Conflict Free Version of Covering Problems on Graphs: Classical and Parameterized

Pallavi Jain[1] Lawqueen Kanesh[1] Pranabendu Misra[2]

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

June 5, 2018

Classical Problem

$\mathcal{Q}$
Input: An instance I.
Question:

Classical Problems: Running Examples

MINIMUM WEIGHT SPANNING TREE (MST)
Input: A Graph $G = (V, E)$, and a weight function $w : E \to \mathbb{Z}^+$
Output: A spanning tree T of G, of minimum weight.

Classical Problems: Running Examples

MINIMUM WEIGHT SPANNING TREE (MST)
Input: A Graph $G = (V, E)$, and a weight function $w : E \to \mathbb{Z}^+$
Output: A spanning tree T of G, of minimum weight.

FEEDBACK VERTEX SET (FVS) **Parameter:** k
Input: A Graph $G = (V, E)$ and a positive integer k.
Question: Does there exist a set $S \subseteq V$ of size at most k such that $G - S$ does not contain any cycle?

Classical Problems

A lot of studies on problems that satisfy additional constraints.

A lot of studies on problems that satisfy additional constraints.

- FVS

Classical Problems

A lot of studies on problems that satisfy additional constraints.

- FVS
  - Solution induces a connected subgraph in G (CONNECTED FVS)

Classical Problems

A lot of studies on problems that satisfy additional constraints.

- FVS
  - Solution induces a connected subgraph in G (Connected FVS)
  - Solution induces an independent set in G (Independent FVS)

Classical Problems

A lot of studies on problems that satisfy additional constraints.

- FVS
  - Solution induces a connected subgraph in G (CONNECTED FVS)
  - Solution induces an independent set in G (INDEPENDENT FVS)
- MST

Classical Problems

A lot of studies on problems that satisfy additional constraints.

- FVS
  - Solution induces a connected subgraph in G (Connected FVS)
  - Solution induces an independent set in G (Independent FVS)
- MST
  - At least (most) k leaves.

Classical Problems

A lot of studies on problems that satisfy additional constraints.

- FVS
  - Solution induces a connected subgraph in G (CONNECTED FVS)
  - Solution induces an independent set in G (INDEPENDENT FVS)
- MST
  - At least (most) k leaves.
  - At least (most) k internal nodes.

Conflict free Problem

Conflict Free-$\mathcal{Q}$ (CF-$\mathcal{Q}$)
Input: An instance I of classical problem $\mathcal{Q}$, a conflict graph H.
Output: A solution to I in $\mathcal{Q}$, which is also an independent set in H.

Running Examples

CF-MST

Input: A Graph $G = (V, E)$, a conflict graph $H = (E, E')$, and a weight function $w : E \to \mathbb{Z}^+$

Output: A spanning tree T of G, with minimum weight such that $E(T)$ is an independent set in H.

Running Examples

CF-MST

Input: A Graph $G = (V, E)$, a conflict graph $H = (E, E')$, and a weight function $w : E \to \mathbb{Z}^+$

Output: A spanning tree T of G, with minimum weight such that $E(T)$ is an independent set in H.

CF-FVS **Parameter:** k

Input: A Graph $G = (V, E)$, a conflict graph $H = (V, E')$ and a positive integer k.

Question: Does there exist a set $S \subseteq V$ of size at most k such that $G - S$ does not contain any cycle and S is an independent set in H?

Conflict free Problems

- Shortest Path

Conflict free Problems

- Shortest Path
- Maximum Flow

Conflict free Problems

- Shortest Path
- Maximum Flow
- Knapsack

Conflict free Problems

- Shortest Path
- Maximum Flow
- Knapsack
- Bin Packing

Conflict free Problems

- Shortest Path
- Maximum Flow
- Knapsack
- Bin Packing
- Scheduling

Conflict free Problems

- Shortest Path
- Maximum Flow
- Knapsack
- Bin Packing
- Scheduling
- Maximum Matching

Conflict free Problems

- Shortest Path
- Maximum Flow
- Knapsack
- Bin Packing
- Scheduling
- Maximum Matching
- Minimum Weight Spanning Tree

Conflict free Problems

- Shortest Path
- Maximum Flow
- Knapsack
- Bin Packing
- Scheduling
- Maximum Matching
- Minimum Weight Spanning Tree
- ...

By Kann, Pferchy , Epstein, Even. . . .

Why Conflict Free Problems?

- Take a detour and introduce vertex deletion graph problems.
- First introduce the notion of *graph properties*.

Graph Property

- A graph property Π is a family of graphs.

Graph Property

- A graph property Π is a family of graphs.

- Π is Edgeless graphs
- Π is Forest
- Π is Bipartite graphs
- Π is Chordal graphs
- Π is Interval graphs

Π-Vertex Deletion

Π-Vertex Deletion

Π - Graph property.

Π-Vertex Deletion

Π - Graph property.

Π-Vertex Deletion **Parameter:** k
Input: A graph G and a non-negative integer k.
Question: Does there exist $S \subseteq V(G)$, such that $|S| \leq k$ and $G - S$ is in Π?

Π-Vertex Deletion

Π - Graph property.

Π-Vertex Deletion **Parameter:** k
Input: A graph G and a non-negative integer k.
Question: Does there exist $S \subseteq V(G)$, such that $|S| \leq k$ and $G - S$ is in Π?

- Π is Edgeless graphs - Vertex Cover (VC).

Π-Vertex Deletion

Π - Graph property.

Π-Vertex Deletion **Parameter:** k
Input: A graph G and a non-negative integer k.
Question: Does there exist $S \subseteq V(G)$, such that $|S| \leq k$ and $G - S$ is in Π?

- Π is Edgeless graphs - Vertex Cover (VC).
- Π is Forest - Feedback Vertex Set (FVS).

Π-Vertex Deletion

Π - Graph property.

Π-Vertex Deletion	**Parameter:** k

Input: A graph G and a non-negative integer k.
Question: Does there exist $S \subseteq V(G)$, such that $|S| \leq k$ and $G - S$ is in Π?

- Π is Edgeless graphs - Vertex Cover (VC).
- Π is Forest - Feedback Vertex Set (FVS).
- Π is Bipartite graphs - Odd Cycle Transversal (OCT).

Π-Vertex Deletion

Π - Graph property.

Π-Vertex Deletion **Parameter:** k
Input: A graph G and a non-negative integer k.
Question: Does there exist $S \subseteq V(G)$, such that $|S| \leq k$ and $G - S$ is in Π?

- Π is Edgeless graphs - Vertex Cover (VC).
- Π is Forest - Feedback Vertex Set (FVS).
- Π is Bipartite graphs - Odd Cycle Transversal (OCT).
- Π is Chordal graphs - Chordal Vertex Deletion (CVD).

Π-Vertex Deletion

Π - Graph property.

> Π-Vertex Deletion **Parameter:** k
> **Input:** A graph G and a non-negative integer k.
> **Question:** Does there exist $S \subseteq V(G)$, such that $|S| \leq k$ and $G - S$ is in Π?

- Π is Edgeless graphs - Vertex Cover (VC).
- Π is Forest - Feedback Vertex Set (FVS).
- Π is Bipartite graphs - Odd Cycle Transversal (OCT).
- Π is Chordal graphs - Chordal Vertex Deletion (CVD).
- Π is Interval graphs - Interval Vertex Deletion (IVD).

CF-Π-VD

CONFLICT FREE Π-VERTEX DELETION (CF-Π-VD) **Parameter:** k
Input: A graph G, a conflict graph H on vertex set $V(G)$ and a non-negative integer k.
Question: Does there exist a set $S \subseteq V(G)$, such that $|S| \leq k$, $G - S$ is in Π and S is an independent set in H?

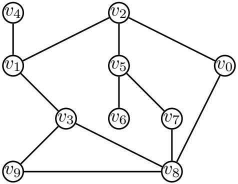
CF-Π-VD

CONFLICT FREE Π-VERTEX DELETION (CF-Π-VD) **Parameter:** k
Input: A graph G, a conflict graph H on vertex set $V(G)$ and a non-negative integer k.
Question: Does there exist a set $S \subseteq V(G)$, such that $|S| \leq k$, $G - S$ is in Π and S is an independent set in H?

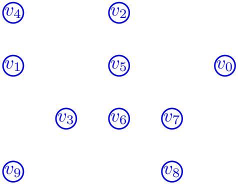
- Π is Edgeless graphs - CF-VC.
- Π is Forest - CF-FVS.
- Π is Bipartite graphs - CF-OCT.
- Π is Chordal graphs - CF-CVD.
- Π is Interval graphs - CF-IVD.

Motivation I: Generalizes Classical Graph Problem

When *H* is edgeless:



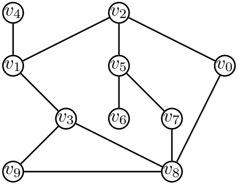
$G = (V, E)$



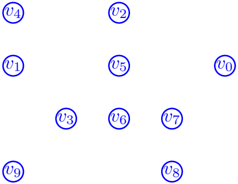
$H = (V, E')$

Motivation I: Generalizes Classical Graph Problem

When H is edgeless:



$G = (V, E)$

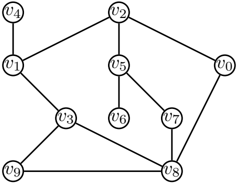


$H = (V, E')$

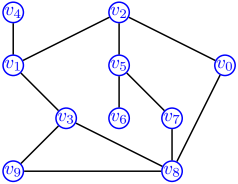
$\mathcal{Q}$=CF-$\mathcal{Q}$.

Motivation II: Generalizes Independence Constraint

When $G = H$



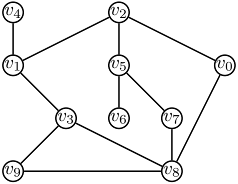
$G = (V, E)$



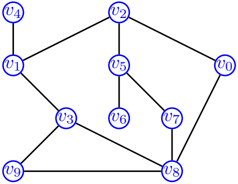
$H = (V, E')$

Motivation II: Generalizes Independence Constraint

When $G = H$



$G = (V, E)$



$H = (V, E')$

CF-$\mathcal{Q}$=Independent-$\mathcal{Q}$.

Models CSP like dichotomy for specific graph problem

- which graph family does the input belongs?

Models CSP like dichotomy for specific graph problem

- which graph family does the input belongs?
- which graph family does the conflict graphs belongs?

Models CSP like dichotomy for specific graph problem

- which graph family does the input belongs?
- which graph family does the conflict graphs belongs?

$(\mathcal{G}, \mathcal{H})$-CF-$\Pi$-VD
Input: A graph $G \in \mathcal{G}$, a conflict graph $H \in \mathcal{H}$ on vertex set $V(G)$ and a non-negative integer k.
Question: Does there exist a set $S \subseteq V(G)$, such that $|S| \leq k$, $G - S$ is in Π and S is an independent set in H?

Framework of Our Study – Parameterized Complexity

Parameterized Complexity : Quick Overview

- Associate a parameter k to an instance I of a classical problem.

Parameterized Complexity : Quick Overview

- Associate a parameter k to an instance I of a classical problem.
- Parameterized problem (I, k) is *fixed-parameter tractable* (FPT) if there is an algorithm that solves it in time $\mathcal{O}(f(k) \cdot |I|^{\mathcal{O}(1)})$.

Parameterized Complexity : Quick Overview

- Associate a parameter k to an instance I of a classical problem.
- 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

Parameterized Complexity : Quick Overview

- Associate a parameter k to an instance I of a classical problem.
- 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
- W[t]-hard, for some $t \in \mathbb{N}$.

Our Results

Forbidden Characterization

- $\mathcal{F}(\Pi)$: Forbidden set characterization of Π.

Forbidden Characterization

- $\mathcal{F}(\Pi)$: Forbidden set characterization of Π.
- Graphs in Π does not contain any graph in $\mathcal{F}(\Pi)$ as an induced subgraph (subgraph).

Forbidden Characterization

- $\mathcal{F}(\Pi)$: Forbidden set characterization of Π.
- Graphs in Π does not contain any graph in $\mathcal{F}(\Pi)$ as an induced subgraph (subgraph).
- VC- $\mathcal{F}(\Pi)$ is an edge (finite).

Forbidden Characterization

- $\mathcal{F}(\Pi)$: Forbidden set characterization of Π.
- Graphs in Π does not contain any graph in $\mathcal{F}(\Pi)$ as an induced subgraph (subgraph).
- VC- $\mathcal{F}(\Pi)$ is an edge (finite).
- FVS- $\mathcal{F}(\Pi)$ is set of all cycles (infinite).

Forbidden Characterization

- $\mathcal{F}(\Pi)$ is **finite**:
  - CF-Π-VD is FPT with running time $(\mathcal{O}(\alpha^k \cdot n \cdot T(m, n)))$, where $T(m, n)$ is time to recognize a graph in Π and α is the size of largest graph in $\mathcal{F}(\Pi)$.

Forbidden Characterization

- $\mathcal{F}(\Pi)$ is **finite**:
  - CF-Π-VD is FPT with running time $(\mathcal{O}(\alpha^k \cdot n \cdot T(m,n)))$, where $T(m,n)$ is time to recognize a graph in Π and α is the size of largest graph in $\mathcal{F}(\Pi)$.
    - Split Vertex Deletion
    - $\mathcal{F}(\Pi)$ is $\{C_4, C_5, 2K_2\}$

Forbidden Characterization

- $\mathcal{F}(\Pi)$ is **finite**:
  - CF-Π-VD is FPT with running time ($\mathcal{O}(\alpha^k \cdot n \cdot T(m,n))$), where $T(m,n)$ is time to recognize a graph in Π and α is the size of largest graph in $\mathcal{F}(\Pi)$.
    - Split Vertex Deletion
    - $\mathcal{F}(\Pi)$ is $\{C_4, C_5, 2K_2\}$
    - $\mathcal{O}(5^k n^{\mathcal{O}(1)})$ time algorithm

Forbidden Characterization

- $\mathcal{F}(\Pi)$ is **finite**:
  - CF-Π-VD is FPT with running time $(\mathcal{O}(\alpha^k \cdot n \cdot T(m,n)))$, where $T(m,n)$ is time to recognize a graph in Π and α is the size of largest graph in $\mathcal{F}(\Pi)$.
    - Split Vertex Deletion
    - $\mathcal{F}(\Pi)$ is $\{C_4, C_5, 2K_2\}$
    - $\mathcal{O}(5^k n^{\mathcal{O}(1)})$ time algorithm
  - Admits $\mathcal{O}(\alpha^2 \alpha! k^\alpha)$ vertex kernel.

Forbidden Characterization

- $\mathcal{F}(\Pi)$ is **finite**:
  - CF-Π-VD is FPT with running time $(\mathcal{O}(\alpha^k \cdot n \cdot T(m, n)))$, where $T(m, n)$ is time to recognize a graph in Π and α is the size of largest graph in $\mathcal{F}(\Pi)$.
    - Split Vertex Deletion
    - $\mathcal{F}(\Pi)$ is $\{C_4, C_5, 2K_2\}$
    - $\mathcal{O}(5^k n^{\mathcal{O}(1)})$ time algorithm
  - Admits $\mathcal{O}(\alpha^2 \alpha! k^\alpha)$ vertex kernel.
- $\mathcal{F}(\Pi)$ is (well behaved) **infinite**: CF-Π-VD is W[1]-hard.

Can we give better than $\mathcal{O}(\alpha^k \cdot n \cdot T(m, n))$ running time algorithm?

Faster Running Time

VC	CF-VC

[1] $\mathcal{O}^\star$ suppresses the polynomial factor in the running time.

Faster Running Time

VC	CF-VC
$2k$-vertex kernel	$2k$-vertex kernel

[1] $\mathcal{O}^\star$ suppresses the polynomial factor in the running time.

Faster Running Time

VC	CF-VC
$2k$-vertex kernel	$2k$-vertex kernel
Factor 2- approximation algorithm	Factor 2- approximation algorithm

[1] $\mathcal{O}^\star$ suppresses the polynomial factor in the running time.

Faster Running Time

VC	CF-VC
$2k$-vertex kernel	$2k$-vertex kernel
Factor 2- approximation algorithm	Factor 2- approximation algorithm
$\mathcal{O}^\star(1.2738^k)$ FPT algorithm [1]	$\mathcal{O}^\star(1.2738^k)$ FPT algorithm

[1] $\mathcal{O}^\star$ suppresses the polynomial factor in the running time.

Faster Running Time

VC	CF-VC
$2k$-vertex kernel	$2k$-vertex kernel
Factor 2- approximation algorithm	Factor 2- approximation algorithm
$\mathcal{O}^{\star}(1.2738^{k})$ FPT algorithm [1]	$\mathcal{O}^{\star}(1.2738^{k})$ FPT algorithm
$\mathcal{O}^{\star}(1.1996^{n})$ exact algorithm	$\mathcal{O}^{\star}(1.1996^{n})$ exact algorithm

[1] $\mathcal{O}^{\star}$ suppresses the polynomial factor in the running time.

Faster Running Time

VC	CF-VC
$2k$-vertex kernel	$2k$-vertex kernel
Factor 2- approximation algorithm	Factor 2- approximation algorithm
$\mathcal{O}^\star(1.2738^k)$ FPT algorithm [1]	$\mathcal{O}^\star(1.2738^k)$ FPT algorithm
$\mathcal{O}^\star(1.1996^n)$ exact algorithm	$\mathcal{O}^\star(1.1996^n)$ exact algorithm
Polynomial time- when graph G is of degree at most 2	NP-complete even when graph G is of degree at most 2 (G is disjoint union of P_3)

[1] $\mathcal{O}^\star$ suppresses the polynomial factor in the running time.

Faster Running Time

VC	CF-VC
$2k$-vertex kernel	$2k$-vertex kernel
Factor 2- approximation algorithm	Factor 2- approximation algorithm
$\mathcal{O}^\star(1.2738^k)$ FPT algorithm [1]	$\mathcal{O}^\star(1.2738^k)$ FPT algorithm
$\mathcal{O}^\star(1.1996^n)$ exact algorithm	$\mathcal{O}^\star(1.1996^n)$ exact algorithm
Polynomial time- when graph G is of degree at most 2	NP-complete even when graph G is of degree at most 2 (G is disjoint union of P_3)
	Polynomial time- G has degree at most one, or both G and H have a perfect matching

[1] $\mathcal{O}^\star$ suppresses the polynomial factor in the running time.

Faster Running Time

Problem Name	Classical	Conflict Free

Faster Running Time

Problem Name	Classical	Conflict Free
d-Hitting Set	$\mathcal{O}^\star((d-0.7262)^k)$ time algorithm	$\mathcal{O}^\star((d-0.7262)^k)$ time algorithm

Faster Running Time

Problem Name	Classical	Conflict Free
d-Hitting Set	$\mathcal{O}^\star((d-0.7262)^k)$ time algorithm	$\mathcal{O}^\star((d-0.7262)^k)$ time algorithm
Split Vertex Deletion	$\mathcal{O}^\star(1.2738^k k^{\mathcal{O}(\log k)})$ time algorithm	$\mathcal{O}^\star(1.2738^k k^{\mathcal{O}(\log k)})$ time algorithm

Faster Running Time

Problem Name	Classical	Conflict Free
d-HITTING SET	$\mathcal{O}^\star((d-0.7262)^k)$ time algorithm	$\mathcal{O}^\star((d-0.7262)^k)$ time algorithm
SPLIT VERTEX DELETION	$\mathcal{O}^\star(1.2738^k k^{\mathcal{O}(\log k)})$ time algorithm	$\mathcal{O}^\star(1.2738^k k^{\mathcal{O}(\log k)})$ time algorithm
FVS IN TOURNAMENTS	$\mathcal{O}^\star(1.618^k)$ time algorithm	$\mathcal{O}^\star(2^k)$ time algorithm

Finite Forbidden Characterization

Theorem

CF-VC *admits a* $2k$*-vertex kernel, a factor* 2*-approximation algorithm, an* $\mathcal{O}^{\star}(1.2738^k)$ FPT *algorithm and a* $\mathcal{O}^{\star}(1.1996^n)$ *exact algorithm.*

Finite Forbidden Characterization

Theorem

CF-VC *admits a* $2k$*-vertex kernel, a factor* 2*-approximation algorithm, an* $\mathcal{O}^\star(1.2738^k)$ FPT *algorithm and a* $\mathcal{O}^\star(1.1996^n)$ *exact algorithm.*

- Polynomial time parameter preserving reduction from CF-VC to MIN ONES 2-SAT.

Finite Forbidden Characterization

Theorem

CF-VC *admits a* $2k$*-vertex kernel, a factor* 2*-approximation algorithm, an* $\mathcal{O}^\star(1.2738^k)$ FPT *algorithm and a* $\mathcal{O}^\star(1.1996^n)$ *exact algorithm.*

- Polynomial time parameter preserving reduction from CF-VC to Min Ones 2-SAT.
- Min Ones 2-SAT can be solved as fast as VC.

Finite Forbidden Characterization

Theorem

CF-VC *admits a* $2k$*-vertex kernel, a factor* 2*-approximation algorithm, an* $\mathcal{O}^\star(1.2738^k)$ FPT *algorithm and a* $\mathcal{O}^\star(1.1996^n)$ *exact algorithm.*

- Polynomial time parameter preserving reduction from CF-VC to Min Ones 2-SAT.
- Min Ones 2-SAT can be solved as fast as VC.
- In polynomial time we can test whether there exists a solution to CF-VC for pair (G, H).

Finite Forbidden Characterization

MIN ONES 2-SAT
Input: A 2-CNF formula ϕ
Output: A satisfying assignment for ϕ formula that minimizes the number of variables that are set to 1

Finite Forbidden Characterization

- Let (G, H, k) be an instance of $\mathrm{CF\text{-}VC}$.

Finite Forbidden Characterization

- Let (G, H, k) be an instance of $\mathrm{CF\text{-}VC}$.
- For every edge $uv \in E(G)$, introduce a clause $(u \vee v)$

Finite Forbidden Characterization

- Let (G, H, k) be an instance of $\mathrm{CF\text{-}VC}$.
- For every edge $uv \in E(G)$, introduce a clause $(u \vee v)$
- for every edge $uv \in E(H)$, introduce a clause $(\bar{u} \vee \bar{v})$ in Φ.

Finite Forbidden Characterization

- Let (G, H, k) be an instance of $\mathrm{CF\text{-}VC}$.
- For every edge $uv \in E(G)$, introduce a clause $(u \vee v)$
- for every edge $uv \in E(H)$, introduce a clause $(\bar{u} \vee \bar{v})$ in Φ.

$$\Phi = \bigwedge_{uv \in E(G)} (u \vee v) \bigwedge_{uv \in E(H)} (\bar{u} \vee \bar{v}).$$

Infinite Forbidden Characterization

Theorem

CF-FVS *is* W[1]*-hard, when parameterized by the solution size.*

Infinite Forbidden Characterization

Theorem

CF-FVS *is* W[1]*-hard, when parameterized by the solution size.*

- Polynomial time parameter preserving reduction from MULTICOLORED INDEPENDENT SET to CF-FVS.

Infinite Forbidden Characterization

Theorem

CF-FVS *is* W[1]*-hard, when parameterized by the solution size.*

- Polynomial time parameter preserving reduction from MULTICOLORED INDEPENDENT SET to CF-FVS.
- MULTICOLORED INDEPENDENT SET is W[1]-hard.

Infinite Forbidden Characterization

Theorem

CF-FVS *is* W[1]*-hard, when parameterized by the solution size.*

- Polynomial time parameter preserving reduction from Multicolored Independent Set to CF-FVS.
- Multicolored Independent Set is W[1]-hard.
- A similar result holds for Conflict Free Odd Cycle Transversal (CF-OCT), Conflict Free Chordal Vertex Deletion (CF-CVD) and Conflict Free Interval Vertex Deletion (CF-IVD)

Infinite Forbidden Characterization

CF-FVS **Parameter:** k

Input: A graph G, a conflict graph H on vertex set $V(G)$, an integer k.

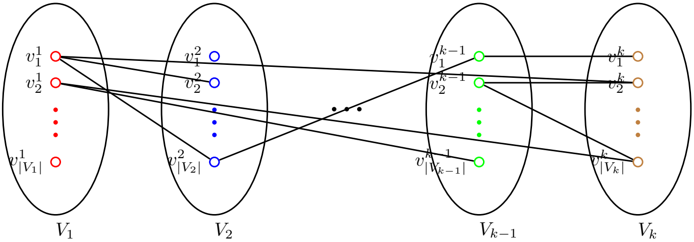
Question: Does there exist $S \subseteq V(G)$, such that $|S| \leq k$, $G - S$ is a forest and S is an independent set in H?

Infinite Forbidden Characterization

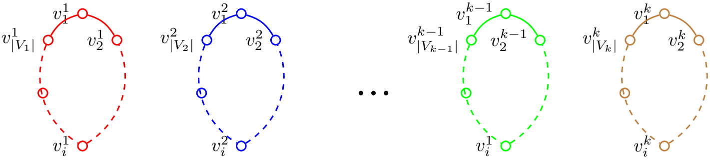
CF-FVS **Parameter:** k
Input: A graph G, a conflict graph H on vertex set $V(G)$, an integer k.
Question: Does there exist $S \subseteq V(G)$, such that $|S| \leq k$, $G - S$ is a forest and S is an independent set in H?

Multicolored Independent Set **Parameter:** k
Input: A graph $G = (V, E)$, and a partition $V_1, V_2, \cdots, V_k$ of V.
Question: Is there a set $S \subseteq V$ such that S is an independent set in G, and for each $i \in [k]$, we have $|S \cap V_i| = 1$?

Infinite Forbidden Characterization



Conflict graph $H = G$



Graph G'

When does CF-Π-VD admits an FPT algorithm?

Infinite Forbidden Characterization

$\mathcal{H}$ is class of:

Infinite Forbidden Characterization

$\mathcal{H}$ is class of:

- d-degenerate graphs, $(\mathcal{D}_d)$

Infinite Forbidden Characterization

$\mathcal{H}$ is class of:

- d-degenerate graphs, $(\mathcal{D}_d)$
- Nowhere Dense graphs

Infinite Forbidden Characterization

$\mathcal{H}$ is class of:

- d-degenerate graphs, $(\mathcal{D}_d)$
- Nowhere Dense graphs

CF-FVS, CF-OCT, Chordal Vertex Deletion and Interval Vertex Deletion are FPT

Independence Covering Lemma [Lokshtanov et al., SODA 2018]

Independence Covering Lemma [Lokshtanov et al., SODA 2018]

k-Independence Covering Family ($\mathcal{F}(H^\star, k)$):

- A family of independent sets of graph $H^\star$

Independence Covering Lemma [Lokshtanov et al., SODA 2018]

k-Independence Covering Family ($\mathcal{F}(H^{\star}, k)$):

- A family of independent sets of graph $H^{\star}$
- For any independent set X in $H^{\star}$ of size at most k there exists a set Y in $\mathcal{F}(H^{\star}, k)$ such that $X \subseteq Y$

Independence Covering Lemma [Lokshtanov et al., SODA 2018]

k-Independence Covering Family ($\mathcal{F}(H^\star, k)$):

- A family of independent sets of graph $H^\star$
- For any independent set X in $H^\star$ of size at most k there exists a set Y in $\mathcal{F}(H^\star, k)$ such that $X \subseteq Y$

Independence Covering Lemma

- Given a d-degenerate graph H^*, integer k.

Independence Covering Lemma [Lokshtanov et al., SODA 2018]

k-Independence Covering Family ($\mathcal{F}(H^{\star}, k)$):

- A family of independent sets of graph $H^{\star}$
- For any independent set X in $H^{\star}$ of size at most k there exists a set Y in $\mathcal{F}(H^{\star}, k)$ such that $X \subseteq Y$

Independence Covering Lemma

- Given a d-degenerate graph H^{*}, integer k.
- There exist two deterministic algorithms running in time polynomial in the family size to compute independence covering families

Independence Covering Lemma [Lokshtanov et al., SODA 2018]

k-Independence Covering Family ($\mathcal{F}(H^\star, k)$):

- A family of independent sets of graph $H^\star$
- For any independent set X in $H^\star$ of size at most k there exists a set Y in $\mathcal{F}(H^\star, k)$ such that $X \subseteq Y$

Independence Covering Lemma

- Given a d-degenerate graph H^*, integer k.
- There exist two deterministic algorithms running in time polynomial in the family size to compute independence covering families
- $\mathcal{F}_1(H^\star, k)$ of size at most $\binom{k(d+1)}{k} 2^{o(k(d+1))} \log n$

Independence Covering Lemma [Lokshtanov et al., SODA 2018]

k-Independence Covering Family ($\mathcal{F}(H^\star, k)$):

- A family of independent sets of graph $H^\star$
- For any independent set X in $H^\star$ of size at most k there exists a set Y in $\mathcal{F}(H^\star, k)$ such that $X \subseteq Y$

Independence Covering Lemma

- Given a d-degenerate graph H^*, integer k.
- There exist two deterministic algorithms running in time polynomial in the family size to compute independence covering families
- $\mathcal{F}_1(H^\star, k)$ of size at most $\binom{k(d+1)}{k} 2^{o(k(d+1))} \log n$
- $\mathcal{F}_2(H^\star, k)$ of size at most $\binom{k^2(d+1)^2}{k} (k(d+1))^{\mathcal{O}(1)} \log n$

Independence Covering Lemma [Lokshtanov et al., SODA 2018]

k-Independence Covering Family ($\mathcal{F}(H^\star, k)$):

- A family of independent sets of graph $H^\star$
- For any independent set X in $H^\star$ of size at most k there exists a set Y in $\mathcal{F}(H^\star, k)$ such that $X \subseteq Y$

Independence Covering Lemma

- Given a d-degenerate graph H^*, integer k.
- There exist two deterministic algorithms running in time polynomial in the family size to compute independence covering families
- $\mathcal{F}_1(H^\star, k)$ of size at most $\binom{k(d+1)}{k} 2^{o(k(d+1))} \log n$
- $\mathcal{F}_2(H^\star, k)$ of size at most $\binom{k^2(d+1)^2}{k} (k(d+1))^{\mathcal{O}(1)} \log n$

A similar lemma exist for Nowhere Dense graphs.

FPT algorithm for $(\star, \mathcal{D}_d)$-CF-FVS

- For conflict graph H, find independence covering family, $\mathcal{F}(H^\star, k)$

FPT algorithm for $(\star, \mathcal{D}_d)$-CF-FVS

- For conflict graph H, find independence covering family, $\mathcal{F}(H^\star, k)$
- For each $Y \in \mathcal{F}(H^\star, k)$, find a solution of FVS contained in Y

FPT algorithm for $(\star, \mathcal{D}_d)$-CF-FVS

- For conflict graph H, find independence covering family, $\mathcal{F}(H^\star, k)$
- For each $Y \in \mathcal{F}(H^\star, k)$, find a solution of FVS contained in Y
  - Solve weighted version of FVS
  - Assign weight $k+1$ to all $v \in V(G) \setminus Y$ and 1 to $v \in Y$.
  - WEIGHTED FVS can be solved in time $\mathcal{O}(3.618^k n^{\mathcal{O}(1)})$

Open Problems:

Open Problems:

- What are other graph classes $\mathcal{G}, \mathcal{H}$ and graph property Π, when $(\mathcal{G}, \mathcal{H})$-CF-$\Pi$-VD is FPT?

Open Problems:

- What are other graph classes $\mathcal{G}, \mathcal{H}$ and graph property Π, when $(\mathcal{G}, \mathcal{H})$-CF-$\Pi$-VD is FPT?
- For what graph classes $\mathcal{G}, \mathcal{H}$ and graph property Π does $(\mathcal{G}, \mathcal{H})$-CF-$\Pi$-VD admits polynomial kernel?

Questions?