A linear-delay algorithm for minimal dominating sets enumeration in split graphs

IMD Seminar

December 1st, 2022

Oscar Defrain

LIS, Aix-Marseille Université

SIAM J. DISCRETE MATH.
Vol. 28, No. 4, pp. 1916–1929

ON THE ENUMERATION OF MINIMAL DOMINATING SETS AND RELATED NOTIONS*

MAMADOU MOUSTAPHA KANTÉ[†], VINCENT LIMOUZY[†], ARNAUD MARY[†], AND LHOUARI NOURINE[†]

Abstract. A dominating set D in a graph is a subset of its vertex set such that each vertex is either in D or has a neighbor in D. In this paper, we are interested in the enumeration of (inclusionwise) minimal dominating sets in graphs, called the DOM-ENUM problem. It is well known that this problem can be polynomially reduced to the TRANS-ENUM problem in hypergraphs, i.e., the problem of enumerating all minimal transversals in a hypergraph. First, we show that the TRANS-ENUM problem can be polynomially reduced to the DOM-ENUM problem. As a consequence there exists an output-polynomial time algorithm for the TRANS-ENUM problem if and only if there exists one for the DOM-ENUM problem. Second, we study the DOM-ENUM problem in some graph classes. We give an output-polynomial time algorithm for the DOM-ENUM problem in split graphs and introduce the completion of a graph to obtain an output-polynomial time algorithm for the DOM-ENUM problem in P_6-free chordal graphs, a proper superclass of split graphs. Finally, we investigate the complexity of the enumeration of (inclusionwise) minimal connected dominating sets and minimal total dominating sets of graphs. We show that there exists an output-polynomial time algorithm for the DOM-ENUM problem (or, equivalently, TRANS-ENUM problem) if and only if there exists one for the following enumeration problems: minimal total dominating sets, minimal total dominating sets in split graphs, minimal connected dominating sets in split graphs, minimal dominating sets in co-bipartite graphs.

Key words. output polynomial time algorithm, minimal (connected, total) dominating set enumeration, hypergraph dualization, transversal problem

AMS subject classifications. 68R05, 68R10, 05C30, 05C69, 05C85, 05C65

DOI. 10.1137/120862612

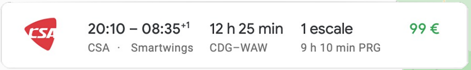
 Optimisation problems

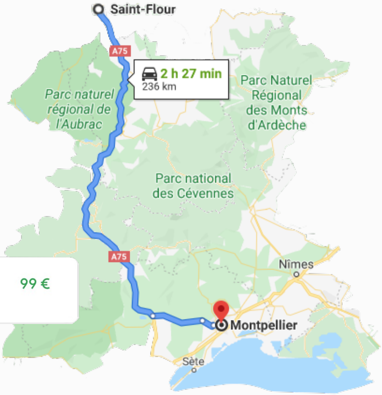
Typical question:

Given input I, find the best solution
from all feasible solutions of I.

Examples:

- shortest path to Montpellier
- cheapest flight to Warsaw
- best answer to a query
- ...

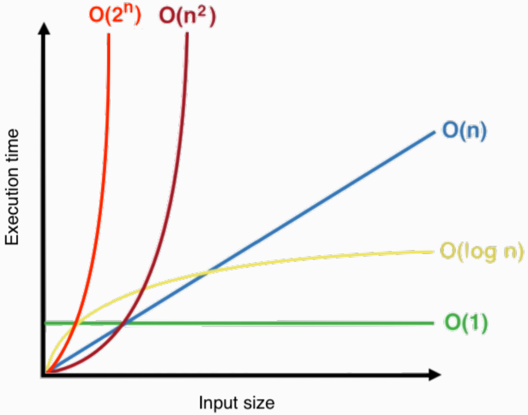




https://www.maps.google.fr

With n the input size, e.g., number of roads in the network

Efficient algorithm: runs in poly(n)-time



based on Daniel Ko's chart

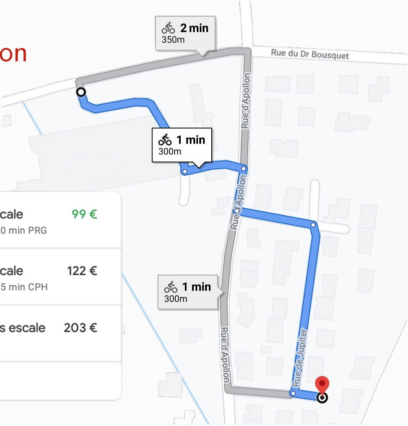
https://medium.com/@dankomong/big-o-notation-using-ruby-a357d85bb9b1

Typical question:

Given input I, list all solutions in I.

Examples:

- bike itineraries to a destination
- flights to Warsaw
- answers to a query
- ...



CSA	20:10 – 08:35[+1] CSA · Smartwings	12 h 25 min CDG–WAW	1 escale 9 h 10 min PRG	99 €
SAS	14:30 – 22:30 SAS	8 h 0 min CDG–WAW	1 escale 4 h 55 min CPH	122 €
AF	13:00 – 15:20 Air France	2 h 20 min CDG–WAW	Sans escale	203 €
	114 autres vols			

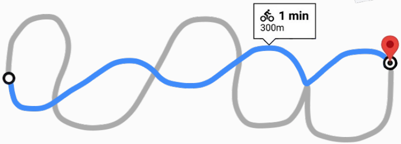
https://www.maps.google.fr

Typical question:

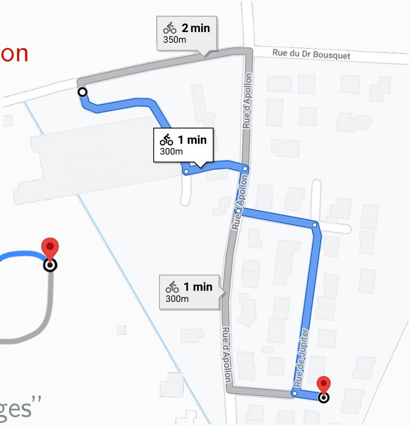
Given input I, list all solutions in I.

Examples:

- bike itineraries to a destination
- flights to Warsaw
- answers to a query
- ...



$2^{m/2}$ different paths
with m the number of "edges"



https://www.maps.google.fr

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

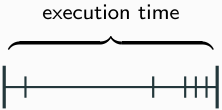
Theorem (Fredman and Khachiyan, 1996)

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

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

Let n be input size, e.g., number of roads in the network

Let d be output size, ≈number of solutions

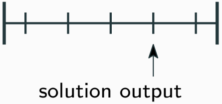


output-polynomial

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

incremental-polynomial

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



polynomial-delay

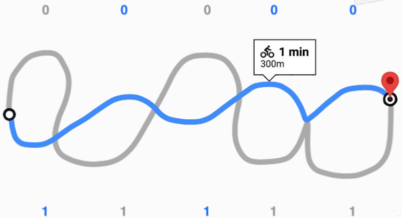
poly(n)*-time between two cons. outputs*

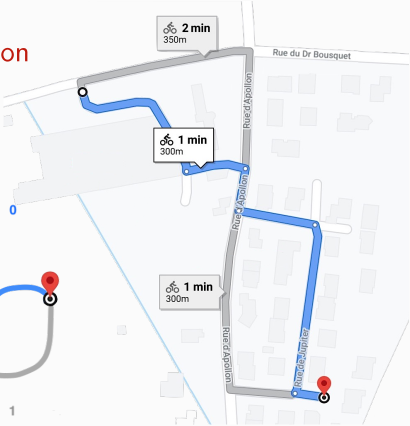
Typical question:

Given input I, list all solutions in I.

Examples:

- bike itineraries to a destination
- flights to Warsaw
- answers to a query
- ...

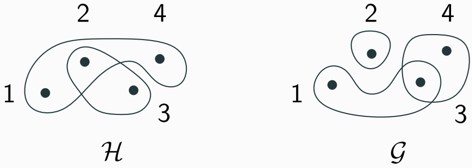




https://www.maps.google.fr

Definitions:

- hypergraph: family of subsets $\mathcal{H} \subseteq 2^X$ on **ground set** X
  called Sperner if $A \not\subset B$ for any two $A, B \in \mathcal{H}$
- transversal of $\mathcal{H}$: $T \subseteq X$ s.t. $T \cap E \neq \emptyset$ for any $E \in \mathcal{H}$
- $Tr(\mathcal{H})$: set of (inclusion-wise) minimal transervals of $\mathcal{H}$
  it is a Sperner hypergraph!
- two Sperner hypergraphs $\mathcal{H}$ and $\mathcal{G}$ are dual if $\mathcal{G} = Tr(\mathcal{H})$
  and $Tr(Tr(\mathcal{H})) = \mathcal{H}$



Hypergraph Dualization

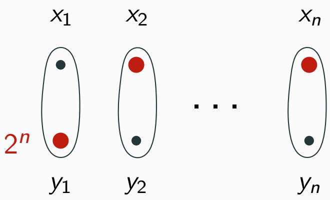
input: two hypergraphs $\mathcal{H}$ and $\mathcal{G}$ on same ground set.

question: are $\mathcal{H}$ and $\mathcal{G}$ dual?

Minimal Transversals Enumeration (Trans-Enum)

input: a hypergraph $\mathcal{H}$.

output: the set $\mathcal{G} = Tr(\mathcal{H})$ of minimal transversals of $\mathcal{H}$.



Hypergraph Dualization

input: two hypergraphs $\mathcal{H}$ and $\mathcal{G}$ on same ground set.

question: are $\mathcal{H}$ and $\mathcal{G}$ dual?

Theorem (Fredman and Khachiyan, 1996)

There is an $N^{o(\log N)}$ quasi-polynomial time algorithm solving Hypergraph Dualization where $N = |\mathcal{H}| + |\mathcal{G}|$.

Rough idea: pick an element x_i with high frequency in φ and ψ, and reduce the problem to the dualization of two separate instances not containing x_i

Yields a quasi-polynomial incremental algorithm for Trans-Enum

Minimal Transversals Enumeration (Trans-Enum)

input: a hypergraph $\mathcal{H}$.

output: the set $\mathcal{G} = Tr(\mathcal{H})$ of minimal transversals of $\mathcal{H}$.

Equivalent to:

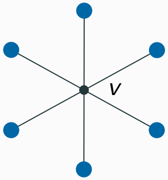
- translating from a positive CNF to a positive DNF
- → enumerating the minimal dominating sets of a graph
- enumerating the minimal set coverings of a hypergraph
- enumerating database repairs

Are harder than Trans-Enum:

- lattice dualization problems
- meet-irreducible/implicational bases translations
- characteristic models/Horn clauses translations

Definitions:

- graph G: a set of vertices $V(G)$, together with a set of edges $E(G) \subseteq \{\{x, y\} \mid x, y \in V(G),\ x \neq y\}$
- stable set: set of pairwise non-adjacent vertices
- clique: set of pairwise adjacent vertices
- $N(v)$: neighborhood of vertex v



- $N(S) = \bigcup_{v \in S} N(v) \setminus S$: 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 neighbor of $v \in D$:

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

Observation

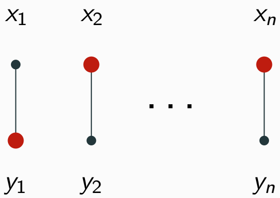
A DS is minimal if and only if it is irredundant.

if all its vertices have a private neighbor.

Minimal DS Enumeration (Dom-Enum)

input: a graph G.

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

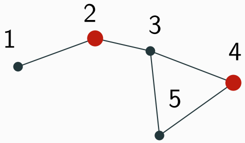


Minimal DS Enumeration (Dom-Enum)

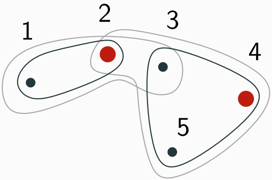
input: a graph G.

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

A particular case of Trans-Enum



G



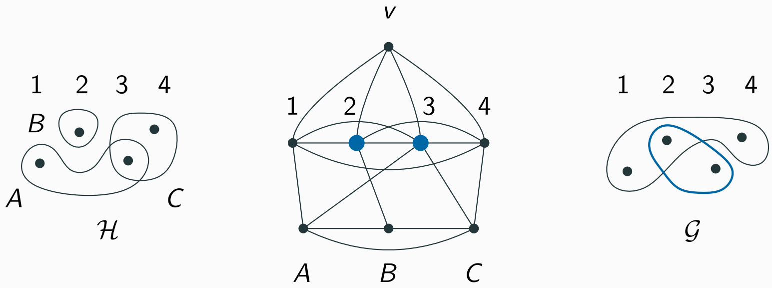
$\mathcal{H}$

Minimal DS Enumeration (Dom-Enum)

input: a graph G.

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

Equivalent to Trans-Enum [Kanté et al., 2014]



Minimal DS Enumeration (Dom-Enum)

input: a graph G.

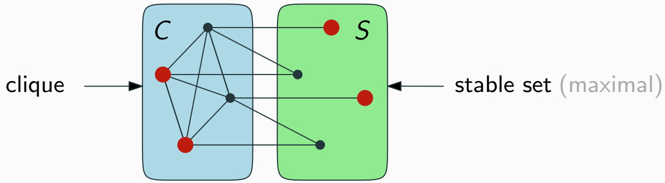
output: the set $\mathcal{D}(G)$ of minimal DS of G.

Equivalent to Trans-Enum in cobipartite graphs

General case: open, best algo in $N^{o(\log N)}$, $N = |G| + |\mathcal{D}(G)|$

Known cases:

- **output poly.:** $\log(n)$-degenerate graphs
- **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



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

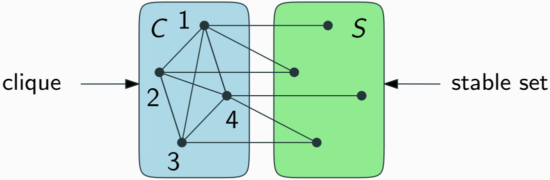
A set $D \subseteq V(G)$ is a minimal DS of G iff D dominates S and every $v \in D$ has a private neighbor in S.

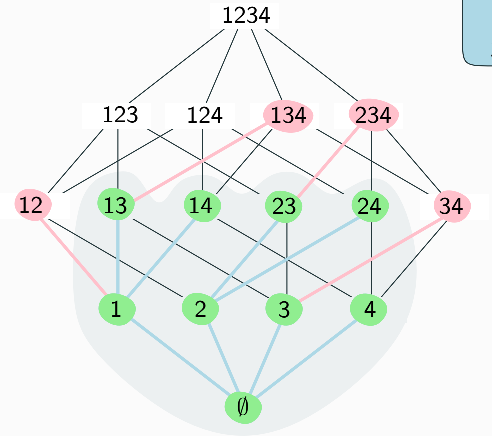
Then: $D \cap S = \{$all vertices not dominated by $D \cap C\}$

Idea: complete every irredundant set $X \subseteq C$ in S

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

→ can be enumerated with linear delay


C
1
2
3
4
S
clique
stable set


1234
123
124
134
234
12
13
14
23
24
34
1
2
3
4
∅

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

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

Summary: co-bipartite graphs are tough, split are easy

What if the graph is bipartite (partitionned into two stable sets)?

What if it is triangle-free? K_t-free (no clique of size t), fixed t ✓

Theorem (Bonamy, D, Heinrich, Pilipczuk, Raymond, 2020)

The minimal DS of any K_t-free graph G can be enumerated in time $O(n^{2^{t+1}} \cdot |\mathcal{D}(G)|^{2^t})$ and poly. space.

Still... poly. delay is open for bipartite graphs