Enumerating minimal dominating sets in K_t-free graphs

Jean-Florent Raymond
(TU Berlin)

IRIF
28/05/19

Joint work with **Marthe Bonamy** (Bordeaux), **Oscar Defrain** (Clermont-Ferrand), **Marc Heinrich** (Lyon), and **Michał Pilipczuk** (Warsaw)

Typical question:

given input I, list all objects of type X in I

Typical question:

given input I, list all objects of type X in I

- cycles, cliques, VCs, dominating sets, etc. of a graph

Typical question:

given input I, list all objects of type X in I

- cycles, cliques, VCs, dominating sets, etc. of a graph
- minimal transversals of an hypergraph

Typical question:

given input I, list all objects of type X in I

- cycles, cliques, VCs, dominating sets, etc. of a graph
- minimal transversals of an hypergraph
- variable assignements satisfying a formula

Typical question:

given input I, list all objects of type X in I

- cycles, cliques, VCs, dominating sets, etc. of a graph
- minimal transversals of an hypergraph
- variable assignements satisfying a formula
- minimal story arc sets in a metabolic network

Typical question:

given input I, list all objects of type X in I

- cycles, cliques, VCs, dominating sets, etc. of a graph
- minimal transversals of an hypergraph
- variable assignements satisfying a formula
- minimal story arc sets in a metabolic network
- trains from Paris to Lyon leaving tomorrow before 10:00

Typical question:

given input I, list all objects of type X in I

- cycles, cliques, VCs, dominating sets, etc. of a graph
- minimal transversals of an hypergraph
- variable assignements satisfying a formula
- minimal story arc sets in a metabolic network
- trains from Paris to Lyon leaving tomorrow before 10:00
- shortest paths from IRIF to Jussieu

Typical question:

given input I, list all objects of type X in I

- cycles, cliques, VCs, dominating sets, etc. of a graph
- minimal transversals of an hypergraph
- variable assignements satisfying a formula
- minimal story arc sets in a metabolic network
- trains from Paris to Lyon leaving tomorrow before 10:00
- shortest paths from IRIF to Jussieu

Problem: possibly many objects!

Input-sensitive: in terms of input size

Input-sensitive: in terms of input size

- [Fomin, Grandoni, and Pyatkin, 2008]:
  $O\left(1.7159^n\right)$-time for Minimal Dominating Set

Input-sensitive: in terms of input size

- [Fomin, Grandoni, and Pyatkin, 2008]:
  $O(1.7159^n)$-time for Minimal Dominating Set
- upper-bounds the number of objects

Two perspectives about complexity

Input-sensitive: in terms of input size

- [Fomin, Grandoni, and Pyatkin, 2008]: $O\left(1.7159^n\right)$-time for Minimal Dominating Set
- upper-bounds the number of objects

Output-sensitive: in terms of input+output size

Input-sensitive: in terms of input size

- [Fomin, Grandoni, and Pyatkin, 2008]:
  $O(1.7159^n)$-time for Minimal Dominating Set
- upper-bounds the number of objects

Output-sensitive: in terms of input+output size

- [Fredman and Khachiyan, 1996]:
  $O(s^{\log s})$-time for Minimal Dominating Set
  $s = \text{poly}\,|G| + |\mathcal{D}(G)|$

"Fast" output-sensitive algorithms

n: input size

- **output-polynomial:** poly$(n + s)$ to get all the solutions

  s: number of solutions

n: input size

- **output-polynomial:** $\text{poly}(n + s)$ to get all the solutions
  s: number of solutions
- **polynomial delay:** $\text{poly}(n)$ for each new solution

n: input size

- **output-polynomial:** $\mathrm{poly}(n+s)$ to get all the solutions
  s: number of solutions
- **polynomial delay:** $\mathrm{poly}(n)$ for each new solution
- **linear delay:** $O(n)$ for each new solution

n: input size

- **output-polynomial:** $\text{poly}(n + s)$ to get all the solutions
  s: number of solutions
- **polynomial delay:** $\text{poly}(n)$ for each new solution
- **linear delay:** $O(n)$ for each new solution
- incremental poly. delay, constant delay,...

"FAST" OUTPUT-SENSITIVE ALGORITHMS

n: input size

- **output-polynomial:** $\text{poly}(n+s)$ to get all the solutions
  s: number of solutions
- **polynomial delay:** $\text{poly}(n)$ for each new solution
- **linear delay:** $O(n)$ for each new solution
- incremental poly. delay, constant delay,...

Note: output space is not counted.

Hypergraph transversal enumeration problem ($\textbf{TransEnum}$)

Input a hypergraph (V, E)

Hypergraph transversal enumeration problem (TransEnum)

Input a hypergraph (V, E)

Output every inclusion-min. $S \subseteq V$ that meets every $e \in E$

Hypergraph transversal enumeration problem (TransEnum)

Input a hypergraph (V, E)

Output every inclusion-min. $S \subseteq V$ that meets every $e \in E$

output-polynomial time algorithm?

Hypergraph transversal enumeration problem (TransEnum)

Input a hypergraph (V, E)

Output every inclusion-min. $S \subseteq V$ that meets every $e \in E$

output-polynomial time algorithm?

- long standing open problem (>35 years)

Hypergraph transversal enumeration problem (TransEnum)

Input a hypergraph (V, E)

Output every inclusion-min. $S \subseteq V$ that meets every $e \in E$

output-polynomial time algorithm?

- long standing open problem (>35 years)
- Fredman-Khachiyan 1996: $s^{O(\log s/\log\log s)}$ (s = |in|+|out|)

Hypergraph transversal enumeration problem (TransEnum)

Input a hypergraph (V, E)

Output every inclusion-min. $S \subseteq V$ that meets every $e \in E$

output-polynomial time algorithm?

- long standing open problem (>35 years)
- Fredman-Khachiyan 1996: $s^{O(\log s / \log \log s)}$ (s = |in|+|out|)
- connections with several problems in logic, database theory, AI, etc.

Hypergraph transversal enumeration problem (TransEnum)

Input a hypergraph (V, E)

Output every inclusion-min. $S \subseteq V$ that meets every $e \in E$

output-polynomial time algorithm?

- long standing open problem (>35 years)
- Fredman-Khachiyan 1996: $s^{O(\log s / \log \log s)}$ $(s = |in| + |out|)$
- connections with several problems in logic, database theory, AI, etc. and **graph theory**

Minimal dominating sets

- $N(v)$: neighbors of v

- $N(v)$: neighbors of v
- dominating set (DS): $D \subseteq V(G)$ s.t. $V(G) = D \cup N(D)$

  "D can see everybody else"

- $N(v)$: neighbors of v
- dominating set (DS): $D \subseteq V(G)$ s.t. $V(G) = D \cup N(D)$

  "D can see everybody else"
- minimal dominating set: inclusion-wise minimal DS

- $N(v)$: neighbors of v
- dominating set (DS): $D \subseteq V(G)$ s.t. $V(G) = D \cup N(D)$

  "D can see everybody else"
- minimal dominating set: inclusion-wise minimal DS

Minimal dominating set enumeration problem (DomEnum)

Input a graph G

Output every minimal dominating set of G

- $N(v)$: neighbors of v
- dominating set (DS): $D \subseteq V(G)$ s.t. $V(G) = D \cup N(D)$

  "D can see everybody else"
- minimal dominating set: inclusion-wise minimal DS

Minimal dominating set enumeration problem (DomEnum)

Input a graph G

Output every minimal dominating set of G

minimal transversals of $\{\{v\} \cup N(v),\ v \in V(G)\}$

- $N(v)$: neighbors of v
- dominating set (DS): $D \subseteq V(G)$ s.t. $V(G) = D \cup N(D)$

  "D can see everybody else"
- minimal dominating set: inclusion-wise minimal DS

Minimal dominating set enumeration problem (DomEnum)

Input a graph G

Output every minimal dominating set of G

minimal transversals of $\{\{v\} \cup N(v),\ v \in V(G)\}$

=

minimal DS of G

- $N(v)$: neighbors of v
- dominating set (DS): $D \subseteq V(G)$ s.t. $V(G) = D \cup N(D)$

  "D can see everybody else"
- minimal dominating set: inclusion-wise minimal DS

Minimal dominating set enumeration problem (DomEnum)

Input a graph G

Output every minimal dominating set of G

minimal transversals of $\{\{v\} \cup N(v),\ v \in V(G)\}$

=

minimal DS of G

$\rightsquigarrow$ DomEnum $\subseteq$ TransEnum

Theorem (Kanté, Limouzy, Mary, Nourine, 2014)

$TransEnum \subseteq (DomEnum$ *in co-bipartite graphs)*

Theorem (Kanté, Limouzy, Mary, Nourine, 2014)

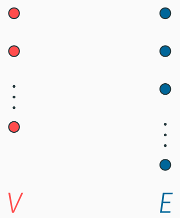
$TransEnum \subseteq (DomEnum$ in co-bipartite graphs)

instance of TransEnum $\mathcal{H} = (V, E) \rightsquigarrow$ instance of DomEnum G

Theorem (Kanté, Limouzy, Mary, Nourine, 2014)

$TransEnum \subseteq (DomEnum$ in co-bipartite graphs)

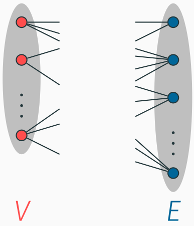
instance of $\mathrm{TransEnum}$ $\mathcal{H} = (V, E) \rightsquigarrow$ instance of $\mathrm{DomEnum}$ G



Theorem (Kanté, Limouzy, Mary, Nourine, 2014)

$\mathit{TransEnum} \subseteq (\mathit{DomEnum}$ *in co-bipartite graphs*$)$

instance of $\mathrm{TransEnum}$ $\mathcal{H} = (V, E) \rightsquigarrow$ instance of $\mathrm{DomEnum}$ G



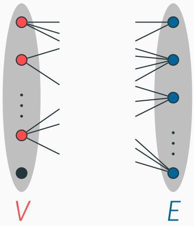
$ve \in E(G)$ si $v \in e$

+ all edges in V and in E

Theorem (Kanté, Limouzy, Mary, Nourine, 2014)

$TransEnum \subseteq (DomEnum$ *in co-bipartite graphs)*

instance of TransEnum $\mathcal{H} = (V, E) \rightsquigarrow$ instance of DomEnum G



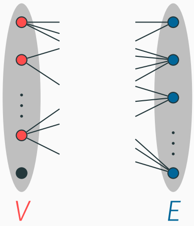
$ve \in E(G)$ si $v \in e$

+ all edges in V and in E

Theorem (Kanté, Limouzy, Mary, Nourine, 2014)

$\mathit{TransEnum} \subseteq (\mathit{DomEnum}$ *in co-bipartite graphs)*

instance of $\mathrm{TransEnum}$ $\mathcal{H} = (V, E) \rightsquigarrow$ instance of $\mathrm{DomEnum}$ G



$\forall D$ minimal dominating set of G,

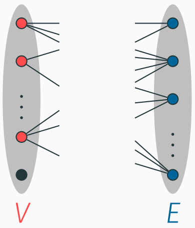
$ve \in E(G)$ si $v \in e$

\+ all edges in V and in E

Theorem (Kanté, Limouzy, Mary, Nourine, 2014)

TransEnum $\subseteq$ (*DomEnum* in co-bipartite graphs)

instance of $\mathrm{TransEnum}$ $\mathcal{H} = (V, E) \rightsquigarrow$ instance of $\mathrm{DomEnum}$ G



$\forall D$ minimal dominating set of G,

- either $D = \{x, y\}$, $x \in V \cup \{\bullet\}$, $y \in E$

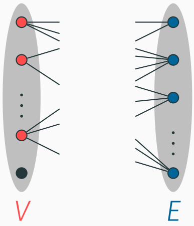
$ve \in E(G)$ si $v \in e$

+ all edges in V and in E

Theorem (Kanté, Limouzy, Mary, Nourine, 2014)

TransEnum $\subseteq$ (*DomEnum in co-bipartite graphs*)

instance of TransEnum $\mathcal{H} = (V, E) \rightsquigarrow$ instance of DomEnum G



$\forall D$ minimal dominating set of G,

- either $D = \{x, y\}$, $x \in V \cup \{\bullet\}$, $y \in E$
- or $D \subseteq V$ and is a transv. of $\mathcal{H}$

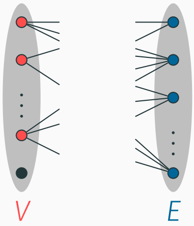
$ve \in E(G)$ si $v \in e$

\+ all edges in V and in E

Theorem (Kanté, Limouzy, Mary, Nourine, 2014)

TransEnum $\subseteq$ (*DomEnum in co-bipartite graphs*)

instance of TransEnum $\mathcal{H} = (V, E) \rightsquigarrow$ instance of DomEnum G



$\forall D$ minimal dominating set of G,

- either $D = \{x, y\}$, $x \in V \cup \{\bullet\}$, $y \in E$
- or $D \subseteq V$ and is a transv. of $\mathcal{H}$

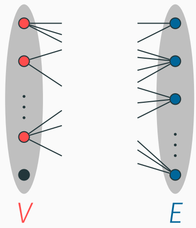
$\forall T$ minimal transversal of $\mathcal{H}$,

$ve \in E(G)$ si $v \in e$
+ all edges in V and in E

Theorem (Kanté, Limouzy, Mary, Nourine, 2014)

TransEnum $\subseteq$ (*DomEnum* *in co-bipartite graphs*)

instance of TransEnum $\mathcal{H} = (V, E) \rightsquigarrow$ instance of DomEnum G



$\forall D$ minimal dominating set of G,

- either $D = \{x, y\}$, $x \in V \cup \{\bullet\}$, $y \in E$
- or $D \subseteq V$ and is a transv. of $\mathcal{H}$

$\forall T$ minimal transversal of $\mathcal{H}$,

- T is a min. dominating set of G

$ve \in E(G)$ si $v \in e$

+ all edges in V and in E

(big) open problem: complexity of DomEnum

(big) open problem: complexity of DomEnum

Known: degenerate graphs, graphs of bounded clique-width or mim-width, split graphs, graphs of girth $\geqslant 7$, etc.

(big) open problem: complexity of DomEnum

Known: degenerate graphs, graphs of bounded clique-width or mim-width, split graphs, graphs of girth $\geqslant 7$, etc.

Question (Kanté et al. 2013): what about bipartite graphs?

(big) open problem: complexity of DomEnum

Known: degenerate graphs, graphs of bounded clique-width or mim-width, split graphs, graphs of girth $\geqslant 7$, etc.

Question (Kanté et al. 2013): what about bipartite graphs?

Theorem (Bonamy, Defrain, Heinrich, Pilipczuk, R., 2019+)

There is an algorithm enumerating minimal dominating sets of K_t-free graphs in output-polynomial time, for every $t \in \mathbb{N}$.

Bicolored graph: pair (G, A) with $A \subseteq V(G)$

Bicolored graph: pair (G, A) with $A \subseteq V(G)$

Minimal dominating set of (G, A): minimal $S \subseteq V(G)$ that dominates A

($G - A$ does not need to be dominated)

Bicolored graph: pair (G, A) with $A \subseteq V(G)$

Minimal dominating set of (G, A): minimal $S \subseteq V(G)$ that dominates A

($G - A$ does not need to be dominated)

Theorem (Bonamy, Defrain, Heinrich, Pilipczuk, R., 2019+)

There is an algorithm enumerating minimal dominating sets of bicolored graphs (G, A) s.t. $G[A]$ is K_t-free in output-polynomial time, for every integer $t \geqslant 1$.

Bicolored graph: pair (G, A) with $A \subseteq V(G)$

Minimal dominating set of (G, A): minimal $S \subseteq V(G)$ that dominates A

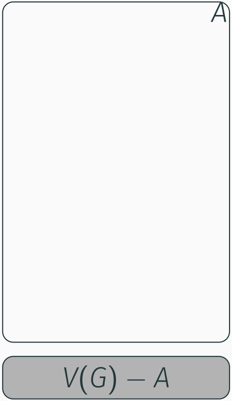
($G - A$ does not need to be dominated)

Theorem (Bonamy, Defrain, Heinrich, Pilipczuk, R., 2019+)

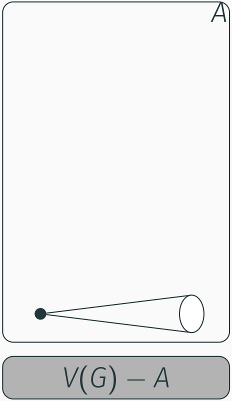
There is an algorithm enumerating minimal dominating sets of bicolored graphs (G, A) s.t. $G[A]$ is K_t-free in output-polynomial time, for every integer $t \geqslant 1$.

- $t = 1$: trivial
- $t > 1$: induction

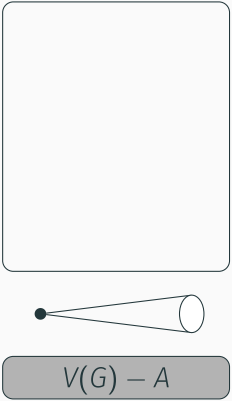
Goal: enumeration of minimal DS in (G, A) s.t. $G[A]$ is K_t-free



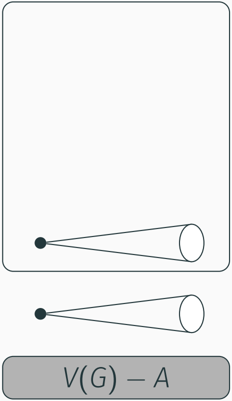
Goal: enumeration of minimal DS in (G, A) s.t. $G[A]$ is K_t-free



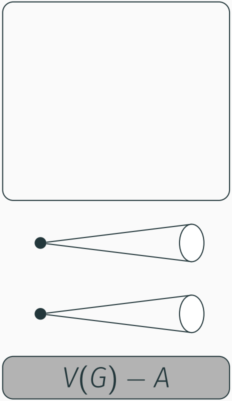
Goal: enumeration of minimal DS in (G, A) s.t. $G[A]$ is K_t-free



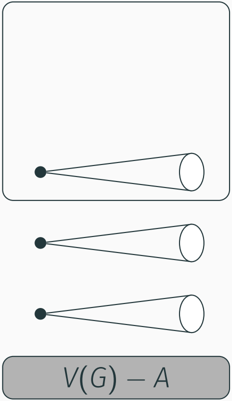
Goal: enumeration of minimal DS in (G, A) s.t. $G[A]$ is K_t-free



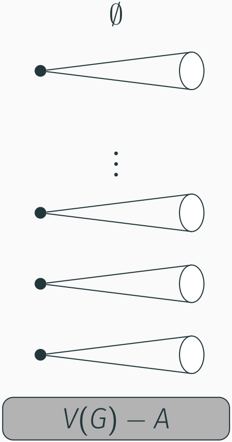
Goal: enumeration of minimal DS in (G, A) s.t. $G[A]$ is K_t-free



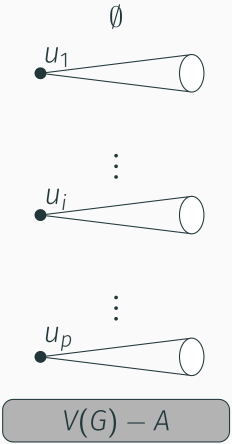
Goal: enumeration of minimal DS in (G, A) s.t. $G[A]$ is K_t-free



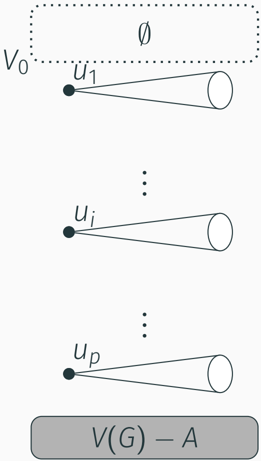
Goal: enumeration of minimal DS in (G, A) s.t. $G[A]$ is K_t-free



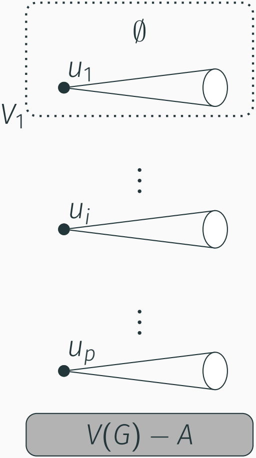
Goal: enumeration of minimal DS in (G, A) s.t. $G[A]$ is K_t-free



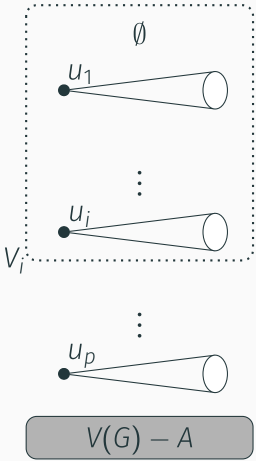
Goal: enumeration of minimal DS in (G, A) s.t. $G[A]$ is K_t-free



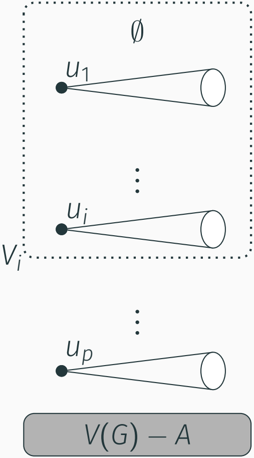
Goal: enumeration of minimal DS in (G, A) s.t. $G[A]$ is K_t-free



Goal: enumeration of minimal DS in (G, A) s.t. $G[A]$ is K_t-free



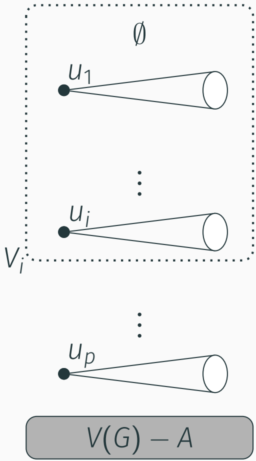
Goal: enumeration of minimal DS in (G, A) s.t. $G[A]$ is K_t-free



Plan:

1. enumerate the minimal DS of V_i
2. use them to enumerate those of V_{i+1}

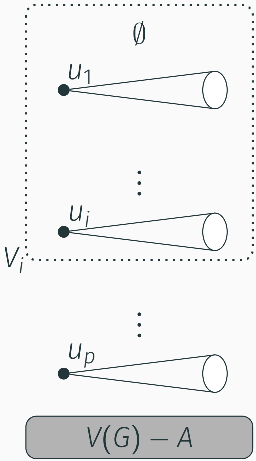
Goal: enumeration of minimal DS in (G, A) s.t. $G[A]$ is K_t-free



Plan:

1. enumerate the minimal DS of V_i with vertices of G
2. use them to enumerate those of V_{i+1}

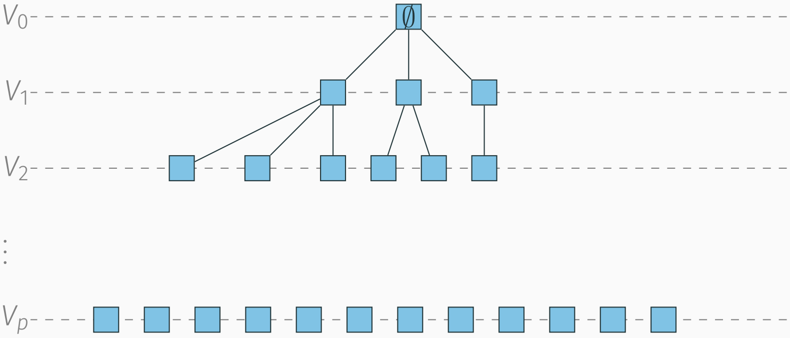
Goal: enumeration of minimal DS in (G, A) s.t. $G[A]$ is K_t-free



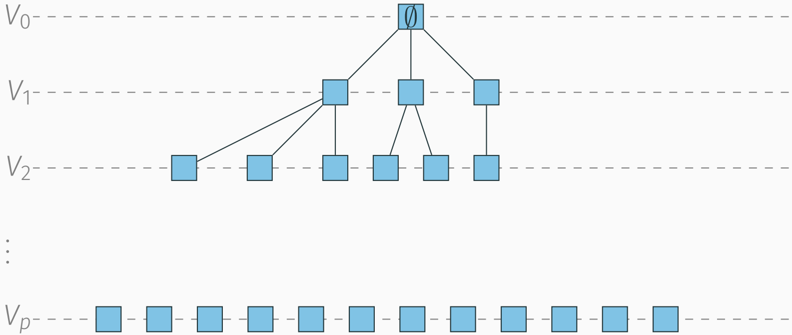
Plan:

1. enumerate the minimal DS of V_i
   with vertices of G
2. use them to enumerate those of V_{i+1}
   extend each minimal DS of V_i
   to minimal DS of V_{i+1}

Growing partial minimal dominating sets

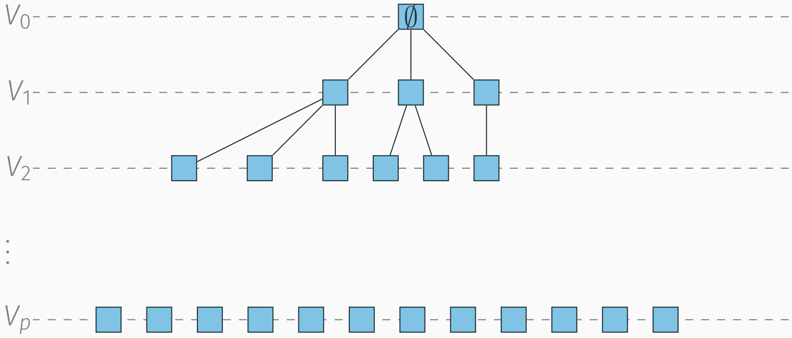


Growing partial minimal dominating sets



Important properties:

- no cycle



Important properties:

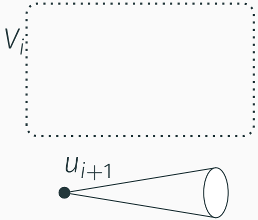
- no cycle
- no leaf before level p

From V_i to V_{i+1}

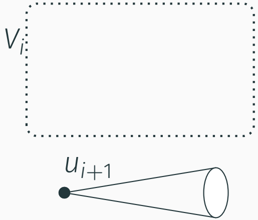
Goal: extend a minimal DS D of V_i to a minimal DS of V_{i+1}

From V_i to V_{i+1}

Goal: extend a minimal DS D of V_i to a minimal DS of V_{i+1}



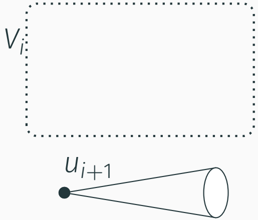
Goal: extend a minimal DS D of V_i to a minimal DS of V_{i+1}



Observation:

- possibly D dominates V_{i+1}
- if not then $D \cup \{u_{i+1}\}$ does.

Goal: extend a minimal DS D of V_i to a minimal DS of V_{i+1}



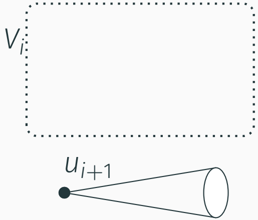
Observation:

- possibly D dominates V_{i+1}
- if not then $D \cup \{u_{i+1}\}$ does.

extension is always possible

From V_i to V_{i+1}

Goal: extend a minimal DS D of V_i to a minimal DS of V_{i+1}



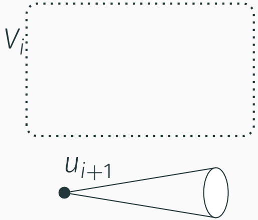
Observation:

- possibly D dominates V_{i+1}
- if not then $D \cup \{u_{i+1}\}$ does.

extension is always possible

candidate extension: minimal X s.t. $D \cup X$ dominates V_{i+1}

Goal: extend a minimal DS D of V_i to a minimal DS of V_{i+1}



Observation:

- possibly D dominates V_{i+1}
- if not then $D \cup \{u_{i+1}\}$ does.

extension is always possible

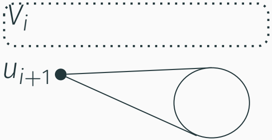
candidate extension: minimal X s.t. $D \cup X$ dominates V_{i+1}

Lemma: $|$candidate extensions of $D| \leqslant |$minimal DS of $(G, A)|$
(so we can try them all even if only few *work*)

Which are the candidate extensions of D?

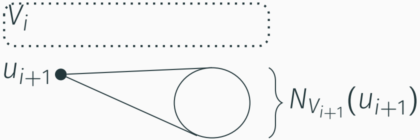
D: minimal DS of V_i

Candidate ext.: minimal X s.t. $D \cup X$ dominates V_{i+1}



D: minimal DS of V_i

Candidate ext.: minimal X s.t. $D \cup X$ dominates V_{i+1}

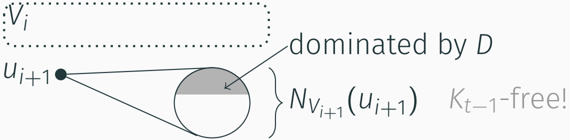


K_{t-1}-free!

Which are the candidate extensions of D?

D: minimal DS of V_i

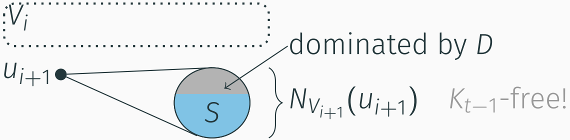
Candidate ext.: minimal X s.t. $D \cup X$ dominates V_{i+1}



Which are the candidate extensions of D?

D: minimal DS of V_i

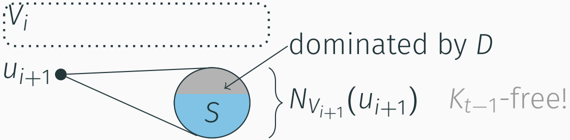
Candidate ext.: minimal X s.t. $D \cup X$ dominates V_{i+1}



Which are the candidate extensions of D?

D: minimal DS of V_i

Candidate ext.: minimal X s.t. $D \cup X$ dominates V_{i+1}

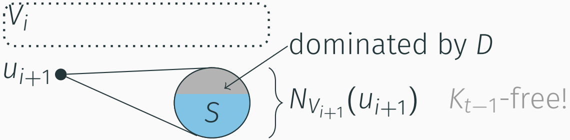


- if u_{i+1} dom. by D: X only has to dominate S

Which are the candidate extensions of D?

D: minimal DS of V_i

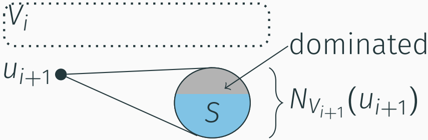
Candidate ext.: minimal X s.t. $D \cup X$ dominates V_{i+1}



- if u_{i+1} dom. by D: X only has to dominate S
  $\rightarrow$ exactly the minimal DS of (G, S)

D: minimal DS of V_i

Candidate ext.: minimal X s.t. $D \cup X$ dominates V_{i+1}



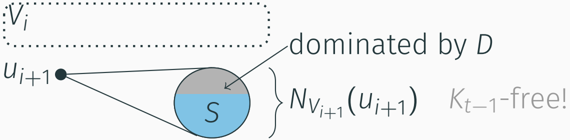
K_{t-1}-free!

- if u_{i+1} dom. by D: X only has to dominate S
  $\rightarrow$ exactly the minimal DS of (G, S)
- or u_{i+1} not dom. by D: X also need to dom. u_{i+1}

Which are the candidate extensions of D?

D: minimal DS of V_i

Candidate ext.: minimal X s.t. $D \cup X$ dominates V_{i+1}

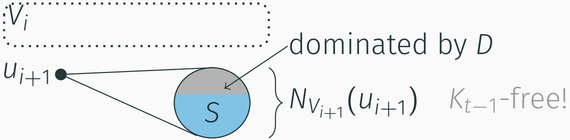


- if u_{i+1} dom. by D: X only has to dominate S
  $\rightarrow$ exactly the minimal DS of (G, S)
- or u_{i+1} not dom. by D: X also need to dom. u_{i+1}
  $$\rightarrow \; X = \{t\} \cup Q \text{ with } \begin{cases} t \in N[u_{i+1}] \\ Q \text{ min DS of } (G, S - N[t]) \end{cases}$$

Which are the candidate extensions of D?

D: minimal DS of V_i

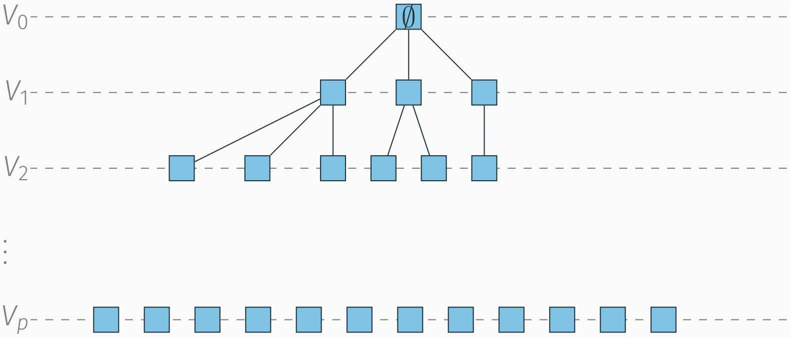
Candidate ext.: minimal X s.t. $D \cup X$ dominates V_{i+1}



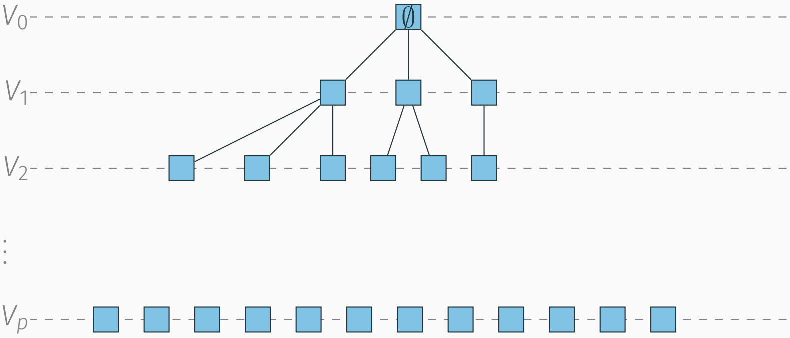
- if u_{i+1} dom. by D: X only has to dominate S
  $\rightarrow$ exactly the minimal DS of (G, S)
- or u_{i+1} not dom. by D: X also need to dom. u_{i+1}
  $\rightarrow \; X = \{t\} \cup Q$ with $\begin{cases} t \in N[u_{i+1}] \\ Q \text{ min DS of } (G, S - N[t]) \end{cases}$

In both cases: subproblem (G, A') where A' is K_{t-1}-free

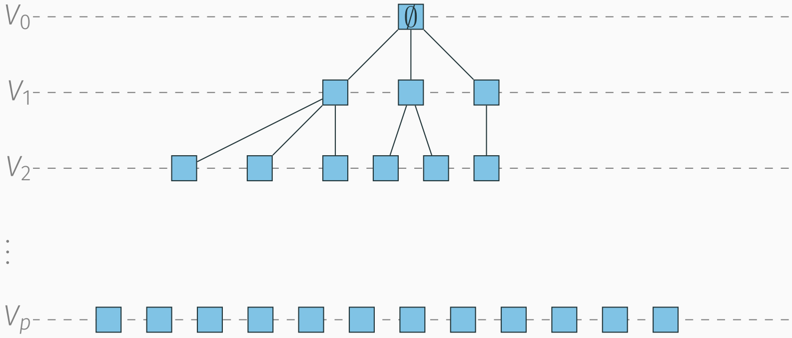
The algorithm



The algorithm

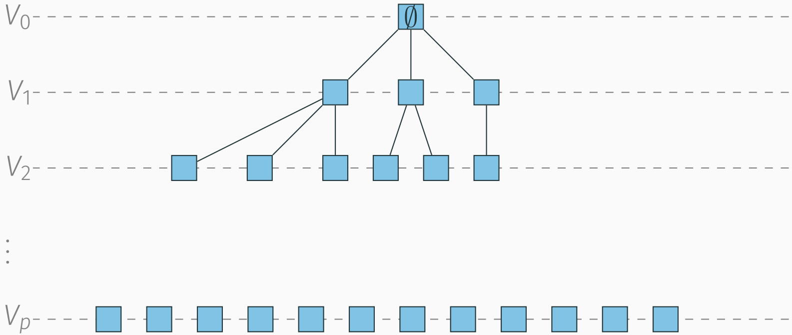


For each D minimal DS of V_i:



For each D minimal DS of V_i:

- compute all candidate extensions;



For each D minimal DS of V_i:

- compute all candidate extensions;
- only keep the $X \cup D$'s that are minimal and children of D.

Theorem (Bonamy, Defrain, Heinrich, Pilipczuk, R., 2019+)

There is an algorithm enumerating minimal dominating sets of bicolored graphs (G, A) s.t. $G[A]$ is K_t-free in output-polynomial time and polynomial space, for every integer $t \geqslant 1$.

Theorem (Bonamy, Defrain, Heinrich, Pilipczuk, R., 2019+)
There is an algorithm enumerating minimal dominating sets of bicolored graphs (G, A) s.t. $G[A]$ is K_t-free in output-polynomial time and polynomial space, for every integer $t \geqslant 1$.

- complexity improvements?

Theorem (Bonamy, Defrain, Heinrich, Pilipczuk, R., 2019+)
There is an algorithm enumerating minimal dominating sets of bicolored graphs (G, A) s.t. $G[A]$ is K_t-free in output-polynomial time and polynomial space, for every integer $t \geqslant 1$.

- complexity improvements?
- extensions to other classes?

Theorem (Bonamy, Defrain, Heinrich, Pilipczuk, R., 2019+)

There is an algorithm enumerating minimal dominating sets of bicolored graphs (G, A) s.t. $G[A]$ is K_t-free in output-polynomial time and polynomial space, for every integer $t \geqslant 1$.

- complexity improvements?
- extensions to other classes?

Theorem (Bonamy, Defrain, Heinrich, R., Pilipczuk, 2019+)

Deciding if a vertex set X can be extended into a minimal dominating set is NP-complete in bipartite graphs.

Theorem (Bonamy, Defrain, Heinrich, Pilipczuk, R., 2019+)

∃ output-polynomial time and polynomial space algo. for:

- *$(K_t + e)$-free graphs;*

Theorem (Bonamy, Defrain, Heinrich, Pilipczuk, R., 2019+)

∃ output-polynomial time and polynomial space algo. for:

- *$(K_t + e)$-free graphs;*
- *-free graphs;*

Extensions

Theorem (Bonamy, Defrain, Heinrich, Pilipczuk, R., 2019+)

$\exists$ *output-polynomial time and polynomial space algo. for:*

- $(K_t + e)$*-free graphs;*
- *-free graphs;*
- *-free graphs.*

The end

Main result: output-polynomial algorithm for the enumeration of minimal dominating sets in K_t-free graphs

Main result: output-polynomial algorithm for the enumeration of minimal dominating sets in K_t-free graphs

Open problems:

Main result: output-polynomial algorithm for the enumeration of minimal dominating sets in K_t-free graphs

Open problems:

- poly delay algo for K_t-free graphs? For bipartite graphs?

Main result: output-polynomial algorithm for the enumeration of minimal dominating sets in K_t-free graphs

Open problems:

- poly delay algo for K_t-free graphs? For bipartite graphs?
- output-poly algo for H-free graphs (C_4, $K_p + K_q$ for $p, q \geqslant 3$, etc.)

Main result: output-polynomial algorithm for the enumeration of minimal dominating sets in K_t-free graphs

Open problems:

- poly delay algo for K_t-free graphs? For bipartite graphs?
- output-poly algo for H-free graphs (C_4, $K_p + K_q$ for $p, q \geqslant 3$, etc.)
- other classes: unit-disk graphs, comparability graphs, etc.

Main result: output-polynomial algorithm for the enumeration of minimal dominating sets in K_t-free graphs

Open problems:

- poly delay algo for K_t-free graphs? For bipartite graphs?
- output-poly algo for H-free graphs (C_4, $K_p + K_q$ for $p, q \geqslant 3$, etc.)
- other classes: unit-disk graphs, comparability graphs, etc.
- output-poly algo or hardness proof for the general case.

Main result: output-polynomial algorithm for the enumeration of minimal dominating sets in K_t-free graphs

Open problems:

- poly delay algo for K_t-free graphs? For bipartite graphs?
- output-poly algo for H-free graphs (C_4, $K_p + K_q$ for $p, q \geqslant 3$, etc.)
- other classes: unit-disk graphs, comparability graphs, etc.
- output-poly algo or hardness proof for the general case.

Thank you!