Important separators and parameterized algorithms

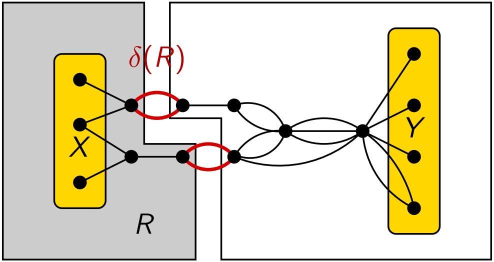
Dániel Marx[1]

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

School on Parameterized Algorithms and Complexity
Będlewo, Poland
August 22, 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.

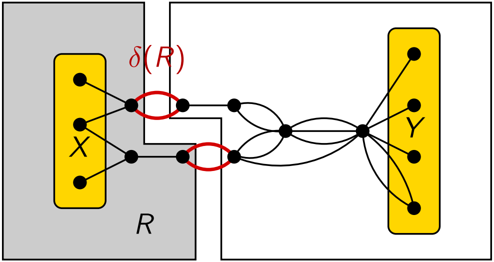
Observation: Every minimal (X, Y)-cut S can be expressed as $S = \delta(R)$ for some $X \subseteq R$ and $R \cap Y = \emptyset$.



Definition

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

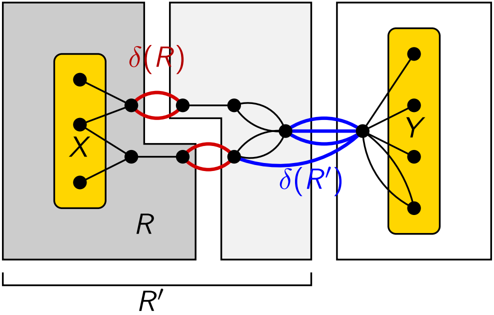
Note: Can be checked in polynomial time if a cut is important ($\delta(R)$ is important if $R = R_{\max}$).



Definition

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

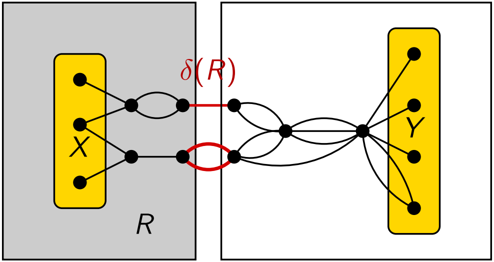
Note: Can be checked in polynomial time if a cut is important ($\delta(R)$ is important if $R = R_{\max}$).



Definition

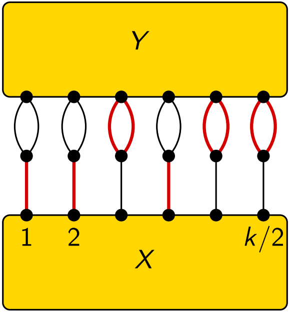
A minimal (X, Y)-cut $\delta(R)$ is **important** if there is no (X, Y)-cut $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$.

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



The number of important cuts can be exponentially large.

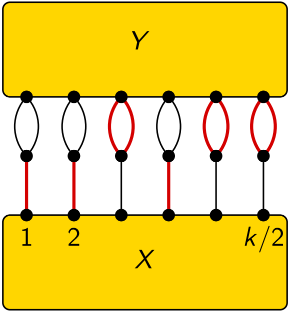
Example:



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.

A new technique used by several results:

- Multicut [M. and Razgon STOC 2011]
- Clustering problems [Lokshtanov and M. ICALP 2011]
- Directed Multiway Cut [Chitnis, Hajiaghayi, M. SODA 2012]
- Directed Multicut in DAGs [Kratsch, Pilipczuk, Pilipczuk, Wahlström ICALP 2012]
- Directed Subset Feedback Vertex Set [Chitnis, Cygan, Hajiaghayi, M. ICALP 2012]
- Parity Multiway Cut [Lokshtanov, Ramanujan ICALP 2012]
- List homomorphism removal problems [Chitnis, Egri, and M. ESA 2013]
- ... more work in progress.

We want to partition objects into clusters subject to certain requirements (typically: related objects are clustered together, bounds on the number or size of the clusters etc.)

(p, q)-CLUSTERING

Input: A graph G, integers p, q.

Find: A partition $(V_1, \ldots, V_m)$ of $V(G)$ such that for every i

- $|V_i| \leq p$ and
- $\delta(V_i) \leq q$.

$\delta(V_i)$: number of edges leaving V_i.

Theorem

(p, q)-CLUSTERING can be solved in time $2^{O(q)} \cdot n^{O(1)}$.

Good cluster: size at most p and at most q edges leaving it.

Necessary condition:

Every vertex is contained in a good cluster.

Good cluster: size at most p and at most q edges leaving it.

Necessary condition:

Every vertex is contained in a good cluster.

But surprisingly, this is also a **sufficient condition!**

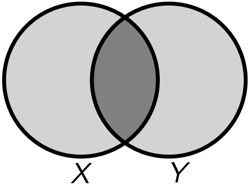
Lemma

Graph G has a (p, q)-clustering if and only if every vertex is in a good cluster.

Lemma

Graph G has a (p, q)-clustering if and only if every vertex is in a good cluster.

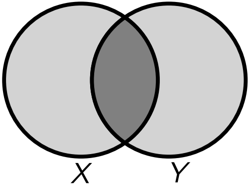
Proof: Find a collection of good clusters covering every vertex and having minimum total size. Suppose two clusters intersect.



Lemma

Graph G has a (p, q)-clustering if and only if every vertex is in a good cluster.

Proof: Find a collection of good clusters covering every vertex and having minimum total size. Suppose two clusters intersect.



$$\delta(X) + \delta(Y) \geq \delta(X \setminus Y) + \delta(Y \setminus X)$$

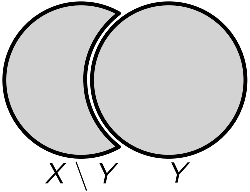
(posimodularity)

$\Rightarrow$ either $\delta(X) \geq \delta(X \setminus Y)$ or $\delta(Y) \geq \delta(Y \setminus X)$ holds.

Lemma

Graph G has a (p, q)-clustering if and only if every vertex is in a good cluster.

Proof: Find a collection of good clusters covering every vertex and having minimum total size. Suppose two clusters intersect.



$$\delta(X) + \delta(Y) \geq \delta(X \setminus Y) + \delta(Y \setminus X)$$

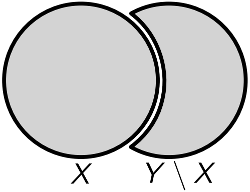
(posimodularity)

If $\delta(X) \geq \delta(X \setminus Y)$, replace X with $X \setminus Y$, strictly decreasing the total size of the clusters.

Lemma

Graph G has a (p, q)-clustering if and only if every vertex is in a good cluster.

Proof: Find a collection of good clusters covering every vertex and having minimum total size. Suppose two clusters intersect.



$$\delta(X) + \delta(Y) \geq \delta(X \setminus Y) + \delta(Y \setminus X)$$

(posimodularity)

If $\delta(Y) \geq \delta(Y \setminus X)$, replace Y with $Y \setminus X$, strictly decreasing the total size of the clusters. QED ■

We have seen:

Lemma

Graph G has a (p, q)-clustering if and only if every vertex is in a good cluster.

All we have to do is to check if a given vertex v is in a good cluster. Trivial to do in time $n^{O(q)}$.

We have seen:

Lemma

Graph G has a (p, q)-clustering if and only if every vertex is in a good cluster.

All we have to do is to check if a given vertex v is in a good cluster. Trivial to do in time $n^{O(q)}$.

We prove next:

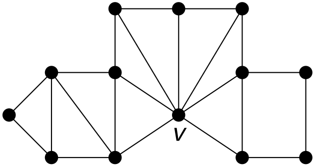
Lemma

We can check in time $2^{O(q)} \cdot n^{O(1)}$ if v is in a good cluster.

Definition

Fix a distinguished vertex v in a graph G. A set $X \subseteq V(G)$ is an **important set** if

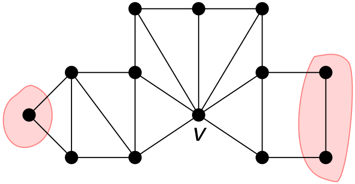
- $v \notin X$,
- there is no set $X \subset X'$ with $v \notin X$ and $\delta(X') \leq \delta(X)$.



Definition

Fix a distinguished vertex v in a graph G. A set $X \subseteq V(G)$ is an **important set** if

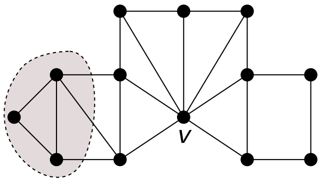
- $v \notin X$,
- there is no set $X \subset X'$ with $v \notin X$ and $\delta(X') \leq \delta(X)$.



Definition

Fix a distinguished vertex v in a graph G. A set $X \subseteq V(G)$ is an **important set** if

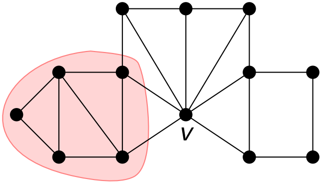
- $v \notin X$,
- there is no set $X \subset X'$ with $v \notin X$ and $\delta(X') \leq \delta(X)$.



Definition

Fix a distinguished vertex v in a graph G. A set $X \subseteq V(G)$ is an **important set** if

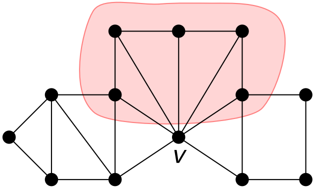
- $v \notin X$,
- there is no set $X \subset X'$ with $v \notin X$ and $\delta(X') \leq \delta(X)$.



Definition

Fix a distinguished vertex v in a graph G. A set $X \subseteq V(G)$ is an **important set** if

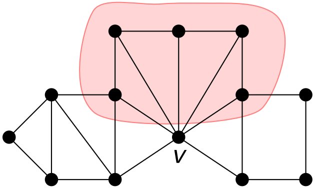
- $v \notin X$,
- there is no set $X \subset X'$ with $v \notin X$ and $\delta(X') \leq \delta(X)$.



Definition

Fix a distinguished vertex v in a graph G. A set $X \subseteq V(G)$ is an **important set** if

- $v \notin X$,
- there is no set $X \subset X'$ with $v \notin X$ and $\delta(X') \leq \delta(X)$.

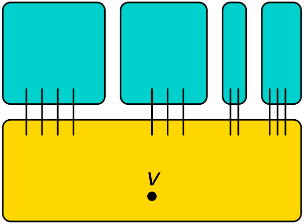


Observation: X is an important set if and only if $\delta(X)$ is an important (x, v)-cut for every $x \in X$.

Consequence: Every vertex is contained in at most 4^k important sets.

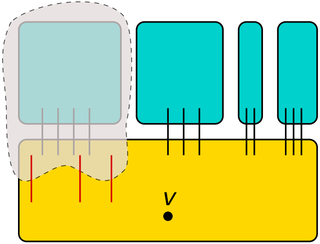
Lemma

If C is a good cluster of minimum size containing v, then every component of $G \setminus C$ is an important set.



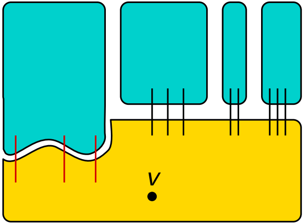
Lemma

If C is a good cluster of minimum size containing v, then every component of $G \setminus C$ is an important set.



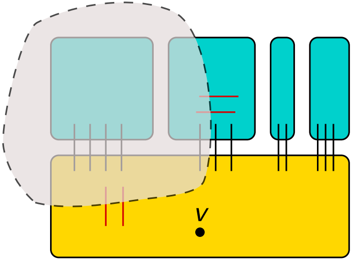
Lemma

If C is a good cluster of minimum size containing v, then every component of $G \setminus C$ is an important set.



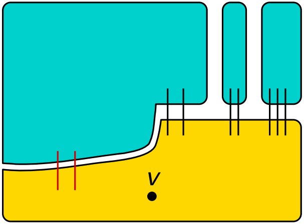
Lemma

If C is a good cluster of minimum size containing v, then every component of $G \setminus C$ is an important set.



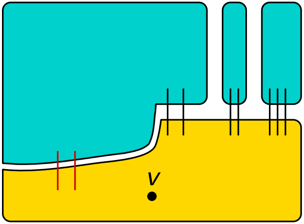
Lemma

If C is a good cluster of minimum size containing v, then every component of $G \setminus C$ is an important set.



Lemma

If C is a good cluster of minimum size containing v, then every component of $G \setminus C$ is an important set.

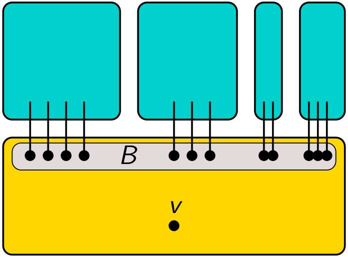


Thus C can be obtained by removing at most q important sets from $V(G)$ (but there are $n^{O(q)}$ possibilities, we cannot try all of them).

- Let $\mathcal{X}$ be the set of all important sets of boundary size at most q in G.
- Let $\mathcal{X}' \subseteq \mathcal{X}$ contain each set with probability $\frac{1}{2}$ independently.
- Let $Z = \bigcup_{X \in \mathcal{X}'} X$.
- Let B be the set of vertices in C with neighbors outside C.

Lemma

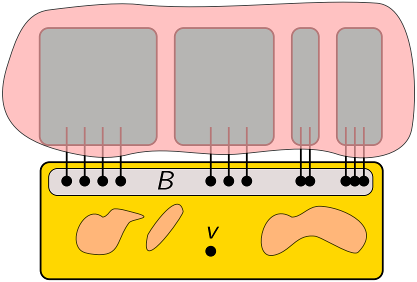
Let C be a good cluster of minimum size containing v. With probability $2^{-2^{O(q)}}$, Z covers $G \setminus C$ and is disjoint from B.



- Let $\mathcal{X}$ be the set of all important sets of boundary size at most q in G.
- Let $\mathcal{X}' \subseteq \mathcal{X}$ contain each set with probability $\frac{1}{2}$ independently.
- Let $Z = \bigcup_{X \in \mathcal{X}'} X$.
- Let B be the set of vertices in C with neighbors outside C.

Lemma

Let C be a good cluster of minimum size containing v. With probability $2^{-2^{O(q)}}$, Z covers $G \setminus C$ and is disjoint from B.



Lemma

Let C be a good cluster of minimum size containing v. With probability $2^{-2^{O(q)}}$, Z covers $G \setminus C$ and is disjoint from B.

Two events:

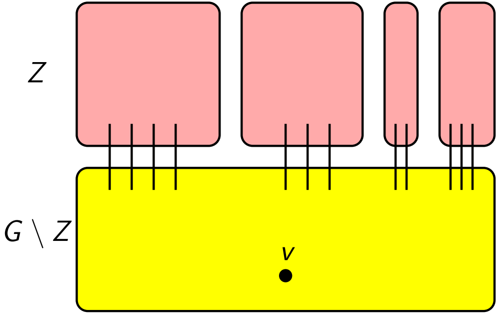
(E1) Z covers $G \setminus C$.
Each of the at most q components is an important set
$\Rightarrow$ all of them are selected by probability at least 2^{-q}.

(E2) Z is disjoint from B.
Each vertex of B is in at most 4^q members of $\mathcal{X}$
$\Rightarrow$ all of them are selected by probability at least 2^{-q4^q}.

The two events are independent (involve different sets of $\mathcal{X}$), thus the claimed probability follows.

Let C be a good cluster of minimum size containing v and assume

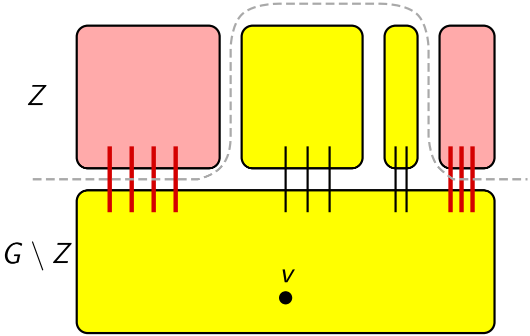
- $G \setminus C$ is covered by Z, and
- Z is disjoint from B (hence no edge going out of C is contained in Z).



Where is the good cluster C in the figure?

Let C be a good cluster of minimum size containing v and assume

- $G \setminus C$ is covered by Z, and
- Z is disjoint from B (hence no edge going out of C is contained in Z).

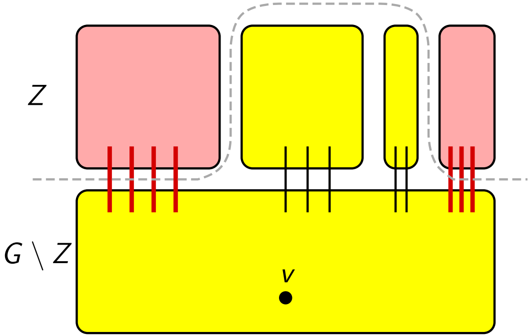


Where is the good cluster C in the figure?

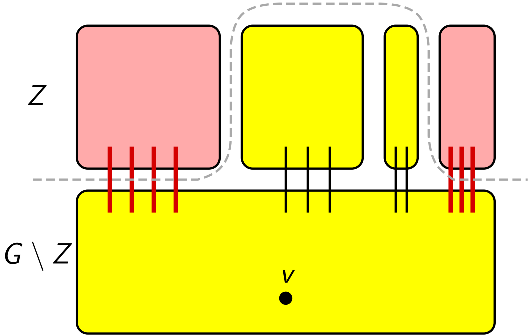
Observe: Components of Z are either fully in the cluster or fully outside the cluster. What is this problem?

Let C be a good cluster of minimum size containing v and assume

- $G \setminus C$ is covered by Z, and
- Z is disjoint from B (hence no edge going out of C is contained in Z).



KNAPSACK!

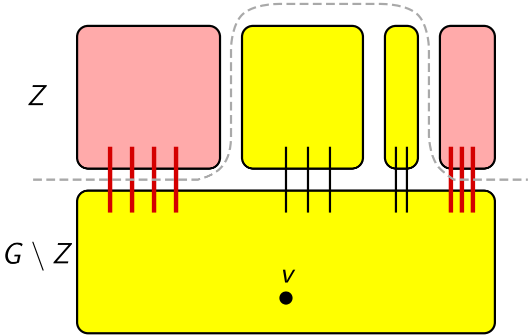


We interpret the components $V_1, \ldots, V_t$ of $G[Z]$ as items:

- V_i has value $\delta(V_i)$ and
- V_i has weight $|V_i|$.

The goal is to select items with total value at least $\delta(Z) - q$ and total weight at most $p - |V(G) \setminus Z|$.

Finding good clusters by KNAPSACK



Standard DP solves it in polynomial time: let $T[i,j]$ be the maximum value of a subset of the first i items having total weight at most j.

Recurrence:

$$T[i,j] = \max\{T[i-1,j], T[i-1,j-|V_i|] + \delta(V_i)\}$$

(p,q)-Clustering

Input: A graph G, integers p, q.

Find: A partition $(V_1, \dots, V_m)$ of $V(G)$ such that for every i

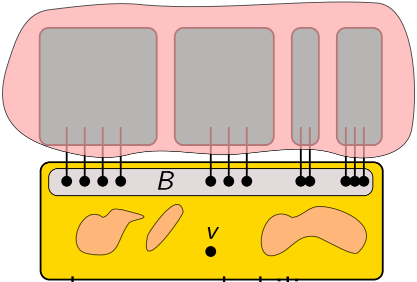
- $|V_i| \leq p$ and
- $\delta(V_i) \leq q$.

- It is sufficient to check for each vertex v if it is in a good cluster.
- Enumerate all the important sets.
- Let Z be the union of random important sets.
- The solution is obtained by extending $G \setminus Z$ with some of the components of $G[Z]$.
- Knapsack.

- Let $\mathcal{X}$ be the set of all important sets of boundary size at most q in G.
- Let $\mathcal{X}' \subseteq \mathcal{X}$ contain each set X with probability $\boxed{4^{-|\delta(X)|}}$.
- Let $Z = \bigcup_{X \in \mathcal{X}'} X$.
- Let B be the set of vertices in C with neighbors outside C.

Lemma

Let C be a good cluster of minimum size containing v. With probability $2^{-O(q)}$, Z covers $G \setminus C$ and is disjoint from B.



Lemma

Let C be a good cluster of minimum size containing v. With probability $2^{-O(q)}$, Z covers $G \setminus C$ and is disjoint from B.

We need to bound the probability of two independent events:

(E1) Z covers $G \setminus C$.

(E2) Z is disjoint from B.

Lemma

Let C be a good cluster of minimum size containing v. With probability $2^{-O(q)}$, Z covers $G \setminus C$ and is disjoint from B.

We need to bound the probability of two independent events:

(E1) Z covers $G \setminus C$.

Probability of selecting every component $K_1, \ldots, K_t$ of $G \setminus C$:

$$\prod_{i=1}^{t} 4^{-|\delta(K_i)|} = 4^{-\sum_{i=1}^{t} |\delta(K_i)|} = 4^{-|\delta(C)|} \geq 4^{-q}.$$

Lemma

Let C be a good cluster of minimum size containing v. With probability $2^{-O(q)}$, Z covers $G \setminus C$ and is disjoint from B.

We need to bound the probability of two independent events:

(E2) Z is disjoint from B.

Recall: $\sum_{S \in \mathcal{S}} 4^{-|S|}$ holds for the set $\mathcal{S}$ of important cuts.

Probability that no important sets containing $w \in B$ is selected:

$$\prod_{\substack{X \in \mathcal{X} \\ w \in X}} (1 - 4^{-|\delta(X)|}) \approx \prod_{\substack{X \in \mathcal{X} \\ w \in X}} \exp\left(-4^{-|\delta(X)|}\right) = \exp\left(-\sum_{\substack{X \in \mathcal{X} \\ w \in X}} 4^{-|\delta(X)|}\right) \geq 1/e.$$

Thus the probability that no vertex of B is covered is $2^{-O(|B|)}$:

$$\prod_{\substack{X \in \mathcal{X} \\ X \cap B \neq \emptyset}} (1 - 4^{-|\delta(X)|}) \geq \prod_{w \in B} \prod_{\substack{X \in \mathcal{X} \\ w \in X}} (1 - 4^{-|\delta(X)|}) = 2^{-O(|B|)} = 2^{-O(q)}.$$

- Randomized $2^{O(q)} \cdot n^{O(1)}$ time algorithm for (p, q)-CLUSTERING.
- Derandomization is possible using standard techniques, but nontrivial to obtain $2^{O(q)}$ running time.
- Parameterization by p: we can get a $2^{O(p)} \cdot n^{O(1)}$ time algorithm.
- Other variants: maximum degree in the cluster is at most p, etc.

Let G be a graph and let $\mathcal{F}$ be a set of subgraphs in G.

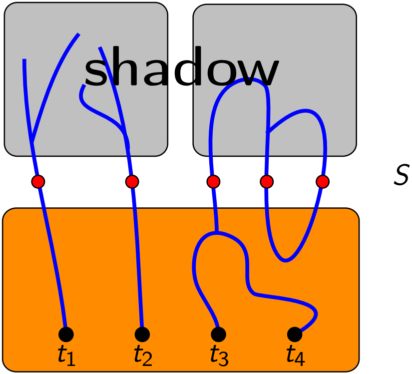
Definition

$\mathcal{F}$-**transversal**: a set of edges of vertices intersecting each subgraph in $\mathcal{F}$ (i.e., "hitting" or "killing" every object in $\mathcal{F}$).

Classical problems formulated as finding a minimum transversal:

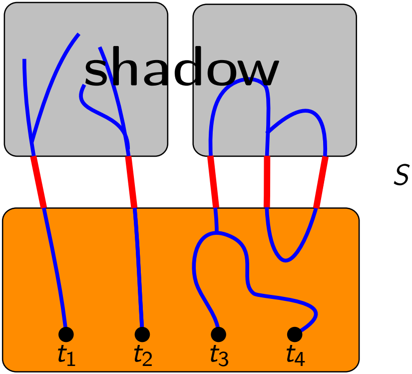
- $s - t$ Cut:
  $\mathcal{F}$ is the set of $s - t$ paths.
- Multiway Cut:
  $\mathcal{F}$ is the set of paths between terminals.
- (Directed) Feedback Vertex Set:
  $\mathcal{F}$ is the set of (directed) cycles.
- Delete edges/vertices to make the graph bipartite:
  $\mathcal{F}$ is the set of odd cycles.
- v is in a (p, q)-cluster:
  $\mathcal{F}$ is the set of all connected graphs of size $p + 1$ containing v.

Let $\mathcal{F}$ be a set of **connected** (not necessarily disjoint!) subgraphs, each **intersecting** a set T of vertices.



The **shadow** of an $\mathcal{F}$-transversal S is the set of vertices not reachable from T in $G \setminus S$.

Let $\mathcal{F}$ be a set of **connected** (not necessarily disjoint!) subgraphs, each **intersecting** a set T of vertices.



The **shadow** of an $\mathcal{F}$-transversal S is the set of vertices not reachable from T in $G \setminus S$.

Shadow: Set of vertices not reachable in $G \setminus S$.

Condition: every $F \in \mathcal{F}$ is **connected** and **intersects** T.

Theorem

In $2^{O(k)} \cdot n^{O(1)}$ time, we can compute a set Z with the following property. If there exists an $\mathcal{F}$-transversal of at most k edges, then with probability $2^{-O(k)}$ there is a minimum $\mathcal{F}$-transversal S with

- the shadow of S is covered by Z and
- no edge of S is contained in Z.

Note: The algorithm **does not** have to know $\mathcal{F}$!

Shadow: Set of vertices not reachable in $G \setminus S$.

Condition: every $F \in \mathcal{F}$ is **connected** and **intersects** T.

Theorem

In $2^{O(k)} \cdot n^{O(1)}$ time, we can compute a set Z with the following property. If there exists an $\mathcal{F}$-transversal of at most k edges, then with probability $2^{-O(k)}$ there is a minimum $\mathcal{F}$-transversal S with

- the shadow of S is covered by Z and
- no edge of S is contained in Z.

Note: The algorithm **does not** have to know $\mathcal{F}$!

Proof idea: we can assume that every component of the shadow is an important set (solution can be pushed towards T). Random selection as in the clustering problem.

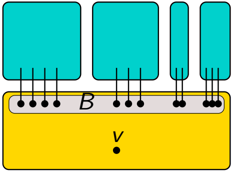
What is this good for?

$\mathcal{F}$ is the set of all connected graphs of size $p+1$ containing v.

v is in a (p, q)-cluster

$\Updownarrow$

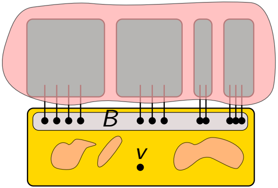
$\mathcal{F}$-transversal of q edges exists.



(p, q)-clusters as $\mathcal{F}$-transversal

$\mathcal{F}$ is the set of all connected graphs of size $p+1$ containing v.

v is in a (p, q)-cluster
$\Updownarrow$
$\mathcal{F}$-transversal of q edges exists.



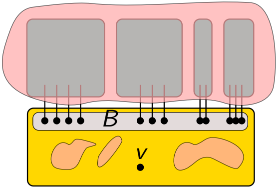
Theorem

In $2^{O(k)} \cdot n^{O(1)}$ time, we can compute a set Z with the following property. If there exists an $\mathcal{F}$-transversal of at most k edges, then with probability $2^{-O(k)}$ there is a minimum $\mathcal{F}$-transversal S with

- the shadow of S is covered by Z and
- no edge of S is contained in Z.

$\mathcal{F}$ is the set of all connected graphs of size $p+1$ containing v.

v is in a (p,q)-cluster
$\Updownarrow$
$\mathcal{F}$-transversal of q edges exists.



Theorem

In $2^{O(k)} \cdot n^{O(1)}$ time, we can compute a set Z with the following property. If there exists an $\mathcal{F}$-transversal of at most k edges, then with probability $2^{-O(k)}$ there is a minimum $\mathcal{F}$-transversal S with

- the shadow of S is covered by Z and
- no edge of S is contained in Z.

Lemma

Let C be a good cluster of minimum size containing v. With probability $2^{-O(q)}$, Z covers $G \setminus C$ and is disjoint from B.

(Directed) Multiway Cut

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

Find: A set S of at most k vertices such that $G \setminus S$ has no (directed) $t_1 - t_2$ path for any $t_1, t_2 \in T$

We have seen:

Theorem

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

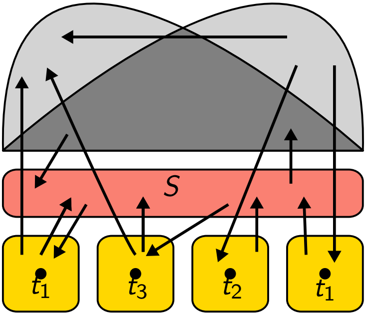
Directed version:

Theorem

Directed Multiway Cut is FPT.

Can be formulated as minimum $\mathcal{F}$-transversal, where $\mathcal{F}$ is the set of directed paths between vertices of T.

Shadow: those vertices of $G \setminus S$ that cannot be reached from T **AND** those vertices of $G \setminus S$ from which T cannot be reached.



Shadow: those vertices of $G \setminus S$ that cannot be reached from T **AND** those vertices of $G \setminus S$ from which T cannot be reached.

Condition: for every $F \in \mathcal{F}$ and every vertex $v \in F$, there is a $T \to v$ and a $v \to T$ path in F.

Theorem

In $f(k) \cdot n^{O(1)}$ time, we can compute a set Z with the following property. If there exists an $\mathcal{F}$-transversal of at most k vertices, then with probability $2^{-O(k^2)}$ there is a minimum $\mathcal{F}$-transversal S with

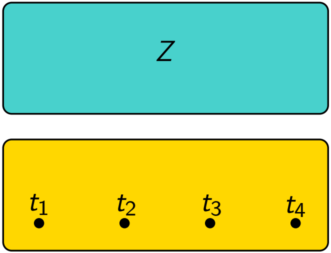
- the shadow of S is covered by Z and
- $S \cap Z = \emptyset$.

Now:

- T: terminals
- $\mathcal{F}$ contains every directed path between two distinct terminals

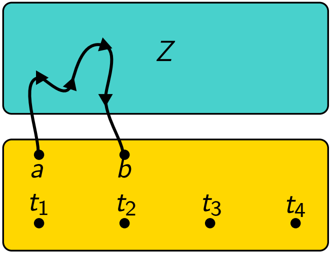
We can assume that Z is disjoint from the solution, so we want to get rid of Z.

- Deleting Z is not a good idea: can make the problem easier.
- To compensate deleting Z, if there is an $a \to b$ path with internal vertices in Z, add a direct $a \to b$ edge.



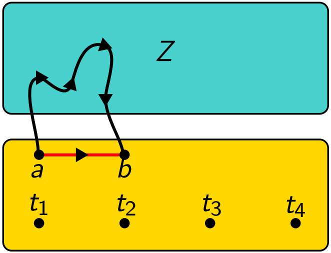
We can assume that Z is disjoint from the solution, so we want to get rid of Z.

- Deleting Z is not a good idea: can make the problem easier.
- To compensate deleting Z, if there is an $a \to b$ path with internal vertices in Z, add a direct $a \to b$ edge.



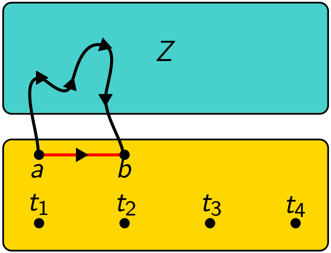
We can assume that Z is disjoint from the solution, so we want to get rid of Z.

- Deleting Z is not a good idea: can make the problem easier.
- To compensate deleting Z, if there is an $a \to b$ path with internal vertices in Z, add a direct $a \to b$ edge.



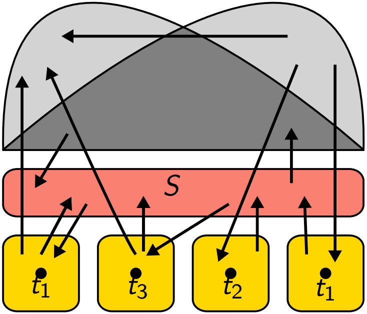
We can assume that Z is disjoint from the solution, so we want to get rid of Z.

- Deleting Z is not a good idea: can make the problem easier.
- To compensate deleting Z, if there is an $a \to b$ path with internal vertices in Z, add a direct $a \to b$ edge.

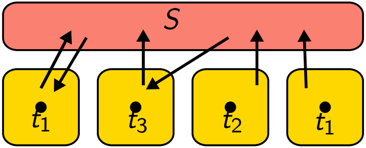


Crucial observation:
S remains a solution (since Z is disjoint from S) and
S is a **shadowless solution** (since Z covers the shadow of S).

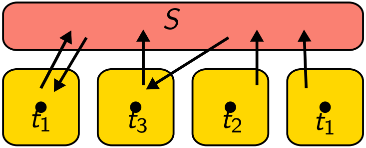
How does a shadowless solution look like?



How does a shadowless solution look like?



How does a shadowless solution look like?



It is an undirected multiway cut in the underlying undirected graph!
$\Rightarrow$ **Problem can be reduced to undirected multiway cut.**

- A simple (but essentially tight) bound on the number of important cuts.
- Algorithmic results: FPT algorithms for
  - Multiway Cut in undirected graphs,
  - Skew Multicut in directed graphs,
  - Directed Feedback Vertex/Edge Set,
  - (p, q)-Clustering,
  - Directed Multiway Cut.