Multicut is FPT

Nicolas Bousquet Jean Daligault Dániel Marx Igor Razgon Stéphan Thomassé

STOC 2011

1. Introduction
   - Parameterized complexity
   - Multicut
2. An FPT algorithm
   - Reducing the instance
   - Left cuts
   - Two different proofs
3. Conclusion

FPT

FPT

A problem parameterized by k is *FPT* (Fixed Parameter Tractable) iff there is an algorithm which runs in time $Poly(n) \cdot f(k)$ for an instance of size n and of parameter k.

FPT

FPT

A problem parameterized by k is *FPT* (Fixed Parameter Tractable) iff there is an algorithm which runs in time $Poly(n) \cdot f(k)$ for an instance of size n and of parameter k.

Interest ?

Confine the combinatorial explosion to a parameter k with $k << n$, where n is the size of the instance.

FPT

FPT

A problem parameterized by k is *FPT* (Fixed Parameter Tractable) iff there is an algorithm which runs in time $Poly(n) \cdot f(k)$ for an instance of size n and of parameter k.

Interest ?

Confine the combinatorial explosion to a parameter k with $k << n$, where n is the size of the instance.

Theorem (Courcelle)

All the problems definable in the Monadic Second Order Logic parameterized by the treewidth are FPT.

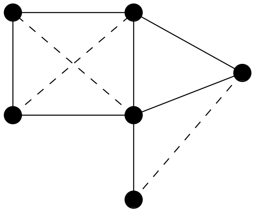
Multicut

Definition

Let $G = (V, E)$ be a graph and R be a set of pairs of vertices. A subset E' of E is a Multicut iff for each pair $xy \in R$ there is no path from x to y in $G' = (V, E \backslash E')$.

Definition

A pair of vertices of R is called a *request*.
A vertex which is in a request is called a *terminal*.



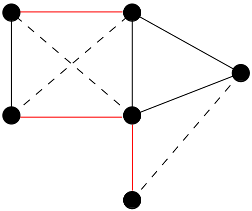
Multicut

Definition

Let $G = (V, E)$ be a graph and R be a set of pairs of vertices. A subset E' of E is a Multicut iff for each pair $xy \in R$ there is no path from x to y in $G' = (V, E \backslash E')$.

Definition

A pair of vertices of R is called a *request*.
A vertex which is in a request is called a *terminal*.



Multicut

Multicut problem

Input : A graph G, a set of requests R, an integer k.
Output : YES iff there exists a Multicut of (G, R) of size at most k.

Multicut

Multicut problem

Input : A graph G, a set of requests R, an integer k.
Output : YES iff there exists a Multicut of (G, R) of size at most k.

Theorem (Chawla, Krauthgamer, Kumar, Rabani, Sivakumar '06)

Multicut has no constant factor approximation algorithm if Khot's Unique Game conjecture holds.

Multicut

Multicut problem

Input : A graph G, a set of requests R, an integer k.
Output : YES iff there exists a Multicut of (G, R) of size at most k.

Theorem (Chawla, Krauthgamer, Kumar, Rabani, Sivakumar '06)

Multicut has no constant factor approximation algorithm if Khot's Unique Game conjecture holds.

Theorem (Gupta '03)

There is an algorithm running in polynomial time giving a solution of the Multicut problem of size at most OPT^2.

Approximation in FPT time

Theorem (Marx, Razgon '09)

There is an algorithm runnung in FPT-time which gives a solution of the Multicut problem of size at most $2 \cdot k$ or answer NO.

Approximation in FPT time

Theorem (Marx, Razgon '09)

There is an algorithm runnung in FPT-time which gives a solution of the Multicut problem of size at most $2 \cdot k$ or answer NO.

Theorem (B.,Daligault, Thomassé and Marx, Razgon '11)

Multicut parameterized by the size of the solution is FPT.

Vertex Multicut

Theorem

If we can solve the Multicut Problem given a Vertex Multicut of size k^2 in FPT time, then we can solve the general problem in FPT time.

Vertex Multicut

Theorem

If we can solve the Multicut Problem given a Vertex Multicut of size k^2 in FPT time, then we can solve the general problem in FPT time.

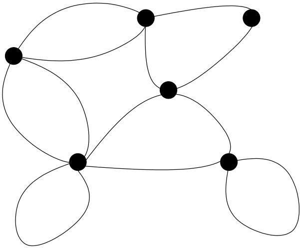
Proof :

Theorem (Gupta '03)

There is an algorithm running in polynomial time giving a solution of the Multicut problem of size at most OPT^2.

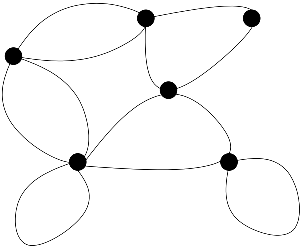
- If the solution given by Gupta is strictly greater than k^2, answer "No".
- Otherwise we have an edge Multicut of size at most k^2. Taking one endpoint of each edge gives a Vertex Multicut.

Some reductions



We want an algorithm which runs in $f(k) \cdot Poly(n)$, hence we can "guess" as much information as we want since the width and the height of the branching process depend only of k.

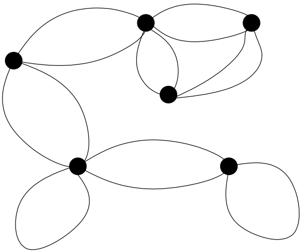
Some reductions



We want an algorithm which runs in $f(k) \cdot Poly(n)$, hence we can "guess" as much information as we want since the width and the height of the branching process depend only of k.

- We can assume that the vertices of the Vertex Multicut are separated by the Multicut.

Some reductions



We want an algorithm which runs in $f(k) \cdot Poly(n)$, hence we can "guess" as much information as we want since the width and the height of the branching process depend only of k.

- We can assume that the vertices of the Vertex Multicut are separated by the Multicut.
- We can assume that the components have one or two attachment vertices.

Left cuts

Let G be a connected graph and x be a vertex called root.

Definition

A *cut* S is a subset of vertices containing x.
The *border* Δ of a cut S is the set of edges with one endpoint in S.
We denote by δ its size.

Left cuts

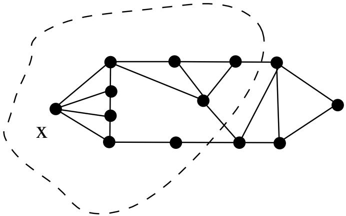
Let G be a connected graph and x be a vertex called root.

Definition

A *cut* S is a subset of vertices containing x.
The *border* Δ of a cut S is the set of edges with one endpoint in S.
We denote by δ its size.

Left cut

A *left cut* S such that if $T \subsetneq S$ then $\delta(T) > \delta(S)$.



Left cuts

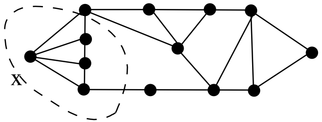
Let G be a connected graph and x be a vertex called root.

Definition

A *cut* S is a subset of vertices containing x.
The *border* Δ of a cut S is the set of edges with one endpoint in S.
We denote by δ its size.

Left cut

A *left cut* S such that if $T \subsetneq S$ then $\delta(T) > \delta(S)$.



Indivisible left cuts

Definition

A cut is *indivisible* S iff $G \backslash S$ is connected.

Theorem

Let y be a vertex. There is a bounded number (in k) of indivisible left cuts of size at most k which separate x from y.

B.,Daligault, Thomassé's proof

Theorem

In a component with one attachment vertex, we can bound (in k) the number of terminal vertices.

B.,Daligault, Thomassé's proof

Theorem

In a component with one attachment vertex, we can bound (in k) the number of terminal vertices.

- The best way to use k edges of the Multicuy in a component is to select a left cut for a terminal vertex.

B.,Daligault, Thomassé's proof

Theorem

In a component with one attachment vertex, we can bound (in k) the number of terminal vertices.

- The best way to use k edges of the Multicuy in a component is to select a left cut for a terminal vertex.
- For each vertex y, we have seen seen that there is a bounded number (in k) of indivisible left cuts of size at most k which separate x from y.
- In the solution, a union of indivisible left cut will be selected.

B.,Daligault, Thomassé's proof

Theorem

In a component with one attachment vertex, we can bound (in k) the number of terminal vertices.

- The best way to use k edges of the Multicuy in a component is to select a left cut for a terminal vertex.
- For each vertex y, we have seen seen that there is a bounded number (in k) of indivisible left cuts of size at most k which separate x from y.
- In the solution, a union of indivisible left cut will be selected.

Theorem

In a component with one attachment vertex, there is a bounded number (in k) of cuts such that if there is a solution of the Multicut problem, there is a solution using one of these cuts.

B., Daligault, Thomassé's proof

- Branch to choose a cut of $\mathcal{L}$ in each component with one attachment vertex.
- Branch (in FPT-time) to reduce the problem using the same kind of reductions for components with 2-attachment vertices.
- Solve the problem on the subdivision of a graph of size k.

Marx, Razgon's proof

Almost 2SAT problem

Input : n variables, a set of clauses of with two literals in each, an integer k.
Parameter : k
Output : YES iff there is an assignation of the variables which satisfy all exept at most k clauses.

Marx, Razgon's proof

Almost 2SAT problem

Input : n variables, a set of clauses of with two literals in each, an integer k.
Parameter : k
Output : YES iff there is an assignation of the variables which satisfy all exept at most k clauses.

Theorem (Razgon, O'Sullivan '09)

Almost 2SAT is Fixed parameter tractable.

Marx, Razgon's proof

Isolated part

Given a solution E', the isolated part is empty iff each connected component of $G(E \backslash E')$ contains a vertex of the Vertex Multicut.

Marx, Razgon's proof

Isolated part

Given a solution E', the isolated part is empty iff each connected component of $G(E \backslash E')$ contains a vertex of the Vertex Multicut.

- When the isolated part is empty, the problem can be reduced to Almost 2SAT.
- The use of random sets with a strange distribution permits to reduce to the previous case.
- This procedure can be derandomized.

1. Introduction
   - Parameterized complexity
   - Multicut

2. An FPT algorithm
   - Reducing the instance
   - Left cuts
   - Two different proofs

3. Conclusion

Polynomial kernels

Definition

A problem parameterized by k has a polynomial kernel iff there is an algorithm running in polynomial time which transforms an instance (n, k) into an instance (n', k') such that :

- $n' \leq Poly(k)$ and $k' \leq k$.
- The new instance is positive iff the original instance is positive.

Polynomial kernels

Definition

A problem parameterized by k has a polynomial kernel iff there is an algorithm running in polynomial time which transforms an instance (n, k) into an instance (n', k') such that :

- $n' \leq Poly(k)$ and $k' \leq k$.
- The new instance is positive iff the original instance is positive.

Theorem (B.,Daligault, Thomassé, Yeo '09)

Multicut in trees parameterized by the size of the solution has a kernel of size $O(k^6)$.

Polynomial kernels

Definition

A problem parameterized by k has a polynomial kernel iff there is an algorithm running in polynomial time which transforms an instance (n, k) into an instance (n', k') such that :

- $n' \leq Poly(k)$ and $k' \leq k$.
- The new instance is positive iff the original instance is positive.

Theorem (B.,Daligault, Thomassé, Yeo '09)

Multicut in trees parameterized by the size of the solution has a kernel of size $O(k^6)$.

Open problem

Does the Multicut problem have a polynomial Kernel ?

Thanks for your attention

Questions ?