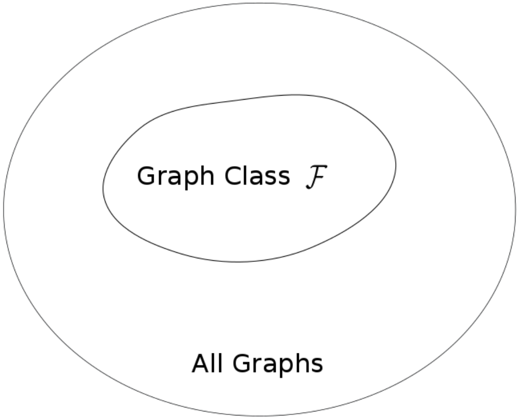
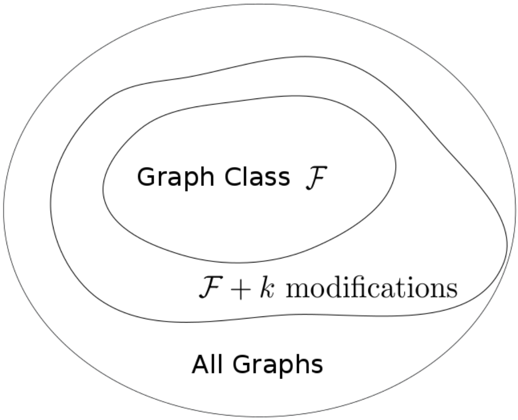
An FPT Algorithm for Contraction to Cactus

R. Krithika[1] Pranabendu Misra [2] and Prafullkumar Tale[1]

3^{rd} July, 2018

[1] The Institute of Mathematical Sciences, HBNI, Chennai, India
[2] University Of Bergen, Bergen, Norway


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	Empty graphs	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

Graph Contraction Problems

Problem Definition

FPT Algorithm

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] $\ldots$

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 : 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] $\ldots$

Graph Contraction Problems

Graph Contraction Problems

$\mathcal{F}$ is a graph class and G/F is graph obtained from G by contracting edges in F

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

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

$\mathcal{F}$ is polynomial time recongnizable graph class.

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

[HvtHL+12]	Tree Contraction	4^k
	Path Contraction	$2^{k+o(k)}$
[GvtHP13]	Planar Contraction	FPT
[CG13]	Clique Contraction	$2^{\mathcal{O}(k \log k)}$
[HvtHLP13]	Bipartite Contraction	FPT
[GM13]		$2^{\mathcal{O}(k^2)}$
[GKPT13]	$\mathcal{F}_{mindeg \geq d}$Contraction	FPT (k, d)

$\mathcal{F}$-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*

- *[LMS13] [CG13]* $\mathcal{F}$ *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*

- *[LMS13] [CG13]* $\mathcal{F}$ *can be characterized as* $P_{\ell+1}$*-free graphs or* C_ℓ*-free graphs for* $\ell \geq 4$.

  P_ℓ *and* C_ℓ *are path and cycle on* ℓ *vertices, respectively.*

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

Theorem

$\mathcal{F}$-EDGE CONTRACTION *is* $W[2]$*-hard if*

- *[LMS13] [CG13]* $\mathcal{F}$ *can be characterized as* $P_{\ell+1}$*-free graphs or* C_ℓ*-free graphs for* $\ell \geq 4$*.*

  P_ℓ *and* C_ℓ *are path and cycle on* ℓ *vertices, respectively.*
- *[ALSZ17]* $\mathcal{F}$ *is Split Graphs*

Starting Point

$\mathcal{F}$-Contraction is FPT when $\mathcal{F}$ is Trees (which are C_3-free) but $W[2]$-hard when $\mathcal{F}$ is family of C_ℓ-free graphs ($\ell \geq 4$).

Starting Point

$\mathcal{F}$-Contraction is FPT when $\mathcal{F}$ is Trees (which are C_3-free) but $W[2]$-hard when $\mathcal{F}$ is family of C_ℓ-free graphs ($\ell \geq 4$).

What are the superclasses of Tree which admits an FPT algorithm?

Problem Definition

Cactus Contraction

Cactus : if every edge is a part of at most one simple cycle.

Cactus Contraction

Cactus : if every edge is a part of at most one simple cycle.

CACTUS CONTRACTION **Parameter:** k

Input: A graph G and an integer k.

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

Cactus Contraction

Cactus : if every edge is a part of at most one simple cycle.

CACTUS CONTRACTION	**Parameter:** k
Input: A graph G and an integer k.	
Question: Does there exist $F \subseteq E(G)$ of size at most k such that G/F is a cactus graph?	

Result:

Cactus Contraction

Cactus : if every edge is a part of at most one simple cycle.

CACTUS CONTRACTION	**Parameter:** k

Input: A graph G and an integer k.

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

Result:

- FPT algorithm running in time $\mathcal{O}(c^k \cdot n^{\mathcal{O}(1)})$ for a 2-connected graph which can be generalized for general graph by working on each of its 2-connected components.

Cactus Contraction

Cactus : if every edge is a part of at most one simple cycle.

CACTUS CONTRACTION	**Parameter:** k

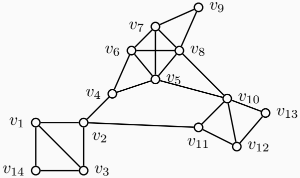
Input: A graph G and an integer k.
Question: Does there exist $F \subseteq E(G)$ of size at most k such that G/F is a cactus graph?

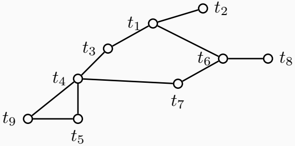
Result:

- FPT algorithm running in time $\mathcal{O}(c^k \cdot n^{\mathcal{O}(1)})$ for a 2-connected graph which can be generalized for general graph by working on each of its 2-connected components.
  Every graph mentioned here is 2-connected.

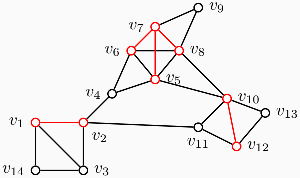
Contraction as a Partition Problem

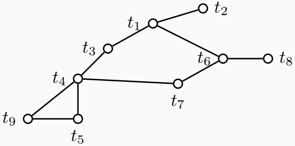
$\mathcal{F}$-Contraction as a Partition Problem



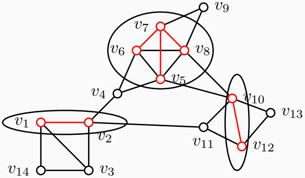


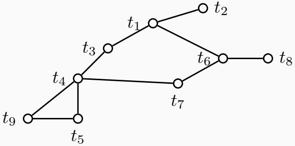
$\mathcal{F}$-Contraction as a Partition Problem





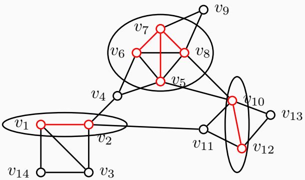
$\mathcal{F}$-Contraction as a Partition Problem

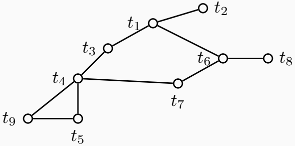




G is contractible to T if there exists a partition of $V(G)$ into

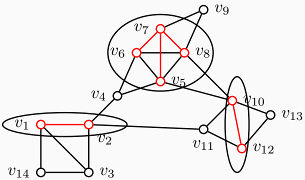
Witness Structure : Definition

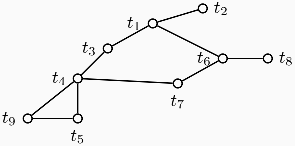




- $\mathcal{W} = \{W(t) \mid t \in V(T)\}$ is called the *T-witness structure* of G

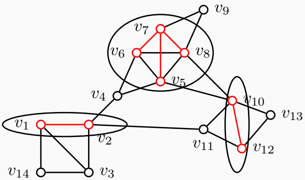
Witness Structure : Definition

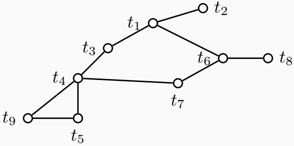




- $\mathcal{W} = \{W(t) \mid t \in V(T)\}$ is called the *T-witness structure* of G

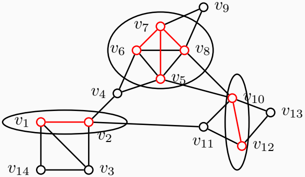
Witness Structure : Definition

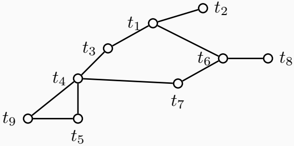




- $\mathcal{W} = \{W(t) \mid t \in V(T)\}$ is called the *T-witness structure* of G

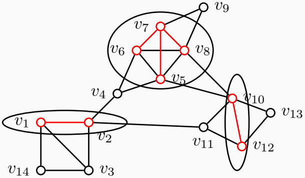
Witness Structure : Observations

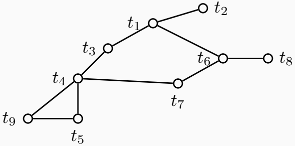




If G is k-contractible to T and $\mathcal{W}$ be its T-witness structure then,

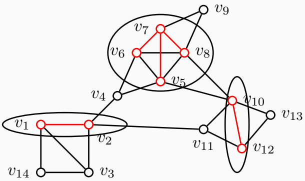
Witness Structure : Observations

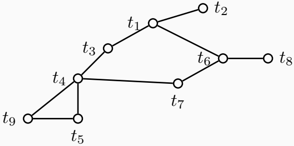




If G is k-contractible to T and $\mathcal{W}$ be its T-witness structure then,

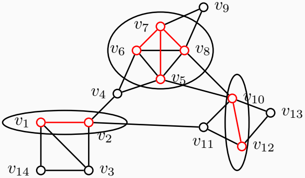
Witness Structure : Observations

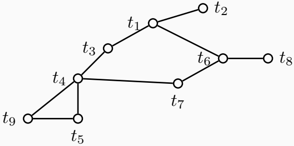




If G is k-contractible to T and $\mathcal{W}$ be its T-witness structure then,

Witness Structure : Observations





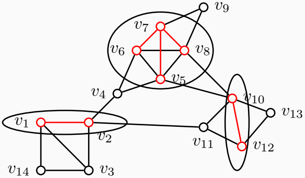
If G is k-contractible to T and $\mathcal{W}$ be its T-witness structure then,

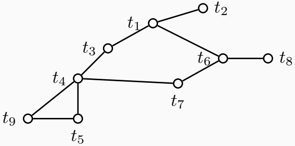
FPT Algorithm

Properties of a cactus T

1. The vertices of T can be properly colored using 3 colors.
2. Every vertex of degree at least 3 is a cut-vertex.
3. The graph obtained from T by subdividing any edge is a cactus.
4. The graph obtained from T by short-circuiting any degree 2 vertex is a cactus.

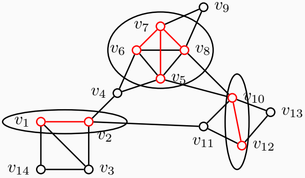
Few Definitions

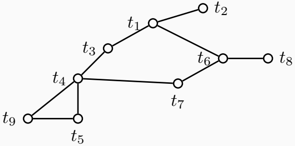




Important Nodes in T

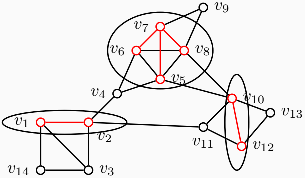
Few Definitions

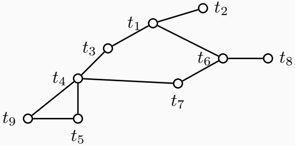




Important Nodes in T

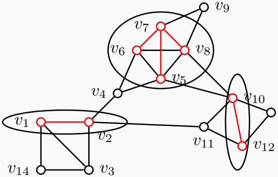
Few Definitions

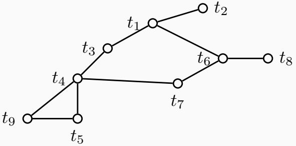




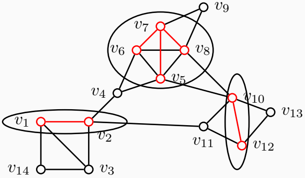
Important Nodes in T

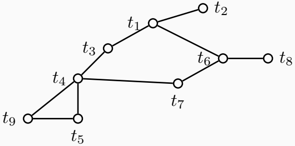
Few Definitions





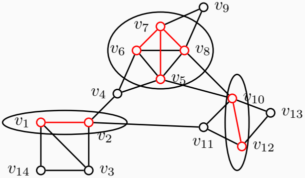
Few Definitions

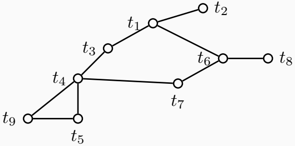




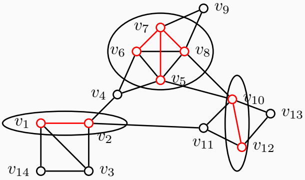
Important vertices in G are the vertices contained in bags

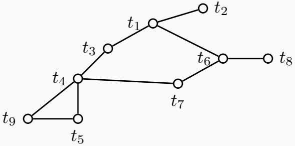
Few Definitions





Important vertices in G are the vertices contained in bags

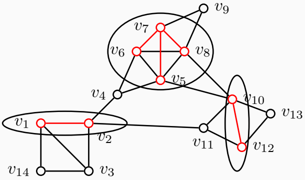


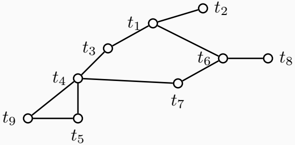


Lemma

There are at most 6k important vertices.

At most $2k$ important vertices contained in big bags (Ex $\{v_1, v_2\}$)

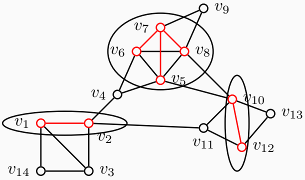


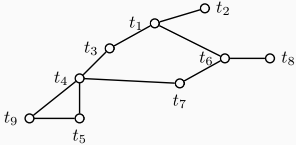


Lemma

There are at most $6k$ important vertices.

At most $2k$ important vertices contained in big bags (Ex $\{v_1, v_2\}$)

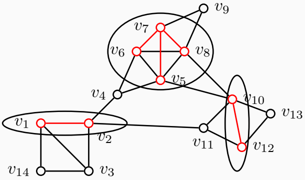


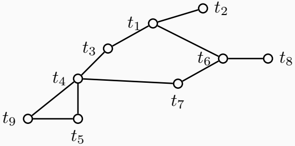


Lemma

There are at most $6k$ important vertices.

Short-circuit all non important terminals

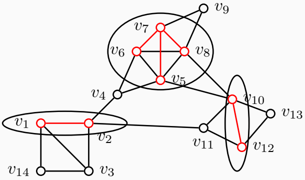


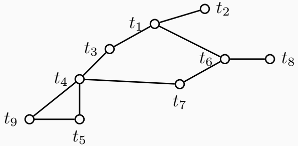


Lemma

There are at most 6k important vertices.

Short-circuit all non important terminals

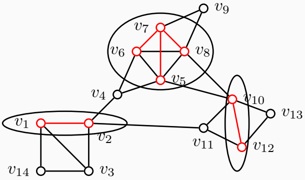




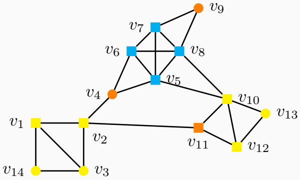
Lemma

There are at most 6k important vertices.

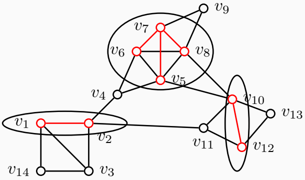
Short-circuit all non important terminals

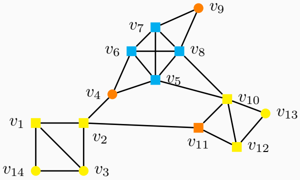


v_9
v_7
v_6
v_8
v_5
v_4
v_{10}
v_{13}
v_1
v_2
v_{11}
v_{12}
v_{14}
v_3

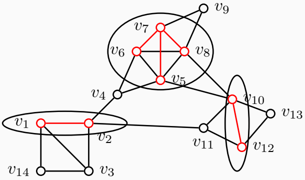


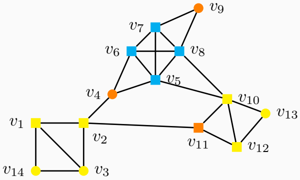
v_9
v_7
v_6
v_8
v_5
v_4
v_{10}
v_{13}
v_1
v_2
v_{11}
v_{12}
v_{14}
v_3





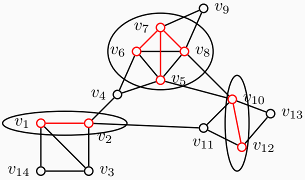
Compatible Coloring $\phi : V(G) \to [3]$ is compatible (wrt $\mathcal{W}$) if

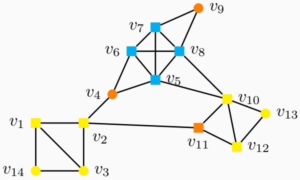




Compatible Coloring $\phi : V(G) \to [3]$ is compatible (wrt $\mathcal{W}$) if

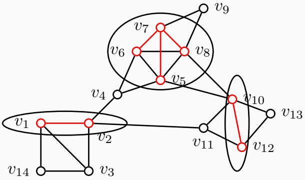
1. Each witness set is monochromatic (Ex. $\{v_5, v_6, v_7, v_8\}$)

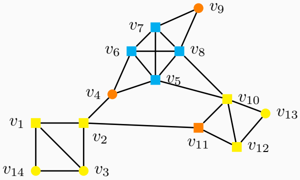




Compatible Coloring $\phi : V(G) \to [3]$ is compatible (wrt $\mathcal{W}$) if

1. Each witness set is monochromatic (Ex. $\{v_5, v_6, v_7, v_8\}$)

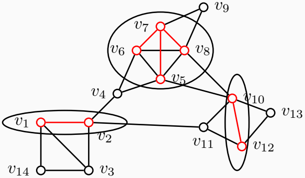


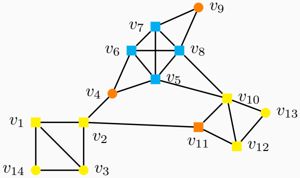


Compatible Coloring $\phi : V(G) \to [3]$ is compatible (wrt $\mathcal{W}$) if

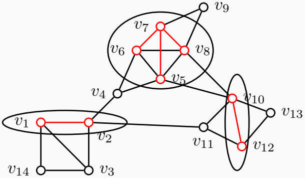
1. Each witness set is monochromatic (Ex. $\{v_5, v_6, v_7, v_8\}$)

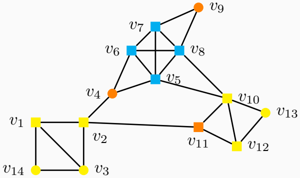
Randomized FPT Algorithm (Big Picture)





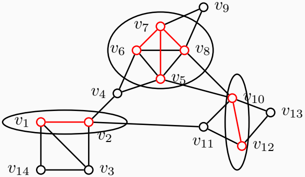
Randomized FPT Algorithm (Big Picture)

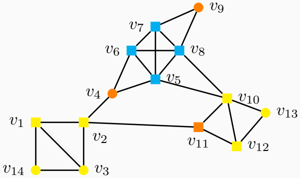




Step 1: Color vertices of input graph uniformly at random with 3

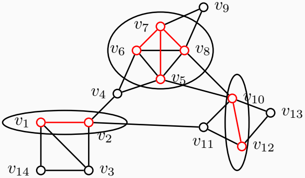
Randomized FPT Algorithm (Big Picture)

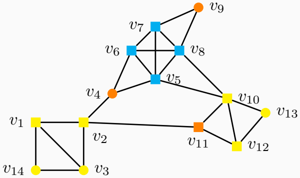




Step 1: Color vertices of input graph uniformly at random with 3

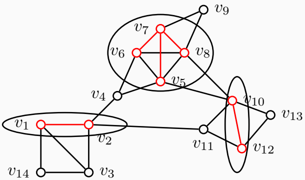
Randomized FPT Algorithm (Big Picture)

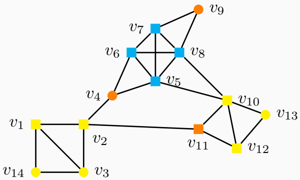




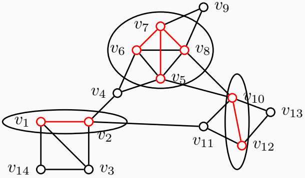
Step 1: Color vertices of input graph uniformly at random with 3

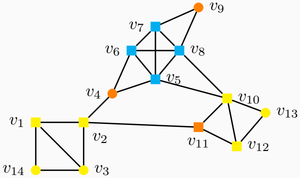
Properties of Compatible Coloring





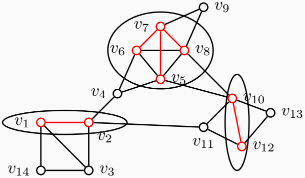
Properties of Compatible Coloring

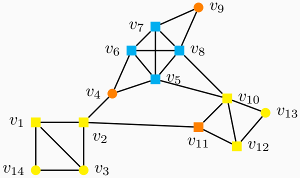




$X \subseteq V(G)$ a colored component of

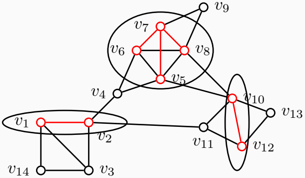
Properties of Compatible Coloring

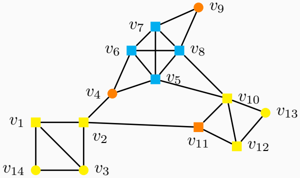




$X \subseteq V(G)$ a colored component of

Properties of Compatible Coloring





$X \subseteq V(G)$ a colored component of

Properties of Compatible Coloring

For every color component X in $\mathcal{X}$:

- all vertices of X are in small bags in $\mathcal{W}$, OR
- X contains exactly one big witness set and the remaining vertices $X \setminus W(t)$ are in small bags.

Properties of Compatible Coloring

For every color component X in $\mathcal{X}$:

- all vertices of X are in small bags in $\mathcal{W}$, OR
- X contains exactly one big witness set and the remaining vertices $X \setminus W(t)$ are in small bags.

Given a coloring ϕ, we are only interested in finding an optimum solution which is compatible with this coloring.

Properties of Compatible Coloring

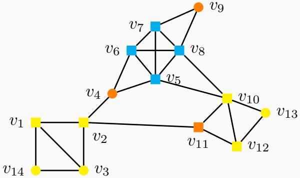
For every color component X in $\mathcal{X}$:

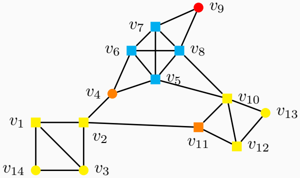
- all vertices of X are in small bags in $\mathcal{W}$, OR
- X contains exactly one big witness set and the remaining vertices $X \setminus W(t)$ are in small bags.

Given a coloring ϕ, we are only interested in finding an optimum solution which is compatible with this coloring.

Hence, for any two components X, Y of ϕ, no edge uv in $E(X, Y)$ is in optimum solution.

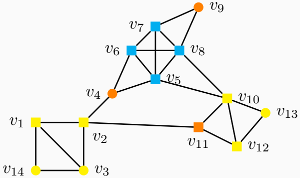
Identifying Vertices in Pendant Cycles and Leaves

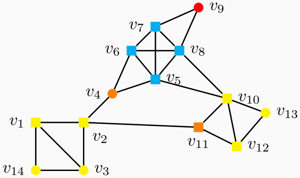




Re-coloring I: For any colored component X in $\mathcal{X}$, if $G - X$

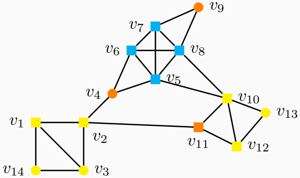
Identifying Vertices in Pendant Cycles and Leaves

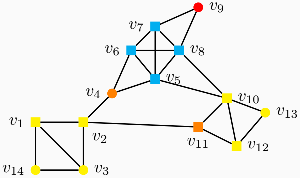




Re-coloring I: For any colored component X in $\mathcal{X}$, if $G - X$

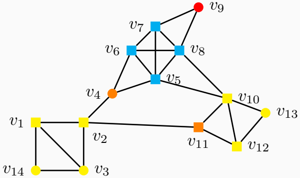
Identifying Vertices in Pendant Cycles and Leaves

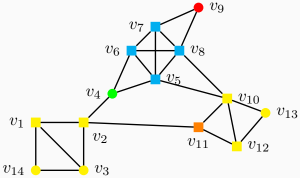




Lemma

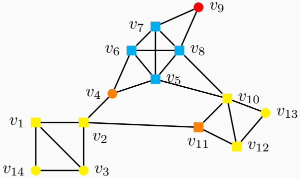
Identifying Vertices in Between Two Big bags

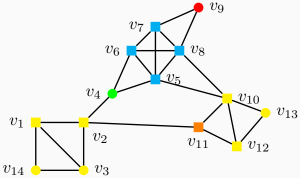




Re-coloring II: For any two colored component Y, Z in $\mathcal{X}$, if

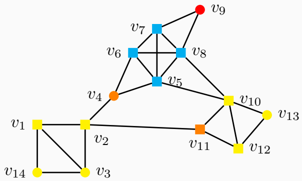
Identifying Vertices in Between Two Big bags

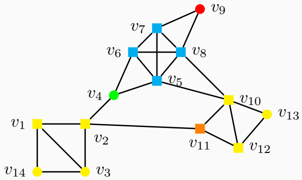




Re-coloring II: For any two colored component Y, Z in $\mathcal{X}$, if

Identifying Vertices in Between Two Big bags





Lemma

Step 2: Extract witness sets out of each colored components of a *compatible* coloring.

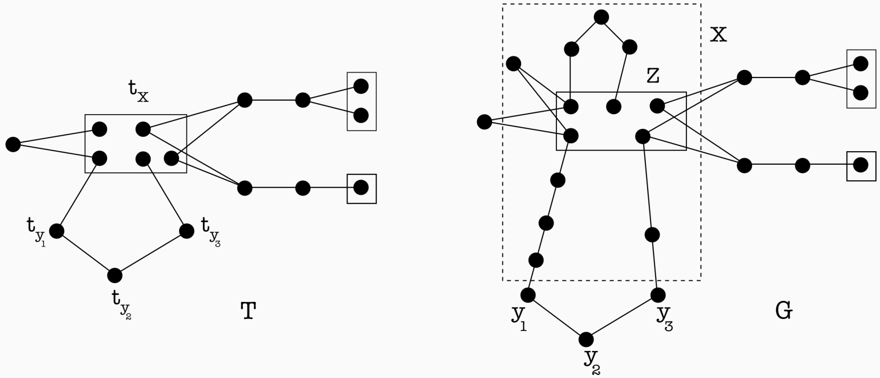
Step 2: Extract witness sets out of each colored components of a *compatible* coloring.

At this stage, every colored component X contains exactly one big witness set $W(t)$

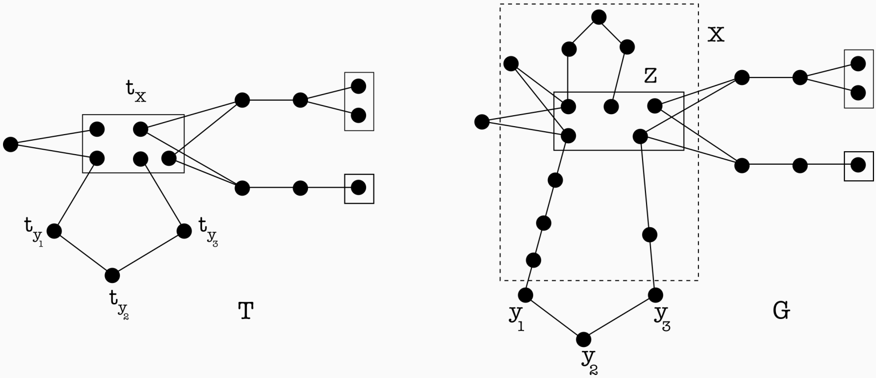
Step 2: Extract witness sets out of each colored components of a *compatible* coloring.

At this stage, every colored component X contains exactly one big witness set $W(t)$ but we don't know it explicitly.

Extract witness sets

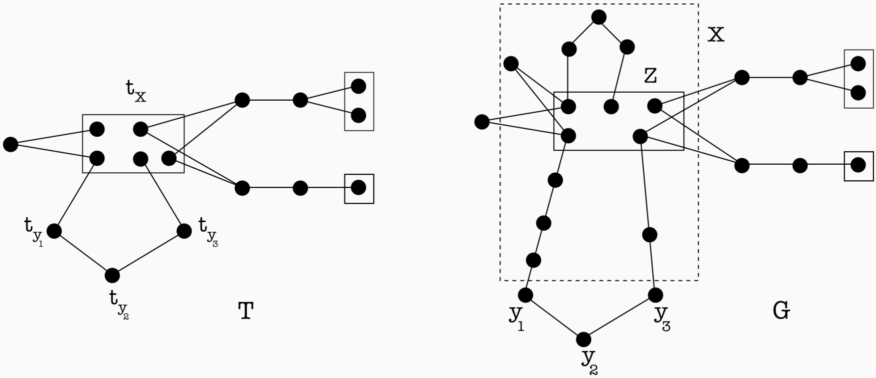


Extract witness sets



Instead of contracting X to t_X, we contract subset Z of X to t_Z.

Extract witness sets



Instead of contracting X to t_X, we contract subset Z of X to t_Z.

We argue that contracting Z is as good as contracting $W(t)$.

Extract witness sets

To argue that contracting Z is as good as contracting $W(t)$, we need to prove:

1. $|Z| \leq |W(t)|$
2. Characterize $W(t)$ in graph $G[X]$ and ensure that set Z has that property.

Once we identify such Z in X, we replace X by Z and singleton set for every vertex in $X \setminus Z$ in $\mathcal{X}$.

Characterize $W(t)$ in graph $G[X]$

A ***core*** of a graph G is a set $Z \subseteq V(G)$ such that every connected component of $G - Z$ is either an isolated vertex or a simple path whose neighborhood is contained in Z.

Characterize $W(t)$ in graph $G[X]$

A ***core*** of a graph G is a set $Z \subseteq V(G)$ such that every connected component of $G - Z$ is either an isolated vertex or a simple path whose neighborhood is contained in Z.

If a core Z is a connected set in G, then we call it a *connected core* of G.

Characterize $W(t)$ in graph $G[X]$

A ***core*** of a graph G is a set $Z \subseteq V(G)$ such that every connected component of $G - Z$ is either an isolated vertex or a simple path whose neighborhood is contained in Z.

If a core Z is a connected set in G, then we call it a *connected core* of G.

Lemma

For a colored component X in $\mathcal{X}$, if $W(t)$ is the big witness set contained in X then $W(t)$ is a connected core of $G[\hat{X}]$.

$\hat{X}$ is superset of X which contains vertices in the connected components of $G - X$ that are either isolated vertices or a simple path in G.

Characterize $W(t)$ in graph $G[X]$

There may be a proper superset of $W(t)$ which is a connected core of $G[\hat{X}]$.

Characterize $W(t)$ in graph $G[X]$

There may be a proper superset of $W(t)$ which is a connected core of $G[\hat{X}]$.

Every vertex in $W(t)$ has at least one of the two responsibility:

Characterize $W(t)$ in graph $G[X]$

There may be a proper superset of $W(t)$ which is a connected core of $G[\hat{X}]$.

Every vertex in $W(t)$ has at least one of the two responsibility:

- part of connected core of $G[\hat{X}]$
- it is in $W(t)$ because of *external constraints*.

Characterize $W(t)$ in graph $G[X]$

There may be a proper superset of $W(t)$ which is a connected core of $G[\hat{X}]$.

Every vertex in $W(t)$ has at least one of the two responsibility:

- part of connected core of $G[\hat{X}]$
- it is in $W(t)$ because of *external constraints*.

We introduce **Marking Scheme** to mark vertices which are in $W(t)$ because of external constraints.

Characterize $W(t)$ in graph $G[X]$

Marking-Scheme For a colored component X in $\mathcal{X}$,

1. If there exists y in $N(X)$ such that $\phi(y) = 5$ then mark all the vertices in $N(y) \cap X$.
2. For a colored component X' in $\mathcal{X}$ which contains a big witness set, mark all vertices in $N(X') \cap X$

Characterize $W(t)$ in graph $G[X]$

Marking-Scheme For a colored component X in $\mathcal{X}$,

1. If there exists y in $N(X)$ such that $\phi(y) = 5$ then mark all the vertices in $N(y) \cap X$.
2. For a colored component X' in $\mathcal{X}$ which contains a big witness set, mark all vertices in $N(X') \cap X$

Lemma

Every marked vertex in X is in big witness set $W(t)$.

Finding Connected Core

Lemma

Given a connected graph G and a subset Q of its vertices, one can compute a minimum connected core of G which has at most k vertices and contains Q in $\mathcal{O}^*(6^k)$ time if it exists.

FPT Algorithm

Input: A *2-connected graph* G and an integer k

Output: A set F of k edges in G such that G/F is a cactus

Generate random coloring $\phi : V(G) \rightarrow \{1, 2, 3\}$ and construct $\mathcal{X}$.

Recoloring I, II.

for *each* $X \in \mathcal{X}$ **do**

  Apply Marking Scheme to obtain marked vertices $Y_X \subseteq X$

  $Z_X \leftarrow$ minimum connected core of $(G[\hat{X}], Y_X)$

  In $\mathcal{X}$, replace X by Z_X and small witness set for every vertex in $X \setminus Z_X$.

if *a spanning forest* F *of* $\mathcal{X}$ *has* $\leq k$ *edges* **then**

  return F

Step 1: Color vertices of input graph uniformly at random with 3 colors.

Randomized FPT Algorithm

Step 1: Color vertices of input graph uniformly at random with 3 colors.

Lemma

Pr[A random coloring is a compatible coloring] is at least $\frac{1}{(6k)^3}$.

Randomized FPT Algorithm

Step 1: Color vertices of input graph uniformly at random with 3 colors.

Lemma

Pr[A random coloring is a compatible coloring] is at least $\frac{1}{(6k)^3}$.

Step 2: Extract witness sets out of each colored components of a *compatible* coloring.

Randomized FPT Algorithm

Step 1: Color vertices of input graph uniformly at random with 3 colors.

Lemma

Pr[A random coloring is a compatible coloring] is at least $\frac{1}{(6k)^3}$.

Step 2: Extract witness sets out of each colored components of a *compatible* coloring.

Lemma

Extracting a witness set from a color class is equivalent of finding its connected core containing some marked vertices which can be done in $\mathcal{O}(6^k)$.

Randomized FPT Algorithm

Step 1: Color vertices of input graph uniformly at random with 3 colors.

Lemma

Pr[A random coloring is a compatible coloring] is at least $\frac{1}{(6k)^3}$.

Step 2: Extract witness sets out of each colored components of a *compatible* coloring.

Lemma

Extracting a witness set from a color class is equivalent of finding its connected core containing some marked vertices which can be done in $\mathcal{O}(6^k)$.

Theorem

CACTUS CONTRACTION *is* FPT *with running time* $\mathcal{O}^*(c^k)$.

Thank you!

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

Leizhen Cai and Chengwei Guo.
Contracting few edges to remove forbidden induced subgraphs.
In *IPEC*, pages 97–109, 2013.

P. A. Golovach, M. Kamiński, D. Paulusma, and D. M. Thilikos.
Increasing the minimum degree of a graph by contractions.
Theoretical Computer Science, 481:74 – 84, 2013.

Sylvain Guillemot and Dániel Marx.
A faster FPT algorithm for bipartite contraction.
Inf. Process. Lett., 113(22–24):906–912, 2013.

Petr A. Golovach, Pim van 't Hof, and Daniel Paulusma.
Obtaining planarity by contracting few edges.
Theoretical Computer Science, 476:38–46, 2013.

- Pinar Heggernes, Pim van 't Hof, Benjamin Lévêque, Daniel Lokshtanov, and Christophe Paul.
  Contracting graphs to paths and trees.
  In *Proceedings of the 6th International Conference on Parameterized and Exact Computation*, IPEC'11, pages 55–66, Berlin, Heidelberg, 2012. Springer-Verlag.
- Pinar Heggernes, Pim van 't Hof, Daniel Lokshtanov, and Christophe Paul.
  Obtaining a bipartite graph by contracting few edges.
  SIAM Journal on Discrete Mathematics, 27(4):2143–2156, 2013.

Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh.
On the hardness of eliminating small induced subgraphs by contracting edges.
In *IPEC*, pages 243–254, 2013.