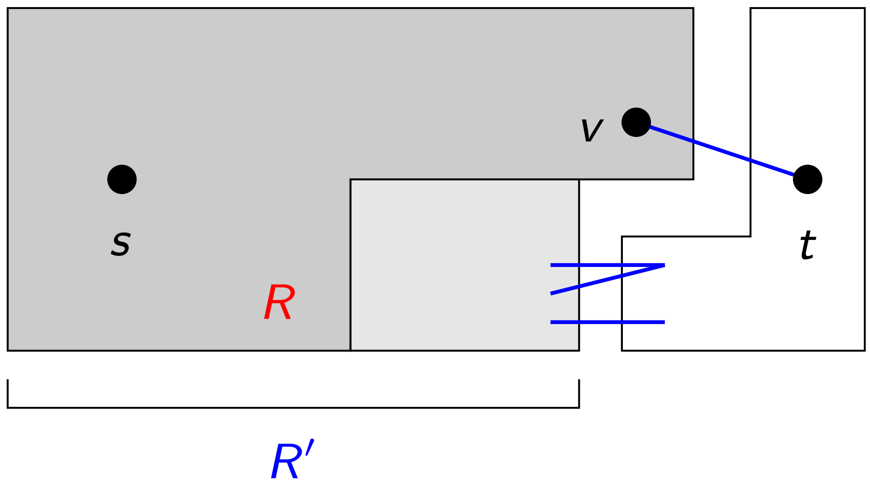


Suppose that $vt \in \delta(R)$ and $|\delta(R)| = k$.

Simple application

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)-separator of size at most k.

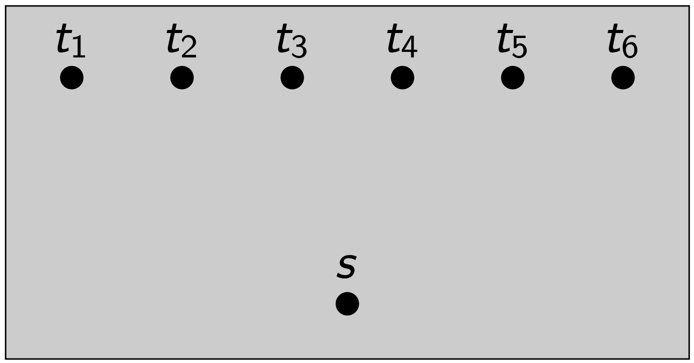


Suppose that $vt \in \delta(R)$ and $|\delta(R)| = k$.
There is an important (s, t)-separator $\delta(R')$ with $R \subseteq R'$ and $|\delta(R')| \leq k$.
Clearly, $vt \in \delta(R')$: $v \in R$, hence $v \in R'$.

Anti isolation

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 $j \neq i$.

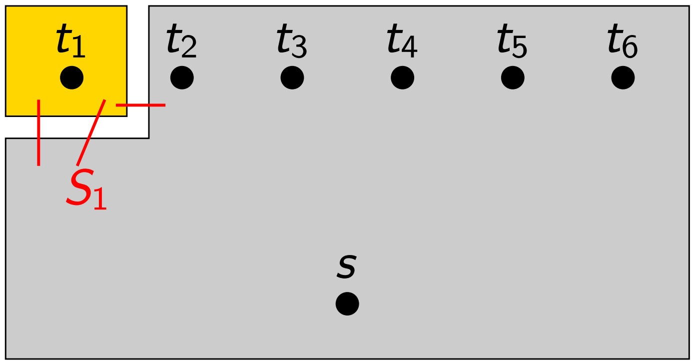
It is possible that n is "large" even if k is "small."



Anti isolation

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 $j \neq i$.

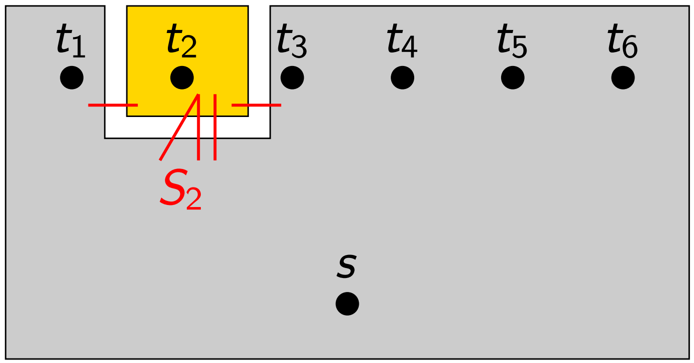
It is possible that n is "large" even if k is "small."



Anti isolation

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 $j \neq i$.

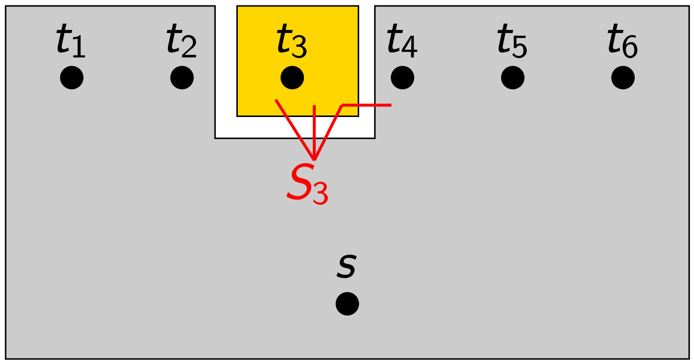
It is possible that n is "large" even if k is "small."



Anti isolation

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 $j \neq i$.

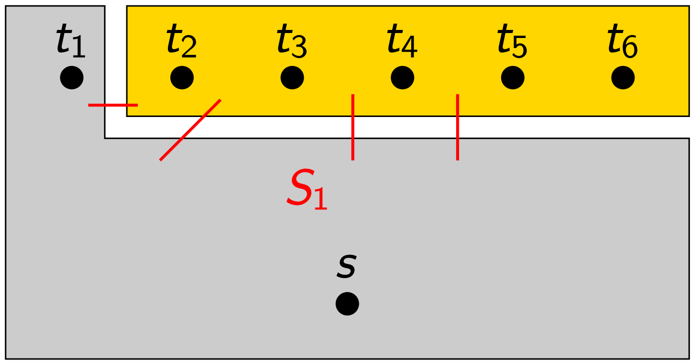
It is possible that n is "large" even if k is "small."



Anti isolation

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."

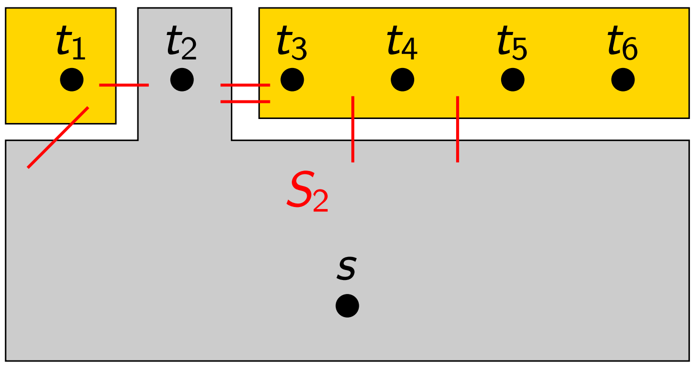


Is the opposite possible, i.e., S_i separates every t_j **except** t_i?

Anti isolation

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."

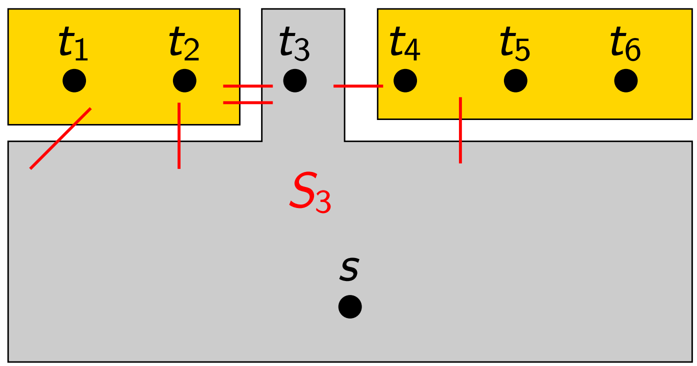


Is the opposite possible, i.e., S_i separates every t_j **except** t_i?

Anti isolation

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."

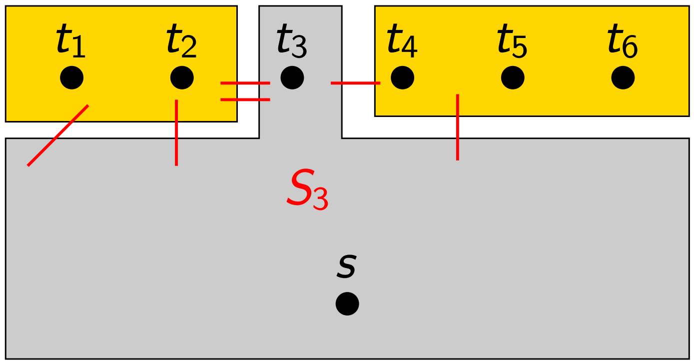


Is the opposite possible, i.e., S_i separates every t_j **except** t_i?

Anti isolation

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 $j \neq i$.

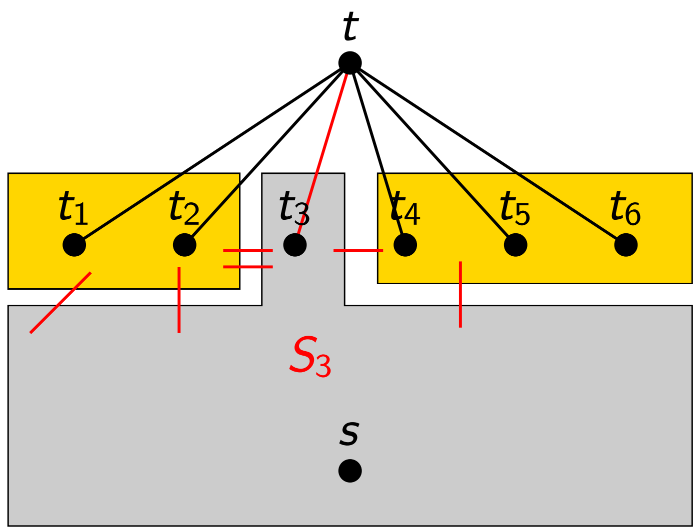
It is possible that n is "large" even if k is "small."



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

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 $n \leq (k+1) \cdot 4^{k+1}$.

Anti isolation

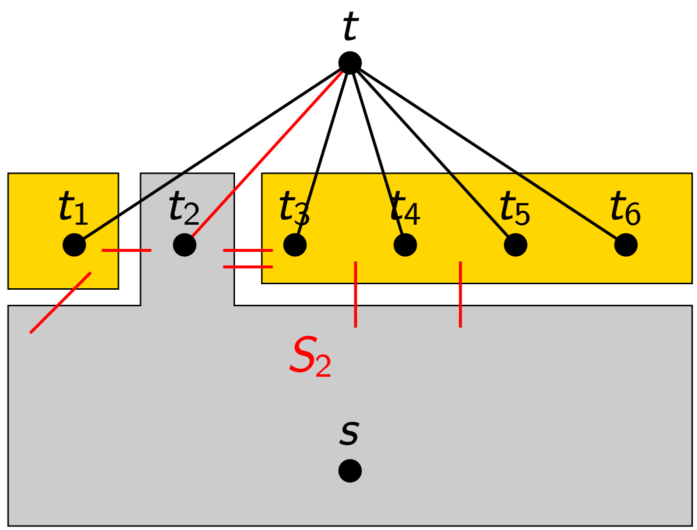


Is the opposite possible, i.e., S_i separates every t_j **except** t_i?

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)-separator of size at most $k + 1$. Use the previous lemma.

Anti isolation

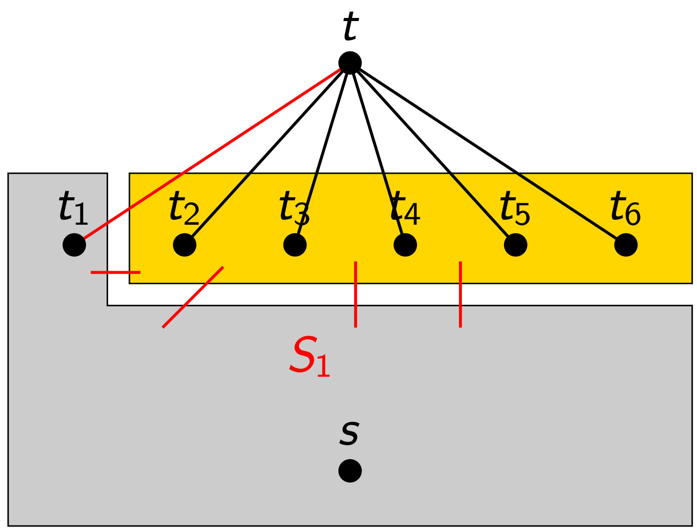


Is the opposite possible, i.e., S_i separates every t_j **except** t_i?

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)-separator of size at most $k+1$. Use the previous lemma.

Anti isolation



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

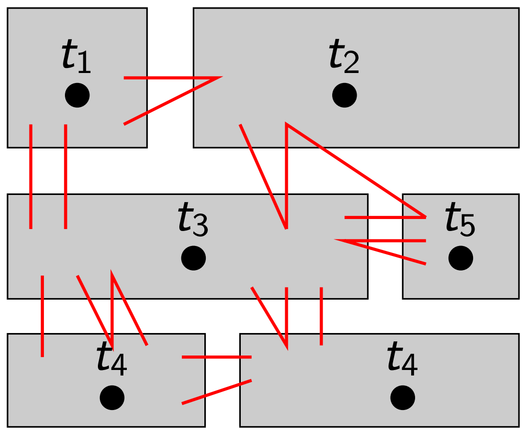
Lemma: If S_i separates t_j from s if and only $j \neq i$ and every S_i has size at most k, then $n \leq (k+1) \cdot 4^{k+1}$.

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

MULTIWAY CUT

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].

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

MULTIWAY CUT

Central notion of parameterized complexity:

> **Definition:** A problem is **fixed-parameter tractable (FPT)** parameterized by k if it can be solved in time $f(k) \cdot n^{O(1)}$ for some function $f(k)$ depending only on k.

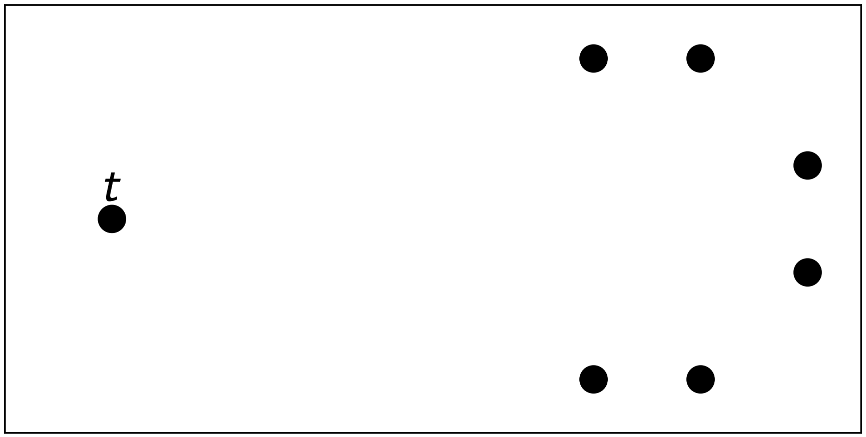
FPT means that the k can be removed from the exponent of n and the combinatorial explosion can be restricted to k.

If $f(k)$ is e.g., 1.2^k, then this can be actually an efficient algorithm!

Theorem: MULTIWAY CUT can be solved in time $4^k \cdot n^{O(1)}$, i.e., it is fixed-parameter tractable (FPT) parameterized by the size k of the solution.

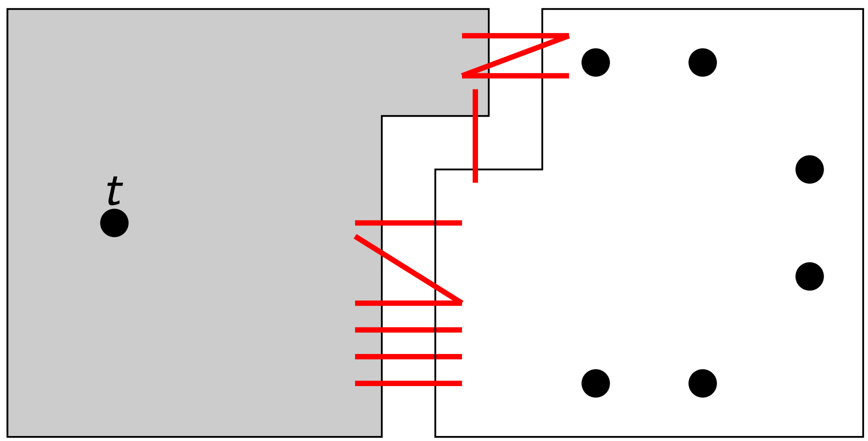
MULTIWAY CUT

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



MULTIWAY CUT

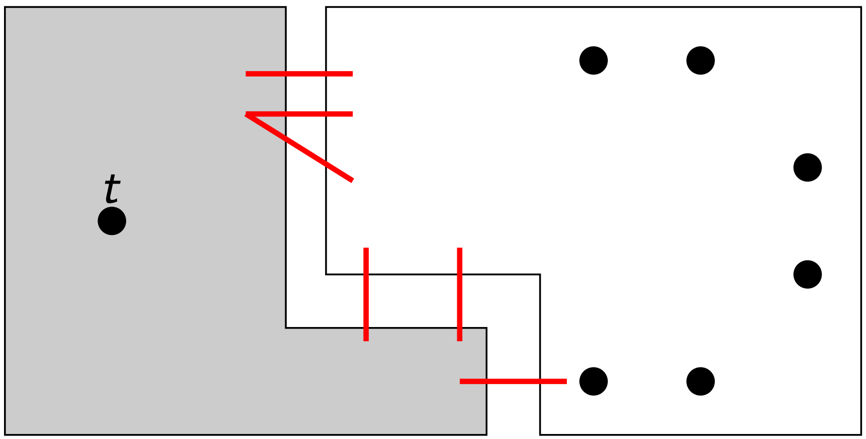
Intuition: Consider a $t \in T$. A subset of the solution S is a $(t, T \setminus t)$-separator.



There are many such separators.

MULTIWAY CUT

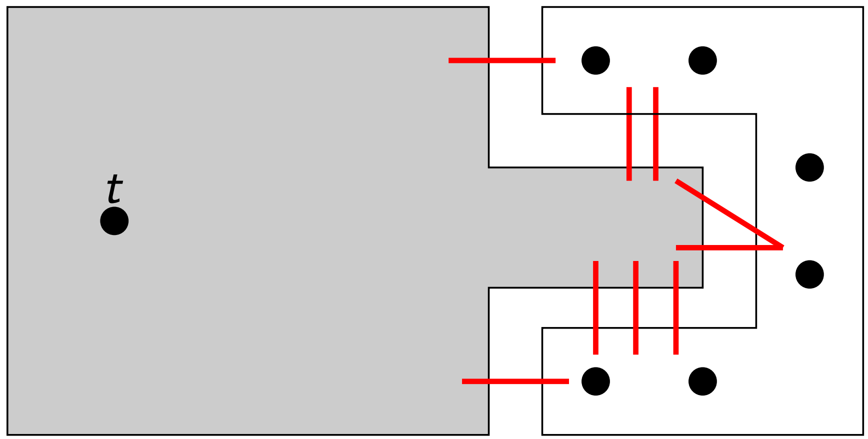
Intuition: Consider a $t \in T$. A subset of the solution S is a $(t, T \setminus t)$-separator.



There are many such separators.

Multiway Cut

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



There are many such separators.

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

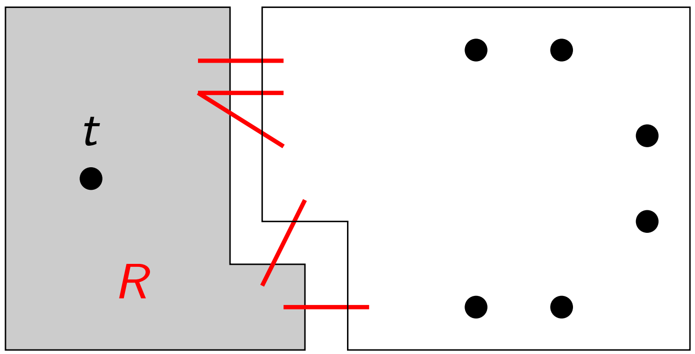
MULTIWAY CUT *and important separators*

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

MULTIWAY CUT *and important separators*

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

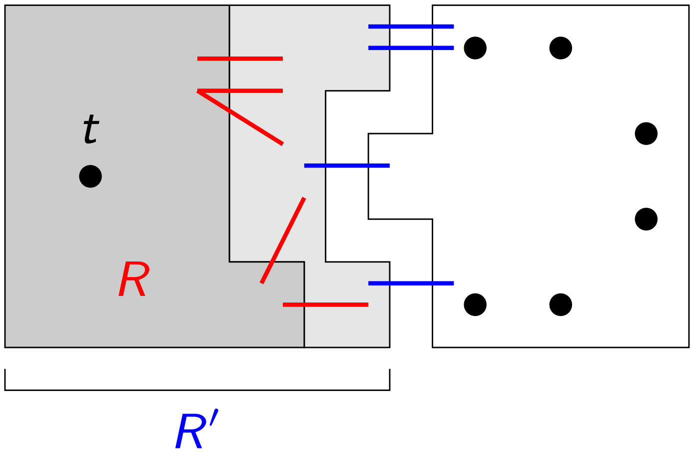
Proof: Let R be the vertices reachable from t in $G \setminus S$ for a solution S.



MULTIWAY CUT *and important separators*

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

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

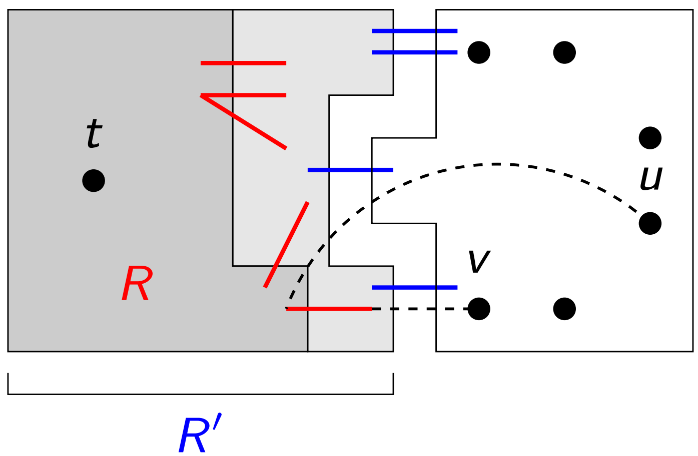


If $\delta(R)$ is not important, then there is an important separator $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$. Replace S with $S' := (S \setminus \delta(R)) \cup \delta(R') \Rightarrow |S'| \leq |S|$

MULTIWAY CUT *and important separators*

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

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



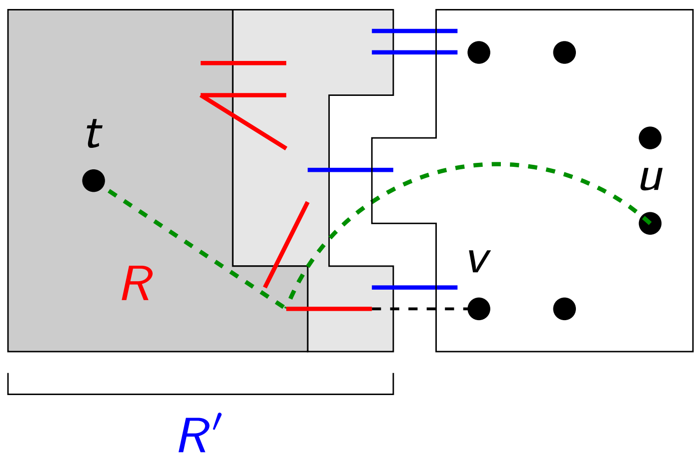
If $\delta(R)$ is not important, then there is an important separator $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$. Replace S with $S' := (S \setminus \delta(R)) \cup \delta(R') \Rightarrow |S'| \leq |S|$

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.

MULTIWAY CUT *and important separators*

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

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



If $\delta(R)$ is not important, then there is an important separator $\delta(R')$ 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.

Algorithm for MULTIWAY CUT

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 $(t, T \setminus t)$ separator 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.

(Better analysis gives 4^k bound on the size of the search tree.)

MULTICUT

MULTICUT	
Input:	Graph G, pairs $(s_1, t_1), \ldots, (s_\ell, t_\ell)$, integer k
Find:	A set S of edges such that $G \setminus S$ has no s_i-t_i path for any i.

Theorem: MULTICUT can be solved in time $f(k,\ell) \cdot n^{O(1)}$ (FPT parameterized by combined parameters k and ℓ).

MULTICUT

MULTICUT

Input: Graph G, pairs $(s_1, t_1), \ldots, (s_\ell, t_\ell)$, integer k

Find: A set S of edges such that $G \setminus S$ has no s_i-t_i path for any i.

Theorem: MULTICUT can be solved in time $f(k, \ell) \cdot n^{O(1)}$ (FPT parameterized by combined parameters k and ℓ).

Proof: The solution partitions $\{s_1, t_1, \ldots, s_\ell, t_\ell\}$ into components. Guess this partition, contract the vertices in a class, and solve MULTIWAY CUT.

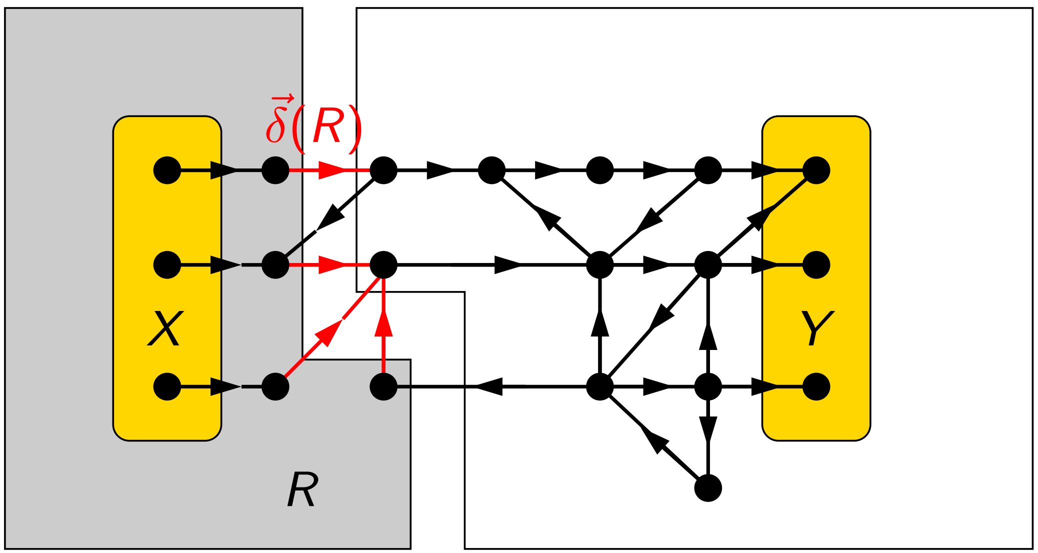
Theorem: [Bousquet, Daligault, Thomassé 2011] [M., Razgon 2011] MULTICUT is FPT parameterized by the size k of the solution.

Directed graphs

Definition: $\vec{\delta}(R)$ is the set of edges leaving R.

Observation: Every inclusionwise-minimal directed (X, Y)-separator S can be expressed as $S = \vec{\delta}(R)$ for some $X \subseteq R$ and $R \cap Y = \emptyset$.

Definition: An (X, Y)-separator $\vec{\delta}(R)$ is **important** if there is no (X, Y)-separator $\vec{\delta}(R')$ with $R \subset R'$ and $|\vec{\delta}(R')| \leq |\vec{\delta}(R)|$.

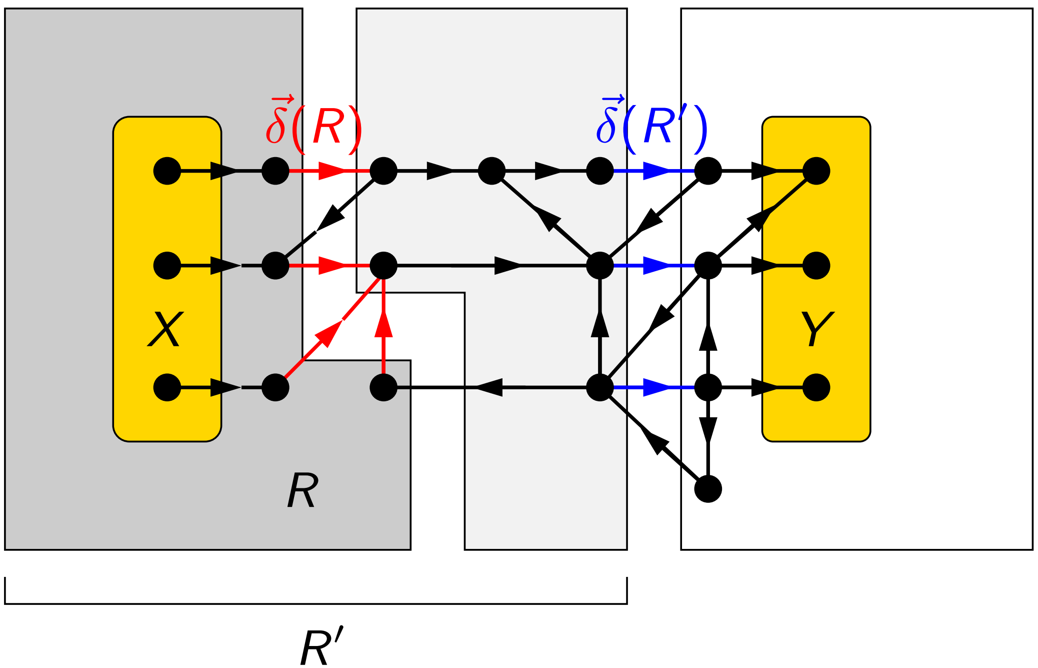


Directed graphs

Definition: $\vec{\delta}(R)$ is the set of edges leaving R.

Observation: Every inclusionwise-minimal directed (X, Y)-separator S can be expressed as $S = \vec{\delta}(R)$ for some $X \subseteq R$ and $R \cap Y = \emptyset$.

Definition: An (X, Y)-separator $\vec{\delta}(R)$ is **important** if there is no (X, Y)-separator $\vec{\delta}(R')$ with $R \subset R'$ and $|\vec{\delta}(R')| \leq |\vec{\delta}(R)|$.



Directed graphs

Definition: $\vec{\delta}(R)$ is the set of edges leaving R.

Observation: Every inclusionwise-minimal directed (X, Y)-separator S can be expressed as $S = \vec{\delta}(R)$ for some $X \subseteq R$ and $R \cap Y = \emptyset$.

Definition: An (X, Y)-separator $\vec{\delta}(R)$ is **important** if there is no (X, Y)-separator $\vec{\delta}(R')$ with $R \subset R'$ and $|\vec{\delta}(R')| \leq |\vec{\delta}(R)|$.

The proof for the undirected case goes through for the directed case:

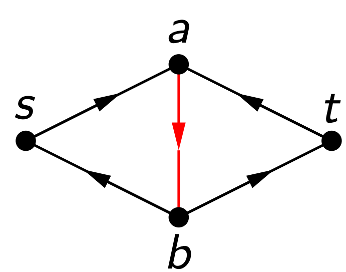
Theorem: There are at most 4^k important directed (X, Y)-separators of size at most k.

DIRECTED MULTIWAY CUT

The undirected approach does not work: the pushing lemma is not true.

Pushing Lemma: [for undirected graphs] Let $t \in T$. The MULTIWAY CUT problem has a solution S that contains an important $(t, T \setminus t)$-separator.

Directed counterexample:



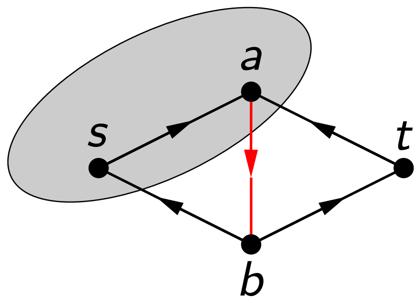
Unique solution with $k = 1$ edges, but it is not an important separator (boundary of $\{s, a\}$, but the boundary of $\{s, a, b\}$ is of the same size).

DIRECTED MULTIWAY CUT

The undirected approach does not work: the pushing lemma is not true.

Pushing Lemma: [for undirected graphs] Let $t \in T$. The MULTIWAY CUT problem has a solution S that contains an important $(t, T \setminus t)$-separator.

Directed counterexample:



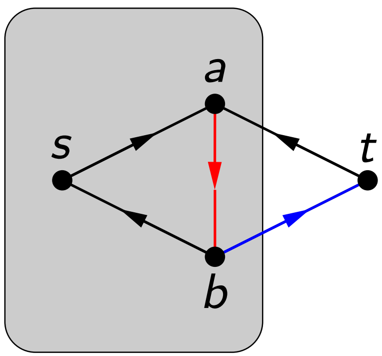
Unique solution with $k = 1$ edges, but it is not an important separator (boundary of $\{s, a\}$, but the boundary of $\{s, a, b\}$ is of the same size).

DIRECTED MULTIWAY CUT

The undirected approach does not work: the pushing lemma is not true.

Pushing Lemma: [for undirected graphs] Let $t \in T$. The MULTIWAY CUT problem has a solution S that contains an important $(t, T \setminus t)$-separator.

Directed counterexample:



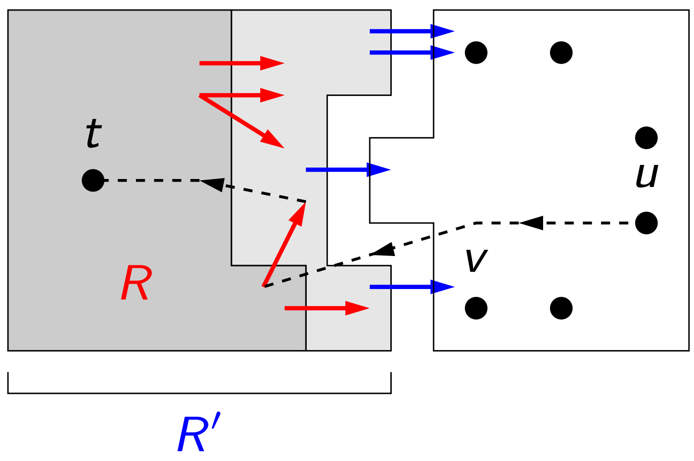
Unique solution with $k = 1$ edges, but it is not an important separator (boundary of $\{s, a\}$, but the boundary of $\{s, a, b\}$ is of the same size).

DIRECTED MULTIWAY CUT

The undirected approach does not work: the pushing lemma is not true.

Pushing Lemma: [for undirected graphs] Let $t \in T$. The MULTIWAY CUT problem has a solution S that contains an important $(t, T \setminus t)$-separator.

Problem in the undirected proof:



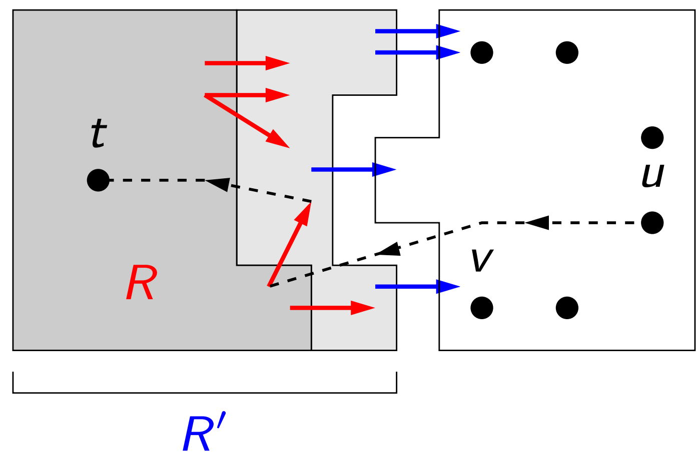
Replacing R by R' cannot create a $t \to u$ path, but can create a $u \to t$ path.

DIRECTED MULTIWAY CUT

The undirected approach does not work: the pushing lemma is not true.

Pushing Lemma: [for undirected graphs] Let $t \in T$. The MULTIWAY CUT problem has a solution S that contains an important $(t, T \setminus t)$-separator.

Problem in the undirected proof:



Replacing R by R' cannot create a $t \to u$ path, but can create a $u \to t$ path.

Theorem: [Chitnis, Hajiaghayi, M. 2011] DIRECTED MULTIWAY CUT is FPT parameterized by the size k of the solution.

DIRECTED MULTICUT

DIRECTED MULTICUT

Input: Graph G, pairs $(s_1, t_1), \ldots, (s_\ell, t_\ell)$, integer k

Find: A set S of edges such that $G \setminus S$ has no $s_i \to t_i$ path for any i.

Theorem: [M. and Razgon 2011] DIRECTED MULTICUT is W[1]-hard parameterized by k.

DIRECTED MULTICUT

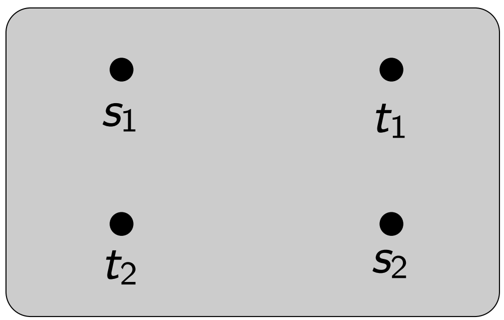
DIRECTED MULTICUT

Input: Graph G, pairs $(s_1, t_1), \ldots, (s_\ell, t_\ell)$, integer k

Find: A set S of edges such that $G \setminus S$ has no $s_i \to t_i$ path for any i.

Theorem: [M. and Razgon 2011] DIRECTED MULTICUT is W[1]-hard parameterized by k.

But the case $\ell = 2$ can be reduced to DIRECTED MULTIWAY CUT:



Directed Multicut

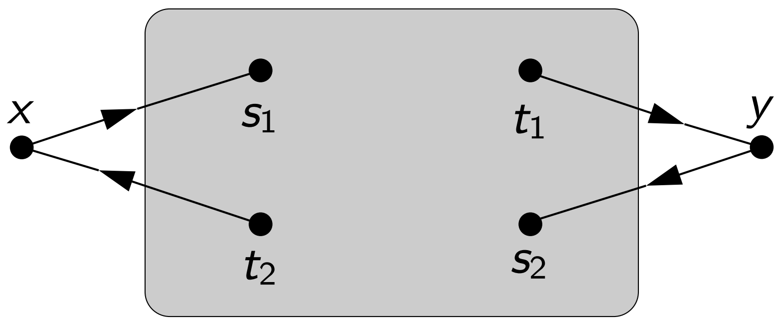
DIRECTED MULTICUT

Input: Graph G, pairs $(s_1, t_1), \dots, (s_\ell, t_\ell)$, integer k

Find: A set S of edges such that $G \setminus S$ has no $s_i \to t_i$ path for any i.

Theorem: [M. and Razgon 2011] DIRECTED MULTICUT is W[1]-hard parameterized by k.

But the case $\ell = 2$ can be reduced to DIRECTED MULTIWAY CUT:



DIRECTED MULTICUT

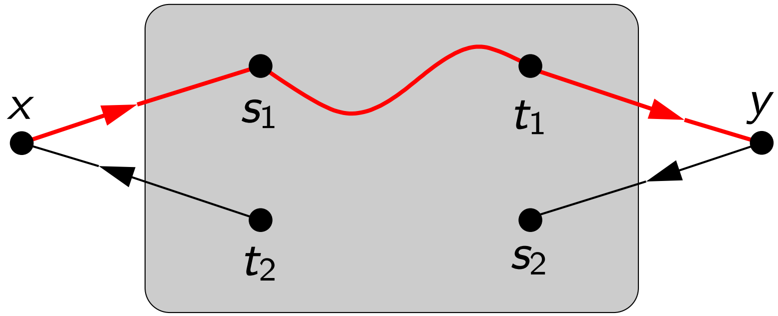
DIRECTED MULTICUT

Input: Graph G, pairs $(s_1, t_1), \dots, (s_\ell, t_\ell)$, integer k

Find: A set S of edges such that $G \setminus S$ has no $s_i \to t_i$ path for any i.

Theorem: [M. and Razgon 2011] DIRECTED MULTICUT is W[1]-hard parameterized by k.

But the case $\ell = 2$ can be reduced to DIRECTED MULTIWAY CUT:



DIRECTED MULTICUT

> DIRECTED MULTICUT
>
> **Input:** Graph G, pairs $(s_1, t_1), \ldots, (s_\ell, t_\ell)$, integer k
>
> **Find:** A set S of edges such that $G \setminus S$ has no $s_i \to t_i$ path for any i.

Theorem: [M. and Razgon 2011] DIRECTED MULTICUT is W[1]-hard parameterized by k.

Corollary: DIRECTED MULTICUT with $\ell = 2$ is FPT parameterized by the size k of the solution.

?

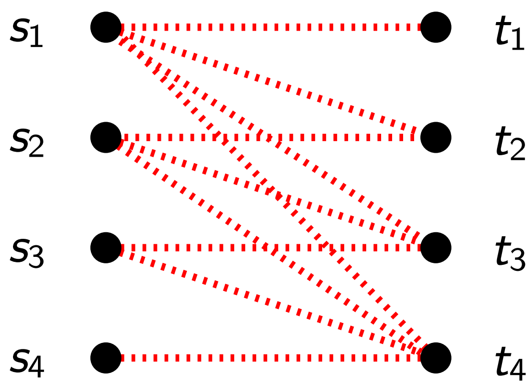
Open: Is DIRECTED MULTICUT with $\ell = 3$ FPT?

Open: Is there an $f(k, \ell) \cdot n^{O(1)}$ algorithm for DIRECTED MULTICUT?

SKEW MULTICUT

SKEW MULTICUT

Input:	Graph G, pairs $(s_1, t_1), \ldots, (s_\ell, t_\ell)$, integer k
Find:	A set S of k directed edges such that $G \setminus S$ contains no $s_i \to t_j$ path for any $i \leq j$.

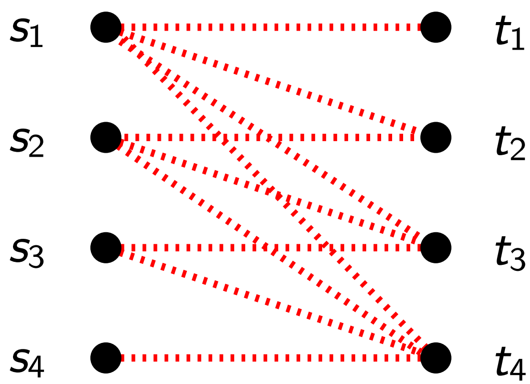


SKEW MULTICUT

SKEW MULTICUT

Input: Graph G, pairs $(s_1, t_1), \ldots, (s_\ell, t_\ell)$, integer k

Find: A set S of k directed edges such that $G \setminus S$ contains no $s_i \to t_j$ path for any $i \leq j$.



Pushing Lemma: SKEW MULTCUT problem has a solution S that contains an important $(s_1, \{t_1, \ldots, t_\ell\})$-separator.

Theorem: [Chen, Liu, Lu, O'Sullivan, Razgon 2008] SKEW MULTICUT can be solved in time $4^k \cdot n^{O(1)}$.

DIRECTED FEEDBACK VERTEX SET

DIRECTED FEEDBACK VERTEX/EDGE SET	
Input:	Directed graph G, integer k
Find:	A set S of k vertices/edges such that $G \setminus S$ is acyclic.

Note: Edge and vertex versions are equivalent, we will consider the edge version here.

Theorem: [Chen, Liu, Lu, O'Sullivan, Razgon 2008] DIRECTED FEEDBACK EDGE SET is FPT parameterized by the size k of the solution.

Solution uses the technique of **iterative compression** introduced by [Reed, Smith, Vetta 2004].

The compression problem

DIRECTED FEEDBACK EDGE SET COMPRESSION	
Input:	Directed graph G, integer k, a set S' of $k+1$ edges such that $G \setminus S'$ is acyclic
Find:	A set S of k edges such that $G \setminus S$ is acyclic.

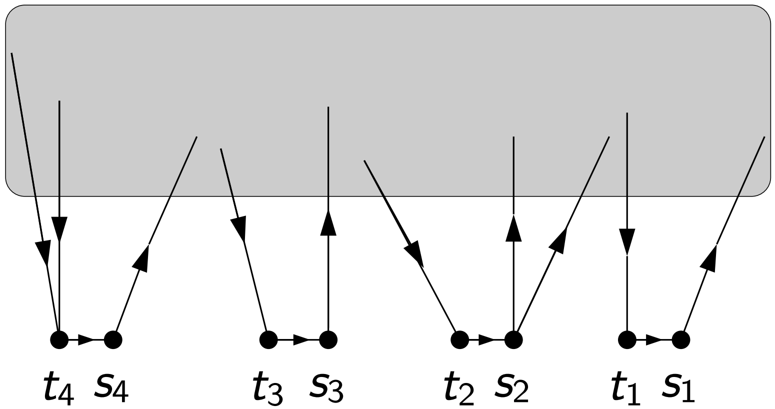
Easier than the original problem, as the extra input S' gives us useful structural information about G.

Lemma: The compression problem is FPT parameterized by k.

The compression problem

Lemma: The compression problem is FPT parameterized by k.

Proof: Let $S' = \{\overrightarrow{t_1 s_1}, \dots, \overrightarrow{t_{k+1} s_{k+1}}\}$.

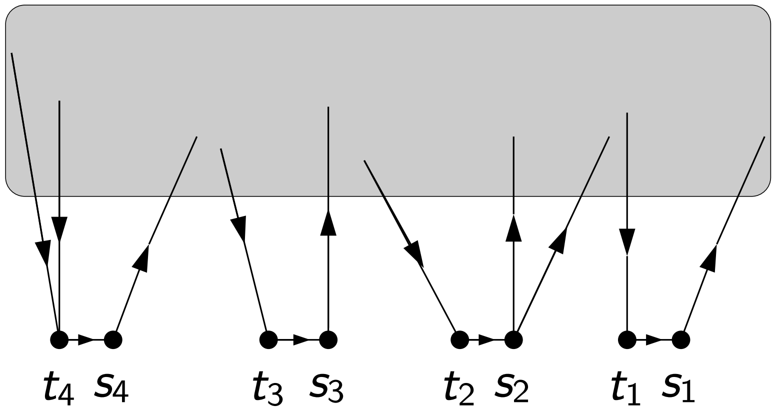


- By guessing and removing $S \cap S'$, we can assume that S and S' are disjoint [2^{k+1} possibilities].
- By guessing the order of $\{s_1, \dots, s_{k+1}\}$ in the acyclic ordering of $G \setminus S$, we can assume that $s_{k+1} < s_k < \cdots < s_1$ in $G \setminus S$ [$(k+1)!$ possibilities].

The compression problem

Lemma: The compression problem is FPT parameterized by k.

Proof: Let $S' = \{\overrightarrow{t_1 s_1}, \ldots, \overrightarrow{t_{k+1} s_{k+1}}\}$.



Claim: Suppose that $S' \cap S = \emptyset$.

$G \setminus S$ is acyclic and has an ordering with $s_{k+1} < s_k < \cdots < s_1$

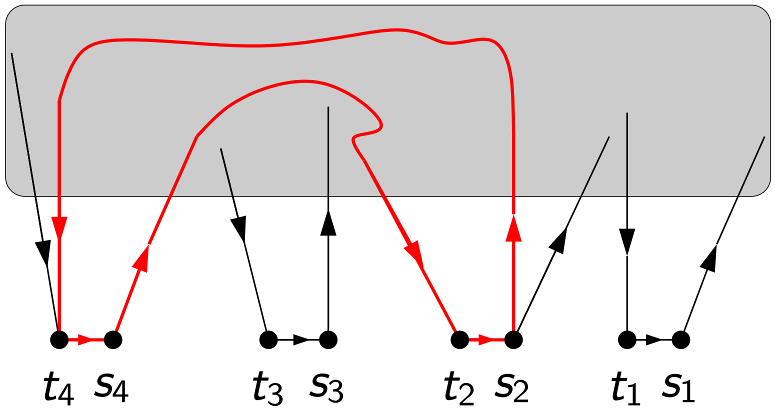
$$\Updownarrow$$

S covers every $s_i \to t_j$ path for every $i \leq j$

The compression problem

Lemma: The compression problem is FPT parameterized by k.

Proof: Let $S' = \{\overrightarrow{t_1 s_1}, \dots, \overrightarrow{t_{k+1} s_{k+1}}\}$.



Claim: Suppose that $S' \cap S = \emptyset$.

$G \setminus S$ is acyclic and has an ordering with $s_{k+1} < s_k < \cdots < s_1$

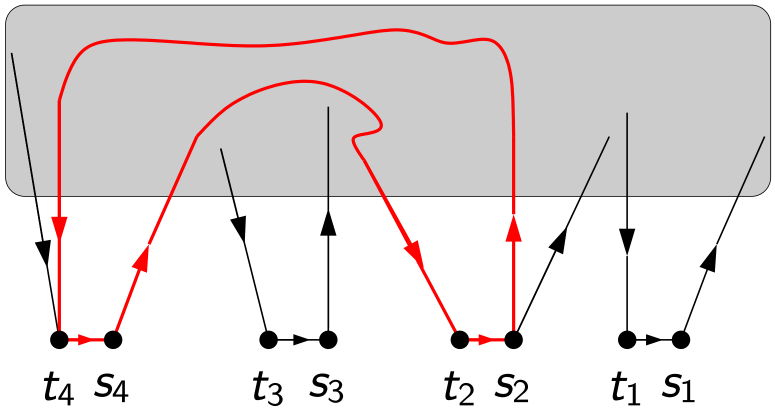
$\Updownarrow$

S covers every $s_i \to t_j$ path for every $i \leq j$

The compression problem

Lemma: The compression problem is FPT parameterized by k.

Proof: Let $S' = \{\overrightarrow{t_1 s_1}, \dots, \overrightarrow{t_{k+1} s_{k+1}}\}$.



Claim: Suppose that $S' \cap S = \emptyset$.

$G \setminus S$ is acyclic and has an ordering with $s_{k+1} < s_k < \cdots < s_1$

$\Updownarrow$

S covers every $s_i \to t_j$ path for every $i \leq j$

$\Rightarrow$ We can solve the compression problem by $2^{k+1} \cdot (k+1)!$ applications of SKEW MULTICUT.

Iterative compression

We have given a $f(k)n^{O(1)}$ algorithm for the following problem:

Directed Feedback Edge Set Compression	
Input:	Directed graph G, integer k, a set S' of $k+1$ edges such that $G \setminus S'$ is acyclic
Find:	A set S of k edges such that $G \setminus S$ is acyclic.

Nice, but how do we get a solution S' of size $k+1$?

Iterative compression

We have given a $f(k)n^{O(1)}$ algorithm for the following problem:

DIRECTED FEEDBACK EDGE SET COMPRESSION	
Input:	Directed graph G, integer k, a set S' of $k+1$ edges such that $G \setminus S'$ is acyclic
Find:	A set S of k edges such that $G \setminus S$ is acyclic.

Nice, but how do we get a solution S' of size $k+1$?

We get it for free!

Useful trick: **iterative compression** (introduced by [Reed, Smith, Vetta 2004] for BIPARTITE DELETION).

Iterative compression

Let $e_1, \ldots, e_m$ be the edges of G and let G_i be the subgraph containing only the first i edges (and all vertices).

For every $i = 1, \ldots, m$, we find a set S_i of k edges such that $G_i \setminus S_i$ is acyclic.

Iterative compression

Let $e_1, \dots, e_m$ be the edges of G and let G_i be the subgraph containing only the first i edges (and all vertices).

For every $i = 1, \dots, m$, we find a set S_i of k edges such that $G_i \setminus S_i$ is acyclic.

- For $i = k$, we have the trivial solution $S_i = \{e_1, \dots, e_k\}$.
- Suppose we have a solution S_i for G_i. Then $S_i \cup \{e_{i+1}\}$ is a solution of size $k + 1$ in the graph G_{i+1}
- Use the compression algorithm for G_{i+1} with the solution $S_i \cup \{e_{i+1}\}$.
  - If there is no solution of size k for G_{i+1}, then we can stop.
  - Otherwise the compression algorithm gives a solution S_{i+1} of size k for G_{i+1}.

We call the compression algorithm m times, everything else is polynomial.
$\Rightarrow$ DIRECTED FEEDBACK EDGE SET is FPT.

Conclusions

- A simple (but essentially tight) bound on the number of important separators.
- Algorithmic results: FPT algorithms for
  - MULTIWAY CUT in undirected graphs,
  - SKEW MULTICUT in directed graphs, and
  - DIRECTED FEEDBACK VERTEX/EDGE SET.