Neighborhood inclusions for minimal dominating sets enumeration

Oscar Defrain and Lhouari Nourine
LIMOS, Université Clermont Auvergne, France

ISAAC 2019
Shanghai, China
December 8–11

Typical question:

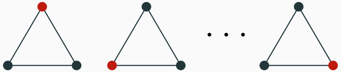
Given input I, list all objects of type X in I.

Examples:

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

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

Output-sensitive: in terms of input+output size

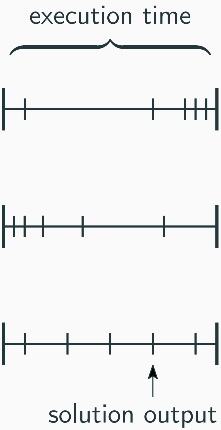
Theorem (Fredman and Khachiyan, 1996)

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

$\rightarrow$ *many techniques (reverse search, backtrack search, etc.)*

“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 dominating sets in G



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

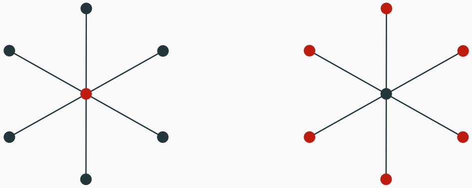
incremental-polynomial
outputs i^{th} *solution in* $\text{poly}(n+i)$*-time*

polynomial-delay
$\text{poly}(n)$*-time between any two outputs*

Minimal dominating sets

- $N[v]$: closed neighborhood of vertex v
- dominating set (DS): $D \subseteq V(G)$ s.t. $V(G) = N[D]$

  "D can see everybody else"
- minimal dominating set: inclusion-wise minimal DS
- private neighbor of $v \in D$: vertex u s.t. $N[u] \cap D = \{v\}$



Observation

A DS is minimal iff all its vertices have a private neighbor.

Minimal DS Enumeration (Dom-Enum)

input: a n-vertex graph G

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

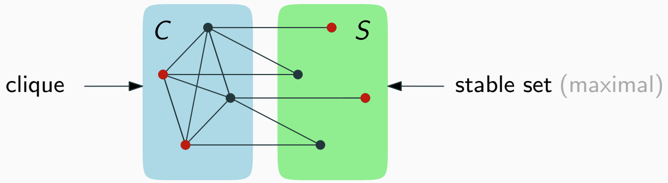
Dream goal: an output-poly. poly(N) algorithm, $N = n + |\mathcal{D}(G)|$

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

Known cases:

- **output-poly.:** $\log(n)$-degenerate graphs, K_t-free graphs for fixed t
- **incr. poly.:** chordal bipartite graphs, bounded conformality graphs
- **poly. delay:** degenerate, line, and chordal graphs
- **linear delay:** permutation and interval graphs, graphs with bounded clique-width, split and P_6-free chordal graphs

Dom-Enum in split graphs (Kanté et al., 2014)



Proposition (Kanté, Limouzy, Mary, and Nourine, 2014)

A set $D \subseteq V(G)$ is a minimal DS of G iff D dominates S and every $v \in D$ has a private neighbor in S.

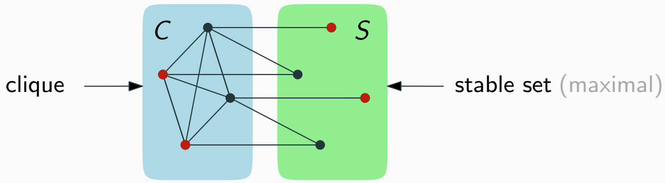
Then: $D \cap S = \{$all vertices not dominated by $D \cap C\}$

Enumeration: complete every set $X \subseteq C$ with priv. neighbors in S into a minimal DS of G

→ the family of such X's is an independence set system

→ can be enumerated with linear delay

Dom-Enum in split graphs (Kanté et al., 2014)



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

There is a linear-delay algorithm enumerating minimal dominating sets in split graphs.

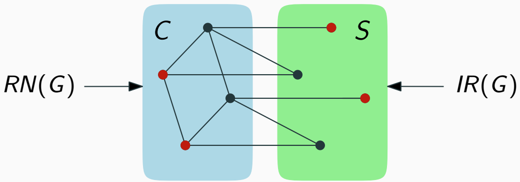
Then: $D \cap S = \{$all vertices not dominated by $D \cap C\}$

Enumeration: complete every set $X \subseteq C$ with priv. neighbors in S into a minimal DS of G

$\rightarrow$ the family of such X's is an independence set system

$\rightarrow$ can be enumerated with linear delay

Redundant and irredundant vertices



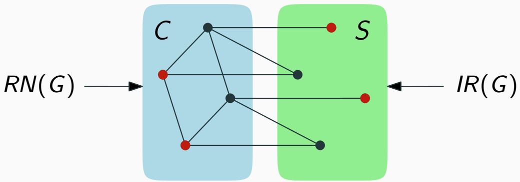
- vertex v is redundant if there exists u s.t. $N[u] \subseteq N[v]$
- vertex v is irredundant otherwise

  is minimal w.r.t. neighborhood inclusion
- $RN(G)$: the set of redundant vertices
- $IR(G)$: the set of irredundant vertices

Proposition

A set $D \subseteq V(G)$ is a minimal DS of G iff D dominates $IR(G)$ and every $v \in D$ has a priv. neighbor in $IR(G)$.

Neighborhood inclusions for Dom-Enum



For $D \subseteq V(G)$:

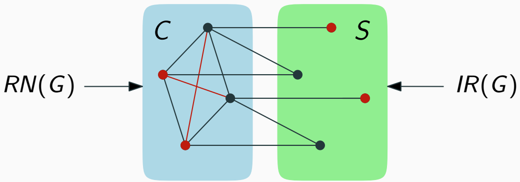
- let $D_{RN} = D \cap RN(G)$ and $D_{IR} = D \cap IR(G)$
- let $\mathcal{D}_{RN}(G) = \{D_{RN} \mid D \in \mathcal{D}(G)\}$ $\rightarrow$ an independence set system whenever G is P_7-free chordal, and an accessible set system whenever G is P_8-free chordal.

Enumeration: check for every set $A \subseteq RN(G)$

$\rightarrow$ whether $A \in \mathcal{D}_{RN}(G)$ (irredundant extension problem)

$\rightarrow$ if so, enumerate every extension $X \subseteq IR(G)$ s.t. $A \cup X \in \mathcal{D}(G)$

Case A: P_6-free chordal graphs

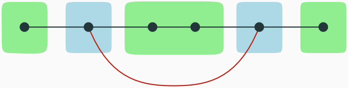


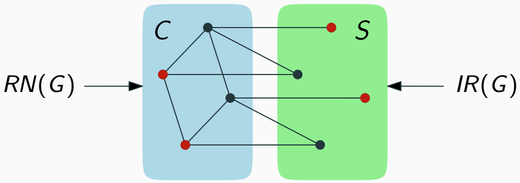
Proposition (Kanté, Limouzy, Mary, and Nourine, 2014)

Let G be a P_6-free chordal graph. Then completing $RN(G)$ into a clique yields a split graph with the same minimal DS.

→ linear-delay algorithm for Dom-Enum in P_6-free chordal graphs

→ does not hold for P_7-free chordal graphs (not even chordal)





Proposition

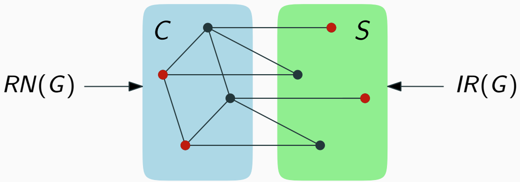
Let G be a P_k-free chordal graph, $k \in \mathbb{N}$. Then the graph $G[IR(G)]$ induced by $IR(G)$ is P_{k-4}-free chordal.

→ linear-delay algorithm for Dom-Enum in P_7-free chordal graphs

→ poly.-delay algorithm for Dom-Enum in P_8-free chordal graphs

- → checking $A \in \mathcal{D}_{RN}(G)$ is linear
- → enumerating X s.t. $A \cup X \in \mathcal{D}(G)$ is polynomial delay using backtrack search technique

Case B: P_7-free and P_8-free chordal graphs



Theorem (D. and Nourine, 2019)

There are linear and polynomial-delay algorithms enumerating minimal dominating sets in P_7-free and P_8-free chordal graphs.

→ linear-delay algorithm for Dom-Enum in P_7-free chordal graphs

→ poly.-delay algorithm for Dom-Enum in P_8-free chordal graphs

  → checking $A \in \mathcal{D}_{RN}(G)$ is linear

  → enumerating X s.t. $A \cup X \in \mathcal{D}(G)$ is polynomial delay using backtrack search technique

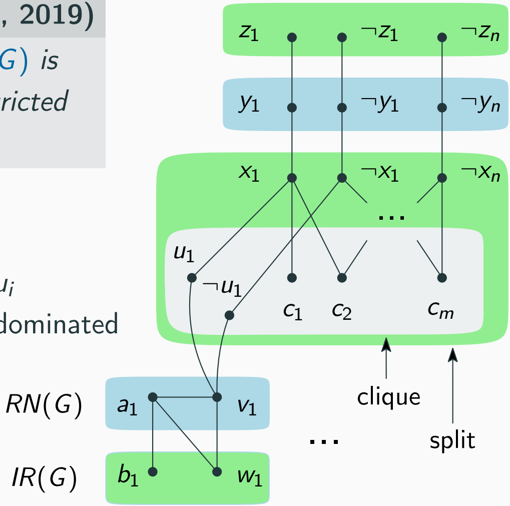
Theorem (D. and Nourine, 2019)

Deciding whether $A \in \mathcal{D}_{RN}(G)$ is NP-complete even when restricted to P_9-free chordal graphs.

→ by reduction from SAT

→ setting $A = RN(G)$

- v_i needs a private u_i or $\neg u_i$
- only $c_1, \ldots, c_m$ are to be dominated



Thank you!