Enumerating minimal dominating sets in the incomparability graphs of bounded dimension posets

Marthe Bonamy, Oscar Defrain,

Piotr Micek, and Lhouari Nourine

LaBRI, CNRS, Université de Bordeaux, France

MIMUW, University of Warsaw, Poland

Jagiellonian University, Cracow, Poland

LIMOS, Université Clermont Auvergne, France

JGA 2020

November 16, 2020

Enumeration problems

Typical question:

Given input I, list all objects of type X in I.

Enumeration problems

Typical question:

Given input I, list all objects of type X in I.

Examples:

- cycles, cliques, stable sets, dominating sets of a graph

Enumeration problems

Typical question:

Given input I, list all objects of type X in I.

Examples:

- cycles, cliques, stable sets, dominating sets of a graph
- transversals of a hypergraph

Enumeration problems

Typical question:

Given input I, list all objects of type X in I.

Examples:

- cycles, cliques, stable sets, dominating sets of a graph
- transversals of a hypergraph
- antichains of a partial order

Enumeration problems

Typical question:

Given input I, list all objects of type X in I.

Examples:

- cycles, cliques, stable sets, dominating sets of a graph
- transversals of a hypergraph
- antichains of a partial order
- variable assignments satisfying a formula

Enumeration problems

Typical question:

Given input I, list all objects of type X in I.

Examples:

- cycles, cliques, stable sets, dominating sets of a graph
- transversals of a hypergraph
- antichains of a partial order
- variable assignments satisfying a formula
- answers to a query

Enumeration problems

Typical question:

Given input I, list all objects of type X in I.

Examples:

- cycles, cliques, stable sets, dominating sets of a graph
- transversals of a hypergraph
- antichains of a partial order
- variable assignments satisfying a formula
- answers to a query
- trains to Paris leaving tomorrow before 10:00
- ...

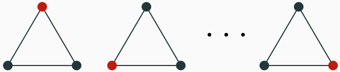
Typical question:

Given input I, list all objects of type X in I.

Examples:

- cycles, cliques, stable sets, dominating sets of a graph
- transversals of a hypergraph
- antichains of a partial order
- variable assignments satisfying a formula
- answers to a query
- trains to Paris leaving tomorrow before 10:00
- ...

Remark: possibly many objects!
$3^{n/3} \approx 1.4422^n$



Two perspectives about complexity

Input-sensitive: in terms of input size

Theorem (Fomin, Grandoni, Pyatkin, and Stepanov, 2008)

There is an $O(1.7159^n)$-time algorithm enumerating all minimal dominating sets in n-vertex graphs.

Two perspectives about complexity

Input-sensitive: in terms of input size

Theorem (Fomin, Grandoni, Pyatkin, and Stepanov, 2008)

There is an $O(1.7159^n)$-time algorithm enumerating all minimal dominating sets in n-vertex graphs.

$\rightarrow$ *basically upper-bounds the number of objects*

Two perspectives about complexity

Input-sensitive: in terms of input size

Theorem (Fomin, Grandoni, Pyatkin, and Stepanov, 2008)

There is an $O(1.7159^n)$-time algorithm enumerating all minimal dominating sets in n-vertex graphs.

$\rightarrow$ *basically upper-bounds the number of objects*

Output-sensitive: in terms of input+output size

Theorem (Fredman and Khachiyan, 1996)

There is a $N^{o(\log N)}$-time algorithm enumerating all minimal dominating sets in n-vertex graphs, where $N = n + |\mathcal{D}(G)|$.

Two perspectives about complexity

Input-sensitive: in terms of input size

Theorem (Fomin, Grandoni, Pyatkin, and Stepanov, 2008)

There is an $O\left(1.7159^n\right)$-time algorithm enumerating all minimal dominating sets in n-vertex graphs.

$\rightarrow$ *basically upper-bounds the number of objects*

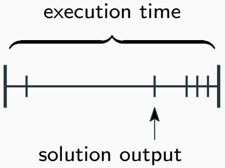
Output-sensitive: in terms of input+output size

Theorem (Fredman and Khachiyan, 1996)

There is a $N^{o(\log N)}$-time algorithm enumerating all minimal dominating sets in n-vertex graphs, where $N = n + |\mathcal{D}(G)|$.

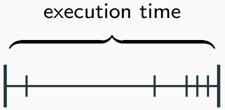
$\rightarrow$ *many techniques (reverse search, flashlight search, ordered generation, proximity search, etc.)*

Let n be input size, e.g., number of vertices of a graph G
Let d be output size, e.g., number of maximal cliques in G

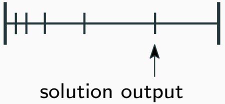


output-polynomial
algo. stops in poly$(n + d)$*-time*

Let n be input size, e.g., number of vertices of a graph G
Let d be output size, e.g., number of maximal cliques in G



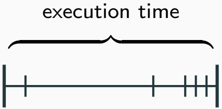
output-polynomial
algo. stops in poly$(n+d)$*-time*



incremental-polynomial
outputs i^{th} *solution in* poly$(n+i)$*-time*

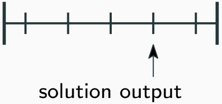
"Fast" output-sensitive algorithms

Let n be input size, e.g., number of vertices of a graph G
Let d be output size, e.g., number of maximal cliques in G



output-polynomial
algo. stops in poly$(n + d)$*-time*

incremental-polynomial
outputs i^{th} *solution in* poly$(n + i)$*-time*

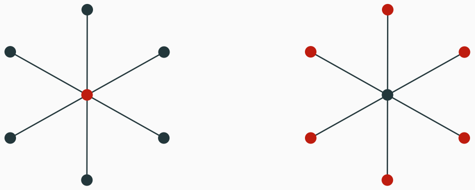


polynomial-delay
poly(n)*-time between two cons. outputs*

Minimal dominating sets

- $N(v)$: neighborhood of vertex v, $N[v] = N(v) \cup \{v\}\}$
- 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



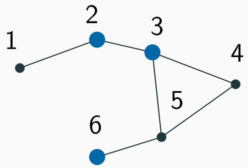
- $N(S) = \bigcup_{v \in S} N(v) \setminus S$: neighborhood of vertex set S
- dominating set (DS): $D \subseteq V(G)$ s.t. $V(G) = D \cup N(D)$

  "D can see everybody else"

- minimal dominating set: inclusion-wise minimal DS
- private neighbors $\mathrm{Priv}(D, v)$ of $v \in D$:

  vertices that are $\begin{cases} \text{dominated by } v, \text{ and} \\ \text{not dominated by } D \setminus \{v\} \end{cases}$ (possibly v)

- irredundant set: $S \subseteq V(G)$ s.t. every $x \in S$ has a priv. neighbor



- $N(S) = \bigcup_{v \in S} N(v) \setminus S$: neighborhood of vertex set S
- 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
- private neighbors $\mathrm{Priv}(D, v)$ of $v \in D$:

  vertices that are $\begin{cases} \text{dominated by } v, \text{ and} \\ \text{not dominated by } D \setminus \{v\} \end{cases}$ (possibly v)
- irredundant set: $S \subseteq V(G)$ s.t. every $x \in S$ has a priv. neighbor

Observation ★

A DS is minimal if and only if it is irredundant.

if all its vertices have a private neighbor.

if $\mathrm{Priv}(D, v) \neq \emptyset$ *for all* $v \in D$

Minimal DS Enumeration (Dom-Enum)

input: a n-vertex graph G.

output: the set $\mathcal{D}(G)$ of minimal DS of G.

Minimal DS Enumeration (Dom-Enum)

input: a n-vertex graph G.

output: the set $\mathcal{D}(G)$ of minimal DS of G.

Dream goal: an output-poly. poly(N) algorithm, $N = n + |\mathcal{D}(G)|$

Minimal DS Enumeration (Dom-Enum)

input: a n-vertex graph G.

output: the set $\mathcal{D}(G)$ of minimal DS of G.

Dream goal: an output-poly. poly(N) algorithm, $N = n + |\mathcal{D}(G)|$

General case: open, best is quasi-polynomial $N^{o(\log N)}$

Minimal DS Enumeration (Dom-Enum)

input: a n-vertex graph G.

output: the set $\mathcal{D}(G)$ of minimal DS of G.

Dream goal: an output-poly. poly(N) algorithm, $N = n + |\mathcal{D}(G)|$

General case: open, best is quasi-polynomial $N^{o(\log N)}$

Hard case: co-bipartite graphs

Minimal DS enumeration (Dom-Enum)

Minimal DS Enumeration (Dom-Enum)

input: a n-vertex graph G.

output: the set $\mathcal{D}(G)$ of minimal DS of G.

Dream goal: an output-poly. poly(N) algorithm, $N = n + |\mathcal{D}(G)|$

General case: open, best is quasi-polynomial $N^{o(\log N)}$

Hard case: co-bipartite graphs

Known cases:

- **output poly.:** log(n)-degenerate graphs, Kt-free graphs

Minimal DS Enumeration (Dom-Enum)

input: a n-vertex graph G.

output: the set $\mathcal{D}(G)$ of minimal DS of G.

Dream goal: an output-poly. poly(N) algorithm, $N = n + |\mathcal{D}(G)|$

General case: open, best is quasi-polynomial $N^{o(\log N)}$

Hard case: co-bipartite graphs

Known cases:

- **output poly.:** log(n)-degenerate graphs, Kt-free graphs
- **incr. poly.:** chordal bipartite graphs, bounded conformality graphs

Minimal DS Enumeration (Dom-Enum)

input: a n-vertex graph G.

output: the set $\mathcal{D}(G)$ of minimal DS of G.

Dream goal: an output-poly. poly(N) algorithm, $N = n + |\mathcal{D}(G)|$

General case: open, best is quasi-polynomial $N^{o(\log N)}$

Hard case: co-bipartite graphs

Known cases:

- **output poly.:** log(n)-degenerate graphs, Kt-free graphs
- **incr. poly.:** chordal bipartite graphs, bounded conformality graphs
- **poly. delay:** degenerate, line, and chordal graphs

Minimal DS Enumeration (Dom-Enum)

input: a n-vertex graph G.

output: the set $\mathcal{D}(G)$ of minimal DS of G.

Dream goal: an output-poly. poly(N) algorithm, $N = n + |\mathcal{D}(G)|$

General case: open, best is quasi-polynomial $N^{o(\log N)}$

Hard case: co-bipartite graphs

Known cases:

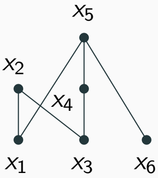
- **output poly.:** log(n)-degenerate graphs, Kt-free graphs
- **incr. poly.:** chordal bipartite graphs, bounded conformality graphs
- **poly. delay:** degenerate, line, and chordal graphs
- **linear delay:** permutation and interval graphs, etc.

Posets & (In)comparability graphs

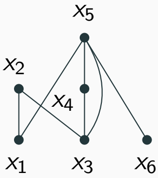
- poset $P = (V, \leq)$: refl., trans., antisymmetric relation $\leq$ on V

  $x \leq x \qquad x \leq y \wedge y \leq x \implies x = y$
- comp(P): graph on V s.t. $uv \in E$ if $u \leq v$ or $v \leq u$
- incomp(P): complementary of comp(P)

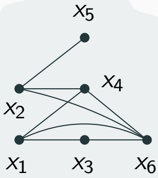
  i.e., an edge for every incomparable pair of elements



$P = (V, \leq)$



$comp(P)$



$incomp(P)$

Minimal DS Enumeration (Dom-Enum)

input: a n-vertex graph G.

output: the set $\mathcal{D}(G)$ of minimal DS of G.

Dream goal: an output-poly. poly(N) algorithm, $N = n + |\mathcal{D}(G)|$

General case: open, best is quasi-polynomial $N^{o(\log N)}$

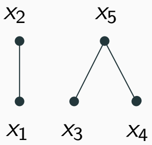
Hard case: co-bipartite graphs, hence incomparability graphs

Known cases: ⚠

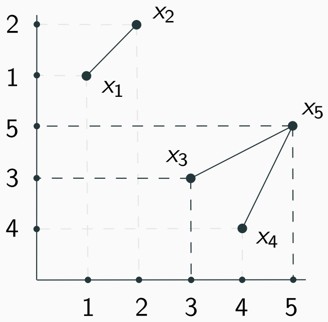
- **output poly.:** log(n)-degenerate graphs, Kt-free graphs
- **incr. poly.:** chordal bipartite graphs, bounded conformality graphs
- **poly. delay:** degenerate, line, and chordal graphs
- **linear delay:** permutation and interval graphs, etc.

Poset dimension of $P = (V, \leq)$:

Least integer d such that elements of P can be embedded into $\mathbb{R}^d$ in such a way that $x \leq y$ in P if and only if the point of x is below the point of y with respect to the product order of $\mathbb{R}^d$

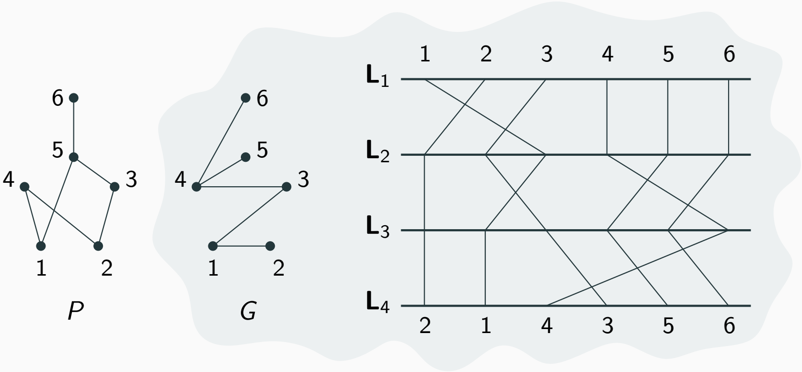


$P = (V, \leq)$



Theorem (Golumbic, Rotem, and Urrutia, 1983)

A graph G is the incomparability graph of a poset of dimension d if and only if it is the intersection graph of the concatenation of d permutation diagrams.



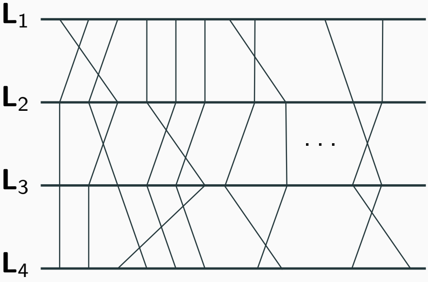
General idea of the algorithm

Observation ★

A DS is minimal if and only if it is irredundant.

if all its vertices have a private neighbor.

- make grow irredundant sets to minimal dominating sets
- ensure that each constructed partial set leads to a solution



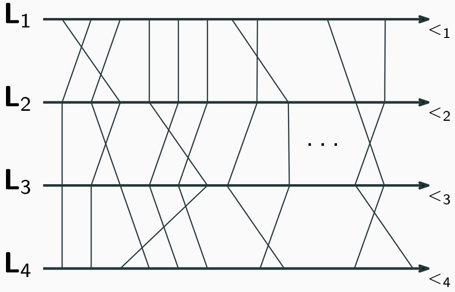
Observation ★

A DS is minimal if and only if it is irredundant.

if all its vertices have a private neighbor.

- make grow irredundant sets to minimal dominating sets
- ensure that each constructed partial set leads to a solution

$\rightarrow$ from left to right!



General idea of the algorithm

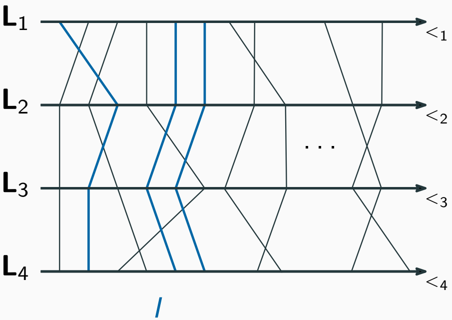
Observation ★

A DS is minimal if and only if it is irredundant.

if all its vertices have a private neighbor.

- make grow irredundant sets to minimal dominating sets
- ensure that each constructed partial set leads to a solution

$\rightarrow$ from left to right!



Definition of "right":

- $\mathsf{R}(I) = \{v \in V \setminus I : \exists j,\ \forall u \in I,\ u <_j v\}$

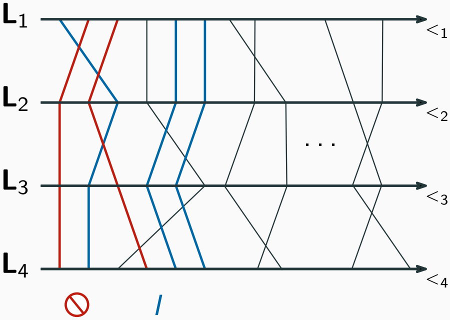
Observation ★

A DS is minimal if and only if it is irredundant.

if all its vertices have a private neighbor.

- make grow irredundant sets to minimal dominating sets
- ensure that each constructed partial set leads to a solution

$\rightarrow$ from left to right!



Definition of "right":

- $\mathsf{R}(I) = \{v \in V \setminus I :$
  $\exists j,\ \forall u \in I,\ u <_j v\}$

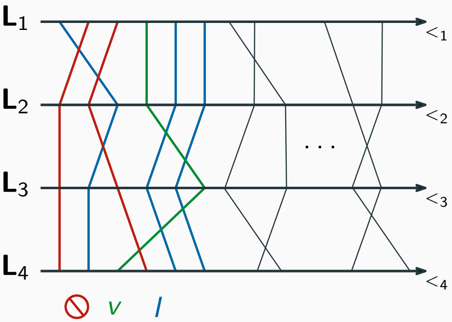
Observation ★

A DS is minimal if and only if it is irredundant.

if all its vertices have a private neighbor.

- make grow irredundant sets to minimal dominating sets
- ensure that each constructed partial set leads to a solution

$\rightarrow$ from left to right!



Definition of "right":

- $\mathsf{R}(I) = \{v \in V \setminus I : \exists j,\ \forall u \in I,\ u <_j v\}$

Observation ★

A DS is minimal if and only if it is irredundant.

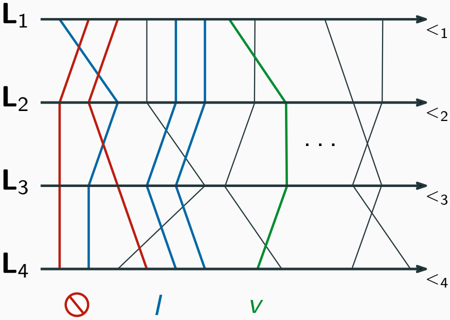
if all its vertices have a private neighbor.

- make grow irredundant sets to minimal dominating sets
- ensure that each constructed partial set leads to a solution

→ from left to right!

Definition of "right":

- $\mathsf{R}(I) = \{v \in V \setminus I :$
  $\exists j,\ \forall u \in I,\ u <_j v\}$



Right-Extension Problem:

Given $I \subseteq V(G)$, decide whether I can be *extended to the right into a min DS*, i.e., whether $\exists D \in \mathcal{D}(G)$ s.t. $I \subseteq D$ and $D \setminus I \subseteq \mathrm{R}(I)$

Right-Extension Problem:

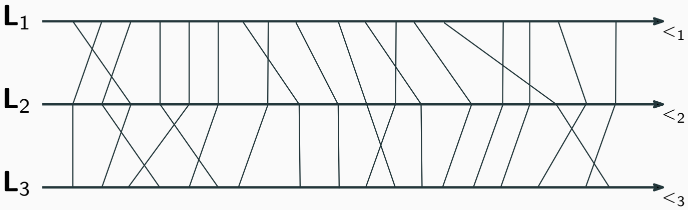
Given $I \subseteq V(G)$, decide whether I can be *extended to the right into a min DS*, i.e., whether $\exists D \in \mathcal{D}(G)$ s.t. $I \subseteq D$ and $D \setminus I \subseteq \mathrm{R}(I)$

Observation $\diamondsuit$

Set $I = \{u_1, \ldots, u_p\}$ *can be extended to the right iff*

$\exists v_1, \ldots, v_p \in \mathrm{Priv}(I, u_1) \times \cdots \times \mathrm{Priv}(I, u_p)$

s.t. $R(I) \setminus N[v_1, \ldots, v_p]$ *dominates* $G - N[I]$



Right-Extension Problem:

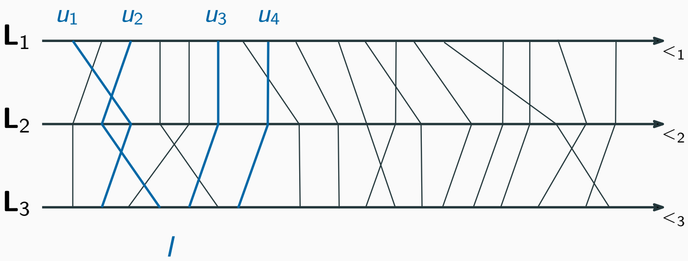
Given $I \subseteq V(G)$, decide whether I can be *extended to the right into a min DS*, i.e., whether $\exists D \in \mathcal{D}(G)$ s.t. $I \subseteq D$ and $D \setminus I \subseteq \mathrm{R}(I)$

Observation $\diamondsuit$

Set $I = \{u_1, \dots, u_p\}$ *can be extended to the right iff*

$\exists v_1, \dots, v_p \in \mathrm{Priv}(I, u_1) \times \cdots \times \mathrm{Priv}(I, u_p)$

s.t. $R(I) \setminus N[v_1, \dots, v_p]$ *dominates* $G - N[I]$



Right-Extension Problem:

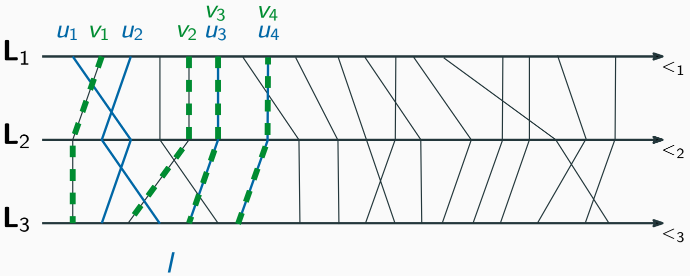
Given $I \subseteq V(G)$, decide whether I can be *extended to the right into a min DS*, i.e., whether $\exists D \in \mathcal{D}(G)$ s.t. $I \subseteq D$ and $D \setminus I \subseteq \mathrm{R}(I)$

Observation ♢

Set $I = \{u_1, \dots, u_p\}$ *can be extended to the right iff*

$$\exists v_1, \dots, v_p \in \mathrm{Priv}(I, u_1) \times \cdots \times \mathrm{Priv}(I, u_p)$$

$$\textit{s.t. } R(I) \setminus N[v_1, \dots, v_p] \textit{ dominates } G - N[I]$$



Right-Extension Problem:

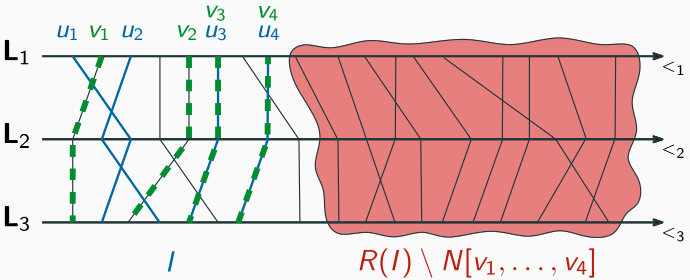
Given $I \subseteq V(G)$, decide whether I can be *extended to the right into a min DS*, i.e., whether $\exists D \in \mathcal{D}(G)$ s.t. $I \subseteq D$ and $D \setminus I \subseteq \mathrm{R}(I)$

Observation ♢

Set $I = \{u_1, \dots, u_p\}$ *can be extended to the right iff*

$$\exists v_1, \dots, v_p \in \mathrm{Priv}(I, u_1) \times \cdots \times \mathrm{Priv}(I, u_p)$$

$$\textit{s.t. } R(I) \setminus N[v_1, \dots, v_p] \textit{ dominates } G - N[I]$$



Right-Extension Problem:

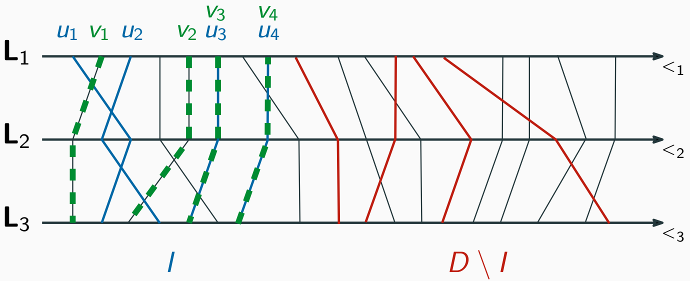
Given $I \subseteq V(G)$, decide whether I can be *extended to the right into a min DS*, i.e., whether $\exists D \in \mathcal{D}(G)$ s.t. $I \subseteq D$ and $D \setminus I \subseteq \mathrm{R}(I)$

Observation $\diamondsuit$

Set $I = \{u_1, \dots, u_p\}$ *can be extended to the right iff*

$$\exists v_1, \dots, v_p \in \mathrm{Priv}(I, u_1) \times \cdots \times \mathrm{Priv}(I, u_p)$$

$$\text{s.t. } R(I) \setminus N[v_1, \dots, v_p] \text{ dominates } G - N[I]$$



Right-Extension Problem:

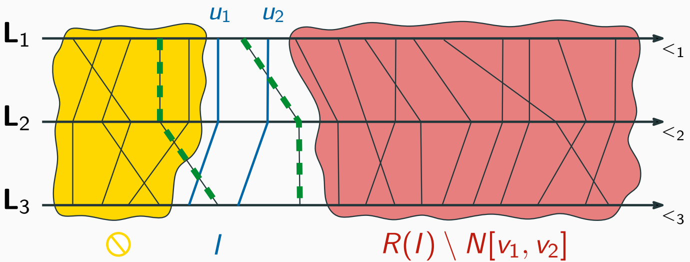
Given $I \subseteq V(G)$, decide whether I can be *extended to the right into a min DS*, i.e., whether $\exists D \in \mathcal{D}(G)$ s.t. $I \subseteq D$ and $D \setminus I \subseteq \mathrm{R}(I)$

Observation $\diamondsuit$

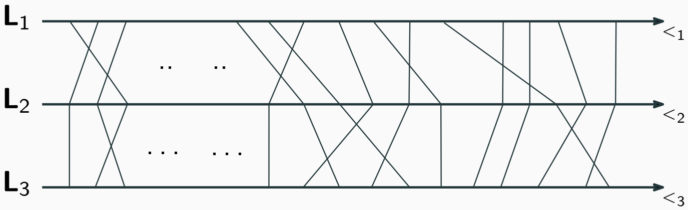
Set $I = \{u_1, \dots, u_p\}$ *can be extended to the right iff*

$\exists v_1, \dots, v_p \in \mathrm{Priv}(I, u_1) \times \dots \times \mathrm{Priv}(I, u_p)$

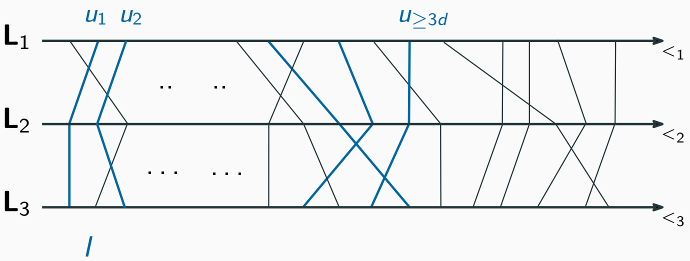
s.t. $R(I) \setminus N[v_1, \dots, v_p]$ *dominates* $G - N[I]$



Extension problem: the unbounded case $l \geq 3d$

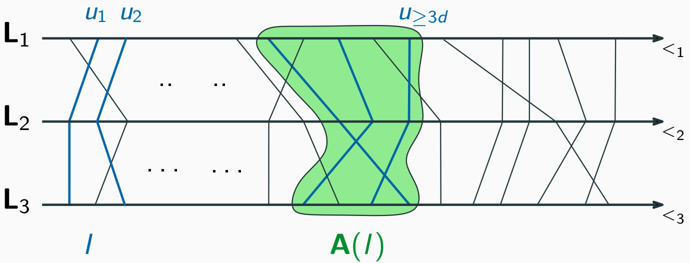


Extension problem: the unbounded case $I \geq 3d$



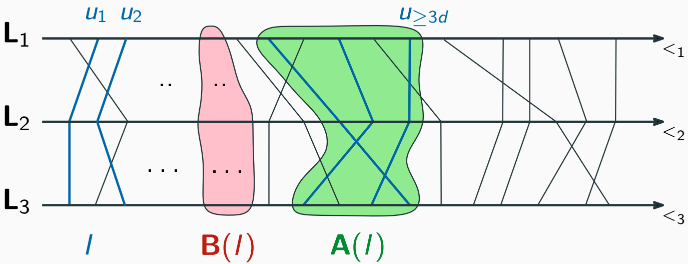
Extension problem: the unbounded case $I \geq 3d$

- 1st layer of I: $\mathbf{A}(I) = (a_1, \dots, a_d)$ so that $a_1 = \mathrm{Max}_{<_1}(I)$, and $\forall i \in \{2, \dots, d\}$, $a_i = \mathrm{Max}_{<_i}(I \setminus \{a_1, \dots, a_{i-1}\})$



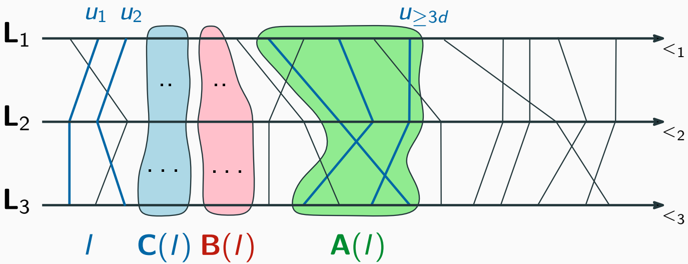
Extension problem: the unbounded case $l \geq 3d$

- 1st layer of I: $\mathbf{A}(I) = (a_1, \ldots, a_d)$ so that $a_1 = \mathrm{Max}_{<_1}(I)$, and $\forall i \in \{2, \ldots, d\}$, $a_i = \mathrm{Max}_{<_i}(I \setminus \{a_1, \ldots, a_{i-1}\})$
- $\mathbf{B}(I) = \mathbf{A}(I \setminus \mathbf{A}(I))$



Extension problem: the unbounded case $I \geq 3d$

- 1st layer of I: $\mathsf{A}(I) = (a_1, \dots, a_d)$ so that $a_1 = \mathrm{Max}_{<_1}(I)$, and $\forall i \in \{2, \dots, d\}$, $a_i = \mathrm{Max}_{<_i}(I \setminus \{a_1, \dots, a_{i-1}\})$
- $\mathsf{B}(I) = \mathsf{A}(I \setminus \mathsf{A}(I))$
- $\mathsf{C}(I) = \mathsf{A}(I \setminus (\mathsf{A}(I) \cup \mathsf{B}(I)))$



Extension problem: the unbounded case $I \geq 3d$

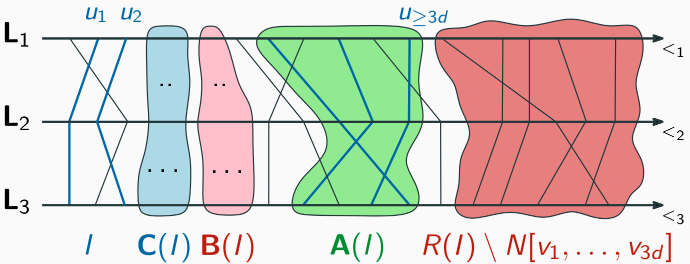
- 1st layer of I: $\mathbf{A}(I) = (a_1, \ldots, a_d)$ so that $a_1 = \mathrm{Max}_{<_1}(I)$, and $\forall i \in \{2, \ldots, d\}$, $a_i = \mathrm{Max}_{<_i}(I \setminus \{a_1, \ldots, a_{i-1}\})$
- $\mathbf{B}(I) = \mathbf{A}(I \setminus \mathbf{A}(I))$
- $\mathbf{C}(I) = \mathbf{A}(I \setminus (\mathbf{A}(I) \cup \mathbf{B}(I)))$

Theorem $\diamondsuit$

Set $I, |I| \geq 3d$ with $A \cup B \cup C = \{u_1, \ldots, u_{3d}\}$ can be ext. iff

$$\exists v_1, \ldots, v_{3d} \in \mathrm{Priv}(I, u_1) \times \cdots \times \mathrm{Priv}(I, u_{3d})$$

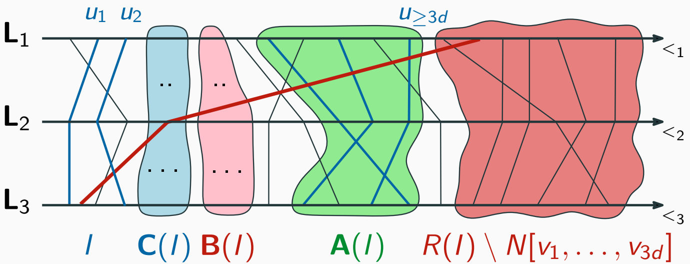
s.t. $R(I) \setminus N[v_1, \ldots, v_{3d}]$ *dominates* $G - N[I]$



- 1st layer of I: $\mathbf{A}(I) = (a_1, \dots, a_d)$ so that $a_1 = \mathrm{Max}_{<_1}(I)$, and $\forall i \in \{2, \dots, d\}$, $a_i = \mathrm{Max}_{<_i}(I \setminus \{a_1, \dots, a_{i-1}\})$
- $\mathbf{B}(I) = \mathbf{A}(I \setminus \mathbf{A}(I))$
- $\mathbf{C}(I) = \mathbf{A}(I \setminus (\mathbf{A}(I) \cup \mathbf{B}(I)))$

Theorem ♢

Set $I, |I| \geq 3d$ with $A \cup B \cup C = \{u_1, \dots, u_{3d}\}$ can be ext. iff
$\exists v_1, \dots, v_{3d} \in \mathrm{Priv}(I, u_1) \times \dots \times \mathrm{Priv}(I, u_{3d})$
s.t. $R(I) \setminus N[v_1, \dots, v_{3d}]$ *dominates* $G - N[I]$

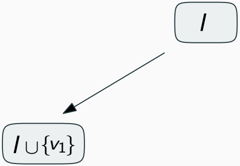


The algorithm

- start with $I = \emptyset$

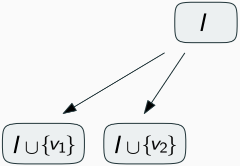
The algorithm

- start with $I = \emptyset$
- for every $v \in \mathsf{R}(I)$, check if $I' = I \cup \{v\}$ *extends to the right into a min DS*, i.e., whether $\exists D \in \mathcal{D}(G)$ s.t. $I' \subseteq D$ and $D \setminus I' \subseteq \mathsf{R}(I')$



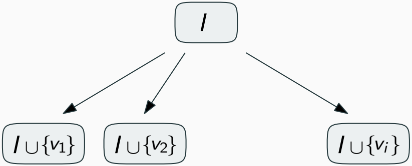
The algorithm

- start with $I = \emptyset$
- for every $v \in \mathsf{R}(I)$, check if $I' = I \cup \{v\}$ *extends to the right into a min DS*, i.e., whether $\exists D \in \mathcal{D}(G)$ s.t. $I' \subseteq D$ and $D \setminus I' \subseteq \mathsf{R}(I')$



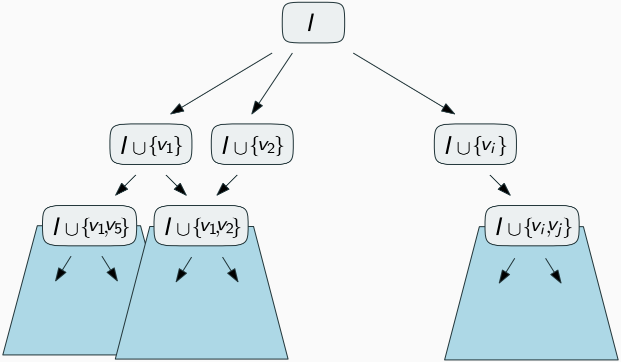
The algorithm

- start with $I = \emptyset$
- for every $v \in \mathsf{R}(I)$, check if $I' = I \cup \{v\}$ *extends to the right into a min DS*, i.e., whether $\exists D \in \mathcal{D}(G)$ s.t. $I' \subseteq D$ and $D \setminus I' \subseteq \mathsf{R}(I')$



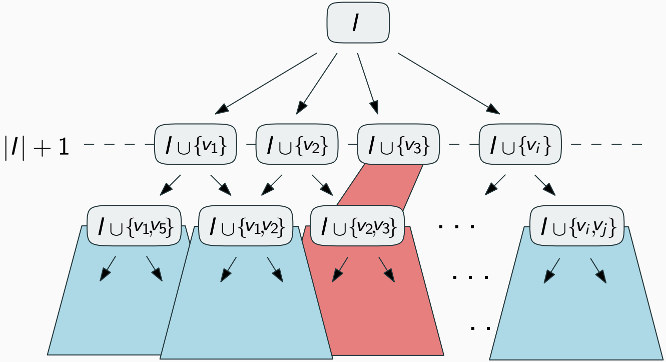
The algorithm

- start with $I = \emptyset$
- for every $v \in \mathrm{R}(I)$, check if $I' = I \cup \{v\}$ *extends to the right into a min DS*, i.e., whether $\exists D \in \mathcal{D}(G)$ s.t. $I' \subseteq D$ and $D \setminus I' \subseteq \mathrm{R}(I')$



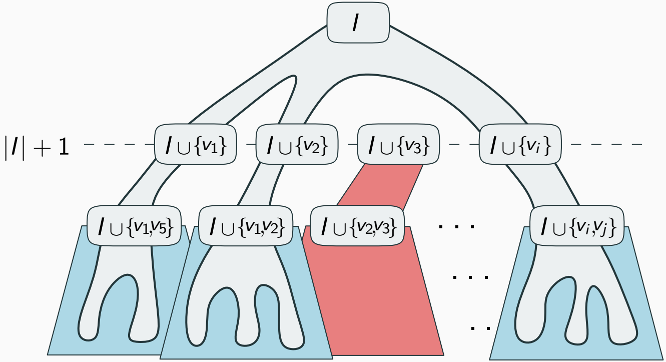
The algorithm

- start with $I = \emptyset$
- for every $v \in \mathsf{R}(I)$, check if $I' = I \cup \{v\}$ *extends to the right into a min DS*, i.e., whether $\exists D \in \mathcal{D}(G)$ s.t. $I' \subseteq D$ and $D \setminus I' \subseteq \mathsf{R}(I')$
- explore if it is the case,



The algorithm

- start with $I = \emptyset$
- for every $v \in \mathrm{R}(I)$, check if $I' = I \cup \{v\}$ *extends to the right into a min DS*, i.e., whether $\exists D \in \mathcal{D}(G)$ s.t. $I' \subseteq D$ and $D \setminus I' \subseteq \mathrm{R}(I')$
- explore if it is the case, AND, if $I = \mathrm{Parent}(I') = I' \setminus \mathrm{Max}_{<_1}(I')$



Main theorem and future work

Theorem (Bonamy, D., Micek, and Nourine)

The set $\mathcal{D}(G)$ of minimal DS of incomp. graphs of posets of dimension d can be enumerated in time $O(n^{3d+4})$ and poly. space.

Main theorem and future work

Theorem (Bonamy, D., Micek, and Nourine)

The set $\mathcal{D}(G)$ of minimal DS of incomp. graphs of posets of dimension d can be enumerated in time $O(n^{3d+4})$ and poly. space.

- complexity improvements? can we get $f(d) \cdot n^{O(1)}$ delay?

Main theorem and future work

Theorem (Bonamy, D., Micek, and Nourine)

The set $\mathcal{D}(G)$ of minimal DS of incomp. graphs of posets of dimension d can be enumerated in time $O(n^{3d+4})$ and poly. space.

- complexity improvements? can we get $f(d) \cdot n^{O(1)}$ delay?
- what about comparability graphs of bounded dimension?

Main theorem and future work

Theorem (Bonamy, D., Micek, and Nourine)

The set $\mathcal{D}(G)$ of minimal DS of incomp. graphs of posets of dimension d can be enumerated in time $O(n^{3d+4})$ and poly. space.

- complexity improvements? can we get $f(d) \cdot n^{O(1)}$ delay?
- what about comparability graphs of bounded dimension?

Theorem (Bonamy, D., Micek, and Nourine)

Minimal DS of comp. graphs of posets of dimension d can be enumerated in incremental-polynomial time and poly. space.

Main theorem and future work

Theorem (Bonamy, D., Micek, and Nourine)

The set $\mathcal{D}(G)$ of minimal DS of incomp. graphs of posets of dimension d can be enumerated in time $O(n^{3d+4})$ and poly. space.

- complexity improvements? can we get $f(d) \cdot n^{O(1)}$ delay?
- what about comparability graphs of bounded dimension?

Theorem (Bonamy, D., Micek, and Nourine)

Minimal DS of comp. graphs of posets of dimension d can be enumerated in incremental-polynomial time and poly. space.

- remaining important cases:

  C_4-free ? ✗

Main theorem and future work

Theorem (Bonamy, D., Micek, and Nourine)

The set $\mathcal{D}(G)$ of minimal DS of incomp. graphs of posets of dimension d can be enumerated in time $O(n^{3d+4})$ and poly. space.

- complexity improvements? can we get $f(d) \cdot n^{O(1)}$ delay?
- what about comparability graphs of bounded dimension?

Theorem (Bonamy, D., Micek, and Nourine)

Minimal DS of comp. graphs of posets of dimension d can be enumerated in incremental-polynomial time and poly. space.

- remaining important cases:

  C_4-free ? ✗

  (general) comparability graphs? ✗

Theorem (Bonamy, D., Micek, and Nourine)

The set $\mathcal{D}(G)$ of minimal DS of incomp. graphs of posets of dimension d can be enumerated in time $O(n^{3d+4})$ and poly. space.

- complexity improvements? can we get $f(d) \cdot n^{O(1)}$ delay?
- what about comparability graphs of bounded dimension?

Theorem (Bonamy, D., Micek, and Nourine)

Minimal DS of comp. graphs of posets of dimension d can be enumerated in incremental-polynomial time and poly. space.

- remaining important cases:

  C_4-free ? ✗

  (general) comparability graphs? ✗

  unit disk? ✗

Theorem (Bonamy, D., Micek, and Nourine)

The set $\mathcal{D}(G)$ of minimal DS of incomp. graphs of posets of dimension d can be enumerated in time $O(n^{3d+4})$ and poly. space.

- complexity improvements? can we get $f(d) \cdot n^{O(1)}$ delay?
- what about comparability graphs of bounded dimension?

Theorem (Bonamy, D., Micek, and Nourine)

Minimal DS of comp. graphs of posets of dimension d can be enumerated in incremental-polynomial time and poly. space.

- remaining important cases:

  C_4-free ? ✗

  (general) comparability graphs? ✗

  unit disk? ✗

Thank you!