Important separators and parameterized algorithms

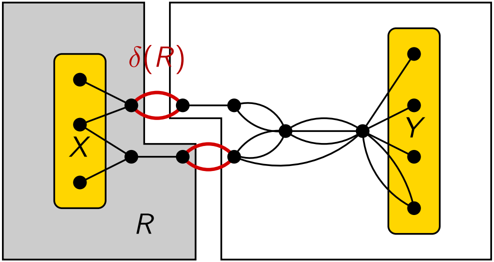
Dániel Marx[1]

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

School on Parameterized Algorithms and Complexity
Będlewo, Poland
August 21, 2014

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.

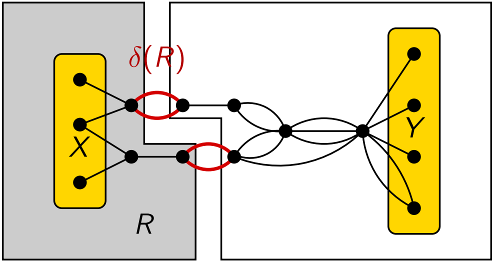
Observation: Every minimal (X, Y)-cut S can be expressed as $S = \delta(R)$ for some $X \subseteq R$ and $R \cap Y = \emptyset$.



Definition

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

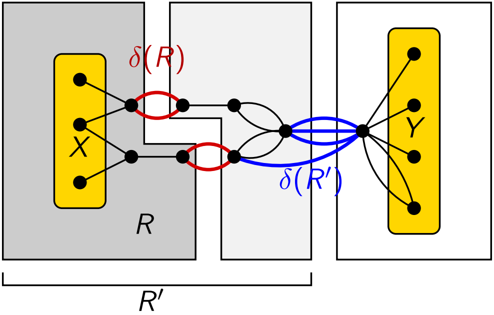
Note: Can be checked in polynomial time if a cut is important ($\delta(R)$ is important if $R = R_{\max}$).



Definition

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

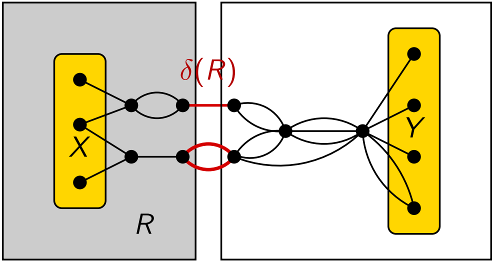
Note: Can be checked in polynomial time if a cut is important ($\delta(R)$ is important if $R = R_{\max}$).



Definition

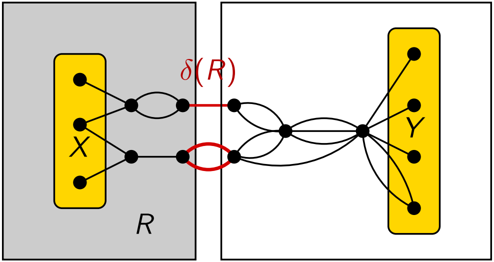
A minimal (X, Y)-cut $\delta(R)$ is **important** if there is no (X, Y)-cut $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$.

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



Theorem

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



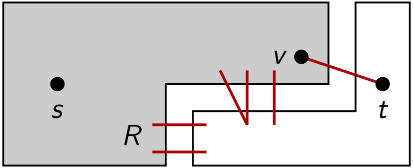
Lemma:

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.

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.

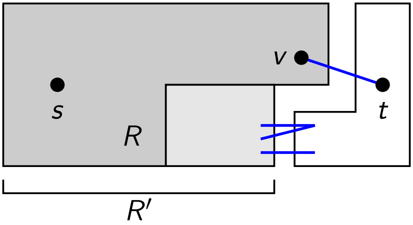


Suppose that $vt \in \delta(R)$ and $|\delta(R)| = k$.

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.

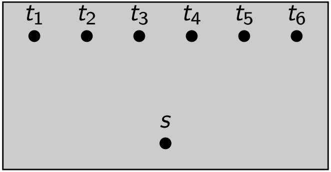
Proof: We show that every such edge is contained in an important (s,t)-cut of size at most k.



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 $v \in R'$.

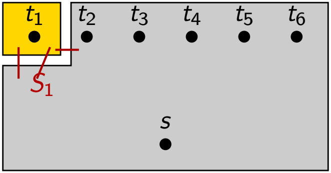
Let $s, t_1, \dots, t_n$ be vertices and $S_1, \dots, S_n$ be sets of at most k edges such that S_i separates t_i from s, but S_i **does not** separate t_j from s for any $j \neq i$.

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



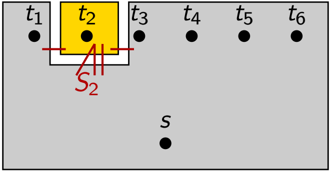
Let $s, t_1, \dots, t_n$ be vertices and $S_1, \dots, S_n$ be sets of at most k edges such that S_i separates t_i from s, but S_i **does not** separate t_j from s for any $j \neq i$.

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



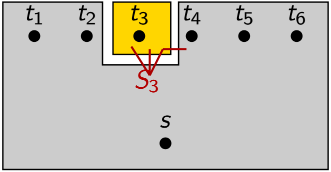
Let $s, t_1, \ldots, t_n$ be vertices and $S_1, \ldots, S_n$ be sets of at most k edges such that S_i separates t_i from s, but S_i **does not** separate t_j from s for any $j \neq i$.

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



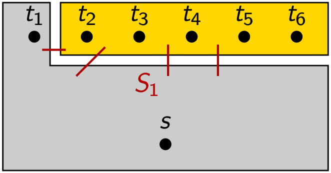
Let $s, t_1, \dots, t_n$ be vertices and $S_1, \dots, S_n$ be sets of at most k edges such that S_i separates t_i from s, but S_i **does not** separate t_j from s for any $j \neq i$.

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 $j \neq i$.

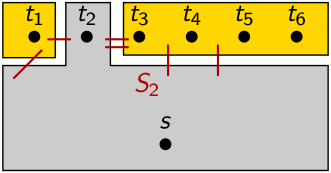
It is possible that n is "large" even if k is "small."



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 $j \neq i$.

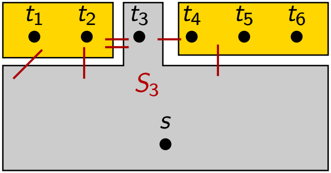
It is possible that n is "large" even if k is "small."



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 $j \neq i$.

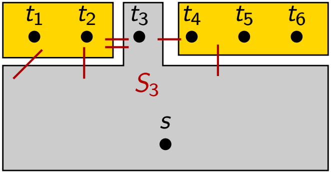
It is possible that n is "large" even if k is "small."



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

Let $s, t_1, \ldots, t_n$ 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 $j \neq i$.

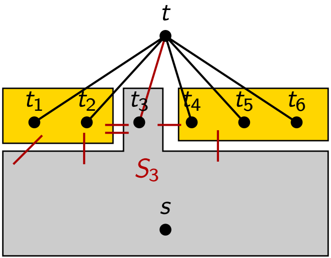
It is possible that n is "large" even if k is "small."



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

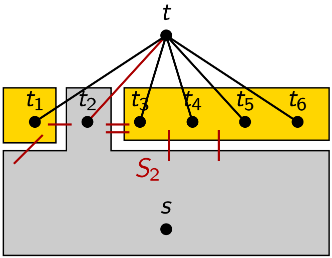


Is the opposite possible, i.e., S_i separates every t_j **except** t_i?

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.

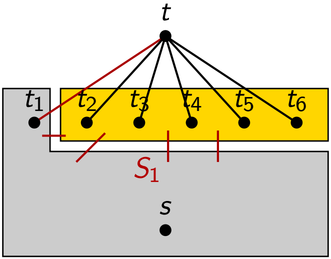


Is the opposite possible, i.e., S_i separates every t_j **except** t_i?

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

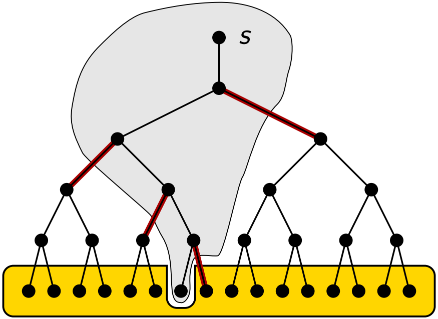
If S_i separates t_j from s if and only $j \neq i$ and every S_i has size at most k, then $n \leq (k+1) \cdot 4^{k+1}$.

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.

Lemma

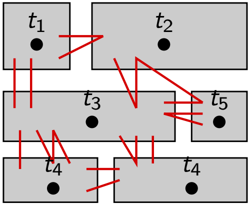
If S_i separates t_j from s if and only $j \neq i$ and every S_i has size at most k, then $n \leq (k+1) \cdot 4^{k+1}$.

Lower bound: in a binary tree of height k, any of the 2^k leaves can be the only reachable leaf after removing k edges.



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.

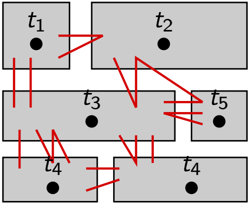
Multiway Cut	
Input:	Graph G, set T of vertices, integer k
Find:	A **multiway cut** S of at most k edges.



Polynomial for $|T| = 2$, but NP-hard for any fixed $|T| \geq 3$.

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.



Theorem

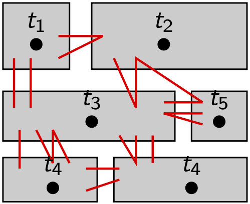
MULTIWAY CUT on planar graphs can be solved in time $2^{O(|T|)} \cdot n^{O(\sqrt{|T|})}$.

Theorem

MULTIWAY CUT on planar graphs is W[1]-hard parameterized by $|T|$.

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 $4^k \cdot n^{O(1)}$, i.e., it is fixed-parameter tractable (FPT) parameterized by the size k of the solution.

Pushing Lemma

Let $t \in T$. The MULTIWAY CUT problem has a solution S that contains an important $(t, T \setminus t)$-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 $(t, T \setminus t)$ cut S of size at most k.
4. Set $G := G \setminus S$ and $k := k - |S|$.
5. Go to step 1.

We can give a 4^k bound on the size of the search tree.

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

Input:	Graph G, pairs (s_1, t_1), ..., (s_ℓ, t_ℓ), 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.

MULTICUT

Input: Graph G, pairs (s_1, t_1), ..., (s_ℓ, t_ℓ), 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.

Much more involved:

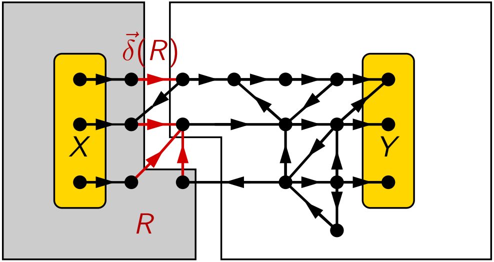
Theorem

MULTICUT is FPT parameterized by the size k of the solution.

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

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

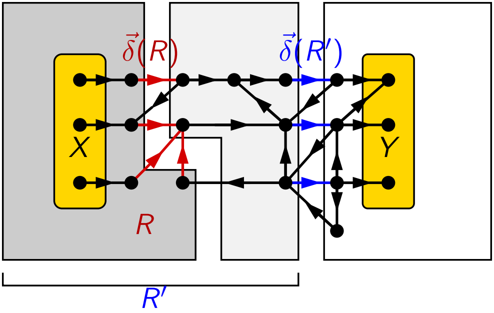
Definition: A minimal (X, Y)-cut $\vec{\delta}(R)$ is **important** if there is no (X, Y)-cut $\vec{\delta}(R')$ with $R \subset R'$ and $|\vec{\delta}(R')| \leq |\vec{\delta}(R)|$.



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

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

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



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

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

Definition: A minimal (X, Y)-cut $\vec{\delta}(R)$ is **important** if there is no (X, Y)-cut $\vec{\delta}(R')$ with $R \subset R'$ and $|\vec{\delta}(R')| \le |\vec{\delta}(R)|$.

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

Theorem

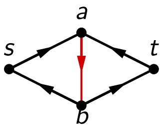
There are at most 4^k important directed (X, Y)-cuts of size at most k.

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 $(t, T \setminus t)$-cut.

Directed counterexample:



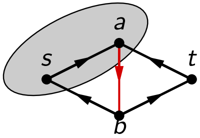
Unique solution with $k = 1$ edges, but it is not an important cut (boundary of $\{s, a\}$, but the boundary of $\{s, a, b\}$ has same size).

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 $(t, T \setminus t)$-cut.

Directed counterexample:



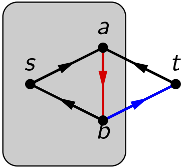
Unique solution with $k = 1$ edges, but it is not an important cut (boundary of $\{s, a\}$, but the boundary of $\{s, a, b\}$ has same size).

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 $(t, T \setminus t)$-cut.

Directed counterexample:



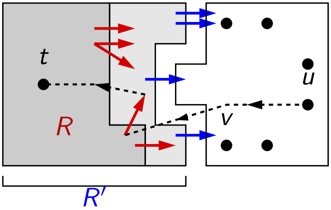
Unique solution with $k = 1$ edges, but it is not an important cut (boundary of $\{s, a\}$, but the boundary of $\{s, a, b\}$ has same size).

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 $(t, T \setminus t)$-cut.

Problem in the undirected proof:



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

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 $(t, T \setminus t)$-cut.

Using additional techniques, one can show:

Theorem

Directed Multiway Cut is FPT parameterized by the size k of the solution.

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

Directed Multicut is W[1]-hard parameterized by k.

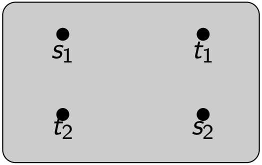
DIRECTED MULTICUT

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

DIRECTED MULTICUT is W[1]-hard parameterized by k.

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



Directed Multicut

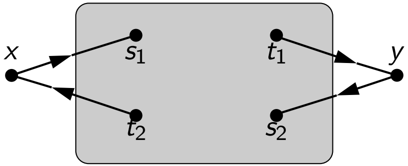
Input: Graph G, pairs (s_1, t_1), ..., (s_ℓ, t_ℓ), integer k

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

Theorem

Directed Multicut is W[1]-hard parameterized by k.

But the case $\ell = 2$ can be reduced to Directed Multiway Cut:



Directed Multicut

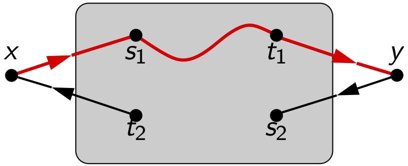
Input: Graph G, pairs (s_1, t_1), ..., (s_ℓ, t_ℓ), integer k

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

Theorem

Directed Multicut is W[1]-hard parameterized by k.

But the case $\ell = 2$ can be reduced to Directed Multiway Cut:



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

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 questions:

?

Is Directed Multicut with $\ell = 3$ FPT?

Is Directed Multicut FPT parameterized by k and ℓ?

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

Directed Multicut is W[1]-hard parameterized by k on DAGs.

Theorem

Directed Multicut is NP-hard for $\ell = 2$ on DAGs.

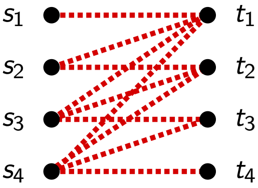
Theorem

Directed Multicut is FPT parameterized by k and ℓ on DAGs.

Skew Multicut

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

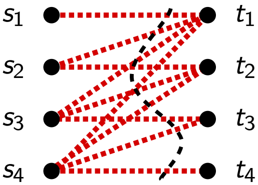
Find: A set S of k directed edges such that $G \setminus S$ contains no $s_i \to t_j$ path for any $i \geq j$.



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 $i \geq j$.



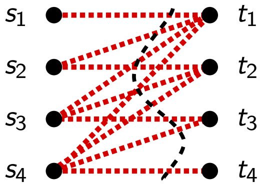
Pushing Lemma

Skew Multicut problem has a solution S that contains an important $(s_\ell, \{t_1, \ldots, t_\ell\})$-cut.

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 $i \geq j$.



Pushing Lemma

Skew Multicut problem has a solution S that contains an important $(s_\ell, \{t_1, \ldots, t_\ell\})$-cut.

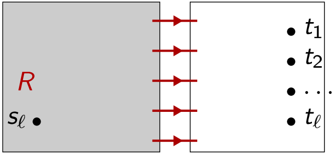
Theorem

Skew Multicut can be solved in time $4^k \cdot n^{O(1)}$.

Pushing Lemma

Skew Multicut problem has a solution S that contains an important $(s_\ell, \{t_1, \dots, t_\ell\})$-cut.

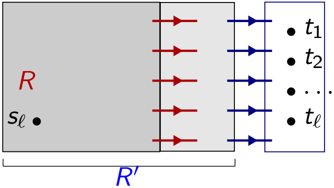
Proof: Similar to the undirected pushing lemma. Let R be the vertices reachable from t in $G \setminus S$ for a solution S.



Pushing Lemma

Skew Multicut problem has a solution S that contains an important $(s_\ell, \{t_1, \ldots, t_\ell\})$-cut.

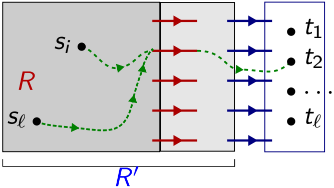
Proof: Similar to the undirected pushing lemma. Let R be the vertices reachable from t in $G \setminus S$ for a solution S.



Pushing Lemma

Skew Multicut problem has a solution S that contains an important $(s_\ell, \{t_1, \ldots, t_\ell\})$-cut.

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

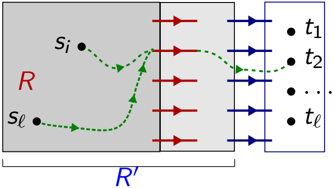


$\delta(R)$ is not important, then there is an important cut $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$. Replace S with $S' := (S \setminus \delta(R)) \cup \delta(R') \Rightarrow |S'| \leq |S|$

Pushing Lemma

Skew Multicut problem has a solution S that contains an important $(s_\ell, \{t_1, \ldots, t_\ell\})$-cut.

Proof: Similar to the undirected pushing lemma. 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 skew multicut: (1) There is no s_ℓ-t_j path in $G \setminus S'$ for any j and (2) a s_i-t_j path in $G \setminus S'$ implies a s_ℓ-t_j path, a contradiction.

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

Directed Feedback Edge Set is FPT parameterized by the size k of the solution.

Solution uses the technique of **iterative compression**.

Directed Feedback Edge Set Compression

Input: Directed graph G, integer k, a set W of $k+1$ edges such that $G \setminus W$ 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 W gives us useful structural information about G.

Lemma

The compression problem is FPT parameterized by k.

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

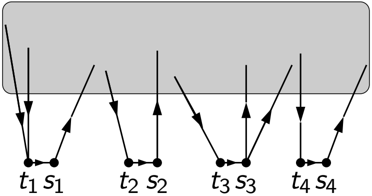
Easier than the original problem, as the extra input W gives us useful structural information about G.

Lemma

The compression problem is FPT parameterized by k.

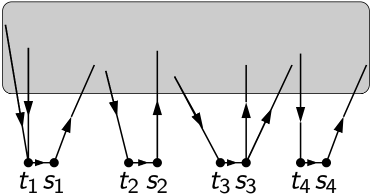
A useful trick for edge deletion problems: we define the compression problem in a way that a solution of $k+1$ **vertices** are given and we have to find a solution of k **edges**.

Proof: Let $W = \{w_1, \ldots, w_{k+1}\}$
Let us split each w_i into an edge $\overrightarrow{t_i s_i}$.



- By guessing the order of $\{w_1, \ldots, w_{k+1}\}$ in the acyclic ordering of $G \setminus S$, we can assume that $w_1 < w_2 < \cdots < w_{k+1}$ in $G \setminus S$ [$(k+1)!$ possibilities].

Proof: Let $W = \{w_1, \ldots, w_{k+1}\}$
Let us split each w_i into an edge $\overrightarrow{t_i s_i}$.



Claim:

$G \setminus S$ is acyclic and has an ordering with $w_1 < w_2 < \cdots < w_{k+1}$

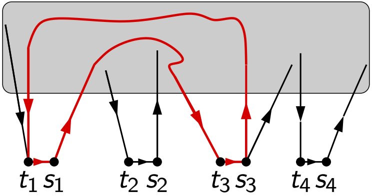
$\Downarrow$

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

$\Downarrow$

$G \setminus S$ is acyclic

Proof: Let $W = \{w_1, \dots, w_{k+1}\}$
Let us split each w_i into an edge $\overrightarrow{t_i s_i}$.



Claim:

$G \setminus S$ is acyclic and has an ordering with $w_1 < w_2 < \cdots < w_{k+1}$

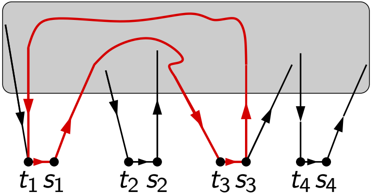
$\Downarrow$

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

$\Downarrow$

$G \setminus S$ is acyclic

Proof: Let $W = \{w_1, \ldots, w_{k+1}\}$
Let us split each w_i into an edge $\overrightarrow{t_i s_i}$.



Claim:

$G \setminus S$ is acyclic and has an ordering with $w_1 < w_2 < \cdots < w_{k+1}$

$$\Downarrow$$

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

$$\Downarrow$$

$G \setminus S$ is acyclic

$\Rightarrow$ We can solve the compression problem by $(k+1)!$ applications of Skew Multicut.

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 W of $k+1$ vertices such that $G \setminus W$ is acyclic
Find:	A set S of k edges such that $G \setminus S$ is acyclic.

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

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 W of $k+1$ vertices such that $G \setminus W$ is acyclic
Find:	A set S of k edges such that $G \setminus S$ is acyclic.

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

We get it for free!

Powerful technique: **iterative compression**.

Let v_1, …, v_n be the edges of G and let G_i be the subgraph induced by $\{v_1, \ldots, v_i\}$.

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

Let $v_1, \ldots, v_n$ be the edges of G and let G_i be the subgraph induced by $\{v_1, \ldots, v_i\}$.

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

- For $i = 1$, we have the trivial solution $S_i = \emptyset$.
- Suppose we have a solution S_i for G_i. Let W_i contain the head of each edge in S_i. Then $W_i \cup \{v_{i+1}\}$ is a set of at most $k + 1$ vertices whose removal makes G_{i+1} acyclic.
- Use the compression algorithm for G_{i+1} with the set $W_i \cup \{v_{i+1}\}$.
  - 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 n times, everything else is polynomial.
$\Rightarrow$ Directed Feedback Edge Set is FPT.

So far we have seen:

- Definition of important cuts.
- Combinatorial bound on the number of important cuts.
- Pushing argument: we can assume that the solution contains an important cut. Solves Multiway Cut, Skew Multicut.
- Iterative compression reduces Directed Feedback Vertex Set to Skew Multicut.

Next:

- Randomized sampling of important separators.