Parameterized algorithms for Steiner Tree and Dominating Set:

bounding the leafage by the vertex leafage

Raul Lopes

LAMSADE - Université Paris Dauphine

March 25, 2022
WALCOM

Joint work with Celina M. H. de Figueiredo (UFRJ), Alexsander A. de Melo (UFRJ), and Ana Silva (UFC).

Chordal graphs

Definition (Chordal graphs)

A graph *G is chordal*

$$\iff$$

G has *no induced cycle* of size *at least* 4.

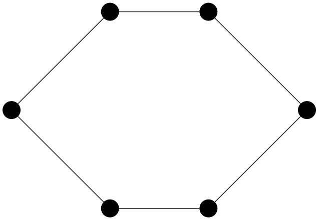
Chordal graphs

Definition (Chordal graphs)

A graph *G is chordal*

$\Longrightarrow$

G has *no induced cycle* of size *at least* 4.



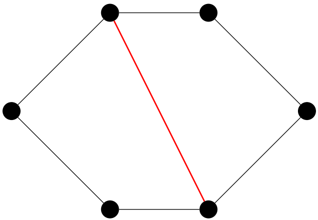
Chordal graphs

Definition (Chordal graphs)

A graph *G is chordal*

$$\Longrightarrow$$

G has *no induced cycle* of size *at least* 4.



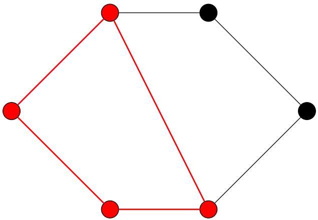
Chordal graphs

Definition (Chordal graphs)

A graph *G is chordal*

$\Longrightarrow$

G has *no induced cycle* of size *at least* 4.



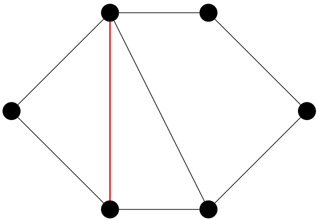
Chordal graphs

Definition (Chordal graphs)

A graph *G is chordal*

$\Longrightarrow$

G has *no induced cycle* of size *at least* 4.



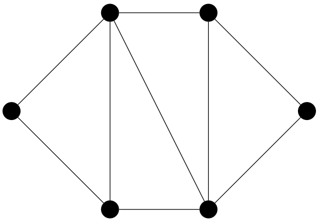
Chordal graphs

Definition (Chordal graphs)

A graph *G is chordal*

$\Longleftrightarrow$

G has *no induced cycle* of size *at least* 4.



Chordal graphs

Definition (Chordal graphs)

A graph *G is chordal*

$\Longrightarrow$

G has *no induced cycle* of size *at least* 4.

Solvable:

- CLIQUE (*polynomial time*).

Hard:

Chordal graphs

Definition (Chordal graphs)

A graph *G is chordal*

$\Longleftrightarrow$

G has *no induced cycle* of size *at least* 4.

Solvable:

- CLIQUE (*polynomial time*).
- VERTEX COLORING (*polynomial time*).

Hard:

Chordal graphs

Definition (Chordal graphs)

A graph *G is chordal*

$$\Longrightarrow$$

G has *no induced cycle* of size *at least* 4.

Solvable:

- CLIQUE (*polynomial time*).
- VERTEX COLORING (*polynomial time*).

Hard:

- k-DOMINATING SET (W[2]-*complete*).

Chordal graphs

Definition (Chordal graphs)

A graph *G is chordal*

$$\Longrightarrow$$

G has *no induced cycle* of size *at least* 4.

Solvable:

- CLIQUE (*polynomial time*).
- VERTEX COLORING (*polynomial time*).

Hard:

- k-DOMINATING SET (W[2]-*complete*).
- k-STEINER TREE (W[2]-*hard*).

Chordal graphs

Definition (Chordal graphs)

A graph *G is chordal*

$$\Longrightarrow$$

G has *no induced cycle* of size *at least* 4.

Solvable:

- CLIQUE (*polynomial time*).
- VERTEX COLORING (*polynomial time*).

Hard:

- k-DOMINATING SET (W[2]-*complete*).
- k-STEINER TREE (W[2]-*hard*).

Chordal graphs

Definition (Chordal graphs)

A graph *G is chordal*

$\Longrightarrow$

G has *no induced cycle* of size *at least* 4.

Solvable:

- CLIQUE (*polynomial time*).
- VERTEX COLORING (*polynomial time*).

Hard:

- k-DOMINATING SET (W[2]-*complete*).
- k-STEINER TREE (W[2]-*hard*).

We show that k-DOMINATING SET and k-STEINER TREE are FPT *in* some *sub-classes of chordal graphs*.

Dominating and Steiner

DOMINATING SET
Input: Graph G, integer k.
Question: Is there $D \subseteq V(G)$ with $|D| \leq k$ s.t. every $v \in V(G) \setminus D$ has a neighbor in D?
D is a *dominating set* of G.

Dominating and Steiner

DOMINATING SET
Input: Graph G, integer k.
Question: Is there $D \subseteq V(G)$ with $|D| \leq k$ s.t. every $v \in V(G) \setminus D$ has a neighbor in D?
D is a *dominating set* of G.

CONNECTED DOMINATING SET
Input: Graph G, integer k.
Question: Is there $D \subseteq V(G)$ with $|D| \leq k$ s.t. every $v \in V(G) \setminus D$ has a neighbor in D and $G[D]$ is connected?
D is a *connected dominating set* of G.

Dominating and Steiner

Dominating Set
Input: Graph G, integer k.
Question: Is there $D \subseteq V(G)$ with $|D| \leq k$ s.t. every $v \in V(G) \setminus D$ has a neighbor in D?
D is a *dominating set* of G.

Connected Dominating Set
Input: Graph G, integer k.
Question: Is there $D \subseteq V(G)$ with $|D| \leq k$ s.t. every $v \in V(G) \setminus D$ has a neighbor in D and $G[D]$ is connected?
D is a *connected dominating set* of G.

Natural parameter $\Longrightarrow$ k (=size of the solution.)

Dominating and Steiner

Dominating Set

Input: Graph G, integer k.

Question: Is there $D \subseteq V(G)$ with $|D| \leq k$ s.t. every $v \in V(G) \setminus D$ has a neighbor in D?

D is a *dominating set* of G.

Connected Dominating Set

Input: Graph G, integer k.

Question: Is there $D \subseteq V(G)$ with $|D| \leq k$ s.t. every $v \in V(G) \setminus D$ has a neighbor in D and $G[D]$ is connected?

D is a *connected dominating set* of G.

Natural parameter $\Longrightarrow$ k (=size of the solution.)

Connected Dominating Set and Steiner Tree have the same complexity in *subclasses of chordal graphs*.

K. White, M. Farber, and W. Pulleybank.
Steiner trees, connected domination and strongly chordal graphs
Networks 15, 1985

Graph G, family of sets $\mathcal{S} = \{S_v \mid v \in V(G)\}$

Definition (Intersection graphs)

G is the *intersection graph of* $\mathcal{S}$

$$\Longrightarrow$$

$$uv \in E(G) \iff S_u \cap S_v \neq \emptyset.$$

Interval graphs: intersection of *subpaths of a path*.

Graph G, family of sets $\mathcal{S} = \{S_v \mid v \in V(G)\}$

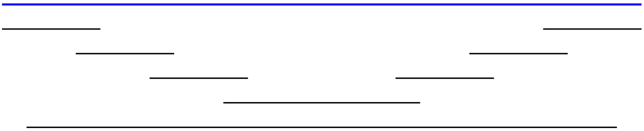
Definition (Intersection graphs)

G is the *intersection graph of* $\mathcal{S}$

$$\Longrightarrow$$

$$uv \in E(G) \iff S_u \cap S_v \neq \emptyset.$$

Interval graphs: intersection of *subpaths of a path*.



Graph G, family of sets $\mathcal{S} = \{S_v \mid v \in V(G)\}$

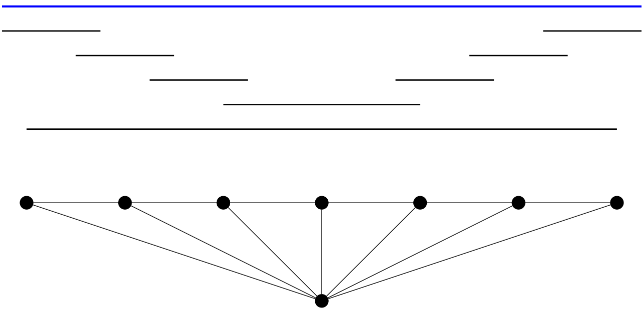
Definition (Intersection graphs)

G is the *intersection graph of* $\mathcal{S}$

$$\Longrightarrow$$

$$uv \in E(G) \iff S_u \cap S_v \neq \emptyset.$$

Interval graphs: intersection of *subpaths of a path*.



Graph G, family of sets $\mathcal{S} = \{S_v \mid v \in V(G)\}$

Definition (Intersection graphs)

G is the *intersection graph of* $\mathcal{S}$

$$\Longrightarrow$$

$$uv \in E(G) \iff S_u \cap S_v \neq \emptyset.$$

Interval graphs: intersection of *subpaths of a path*.

Graph G, family of sets $\mathcal{S} = \{S_v \mid v \in V(G)\}$

Definition (Intersection graphs)

G is the *intersection graph of* $\mathcal{S}$

$$\Longrightarrow$$

$$uv \in E(G) \iff S_u \cap S_v \neq \emptyset.$$

Interval graphs: intersection of *subpaths of a path*.

$$\subset$$

Rooted directed path graphs (RDV): intersection of *directed paths* of an *out-tree*.

Graph G, family of sets $\mathcal{S} = \{S_v \mid v \in V(G)\}$

Definition (Intersection graphs)

G is the *intersection graph of $\mathcal{S}$*

$$\Longrightarrow$$

$$uv \in E(G) \iff S_u \cap S_v \neq \emptyset.$$

Interval graphs: intersection of *subpaths of a path*.

$\subset$

Rooted directed path graphs (RDV): intersection of *directed paths* of an *out-tree*.

$\subset$

Directed path graphs (DV): intersection of *directed paths* of an *oriented tree*.

Graph G, family of sets $\mathcal{S} = \{S_v \mid v \in V(G)\}$

Definition (Intersection graphs)

G is the *intersection graph of $\mathcal{S}$*

$$\Longrightarrow$$

$$uv \in E(G) \iff S_u \cap S_v \neq \emptyset.$$

Interval graphs: intersection of *subpaths of a path*.

$\subset$

Rooted directed path graphs (RDV): intersection of *directed paths* of an *out-tree*.

$\subset$

Directed path graphs (DV): intersection of *directed paths* of an *oriented tree*.

$\subset$

Undirected path graphs (UV): intersection of *paths* of a *tree*.

Graph G, family of sets $\mathcal{S} = \{S_v \mid v \in V(G)\}$

Definition (Intersection graphs)

G is the *intersection graph of* $\mathcal{S}$

$$\Longrightarrow$$

$$uv \in E(G) \iff S_u \cap S_v \neq \emptyset.$$

Interval graphs: intersection of *subpaths of a path*.

$\subset$

Rooted directed path graphs (RDV): intersection of *directed paths* of an *out-tree*.

$\subset$

Directed path graphs (DV): intersection of *directed paths* of an *oriented tree*.

$\subset$

Undirected path graphs (UV): intersection of *paths* of a *tree*.

$\subset$

Chordal graphs: intersection of *subtrees* of a *tree*.

	RDV	DV	UV	Chordal
Dominating	Poly			
Steiner Tree	Poly			

For RDV:

K.S. Booth and J.H. Johnson.
Dominating sets in chordal graphs.
SIAM Journal on Computing 11(1), 1982

K. White, M. Farber, and W. Pulleybank.
Steiner trees, connected domination and strongly chordal graphs.
Networks 15, 1985

	RDV	DV	UV	Chordal
DOMINATING	Poly		NP-c	
STEINER TREE	Poly		NP-c	

For UV:

K.S. Booth and J.H. Johnson.
Dominating sets in chordal graphs.
SIAM Journal on Computing 11(1), 1982

C.H.H. Figueiredo, A.A. Melo, D. Sasaki, and A. Silva.
Revising Johnson's table for the 21st century.
Discrete Applied Mathematics, 2021

	RDV	DV	UV	Chordal
DOMINATING	Poly		NP-c	W[2]-h on k
STEINER TREE	Poly		NP-c	W[2]-h on k

For Chordal:

V. Raman and S. Saurabh.
Short cycles make W-hard problems hard: FPT algorithms for W-hard problems in graphs with no short cycles.
Algorithmica 52(2), 2008.

	RDV	DV	UV	Chordal
Dominating	Poly	Open	NP-c	W[2]-h on k
Steiner Tree	Poly	Open	NP-c	W[2]-h on k

What is between UV and Chordal?

	RDV	DV	UV	Chordal
Dominating	Poly	Open	NP-c	W[2]-h on k
Steiner Tree	Poly	Open	NP-c	W[2]-h on k

What is between UV and Chordal?

- *Leafage*.
- *Vertex leafage*.

Examples

Nodes: $V(T)$, *Vertices:* $V(G)$.

Definition (Tree model)

$\mathcal{T} = (T, \{T_v \mid v \in V(G)\})$ is a *tree model* of G

$$\Longrightarrow$$

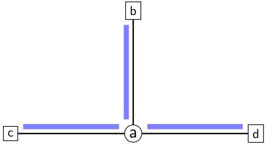
Each $T_v \subseteq T$, and $uv \in E(G) \iff V(T_u) \cap V(T_v) \neq \emptyset$.

Examples

Nodes: $V(T)$, *Vertices:* $V(G)$. Square node $a \implies \exists v \mid T_v = \{a\}$.

Definition (Tree model)

$$\mathcal{T} = (T, \{T_v \mid v \in V(G)\}) \text{ is a } \textit{tree model} \text{ of } G$$
$$\implies$$
$$\text{Each } T_v \subseteq T, \text{ and } uv \in E(G) \iff V(T_u) \cap V(T_v) \neq \emptyset.$$



$T_u = \{b\}$, $T_v = \{c\}$, $T_w = \{d\}$,
$T_{blue} = \{a, b, c, d\}$.

Examples

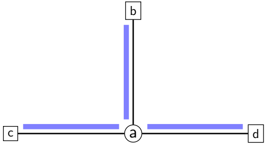
Nodes: $V(T)$, *Vertices:* $V(G)$. Square node $a \implies \exists v \mid T_v = \{a\}$.

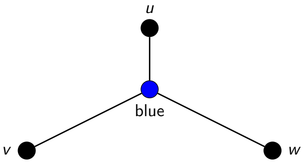
Definition (Tree model)

$\mathcal{T} = (T, \{T_v \mid v \in V(G)\})$ is a *tree model* of G

$\implies$

Each $T_v \subseteq T$, and $uv \in E(G) \iff V(T_u) \cap V(T_v) \neq \emptyset$.





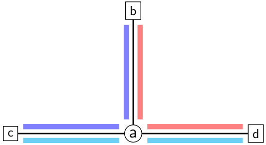
$T_u = \{b\}$, $T_v = \{c\}$, $T_w = \{d\}$,
$T_{blue} = \{a, b, c, d\}$.

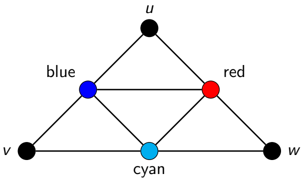
Examples

Nodes: $V(T)$, *Vertices:* $V(G)$. Square node $a \implies \exists v \mid T_v = \{a\}$.

Definition (Tree model)

$$\mathcal{T} = (T, \{T_v \mid v \in V(G)\}) \text{ is a } \textit{tree model} \text{ of } G$$
$$\Longrightarrow$$
$$\text{Each } T_v \subseteq T, \text{ and } uv \in E(G) \iff V(T_u) \cap V(T_v) \neq \emptyset.$$





$T_u = \{b\}$, $T_v = \{c\}$, $T_w = \{d\}$,
$T_{blue} = \{b, c\}$, $T_{red} = \{b, d\}$, $T_{cyan} = \{c, a, d\}$.

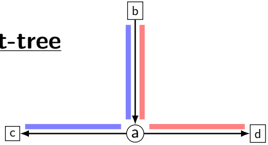
Examples

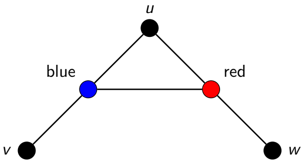
Nodes: $V(T)$, *Vertices:* $V(G)$. Square node $a \implies \exists v \mid T_v = \{a\}$.

Definition (Tree model)

$$\mathcal{T} = (T, \{T_v \mid v \in V(G)\}) \text{ is a } \textit{tree model} \text{ of } G$$
$$\implies$$
$$\text{Each } T_v \subseteq T, \text{ and } uv \in E(G) \iff V(T_u) \cap V(T_v) \neq \emptyset.$$

Out-tree





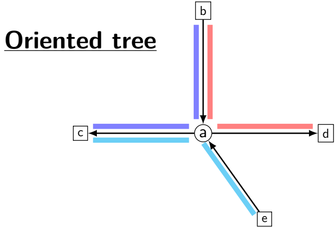
$T_u = \{b\}$, $T_v = \{c\}$, $T_w = \{d\}$,
$T_{blue} = \{b, a, c\}$, $T_{red} = \{b, a, d\}$.

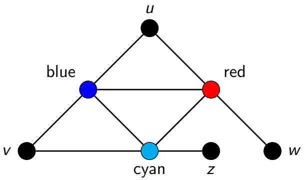
Examples

Nodes: $V(T)$, *Vertices:* $V(G)$. Square node $a \implies \exists v \mid T_v = \{a\}$.

Definition (Tree model)

$$\mathcal{T} = (T, \{T_v \mid v \in V(G)\}) \text{ is a } \textit{tree model} \text{ of } G$$
$$\implies$$
$$\text{Each } T_v \subseteq T \text{, and } uv \in E(G) \iff V(T_u) \cap V(T_v) \neq \emptyset.$$





$T_u = \{b\}$, $T_v = \{c\}$, $T_w = \{d\}$,
$T_{blue} = \{b, a, c\}$, $T_{red} = \{b, a, d\}$. $T_{cyan} = \{e, a, d\}$.

Leafage and vertex leafage

Let $\mathcal{T} = (T, \{T_v \mid v \in V(G)\})$ be a *tree-model* of G.

Definition (Leafage, vertex leafage of a model)

Definition (Leafage, vertex leafage of a graph)

Leafage and vertex leafage

Let $\mathcal{T} = (T, \{T_v \mid v \in V(G)\})$ be a *tree-model* of G.

Definition (Leafage, vertex leafage of a model)

- Leafage $\ell(\mathcal{T}) = \#$*leaves* of T.

Definition (Leafage, vertex leafage of a graph)

Leafage and vertex leafage

Let $\mathcal{T} = (T, \{T_v \mid v \in V(G)\})$ be a *tree-model* of G.

Definition (Leafage, vertex leafage of a model)

- Leafage $\ell(\mathcal{T}) = \#$*leaves* of T.

Definition (Leafage, vertex leafage of a graph)

- Leafage $\ell(G) = \min k$ s.t G has a tree model $\mathcal{T}'$ w. $\ell(\mathcal{T}') = k$.

Leafage and vertex leafage

Let $\mathcal{T} = (T, \{T_v \mid v \in V(G)\})$ be a *tree-model* of G.

Definition (Leafage, vertex leafage of a model)

- Leafage $\ell(\mathcal{T}) = \#$*leaves* of T.
- Vertex leafage $\mathrm{v}\ell(\mathcal{T}) = \max\limits_{v \in V(G)} \{\#leaves T_v\}$.

Definition (Leafage, vertex leafage of a graph)

- Leafage $\ell(G) = \min k$ s.t G has a tree model $\mathcal{T}'$ w. $\ell(\mathcal{T}') = k$.

Leafage and vertex leafage

Let $\mathcal{T} = (T, \{T_v \mid v \in V(G)\})$ be a *tree-model* of G.

Definition (Leafage, vertex leafage of a model)

- Leafage $\ell(\mathcal{T}) = \#\mathit{leaves}$ of T.
- Vertex leafage $\mathrm{v}\ell(\mathcal{T}) = \max_{v \in V(G)} \{\#\mathit{leaves} T_v\}$.

Definition (Leafage, vertex leafage of a graph)

- Leafage $\ell(G) = \min k$ s.t G has a tree model $\mathcal{T}'$ w. $\ell(\mathcal{T}') = k$.
- Vertex leafage $\mathrm{v}\ell(G) = \min k$ s.t G has a tree model $\mathcal{T}'$ w. $\mathrm{v}\ell(\mathcal{T}') = k$.

- $\mathrm{v}\ell(G) \leq \ell(G) \implies \ell(G)$ is a **stronger** parameter.

Leafage and vertex leafage

Let $\mathcal{T} = (T, \{T_v \mid v \in V(G)\})$ be a *tree-model* of G.

Definition (Leafage, vertex leafage of a model)

- Leafage $\ell(\mathcal{T}) = \#leaves$ of T.
- Vertex leafage $\mathrm{v}\ell(\mathcal{T}) = \max_{v \in V(G)} \{\#leaves T_v\}$.

Definition (Leafage, vertex leafage of a graph)

- Leafage $\ell(G) = \min k$ s.t G has a tree model $\mathcal{T}'$ w. $\ell(\mathcal{T}') = k$.
- Vertex leafage $\mathrm{v}\ell(G) = \min k$ s.t G has a tree model $\mathcal{T}'$ w. $\mathrm{v}\ell(\mathcal{T}') = k$.

- $\mathrm{v}\ell(G) \leq \ell(G) \implies \ell(G)$ is a **stronger** parameter.
- *Interval graphs:* chordal graphs with $\ell(G) \leq 2$.

Leafage and vertex leafage

Let $\mathcal{T} = (T, \{T_v \mid v \in V(G)\})$ be a *tree-model* of G.

Definition (Leafage, vertex leafage of a model)

- Leafage $\ell(\mathcal{T}) = \#\mathit{leaves}$ of T.
- Vertex leafage $\mathrm{v}\ell(\mathcal{T}) = \max\limits_{v \in V(G)} \{\#\mathit{leaves} T_v\}$.

Definition (Leafage, vertex leafage of a graph)

- Leafage $\ell(G) = \min k$ s.t G has a tree model $\mathcal{T}'$ w. $\ell(\mathcal{T}') = k$.
- Vertex leafage $\mathrm{v}\ell(G) = \min k$ s.t G has a tree model $\mathcal{T}'$ w. $\mathrm{v}\ell(\mathcal{T}') = k$.

- $\mathrm{v}\ell(G) \leq \ell(G) \implies \ell(G)$ is a **stronger** parameter.
- *Interval graphs:* chordal graphs with $\ell(G) \leq 2$.
- *Undirected path graphs:* chordal graphs with $\mathrm{v}\ell(G) \leq 2$.

Definition (Leafage, vertex leafage of a graph)

- Leafage $\ell(G) = \min k$ s.t G has a tree model $\mathcal{T}$ w. $\ell(\mathcal{T}') = k$.
- Vertex leafage $\mathrm{v}\ell(G) = \min k$ s.t G has a tree model $\mathcal{T}$ w. $\mathrm{v}\ell(\mathcal{T}') = k$.

Definition (Leafage, vertex leafage of a graph)

- Leafage $\ell(G) = \min k$ s.t G has a tree model $\mathcal{T}$ w. $\ell(\mathcal{T}') = k$.
- Vertex leafage $v\ell(G) = \min k$ s.t G has a tree model $\mathcal{T}$ w. $v\ell(\mathcal{T}') = k$.

- Find tree model with $\ell(G)$ leaves: *polynomial time*.

M. Habiba and L. Stacho.
Polynomial-time algorithm for the leafage of chordal graphs.
ESA, 2009.

Definition (Leafage, vertex leafage of a graph)

- Leafage $\ell(G) = \min k$ s.t G has a tree model $\mathcal{T}$ w. $\ell(\mathcal{T}') = k$.
- Vertex leafage $v\ell(G) = \min k$ s.t G has a tree model $\mathcal{T}$ w. $v\ell(\mathcal{T}') = k$.

- Find tree model with $\ell(G)$ leaves: *polynomial time*.
- Decide if $v\ell(G) \leq 3$: NP-complete.

M. Habiba and L. Stacho.
Polynomial-time algorithm for the leafage of chordal graphs.
ESA, 2009.

S. Chaplick and J. Stacho.
The vertex leafage of chordal graphs.
Discrete Applied Mathematics 168, 2014.

Definition (Leafage, vertex leafage of a graph)

- Leafage $\ell(G) = \min k$ s.t G has a tree model $\mathcal{T}$ w. $\ell(\mathcal{T}') = k$.
- Vertex leafage $v\ell(G) = \min k$ s.t G has a tree model $\mathcal{T}$ w. $v\ell(\mathcal{T}') = k$.

- Find tree model with $\ell(G)$ leaves: *polynomial time*.
- Decide if $v\ell(G) \leq 3$: NP-complete.
- DOMINATING SET is FPT w.r.t. $\ell(G)$.

M. Habiba and L. Stacho.
Polynomial-time algorithm for the leafage of chordal graphs.
ESA, 2009.

S. Chaplick and J. Stacho.
The vertex leafage of chordal graphs.
Discrete Applied Mathematics 168, 2014.

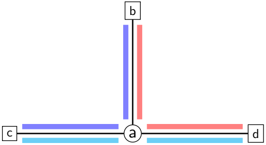
F.V. Foming, P.A. Golovach, and J.F. Raymond.
On the tractability of optimization problems on H-graphs.
Algorithmica 89(2), 2020.

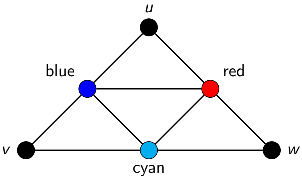
Bounding the leafage by the vertex leafage

- *Minimal tree model* w. host tree $T \implies$ ensures that $\exists v \mid V(T_v) = \{x\}$ for every *leaf* $x \in V(T)$.
  - v is *leafy* vertex.

Bounding the leafage by the vertex leafage

- *Minimal tree model* w. host tree $T \implies$ ensures that $\exists v \mid V(T_v) = \{x\}$ for every *leaf* $x \in V(T)$.
  - v is *leafy* vertex.

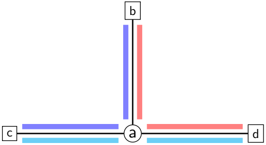


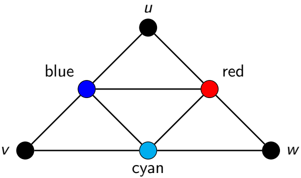


- u, v, w are *leafy vertices*.

Bounding the leafage by the vertex leafage

- *Minimal tree model* w. host tree T $\implies$ ensures that $\exists v \mid V(T_v) = \{x\}$ for every *leaf* $x \in V(T)$.
  - v is *leafy* vertex.



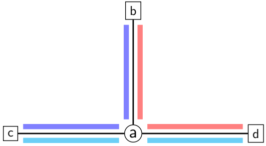


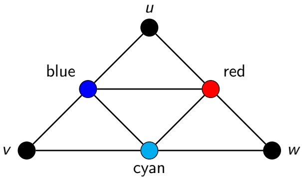
- u, v, w are *leafy vertices*.

- Every *dominating set* D must dominate all *leafy vertices*.

Bounding the leafage by the vertex leafage

- *Minimal tree model* w. host tree $T \implies$ ensures that $\exists v \mid V(T_v) = \{x\}$ for every *leaf* $x \in V(T)$.
  - v is *leafy* vertex.



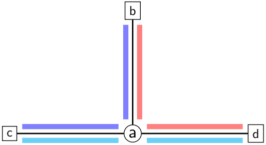


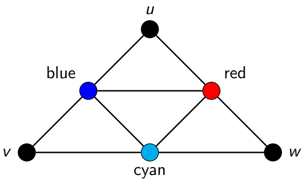
- u, v, w are *leafy vertices*.

- Every *dominating set* D must dominate all *leafy vertices*.
  - For each leafy v with $T_v = \{x\}$, $\exists u \in D$ s.t. $x \in V(T_u)$.

Bounding the leafage by the vertex leafage

- *Minimal tree model* w. host tree $T \implies$ ensures that $\exists v \mid V(T_v) = \{x\}$ for every *leaf* $x \in V(T)$.
  - v is *leafy* vertex.



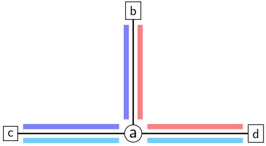


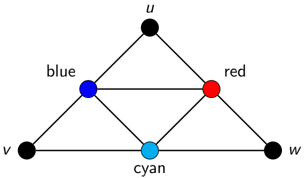
- u, v, w are *leafy vertices*.

- Every *dominating set* D must dominate all *leafy vertices*.
  - For each leafy v with $T_v = \{x\}$, $\exists u \in D$ s.t. $x \in V(T_u)$.
- Each T_u can cover at most v$\ell(G)$ leaves.

Bounding the leafage by the vertex leafage

- *Minimal tree model* w. host tree T $\implies$ ensures that $\exists v \mid V(T_v) = \{x\}$ for every *leaf* $x \in V(T)$.
  - v is *leafy* vertex.



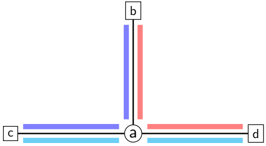


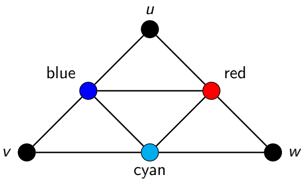
- u, v, w are *leafy vertices*.

- Every *dominating set* D must dominate all *leafy vertices*.
  - For each leafy v with $T_v = \{x\}$, $\exists u \in D$ s.t. $x \in V(T_u)$.
- Each T_u can cover at most $\mathrm{v}\ell(G)$ leaves.
  - $\ell(G) \geq k \cdot \mathrm{v}\ell(G) + 1$, instance of k-Dominating Set is **negative**.

Bounding the leafage by the vertex leafage

- *Minimal tree model* w. host tree $T \implies$ ensures that $\exists v \mid V(T_v) = \{x\}$ for every *leaf* $x \in V(T)$.
  - v is *leafy* vertex.



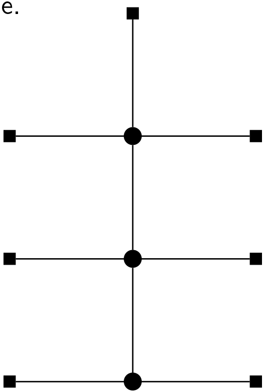


- u, v, w are *leafy vertices*.

- Every *dominating set* D must dominate all *leafy vertices*.
  - For each leafy v with $T_v = \{x\}$, $\exists u \in D$ s.t. $x \in V(T_u)$.
- Each T_u can cover at most $\mathrm{v}\ell(G)$ leaves.
  - $\ell(G) \geq k \cdot \mathrm{v}\ell(G) + 1$, instance of k-Dominating Set is **negative**.
- Every positive instance of k-Dominating Set has $\ell(G) \leq k \cdot \mathrm{v}\ell(G)$.

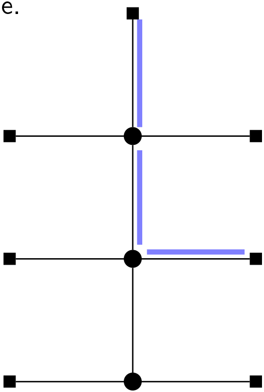
Every positive instance of k-DOMINATING SET has $\ell(G) \leq k \cdot \mathrm{v}\ell(G)$.

- *Square nodes* are associated with *leafy vertices*.
- $\mathrm{v}\ell(G) = 2$ in the example.



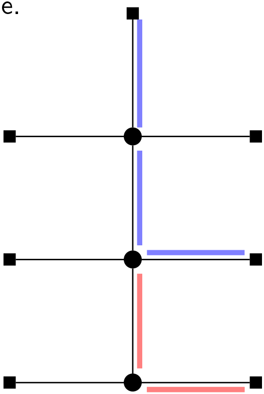
Every positive instance of k-Dominating Set has $\ell(G) \leq k \cdot \mathsf{v}\ell(G)$.

- *Square nodes* are associated with *leafy vertices*.
- $\mathsf{v}\ell(G) = 2$ in the example.



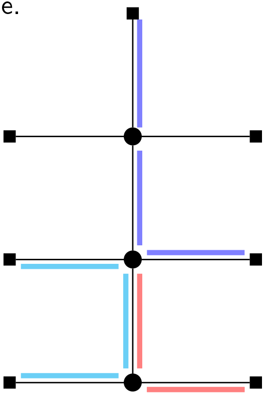
Every positive instance of k-Dominating Set has $\ell(G) \leq k \cdot \mathsf{v}\ell(G)$.

- *Square nodes* are associated with *leafy vertices*.
- $\mathsf{v}\ell(G) = 2$ in the example.



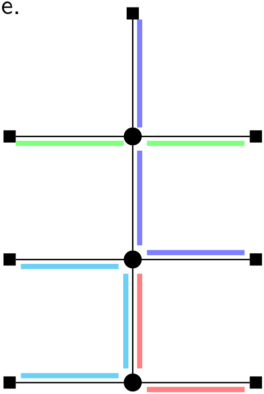
Every positive instance of k-Dominating Set has $\ell(G) \leq k \cdot \mathrm{v}\ell(G)$.

- *Square nodes* are associated with *leafy vertices*.
- $\mathrm{v}\ell(G) = 2$ in the example.



Every positive instance of k-Dominating Set has $\ell(G) \leq k \cdot \mathrm{v}\ell(G)$.

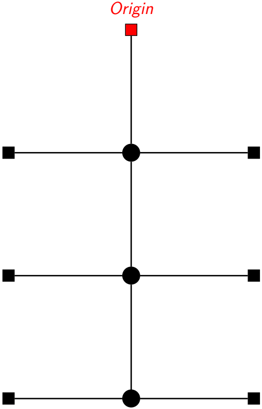
- *Square nodes* are associated with *leafy vertices*.
- $\mathrm{v}\ell(G) = 2$ in the example.



k-Dominating Set in Undirected path graphs

Every *positive* instance of k-Dominating Set has $\ell(G) \leq 2k$.

- *Square nodes* are associated with *leafy vertices*.

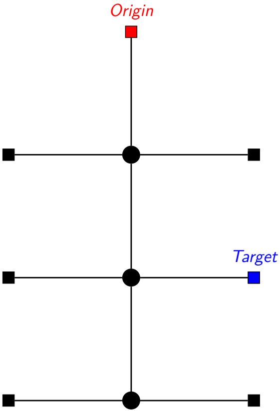


- From *red vertex*: $\leq 2k - 1$ "directions" (other leaves).

k-Dominating Set in Undirected path graphs

Every *positive* instance of k-Dominating Set has $\ell(G) \leq 2k$.

- *Square nodes* are associated with *leafy vertices*.

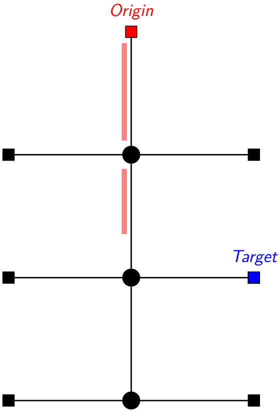


- From *red vertex*: $\leq 2k - 1$ "directions" (other leaves).
- Pick *target*.

k-Dominating Set in Undirected path graphs

Every *positive* instance of k-Dominating Set has $\ell(G) \leq 2k$.

- *Square nodes* are associated with *leafy vertices*.

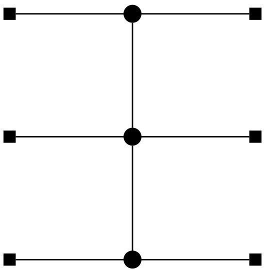


- From *red vertex*: $\leq 2k - 1$ "directions" (other leaves).
- Pick *target*.
- Choose longest path T_v from *red* origin to *blue* target.

k-Dominating Set in Undirected path graphs

Every *positive* instance of k-Dominating Set has $\ell(G) \leq 2k$.

- *Square nodes* are associated with *leafy vertices*.

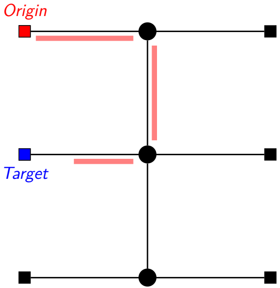


- From *red vertex*: $\leq 2k - 1$ "directions" (other leaves).
- Pick *target*.
- Choose longest path T_v from *red* origin to *blue* target.
- Mark all u with $V(T_u) \cap V(T_v) \neq \emptyset$ as *dominated*, reduce instance.

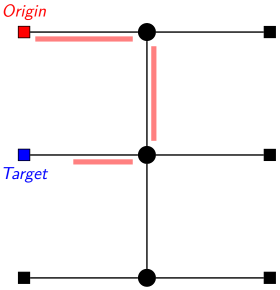
k-Dominating Set in Undirected path graphs

Every *positive* instance of k-Dominating Set has $\ell(G) \leq 2k$.

- *Square nodes* are associated with *leafy vertices*.

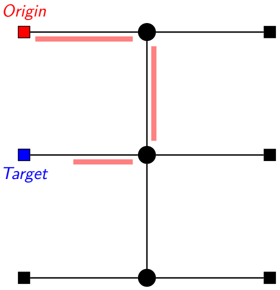


- From *red vertex*: $\leq 2k - 1$ "directions" (other leaves).
- Pick *target*.
- Choose longest path T_v from *red* origin to *blue* target.
- Mark all u with $V(T_u) \cap V(T_v) \neq \emptyset$ as *dominated*, reduce instance.
- Repeat for new *origin*,*target* pair of not-dominated leafy vertices.



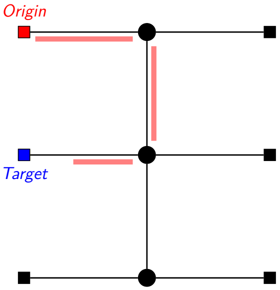
- From *red vertex*: $\leq 2k - 1$ "directions" (other leaves).
- Pick *target*.
- Choose longest path T_v from *red* origin to *blue* target.
- Mark all u with $V(T_u) \cap V(T_v) \neq \emptyset$ as *dominated*, reduce instance.
- Repeat for new *origin*,*target* pair of not-dominated leafy vertices.

- **Bounded search tree:** $(2k)^2$ choices of *origin*,*target* pairs.



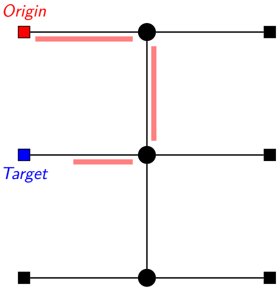
- From *red vertex*: $\leq 2k - 1$ "directions" (other leaves).
- Pick *target*.
- Choose longest path T_v from *red* origin to *blue* target.
- Mark all u with $V(T_u) \cap V(T_v) \neq \emptyset$ as *dominated*, reduce instance.
- Repeat for new *origin*,*target* pair of not-dominated leafy vertices.

- **Bounded search tree:** $(2k)^2$ choices of *origin*,*target* pairs.
  - $(2k)^2$ *choices* $\implies$ new instance on $\leq k - 1$ *leaves* $\implies$ $(2k - 2)^2$ *choices...*



- From *red vertex*: $\leq 2k-1$ “directions” (other leaves).
- Pick *target*.
- Choose longest path T_v from *red* origin to *blue* target.
- Mark all u with $V(T_u) \cap V(T_v) \neq \emptyset$ as *dominated*, reduce instance.
- Repeat for new *origin*,*target* pair of not-dominated leafy vertices.

- **Bounded search tree:** $(2k)^2$ choices of *origin*,*target* pairs.
  - $(2k)^2$ *choices* $\implies$ new instance on $\leq k-1$ *leaves* $\implies$ $(2k-2)^2$ *choices...*
- *Budget* $k \implies$ *height* $\leq k$.



- From *red vertex*: $\leq 2k-1$ "directions" (other leaves).
- Pick *target*.
- Choose longest path T_v from *red* origin to *blue* target.
- Mark all u with $V(T_u) \cap V(T_v) \neq \emptyset$ as *dominated*, reduce instance.
- Repeat for new *origin*,*target* pair of not-dominated leafy vertices.

- **Bounded search tree:** $(2k)^2$ choices of *origin*,*target* pairs.
  - $(2k)^2$ *choices* $\implies$ new instance on $\leq k-1$ *leaves* $\implies$ $(2k-2)^2$ *choices...*
- *Budget* $k \implies$ *height* $\leq k$.
- $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ algorithm.

$2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ algorithm for k-Dominating Set in *undirected path graphs*.

$\mathcal{O}^*\left(2^{k \cdot \mathrm{v}\ell(G)}\right)$ for k-Steiner Tree and k-Connected Dominating Set.

$2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ algorithm for k-Dominating Set in *undirected path graphs*.

$\mathcal{O}^* \left(2^{k \cdot \mathrm{v}\ell(G)}\right)$ for k-Steiner Tree and k-Connected Dominating Set.

- Beats the alternative: *bound* $\ell(G)$ and apply algorithm by Fomin et al. (2020) yields $2^{\mathcal{O}(k^2)} \cdot n^{\mathcal{O}(1)}$ algorithm.

F.V. Foming, P.A. Golovach, and J.F. Raymond.
On the tractability of optimization problems on H-graphs.
Algorithmica 89(2), 2020.

$2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ algorithm for k-Dominating Set in *undirected path graphs*.

$\mathcal{O}^*\left(2^{k \cdot \mathrm{v}\ell(G)}\right)$ for k-Steiner Tree and k-Connected Dominating Set.

- Beats the alternative: *bound* $\ell(G)$ and apply algorithm by Fomin et al. (2020) yields $2^{\mathcal{O}(k^2)} \cdot n^{\mathcal{O}(1)}$ algorithm.

	RDV	DV	UV	Chordal
Dominating	Poly	Open	FPT	W[2]-hard
Steiner Tree	Poly	Open	FPT	W[2]-hard

$2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ algorithm for k-DOMINATING SET in *undirected path graphs*.

$\mathcal{O}^*\left(2^{k \cdot \mathrm{v}\ell(G)}\right)$ for k-STEINER TREE and k-CONNECTED DOMINATING SET.

- Beats the alternative: *bound* $\ell(G)$ and apply algorithm by Fomin et al. (2020) yields $2^{\mathcal{O}(k^2)} \cdot n^{\mathcal{O}(1)}$ algorithm.

	RDV	DV	UV	Chordal
DOMINATING	Poly	Open	FPT	W[2]-hard
STEINER TREE	Poly	Open	FPT	W[2]-hard

- **We show:** DOMINATING SET, STEINER TREE, CONNECTED DOMINATING SET are FPT when parameterized by $k + \mathrm{v}\ell(G)$ when tree model with *optimum vertex leafage* is given.

$2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ algorithm for k-Dominating Set in *undirected path graphs*.

$\mathcal{O}^*\left(2^{k \cdot \mathrm{v}\ell(G)}\right)$ for k-Steiner Tree and k-Connected Dominating Set.

- Beats the alternative: *bound* $\ell(G)$ and apply algorithm by Fomin et al. (2020) yields $2^{\mathcal{O}(k^2)} \cdot n^{\mathcal{O}(1)}$ algorithm.

	RDV	DV	UV	Chordal
Dominating	Poly	Open	FPT	W[2]-hard
Steiner Tree	Poly	Open	FPT	W[2]-hard

- **We show:** Dominating Set, Steiner Tree, Connected Dominating Set are FPT when parameterized by $k + \mathrm{v}\ell(G)$ when tree model with *optimum vertex leafage* is given.
- Undirected path graphs are recognizable in polynomial time.

F. Gavril.
A recognition algorithm for the intersection graphs of paths in trees.
Discrete Mathematics 23(3), 1978.

	RDV	DV	UV	Chordal
Dominating	Poly	Open	FPT	W[2]-hard
Steiner Tree	Poly	Open	FPT	W[2]-hard

- FPT w.r.t. $\ell(G) \implies$ FPT w.r.t. $k + \mathrm{v}\ell(G)$.

	RDV	DV	UV	Chordal
Dominating	Poly	Open	FPT	W[2]-hard
Steiner Tree	Poly	Open	FPT	W[2]-hard

- FPT w.r.t. $\ell(G)$ $\implies$ FPT w.r.t. $k + \mathrm{v}\ell(G)$.
  - Provided a tree model with optimal vertex leafage is given.

	RDV	DV	UV	Chordal
Dominating	Poly	Open	FPT	W[2]-hard
Steiner Tree	Poly	Open	FPT	W[2]-hard

- FPT w.r.t. $\ell(G) \implies$ FPT w.r.t. $k + \mathrm{v}\ell(G)$.
  - Provided a tree model with optimal vertex leafage is given.
- Works for class of Min-LC-VSP$_{\sigma,\rho}$ problems.

	RDV	DV	UV	Chordal
DOMINATING	Poly	Open	FPT	W[2]-hard
STEINER TREE	Poly	Open	FPT	W[2]-hard

- FPT w.r.t. $\ell(G) \implies$ FPT w.r.t. $k + \mathrm{v}\ell(G)$.
  - Provided a tree model with optimal vertex leafage is given.
- Works for class of $\textsc{Min-LC-VSP}_{\sigma,\rho}$ problems.
- Regarding DV, we believe that

	RDV	DV	UV	Chordal
Dominating	Poly	Open	FPT	W[2]-hard
Steiner Tree	Poly	Open	FPT	W[2]-hard

- FPT w.r.t. $\ell(G) \implies$ FPT w.r.t. $k + \mathrm{v}\ell(G)$.
  - Provided a tree model with optimal vertex leafage is given.
- Works for class of $\textsc{Min-LC-VSP}_{\sigma,\rho}$ problems.
- Regarding DV, we believe that
  - Dominating Set is FPT when parameterized by the number c of source nodes of the tree model ($c = 1 \implies$ RDV), and

	RDV	DV	UV	Chordal
Dominating Set	Poly	Open	FPT	W[2]-hard
Steiner Tree	Poly	Open	FPT	W[2]-hard

- FPT w.r.t. $\ell(G) \implies$ FPT w.r.t. $k + \mathsf{v}\ell(G)$.
  - Provided a tree model with optimal vertex leafage is given.
- Works for class of Min-LC-VSP$_{\sigma,\rho}$ problems.
- Regarding DV, we believe that
  - Dominating Set is FPT when parameterized by the number c of source nodes of the tree model ($c = 1 \implies$ RDV), and
  - Dominating Set is NP-complete in DV $\implies$ **separation** of DV and RDV.

Appendix

$\textsc{Min-LC-VSP}_{\sigma,\rho}$

Graph G, subsets $\sigma, \rho \subseteq \{0, \cdots, n-1\}$

Definition ((σ, ρ)-sets)

S is a (σ, ρ)-set

$$\Longrightarrow$$

$|N(v) \cap S| \in \sigma$ for every $v \in S$ and $|N(v) \cap S| \in \rho$ for every $v \notin S$.

$\textsc{Min-LC-VSP}_{\rho,\sigma}$
Input: Graph G, integer k.
Question: Is there a (σ, ρ)-set X with $|X| \leq k$?

Dominating and Steiner

DOMINATING SET
Input: Graph G, integer k.
Question: Is there $D \subseteq V(G)$ with $|D| \leq k$ s.t. every $v \in V(G) \setminus D$ has a neighbor in D?
D is a *dominating set* of G.

STEINER TREE
Input: Graph G, set $X \subseteq V(G)$, integer k.
Question: Is there $S \subseteq V(G)$ with $|S| \leq k$ s.t. $G[X \cup S]$ is connected?
S is *a Steiner set*.

CONNECTED DOMINATING SET
Input: Graph G, integer k.
Question: Is there $D \subseteq V(G)$ with $|D| \leq k$ s.t. every $v \in V(G) \setminus D$ has a neighbor in D and $G[D]$ is connected?
D is a *connected dominating set* of G.

Natural parameter $\Longrightarrow$ k (=size of the solution.)

XP and FPT

Problem with input size n, associated *parameter* k:

- XP problem $\Rightarrow f(k) \cdot n^{g(k)}$ time algorithm.
  - Example: $\mathcal{O}(n^k)$.

XP and FPT

Problem with input size n, associated *parameter* k:

- XP problem $\Rightarrow f(k) \cdot n^{g(k)}$ time algorithm.
  - Example: $\mathcal{O}(n^k)$.
- FPT problem $\Rightarrow f(k) \cdot n^c$ time algorithm.
  - Example: $\mathcal{O}(2^k \cdot n^2)$ (c independent of k).

XP and FPT

Problem with input size n, associated *parameter* k:

- XP problem $\Rightarrow f(k) \cdot n^{g(k)}$ time algorithm.
  - Example: $\mathcal{O}(n^k)$.
- FPT problem $\Rightarrow f(k) \cdot n^c$ time algorithm.
  - Example: $\mathcal{O}(2^k \cdot n^2)$ (c independent of k).
- $\mathrm{W}[1]$-hard problem $\Rightarrow$ strong evidence that it is *not* FPT.

XP and FPT

Problem with input size n, associated *parameter* k:

- XP problem $\Rightarrow f(k) \cdot n^{g(k)}$ time algorithm.
  - Example: $\mathcal{O}(n^k)$.
- FPT problem $\Rightarrow f(k) \cdot n^c$ time algorithm.
  - Example: $\mathcal{O}(2^k \cdot n^2)$ (c independent of k).
- $\mathrm{W}[1]$-hard problem $\Rightarrow$ strong evidence that it is *not* FPT.
- Hardness hierarchy:

$$\mathrm{W}[1] \subseteq \mathrm{W}[2] \subseteq \cdots \subseteq \mathrm{W}[t] \subseteq \mathrm{XP}$$