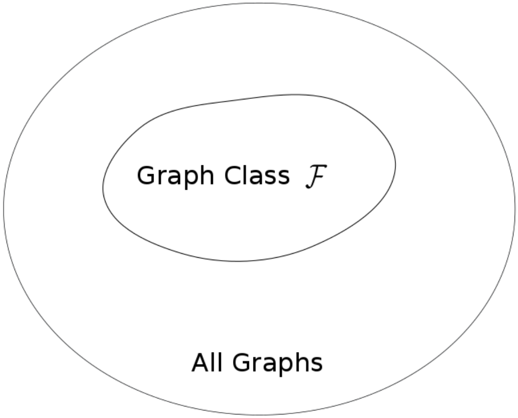
Paths to Trees and Cacti

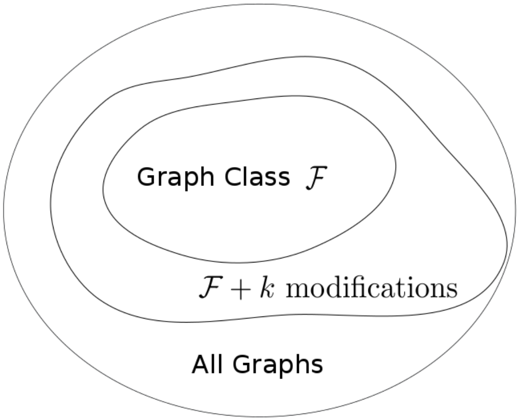
A. Agrawal[1] L. Kanesh[2] S. Saurabh[1,2] and P. Tale[2]

May 24, 2017

[1] University Of Bergen, Bergen, Norway

[2] The Institute of Mathematical Sciences, HBNI, Chennai, India


Graph Class $\mathcal{F}$
All Graphs


Graph Class $\mathcal{F}$
$\mathcal{F} + k$ modifications
All Graphs

Graph Modification Problems

$\mathcal{F}$-MODIFICATION
Input: A graph G
Question: Can we obtain a graph in $\mathcal{F}$ by *some* modifications in the graph G?

Modification allowed

- Vertex Deletion
- Edge Deletion
- Edge Addition
- Edge Contraction

Graph Modification Problems

$\mathcal{F}$-Modification
Input: A graph G
Question: Can we obtain a graph in $\mathcal{F}$ by *some* modifications in the graph G?

Modification allowed

- Vertex Deletion
- Edge Deletion
- Edge Addition
- Edge Contraction

$\mathcal{F}$-Modification : Generalization of many NP-hard problems

$\mathcal{F}$-Modification : Generalization of many NP-hard problems

Graph Problem	$\mathcal{F}$	Modification
Vertex Cover	Independent Sets	Vertex Deletion
Feedback vertex set	Forests	Vertex Deletion
Odd cycle transversal	Bipartite Graphs	Vertex Deletion
Minimum Fill-In	Chordal Graphs	Edge Addition
Edge Bipartization	Bipartite Graphs	Edge Deletion
Cluster Editing	Cluster Graphs	Edge Addition & Deletion
Tree Contraction	Trees	Edge Contraction

Outline

Parameterized Complexity & Contraction Problems

TREE CONTRACTION with additional parameter

Contraction as a Partition Problem

Kernel for BOUNDED TREE CONTRACTION

Kernel Lower Bounds

Parameterized Complexity & Contraction Problems

Parameterized Complexity : Quick Overview

- Goal : Find **better** ways to solve NP-hard problems.
- Associate (*small*) parameter k to each instance I.
- Restrict the combinatorial explosion to the parameter k.
- Parameterized problem (I, k) is *fixed-parameter tractable* (FPT) if there is an algorithm that solves it in time $\mathcal{O}(f(k) \cdot |I|^{\mathcal{O}(1)})$.
- Not all problems (for given parameter) admit such an algorithm Hierarchy of classes : FPT $\subseteq$ W[1] $\subseteq$ W[2] $\ldots$

Parameterized Complexity : Quick Overview

- Goal : Find **better** ways to solve NP-hard problems.
- Associate (*small*) parameter k to each instance I.
- Restrict the combinatorial explosion to the parameter k.
- Parameterized problem (I, k) is *fixed-parameter tractable* (FPT) if there is an algorithm that solves it in time $\mathcal{O}(f(k) \cdot |I|^{\mathcal{O}(1)})$.
- Not all problems (for given parameter) admit such an algorithm Hierarchy of classes : FPT $\subseteq$ W[1] $\subseteq$ W[2] $\dots$

Parameterized Complexity : Quick Overview

- Goal : Find **better** ways to solve NP-hard problems.
- Associate (*small*) parameter k to each instance I.
- Restrict the combinatorial explosion to the parameter k.
- Parameterized problem (I, k) is *fixed-parameter tractable* (FPT) if there is an algorithm that solves it in time $\mathcal{O}(f(k) \cdot |I|^{\mathcal{O}(1)})$.
- Not all problems (for given parameter) admit such an algorithm Hierarchy of classes : FPT $\subseteq$ W[1] $\subseteq$ W[2] $\ldots$

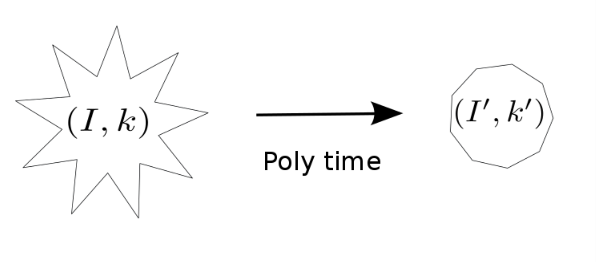
Parameterized Complexity : Quick Overview

- Goal : Find **better** ways to solve NP-hard problems.
- Associate (*small*) parameter k to each instance I.
- Restrict the combinatorial explosion to the parameter k.
- Parameterized problem (I, k) is *fixed-parameter tractable (*FPT) if there is an algorithm that solves it in time $\mathcal{O}(f(k) \cdot |I|^{\mathcal{O}(1)})$.
- Not all problems (for given parameter) admit such an algorithm Hierarchy of classes : FPT $\subseteq$ W[1] $\subseteq$ W[2] $\ldots$

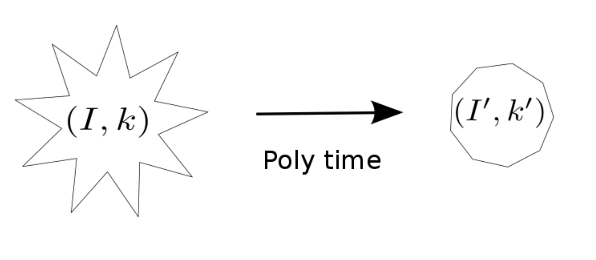
Parameterized Complexity : Quick Overview

- Goal : Find **better** ways to solve NP-hard problems.
- Associate (*small*) parameter k to each instance I.
- Restrict the combinatorial explosion to the parameter k.
- Parameterized problem (I, k) is *fixed-parameter tractable (*FPT) if there is an algorithm that solves it in time $\mathcal{O}(f(k) \cdot |I|^{\mathcal{O}(1)})$.
- Not all problems (for given parameter) admit such an algorithm Hierarchy of classes : $\text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \dots$

Kernelization : Quick Overview



Kernelization : Quick Overview



Parameterized problem (I, k) admits a $h(k)$-kernel if there exists a poly-time algorithm $\mathcal{A}$ which given an input (I, k) outputs (I', k') such that

- $|I'| + k' \leq h(k)$
- (I, k) is YES instance iff (I', k') is YES instance

Kernelization : Quick Overview

- Mathematical analysis of pre-processing.

Kernelization : Quick Overview

- Mathematical analysis of pre-processing.
- Goal: Reduce the size of input instance (in polynomial time) without changing the answer.

Kernelization : Quick Overview

- Mathematical analysis of pre-processing.
- Goal: Reduce the size of input instance (in polynomial time) without changing the answer.
- Not all problems (for given parameter) admit a kernel.

Kernelization : Quick Overview

- Mathematical analysis of pre-processing.
- Goal: Reduce the size of input instance (in polynomial time) without changing the answer.
- Not all problems (for given parameter) admit a kernel.
- Not all problems (for given parameter) admit a polynomial kernel.

Kernelization : Quick Overview

- Mathematical analysis of pre-processing.
- Goal: Reduce the size of input instance (in polynomial time) without changing the answer.
- Not all problems (for given parameter) admit a kernel.
- Not all problems (for given parameter) admit a polynomial kernel.
- *Compression* : If (I', k') is an instance of different problem.

FPT and Kernelization

Problem	FPT	Kernel
VERTEX COVER	$\mathcal{O}(1.27^k \cdot n^2)$	$2k^2$
FEEDBACK VERTEX SET	$\mathcal{O}(3.6181^k \cdot n^c)$	$4k^2$
INDEPENDENT SET	No $f(k) \cdot \lvert I\rvert^{\mathcal{O}(1)}$	No $h(k)$
COLORING	No $f(k) \cdot \lvert I\rvert^{\mathcal{O}(1)}$	No $h(k)$

$\mathcal{F}$-Contraction

$\mathcal{F}$-CONTRACTION **Parameter:** k

Input: A graph G and integer k

Question: Does there exist $F \subseteq E(G)$ of size at most k such that G/F is in $\mathcal{F}$?

$\mathcal{F}$-Contraction : Parameterized Complexity

[Heggernes et al., 2012]	TREE CONTRACTION PATH CONTRACTION	4^k $2^{k+o(k)}$

$\mathcal{F}$-Contraction : Parameterized Complexity

[Heggernes et al., 2012]	Tree Contraction Path Contraction	4^k $2^{k+o(k)}$
[Golovach et al., 2013]	Planar Contraction	FPT
[Cai and Guo, 2013]	Clique Contraction	$2^{\mathcal{O}(k \log k)}$
[Heggernes et al., 2013] [Guillemot and Marx, 2013]	Bipartite Contraction	FPT $2^{\mathcal{O}(k^2)}$

$\mathcal{F}$-Contraction : Parameterized Complexity

Theorem

$\mathcal{F}$-Edge Contraction *is* $W[2]$-*hard if*

$\mathcal{F}$-Contraction : Parameterized Complexity

Theorem

$\mathcal{F}$-Edge Contraction *is* $W[2]$*-hard if*

- *[Lokshtanov et al., 2013] [Cai and Guo, 2013]* $\mathcal{F}$ *can be characterized as* $P_{\ell+1}$*-free graphs or* C_ℓ*-free graphs for* $\ell \geq 4$

$\mathcal{F}$-Contraction : Parameterized Complexity

Theorem

$\mathcal{F}$-Edge Contraction *is* $W[2]$*-hard if*

- *[Lokshtanov et al., 2013] [Cai and Guo, 2013]* $\mathcal{F}$ *can be characterized as* $P_{\ell+1}$*-free graphs or* C_ℓ*-free graphs for* $\ell \geq 4$
- *[Agrawal et al., 2017]* $\mathcal{F}$ *is Split Graphs*

Starting Point

Theorem

[Heggernes et al., 2012] TREE CONTRACTION *does not admit a polynomial kernel unless NP $\subseteq$ coNP/poly and* PATH CONTRACTION *admits a linear vertex kernel.*

Starting Point

Theorem

[Heggernes et al., 2012] TREE CONTRACTION *does not admit a polynomial kernel unless NP* $\subseteq$ *coNP/poly and* PATH CONTRACTION *admits a linear vertex kernel.*

Why is there a polynomial kernel for PATHS *but not for* TREES*?*

Tree Contraction with additional parameter

Tree Contraction with additional parameter

Bounded Tree Contraction **Parameter:** $k + \ell$

Input: A graph G and integers k, ℓ

Question: Does there exist $F \subseteq E(G)$ of size at most k such that G/F is a tree with at most ℓ leaves?

Tree Contraction with additional parameter

BOUNDED TREE CONTRACTION **Parameter:** $k + \ell$

Input: A graph G and integers k, ℓ

Question: Does there exist $F \subseteq E(G)$ of size at most k such that G/F is a tree with at most ℓ leaves?

Results:

- A kernel with $\mathcal{O}(k\ell)$ vertices and $\mathcal{O}(k^2 + k\ell)$ edges.

Tree Contraction with additional parameter

BOUNDED TREE CONTRACTION	**Parameter:** $k + \ell$

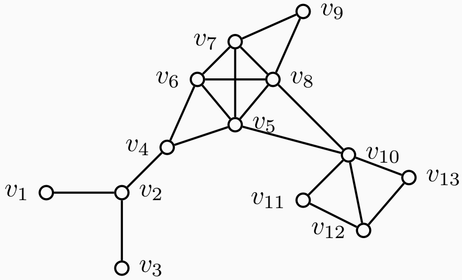
Input: A graph G and integers k, ℓ

Question: Does there exist $F \subseteq E(G)$ of size at most k such that G/F is a tree with at most ℓ leaves?

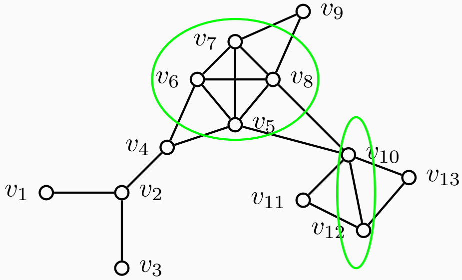
Results:

- A kernel with $\mathcal{O}(k\ell)$ vertices and $\mathcal{O}(k^2 + k\ell)$ edges.
- It does not admit better kernel unless NP $\subseteq$ coNP/poly.

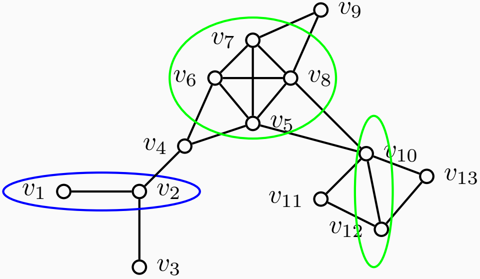
Example : $k = 5, \ell = 4$



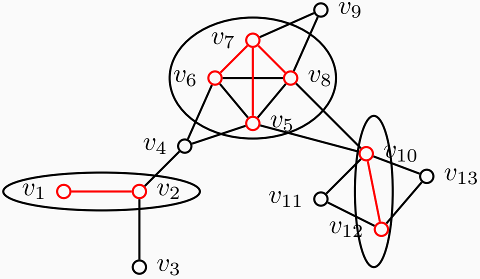
Example : $k = 5, \ell = 4$

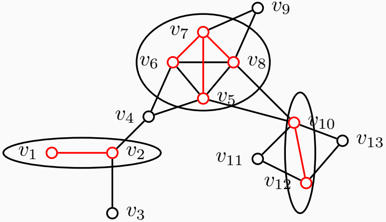


Example : $k = 5, \ell = 4$

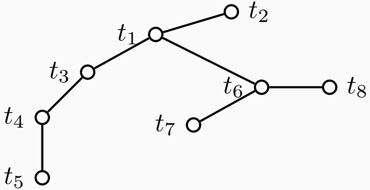


Example : $k = 5, \ell = 4$





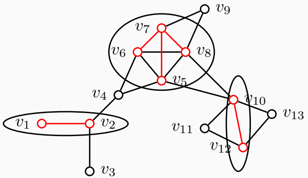
v_9
v_7
v_6
v_8
v_5
v_4
v_{10}
v_{13}
v_1
v_2
v_{11}
v_{12}
v_3

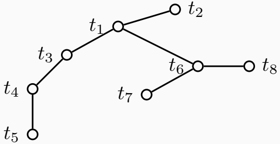


t_2
t_1
t_3
t_6
t_8
t_4
t_7
t_5

Contraction as a Partition Problem

$\mathcal{F}$-Contraction as a Partition Problem

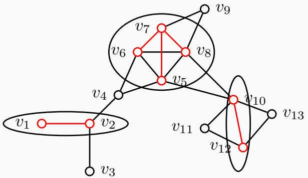


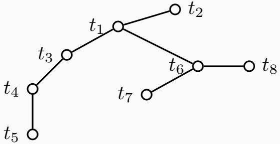


G is contractible to T if there exists a partition of $V(G)$ into $W(t_1), W(t_2), \ldots W(t_{|V(T)|})$ s.t.

- $\forall\ t \in V(T)$, $G[W(t)]$ is connected
- $t_i t_j \in E(T)$ iff $W(t_i)$ and $W(t_j)$ are adjacent in G

$\mathcal{F}$-Contraction as a Partition Problem





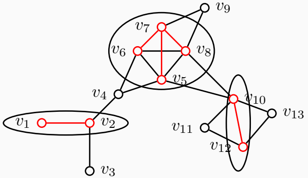
$W(t_1) = \{v_5, v_6, v_7, v_8\}$ $\quad$ $W(t_2) = \{v_9\}$

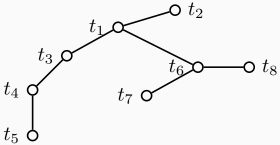
$W(t_3) = \{v_4\}$ $\quad$ $W(t_4) = \{v_1, v_2\}$

$W(t_5) = \{v_3\}$ $\quad$ $W(t_6) = \{v_{10}, v_{12}\}$

$W(t_7) = \{v_{11}\}$ $\quad$ $W(t_8) = \{v_{13}\}$

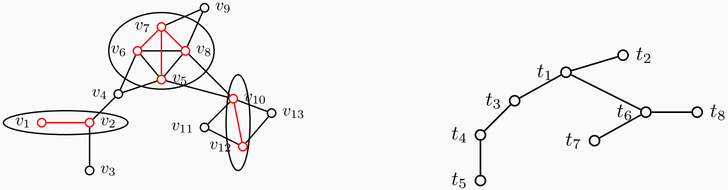
Witness Structure : Definition





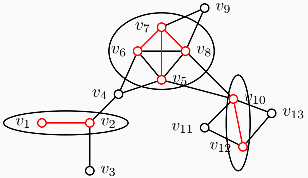
- $\mathcal{W} = \{W(t) \mid t \in V(T)\}$ is called the *T-witness structure* of G

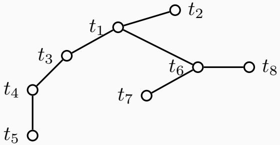
Witness Structure : Definition



- $\mathcal{W} = \{W(t) \mid t \in V(T)\}$ is called the *T-witness structure* of G
- *Big-witness set* if $|W(t)| > 1$ e.g. $W(t_1), W(t_6), W(t_4)$

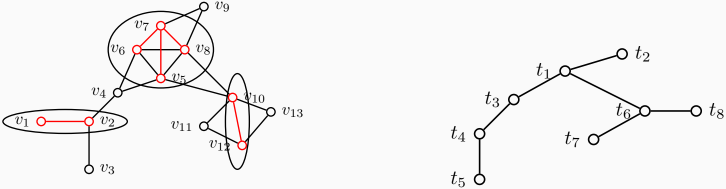
Witness Structure : Definition





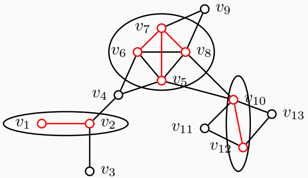
- $\mathcal{W} = \{W(t) \mid t \in V(T)\}$ is called the *T-witness structure* of G
- *Big-witness set* if $|W(t)| > 1$ e.g. $W(t_1), W(t_6), W(t_4)$
- $k = \sum_{t \in V(T)}(|W(t)| - 1)$
  We say G is *k-contractable* to graph T

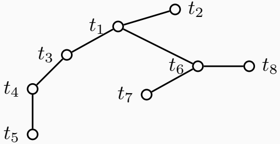
Witness Structure : Observations



If G is k-contractible to H and $\mathcal{W}$ be its H-witness structure then,

Witness Structure : Observations

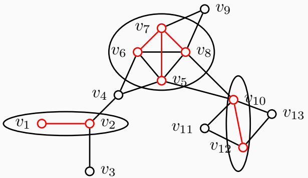


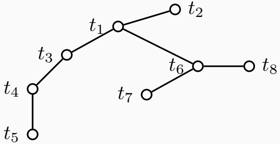


If G is k-contractible to H and $\mathcal{W}$ be its H-witness structure then,

- $|V(G)| \leq |V(H)| + k$;

Witness Structure : Observations

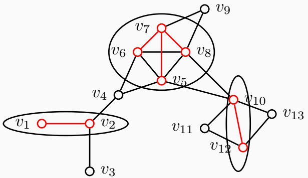


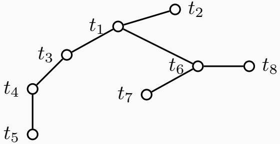


If G is k-contractible to H and $\mathcal{W}$ be its H-witness structure then,

- $|V(G)| \leq |V(H)| + k$;
- No witness set in $\mathcal{W}$ contains more than $k + 1$ vertices;

Witness Structure : Observations

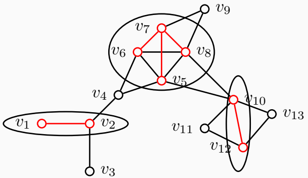


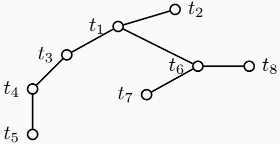


If G is k-contractible to H and $\mathcal{W}$ be its H-witness structure then,

- $|V(G)| \leq |V(H)| + k$;
- No witness set in $\mathcal{W}$ contains more than $k + 1$ vertices;
- $\mathcal{W}$ has at most k big witness sets;

Witness Structure : Observations





If G is k-contractible to H and $\mathcal{W}$ be its H-witness structure then,

- $|V(G)| \leq |V(H)| + k$;
- No witness set in $\mathcal{W}$ contains more than $k + 1$ vertices;
- $\mathcal{W}$ has at most k big witness sets;
- Union of big witness sets in $\mathcal{W}$ contains at most $2k$ vertices.

Properties of $\mathcal{F}$

$$\mathcal{F} = \{T \mid T \text{ is a tree and \#leaves in it is at most } \ell\}$$

Properties of $\mathcal{F}$

$$\mathcal{F} = \{T \mid T \text{ is a tree and \#leaves in it is at most } \ell\}$$

Prop. 1 $\mathcal{F}$ is closed under edge contraction

Properties of $\mathcal{F}$

$$\mathcal{F} = \{T \mid T \text{ is a tree and \#leaves in it is at most } \ell\}$$

Prop. 1 $\mathcal{F}$ is closed under edge contraction

Prop. 2 $\mathcal{F}$ is closed under "uncontracting" internal vertex

Properties of $\mathcal{F}$

$$\mathcal{F} = \{T|\ T \text{ is a tree and \#leaves in it is at most } \ell\}$$

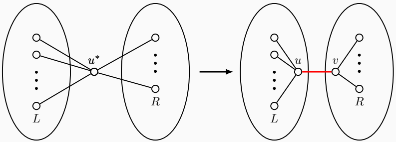
Prop. 1 $\mathcal{F}$ is closed under edge contraction

Prop. 2 $\mathcal{F}$ is closed under "uncontracting" internal vertex

Let $L \cup R$ be a parition of $N(u^*)$.

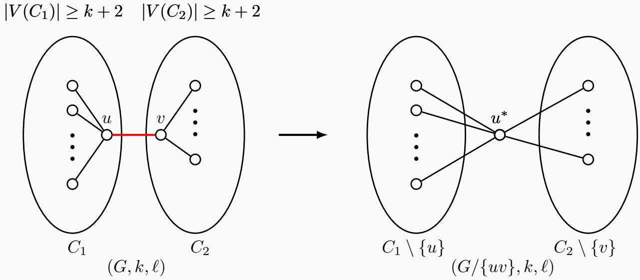
Delete u^* and edge uv s.t. $N(u) = L$ and $N(v) = R$.

Resulting graph is in $\mathcal{F}$



Kernel for Bounded Tree Contraction

Reduction Rule



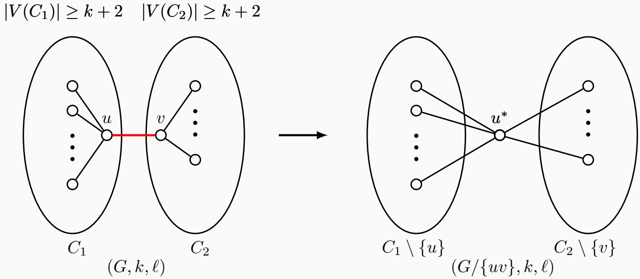
Reduction Rule

Let C_1, C_2 be the connected components in $G - \{uv\}$.

If $|V(C_1)|, |V(C_2)| \geq k+2$ then contract the edge uv.

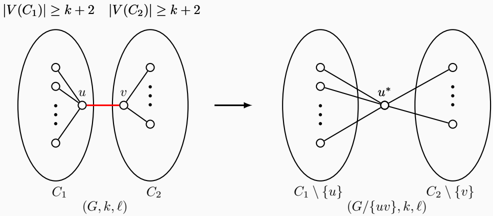
The resulting instance is $(G/\{uv\}, k, \ell)$.

Reduction Rule is sound



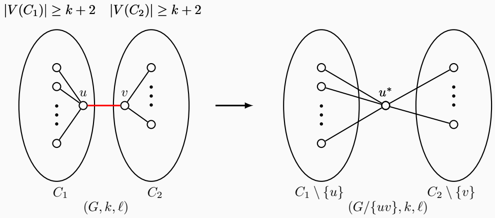
By **Prop 1**, $\mathcal{F}$ is closed under edge contraction

Reduction Rule is complete



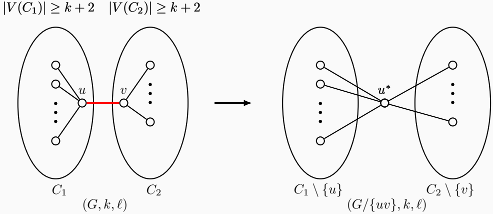
- To Prove : uv is not in any minimal solution of size at most k.
- Assume F is a minimal solution of size at most k and $uv \in F$.
- C_1, C_2 are too big to be contained in a witness set.
- t_{uv} is not a leaf in G/F where $u, v \in W(t_{uv})$.
- By **Prop 2**, *uncontract* node t_{uv} and resulting graph is in $\mathcal{F}$.
- This contradicts the minimality of F.

Reduction Rule is complete



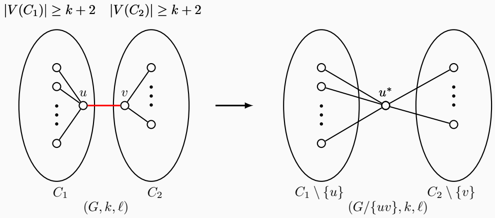
- To Prove : uv is not in any minimal solution of size at most k.
- Assume F is a minimal solution of size at most k and $uv \in F$.
- C_1, C_2 are too big to be contained in a witness set.
- t_{uv} is not a leaf in G/F where $u, v \in W(t_{uv})$.
- By **Prop 2**, *uncontract* node t_{uv} and resulting graph is in $\mathcal{F}$.
- This contradicts the minimality of F.

Reduction Rule is complete



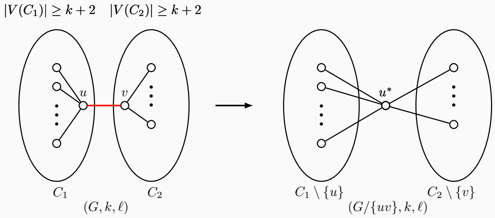
- To Prove : uv is not in any minimal solution of size at most k.
- Assume F is a minimal solution of size at most k and $uv \in F$.
- C_1, C_2 are too big to be contained in a witness set.
- t_{uv} is not a leaf in G/F where $u, v \in W(t_{uv})$.
- By **Prop 2**, *uncontract* node t_{uv} and resulting graph is in $\mathcal{F}$.
- This contradicts the minimality of F.

Reduction Rule is complete



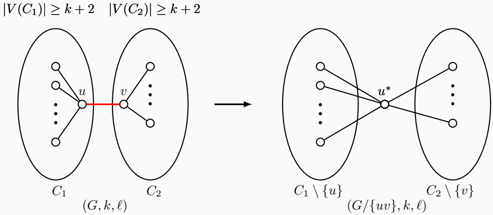
- To Prove : uv is not in any minimal solution of size at most k.
- Assume F is a minimal solution of size at most k and $uv \in F$.
- C_1, C_2 are too big to be contained in a witness set.
- t_{uv} is not a leaf in G/F where $u, v \in W(t_{uv})$.
- By **Prop 2**, *uncontract* node t_{uv} and resulting graph is in $\mathcal{F}$.
- This contradicts the minimality of F.

Reduction Rule is complete



- To Prove : uv is not in any minimal solution of size at most k.
- Assume F is a minimal solution of size at most k and $uv \in F$.
- C_1, C_2 are too big to be contained in a witness set.
- t_{uv} is not a leaf in G/F where $u, v \in W(t_{uv})$.
- By **Prop 2**, *uncontract* node t_{uv} and resulting graph is in $\mathcal{F}$.
- This contradicts the minimality of F.

Reduction Rule is complete



- To Prove : uv is not in any minimal solution of size at most k.
- Assume F is a minimal solution of size at most k and $uv \in F$.
- C_1, C_2 are too big to be contained in a witness set.
- t_{uv} is not a leaf in G/F where $u, v \in W(t_{uv})$.
- By **Prop 2**, *uncontract* node t_{uv} and resulting graph is in $\mathcal{F}$.
- This contradicts the minimality of F.

Kernel for Bounded TC : Bounding $|V(G)|$

- Apply Reduction Rule Exhaustively.
- We bound $V(T)$ and then apply $|V(G)| \leq |V(T)| + k$.
- $|V(T)| \leq \#\text{leaves} \times$ maximum dist. between root and a leaf.
- $P = \{t_1, t_2, \ldots, t_q\}$ be the longest path from root to a leaf.
- If $q \leq 2k + 5$ then $|V(T)| \leq \mathcal{O}(k\ell)$.

Kernel for Bounded TC : Bounding $|V(G)|$

- Apply Reduction Rule Exhaustively.
- We bound $V(T)$ and then apply $|V(G)| \leq |V(T)| + k$.
- $|V(T)| \leq \#$leaves $\times$ maximum dist. between root and a leaf.
- $P = \{t_1, t_2, \ldots, t_q\}$ be the longest path from root to a leaf.
- If $q \leq 2k + 5$ then $|V(T)| \leq \mathcal{O}(k\ell)$.

Kernel for Bounded TC : Bounding $|V(G)|$

- Apply Reduction Rule Exhaustively.
- We bound $V(T)$ and then apply $|V(G)| \leq |V(T)| + k$.
- $|V(T)| \leq$ #leaves $\times$ maximum dist. between root and a leaf.
- $P = \{t_1, t_2, \ldots, t_q\}$ be the longest path from root to a leaf.
- If $q \leq 2k + 5$ then $|V(T)| \leq \mathcal{O}(k\ell)$.

Kernel for Bounded TC : Bounding $|V(G)|$

- Apply Reduction Rule Exhaustively.
- We bound $V(T)$ and then apply $|V(G)| \leq |V(T)| + k$.
- $|V(T)| \leq \#$leaves $\times$ maximum dist. between root and a leaf.
- $P = \{t_1, t_2, \ldots, t_q\}$ be the longest path from root to a leaf.
- If $q \leq 2k + 5$ then $|V(T)| \leq \mathcal{O}(k\ell)$.

Kernel for Bounded TC : Bounding $|V(G)|$

- Apply Reduction Rule Exhaustively.
- We bound $V(T)$ and then apply $|V(G)| \leq |V(T)| + k$.
- $|V(T)| \leq \#\text{leaves} \times$ maximum dist. between root and a leaf.
- $P = \{t_1, t_2, \ldots, t_q\}$ be the longest path from root to a leaf.
- If $q \leq 2k + 5$ then $|V(T)| \leq \mathcal{O}(k\ell)$.

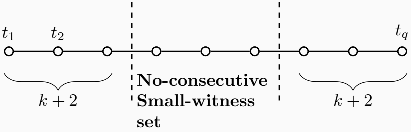
Kernel for Bounded TC : Bounding $|V(G)|$

- If $q > 2k + 5$ then partition into *left, right* and *middle portion*
- No two consecutive small witness set in *middle portion*
- There are at most k big witness set

Kernel for Bounded TC : Bounding $|V(G)|$

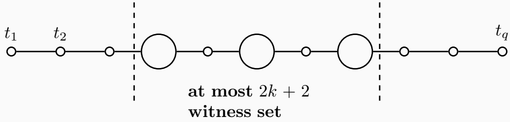
- If $q > 2k + 5$ then partition into *left*, *right* and *middle portion*
- No two consecutive small witness set in *middle portion*
- There are at most k big witness set

Kernel for Bounded TC : Bounding $|V(G)|$



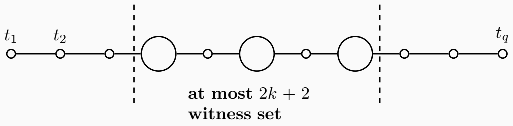
- If $q > 2k + 5$ then partition into *left*, *right* and *middle portion*
- No two consecutive small witness set in *middle portion*
- There are at most k big witness set

Kernel for Bounded TC : Bounding $|V(G)|$



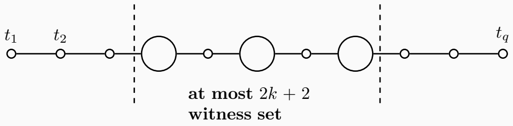
- There are at most k big witness set
- $q \leq 4k + 4$ and $|V(T)| \leq \ell(4k + 4)$.
- $|V(G)|$ is at most $\mathcal{O}(k\ell)$.

Kernel for Bounded TC : Bounding $|V(G)|$



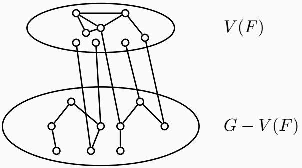
- There are at most k big witness set
- $q \leq 4k + 4$ and $|V(T)| \leq \ell(4k + 4)$.
- $|V(G)|$ is at most $\mathcal{O}(k\ell)$.

Kernel for Bounded TC : Bounding $|V(G)|$

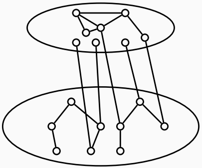


- There are at most k big witness set
- $q \leq 4k + 4$ and $|V(T)| \leq \ell(4k + 4)$.
- $|V(G)|$ is at most $\mathcal{O}(k\ell)$.

Kernel for Bounded TC: Bounding $|E(G)|$

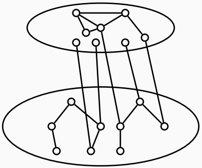


Kernel for Bounded TC: Bounding $|E(G)|$



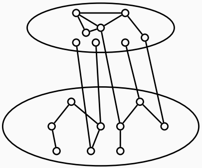
- $G - V(F)$ is a forest and $|V(F)| \leq 2k$

Kernel for Bounded TC: Bounding $|E(G)|$



- $G - V(F)$ is a forest and $|V(F)| \leq 2k$
- $\text{Max\_deg}(T) \leq \ell \Rightarrow \text{Max\_deg}(G) \leq \ell + k$.

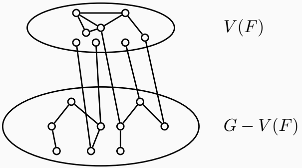
Kernel for Bounded TC: Bounding $|E(G)|$



- $G - V(F)$ is a forest and $|V(F)| \leq 2k$
- Max_deg$(T) \leq \ell \Rightarrow$ Max_deg$(G) \leq \ell + k$.

- \# edges contained in $V(F)$ is $\mathcal{O}(k^2)$

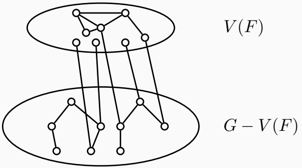
Kernel for Bounded TC: Bounding $|E(G)|$



- $G - V(F)$ is a forest and $|V(F)| \leq 2k$
- $\text{Max\_deg}(T) \leq \ell \Rightarrow \text{Max\_deg}(G) \leq \ell + k$.

- # edges contained in $V(F)$ is $\mathcal{O}(k^2)$
- # edges contained in $V(G) \setminus V(F)$ is $\mathcal{O}(k\ell)$

Kernel for Bounded TC: Bounding $|E(G)|$



- $G - V(F)$ is a forest and $|V(F)| \leq 2k$
- $\text{Max\_deg}(T) \leq \ell \Rightarrow \text{Max\_deg}(G) \leq \ell + k$.

- # edges contained in $V(F)$ is $\mathcal{O}(k^2)$
- # edges contained in $V(G) \setminus V(F)$ is $\mathcal{O}(k\ell)$
- # edges across $V(F)$ and $V(G) \setminus V(F)$ is $\mathcal{O}(k^2 + k\ell)$

Kernel for Bounded TC

- Apply Reduction Rule exhaustively.

Kernel for Bounded TC

- Apply Reduction Rule exhaustively.
- Each application of Reduction Rule contracts an edge and hence it can be applied at most m many times.

Kernel for Bounded TC

- Apply Reduction Rule exhaustively.
- Each application of Reduction Rule contracts an edge and hence it can be applied at most m many times.
- Get *reduced* instance in polynomial time.

Kernel for Bounded TC

- Apply Reduction Rule exhaustively.
- Each application of Reduction Rule contracts an edge and hence it can be applied at most m many times.
- Get *reduced* instance in polynomial time.
- If vertices or edges are not bounded, return a NO instance.

Kernel for Bounded TC

- Apply Reduction Rule exhaustively.
- Each application of Reduction Rule contracts an edge and hence it can be applied at most m many times.
- Get *reduced* instance in polynomial time.
- If vertices or edges are not bounded, return a NO instance.

Kernel for Bounded TC

- Apply Reduction Rule exhaustively.
- Each application of Reduction Rule contracts an edge and hence it can be applied at most m many times.
- Get *reduced* instance in polynomial time.
- If vertices or edges are not bounded, return a NO instance.

Theorem

BOUNDED TREE CONTRACTION *admits a kernel of size* $\mathcal{O}(k^2 + k\ell)$.

Extending to other problem

Definition

An *out-tree* T is a digraph where each vertex has in-degree at most 1 and underlying undirected graph is a tree.

Extending to other problem

Definition

An *out-tree* T is a digraph where each vertex has in-degree at most 1 and underlying undirected graph is a tree.

- Root has in-degree 0; all directed paths are away from the root.

Extending to other problem

Definition

An *out-tree* T is a digraph where each vertex has in-degree at most 1 and underlying undirected graph is a tree.

- Root has in-degree 0; all directed paths are away from the root.
- Similar algorithm with adapation of witness structure to directed graph.

Extending to other problem

Definition

An *out-tree* T is a digraph where each vertex has in-degree at most 1 and underlying undirected graph is a tree.

- Root has in-degree 0; all directed paths are away from the root.
- Similar algorithm with adapation of witness structure to directed graph.

Extending to other problem

Definition

An *out-tree* T is a digraph where each vertex has in-degree at most 1 and underlying undirected graph is a tree.

- Root has in-degree 0; all directed paths are away from the root.
- Similar algorithm with adapation of witness structure to directed graph.

Theorem

Bounded Out-Tree Contraction *admits a kernel of size* $\mathcal{O}(k^2 + k\ell)$.

Extending to other problem

Definition

A *cactus* is an undirected graph such that every edge is contained in at most one cycle.

Extending to other problem

Definition

A *cactus* is an undirected graph such that every edge is contained in at most one cycle.

- *block* : A maximal 2-connected subgraph.

Extending to other problem

Definition

A *cactus* is an undirected graph such that every edge is contained in at most one cycle.

- *block* : A maximal 2-connected subgraph.
- *Block decomposition* of a connected graph is unique and is a tree.

Extending to other problem

Definition

A *cactus* is an undirected graph such that every edge is contained in at most one cycle.

- *block* : A maximal 2-connected subgraph.
- *Block decomposition* of a connected graph is unique and is a tree.
- A block in a cactus can be either a cycle or an edge.

Extending to other problem

Definition

A *cactus* is an undirected graph such that every edge is contained in at most one cycle.

- *block* : A maximal 2-connected subgraph.
- *Block decomposition* of a connected graph is unique and is a tree.
- A block in a cactus can be either a cycle or an edge.
- # *leaves* in a cactus is defined to be # leaves in its block decomposition.

Extending to other problem

Definition

A *cactus* is an undirected graph such that every edge is contained in at most one cycle.

- *block* : A maximal 2-connected subgraph.
- *Block decomposition* of a connected graph is unique and is a tree.
- A block in a cactus can be either a cycle or an edge.
- # *leaves* in a cactus is defined to be # leaves in its block decomposition.

Extending to other problem

Definition

A *cactus* is an undirected graph such that every edge is contained in at most one cycle.

- *block* : A maximal 2-connected subgraph.
- *Block decomposition* of a connected graph is unique and is a tree.
- A block in a cactus can be either a cycle or an edge.
- # *leaves* in a cactus is defined to be # leaves in its block decomposition.

Theorem

Bounded Cactus Contraction *admits a kernel of size* $\mathcal{O}(k^2 + k\ell)$.

Kernel Lower Bounds

DOMINATING SET **Parameter:** k

Input: Graph G and integer k

Question: Does there exists $X \subseteq V(G)$ of size at most k, such that for each $v \in V(G)$, $X \cap N[v] \neq \emptyset$?

Kernel Lower Bounds

DOMINATING SET	**Parameter:** k

Input: Graph G and integer k

Question: Does there exists $X \subseteq V(G)$ of size at most k, such that for each $v \in V(G)$, $X \cap N[v] \neq \emptyset$?

[Jansen and Pieterse, 2015] proved that DOMINATING SET does not admit a compression of bit size $\mathcal{O}(n^{2-\epsilon})$, for any $\epsilon > 0$ unless NP $\subseteq$ coNP/poly.

Kernel Lower Bounds

Red-Blue Dominating Set **Parameter:** k

Input: Bipartite Graph $G := (R \cup B; E)$ and integer k

Question: Does there exists $X \subseteq R$ of size at most k, such that for each $v \in B$, $X \cap N[v] \neq \emptyset$?

Kernel Lower Bounds

RED-BLUE DOMINATING SET	**Parameter:** k

Input: Bipartite Graph $G := (R \cup B; E)$ and integer k
Question: Does there exists $X \subseteq R$ of size at most k, such that for each $v \in B$, $X \cap N[v] \neq \emptyset$?

By [Jansen and Pieterse, 2015]; RED-BLUE DOMINATING SET does not admit a polynomial compression of bit size $\mathcal{O}(n^{2-\epsilon})$, for any $\epsilon > 0$ unless NP $\subseteq$ coNP/poly.

Kernel Lower Bounds

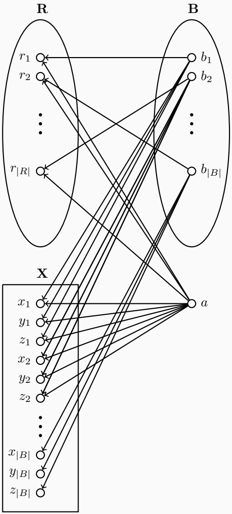


Figure 1: From RBDS to BOUNDED TC

Kernel Lower Bounds

Theorem

BOUNDED TC *does not admit a compression of size* $\mathcal{O}((k^2 + k\ell)^{1-\epsilon})$*, for any* $\epsilon > 0$.

Theorem

BOUNDED CC *does not admit a compression of size* $\mathcal{O}((k^2 + k\ell)^{1-\epsilon})$*, for any* $\epsilon > 0$.

Theorem

BOUNDED OTC *does not admit a compression of size* $\mathcal{O}((k^2 + k\ell)^{1-\epsilon})$*, for any* $\epsilon > 0$.

Thank you!

References

Agrawal, A., Lokshtanov, D., Saurabh, S., and Zehavi, M. (2017).
Split contraction: The untold story.
In *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, Hannover, Germany*, pages 5:1–5:14.

Cai, L. and Guo, C. (2013).
Contracting few edges to remove forbidden induced subgraphs.
In *IPEC*, pages 97–109.

Golovach, P. A., van 't Hof, P., and Paulusma, D. (2013).
Obtaining planarity by contracting few edges.
Theoretical Computer Science, 476:38–46.

Guillemot, S. and Marx, D. (2013).
A faster FPT algorithm for bipartite contraction.