Parameterizations of Locating-Dominating Sets

Dipayan Chakraborty[a,b], Florent Foucaud[a], Diptapriyo Majumdar[c] and Prafullkumar Tale[d]

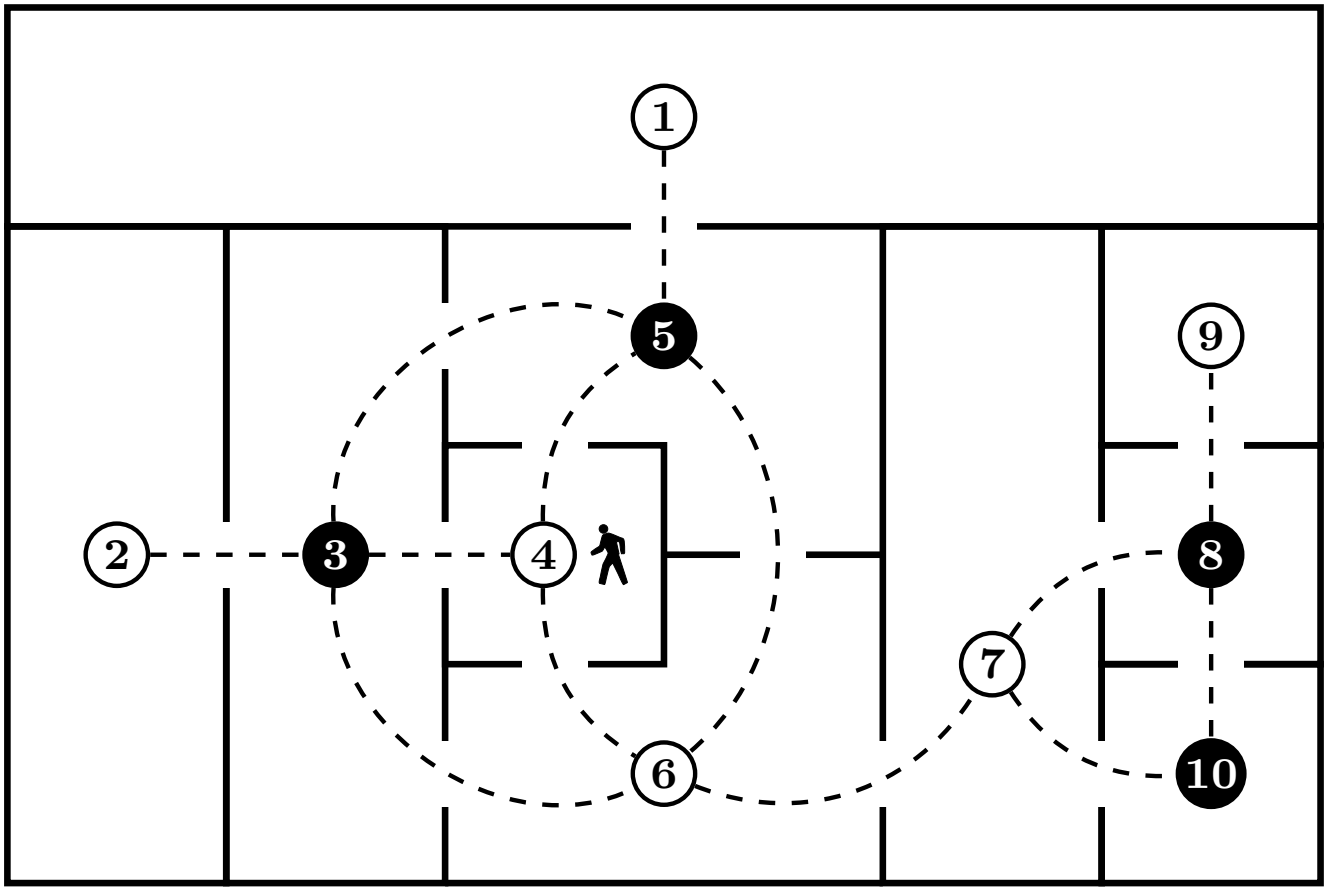
[a]Université Clermont Auvergne, CNRS, Mines Saint-Étienne, Clermont Auvergne INP, LIMOS, 63000 Clermont-Ferrand, France

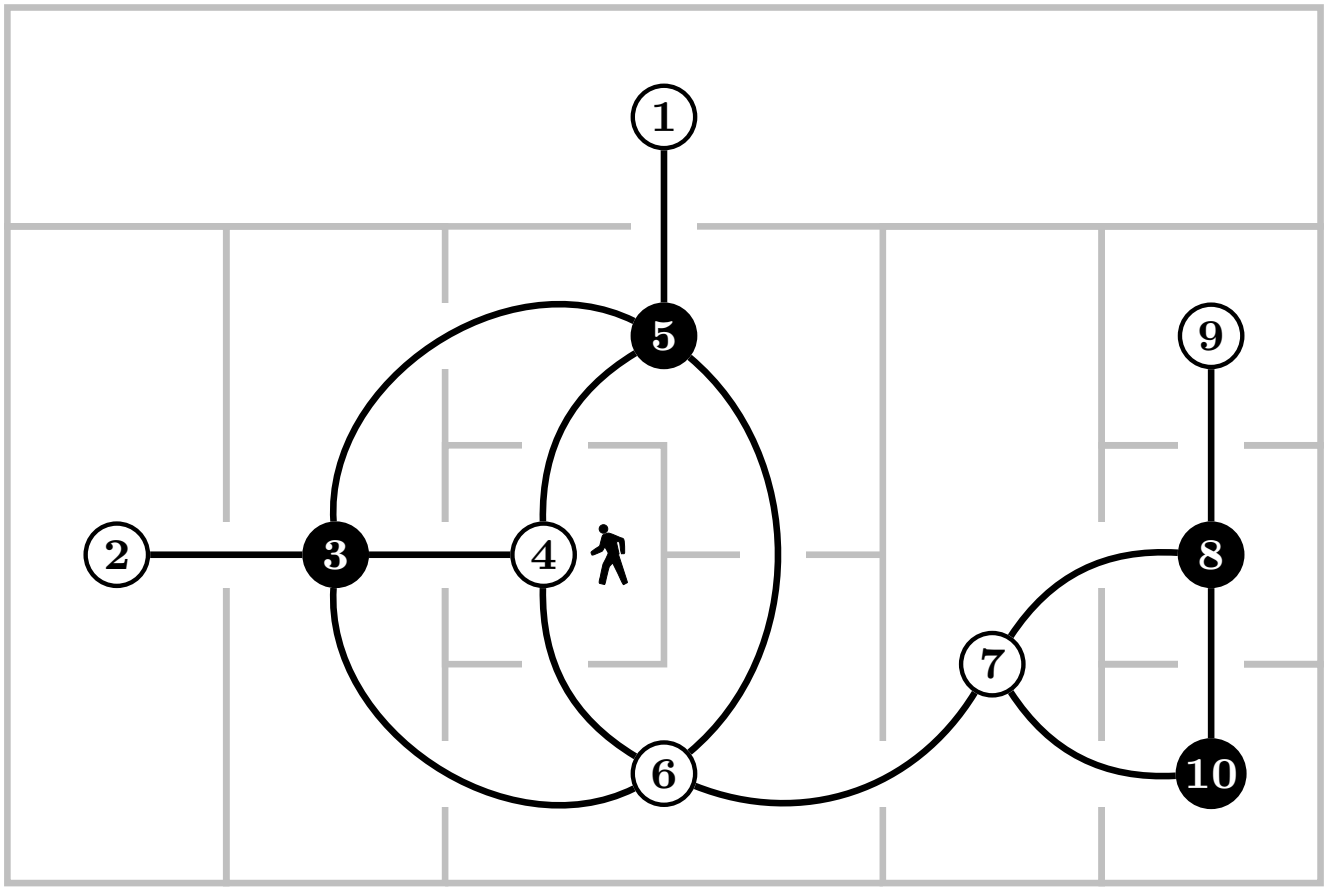
[b]Department of Mathematics and Applied Mathematics, University of Johannesburg, Auckland Park, 2006, South Africa

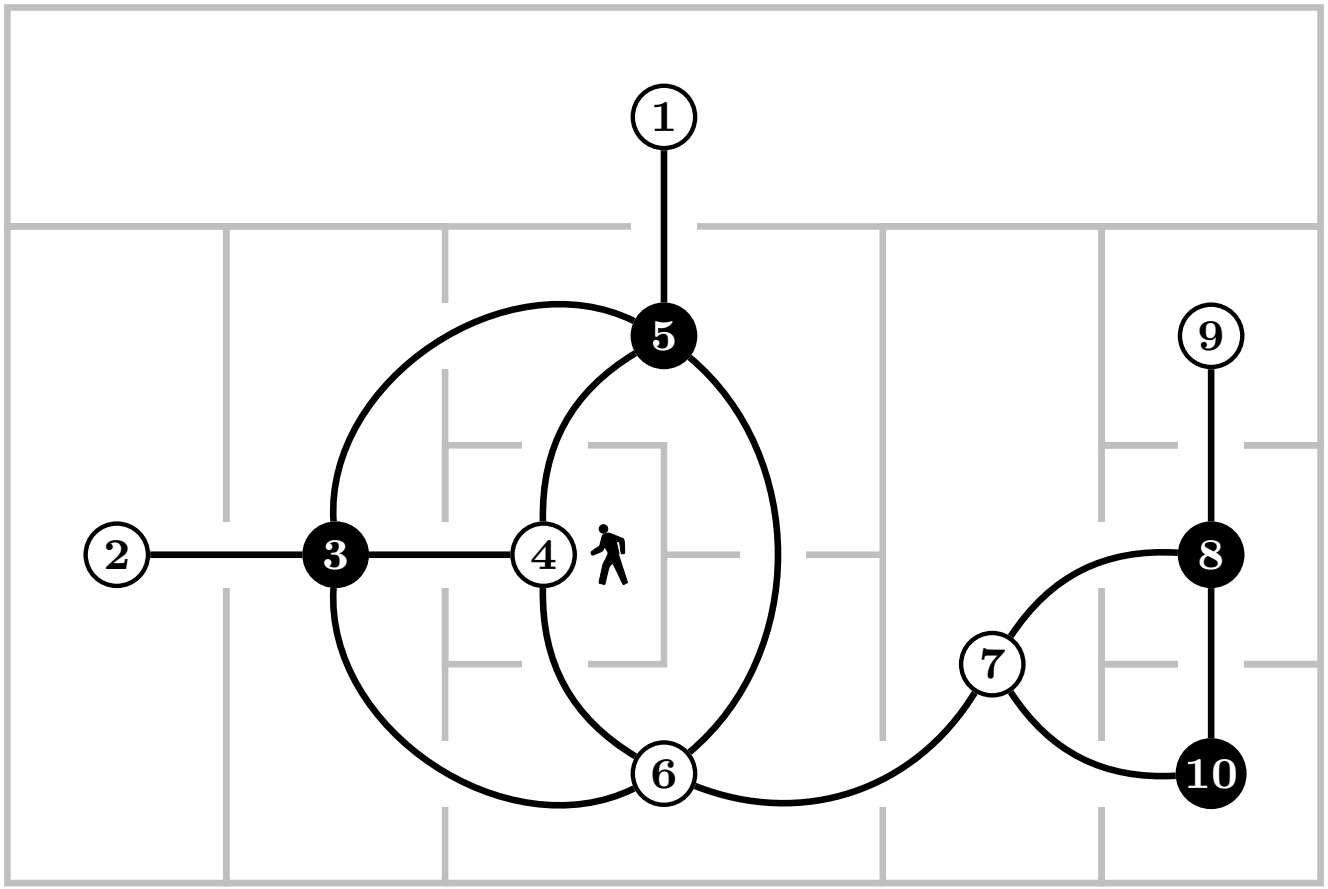
[c]Indraprastha Institute of Information Technology Delhi, Delhi, India

[d]Indian Institute of Science Education and Research Pune, Pune, India

Introduction: Locating dominating sets in graphs


1
5
9
2
3
4
8
7
6
10


1
2
3
4
5
6
7
8
9
10



Graph $G = (V, E)$

Open neighborhood:

$N(v) = \{u : uv \in E\}$

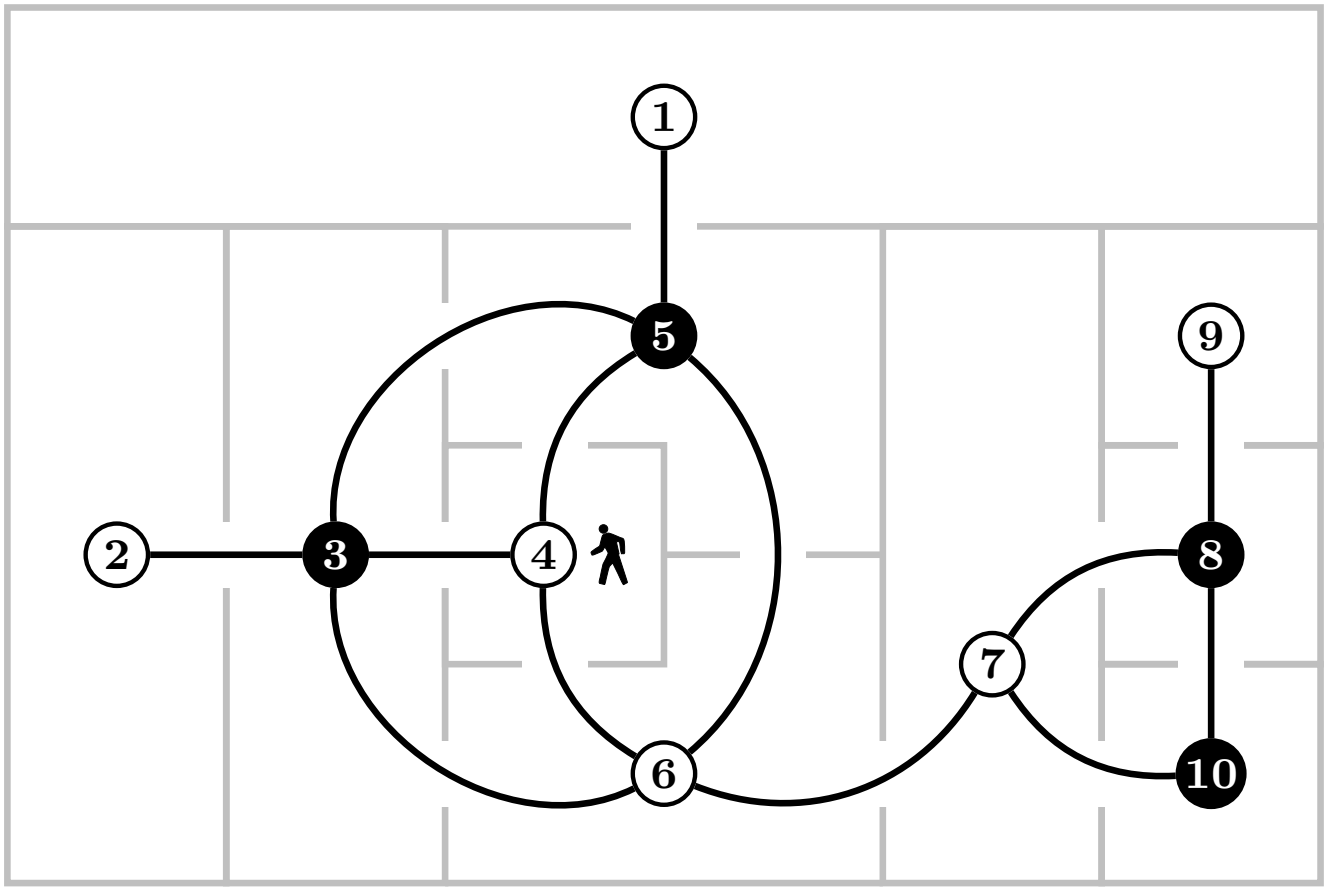
$N(7) = \{6, 8, 10\}$

Closed neighborhood:

$N[v] = N(v) \cup \{v\}$

$N[7] = \{6, 8, 10, 7\}$

C: set of black vertices



(1) A detector can monitor upto distance 1

Graph $G = (V, E)$

Open neighborhood:

$N(v) = \{u : uv \in E\}$

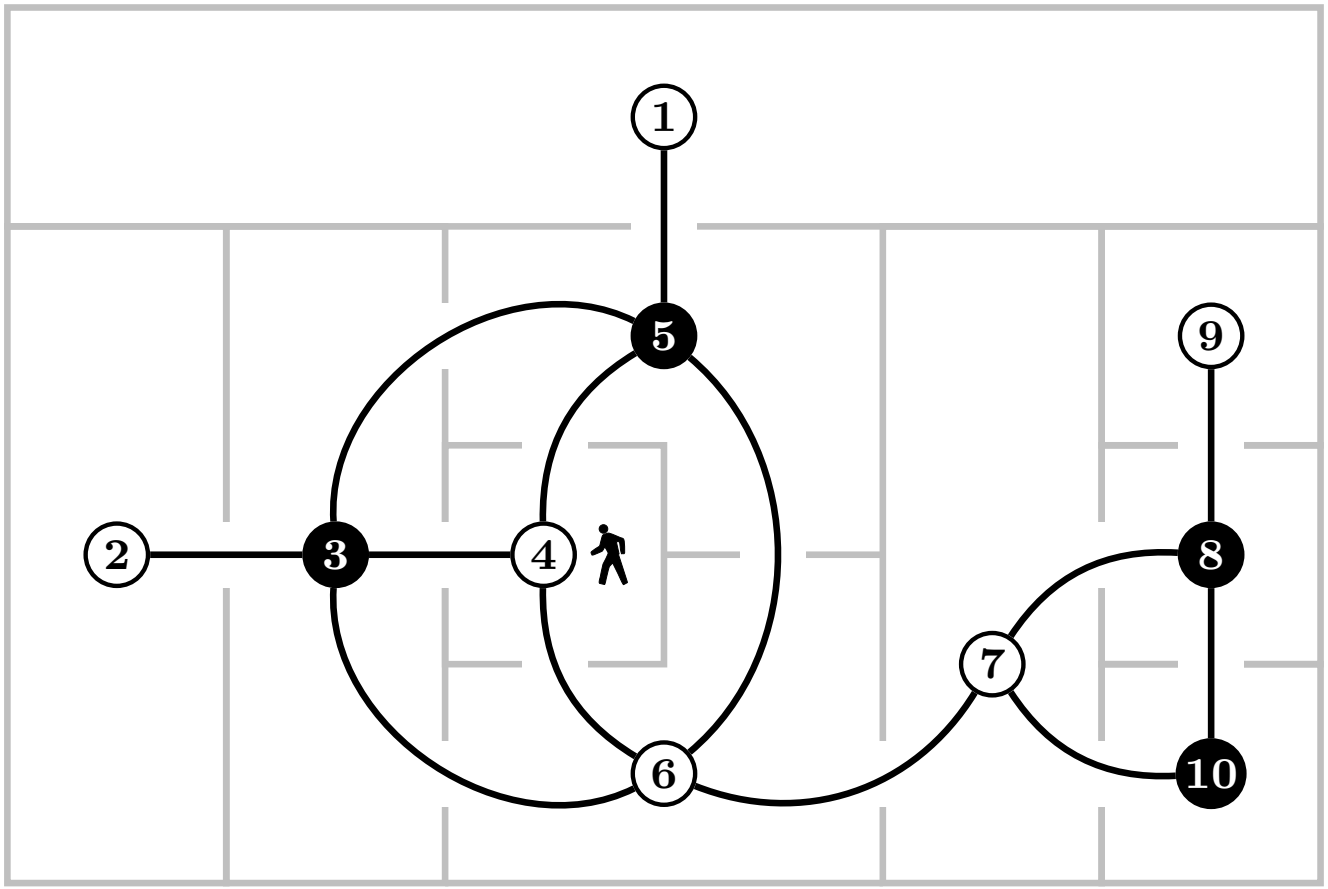
$N(7) = \{6, 8, 10\}$

Closed neighborhood:

$N[v] = N(v) \cup \{v\}$

$N[7] = \{6, 8, 10, 7\}$

C: set of black vertices



Graph $G = (V, E)$

Open neighborhood:

$N(v) = \{u : uv \in E\}$

$N(7) = \{6, 8, 10\}$

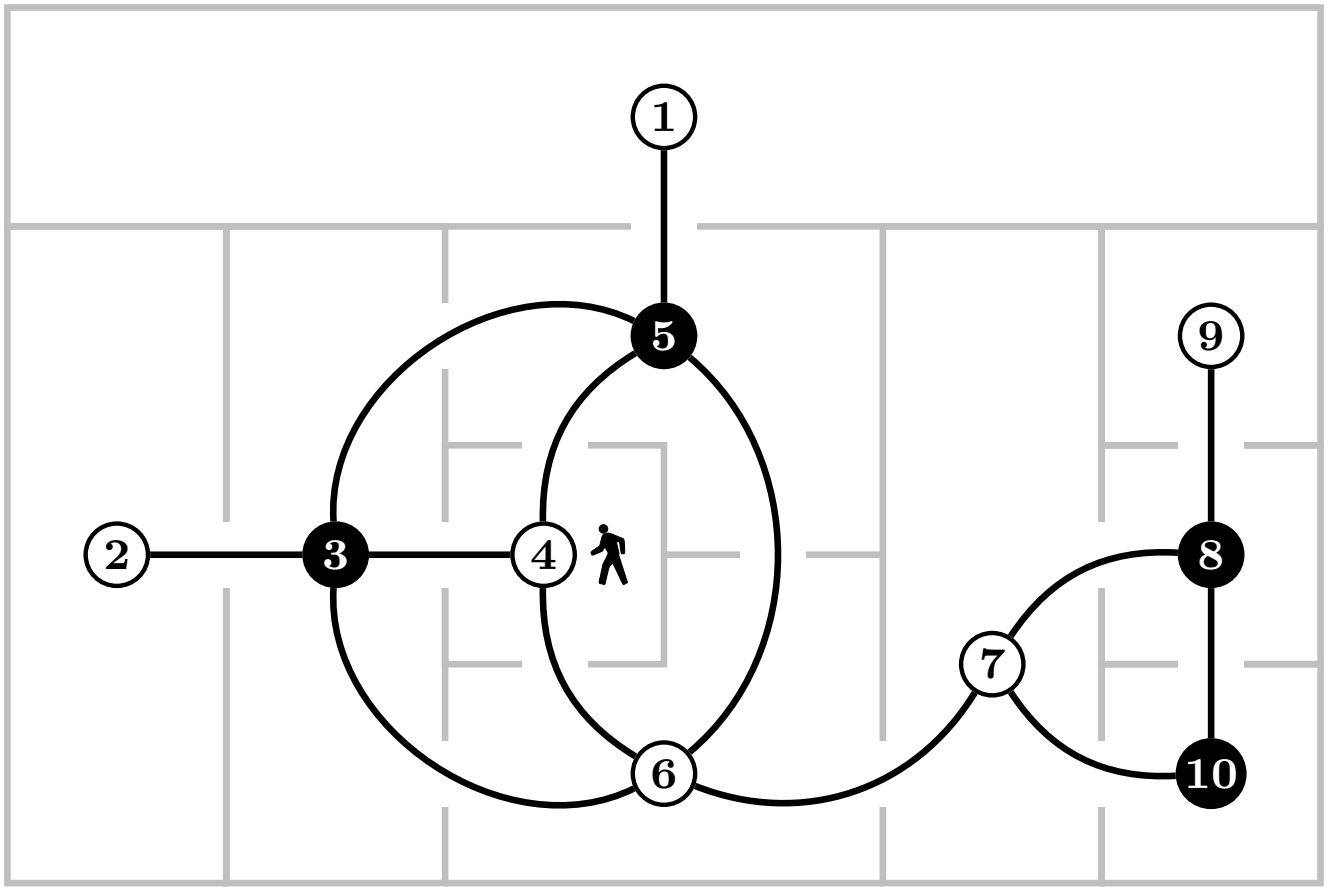
Closed neighborhood:

$N[v] = N(v) \cup \{v\}$

$N[7] = \{6, 8, 10, 7\}$

C: set of black vertices

Dominating set: A set $C \subseteq V$ such that $N[v] \cap C \neq \emptyset$ for all $v \in V$



Graph $G = (V, E)$

Open neighborhood:

$N(v) = \{u : uv \in E\}$

$N(7) = \{6, 8, 10\}$

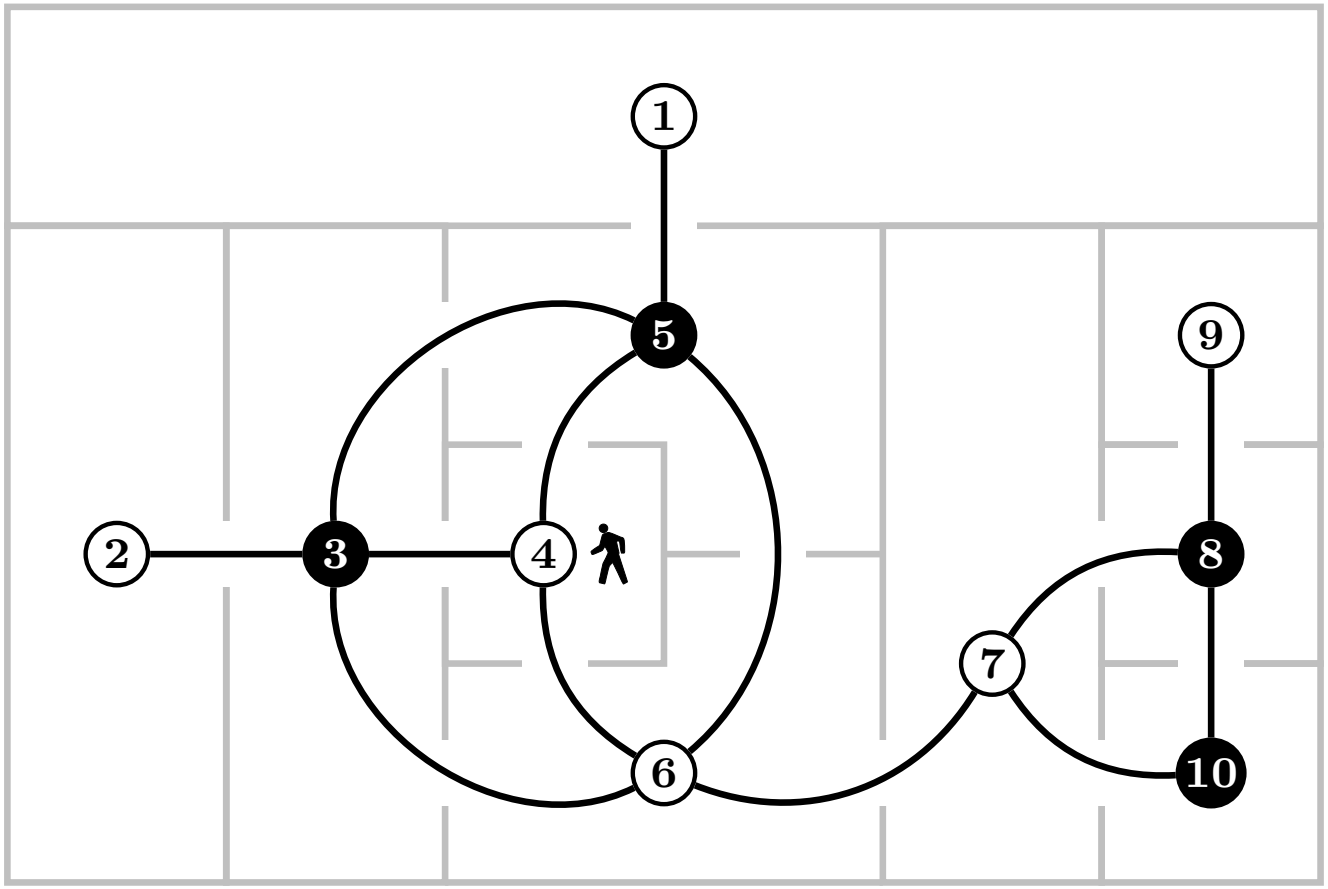
Closed neighborhood:

$N[v] = N(v) \cup \{v\}$

$N[7] = \{6, 8, 10, 7\}$

C: set of black vertices

Dominating set: A set $C \subseteq V$ such that $N[v] \cap C \neq \emptyset$ for all $v \in V$



Graph $G = (V, E)$

Open neighborhood:

$N(v) = \{u : uv \in E\}$

$N(7) = \{6, 8, 10\}$

Closed neighborhood:

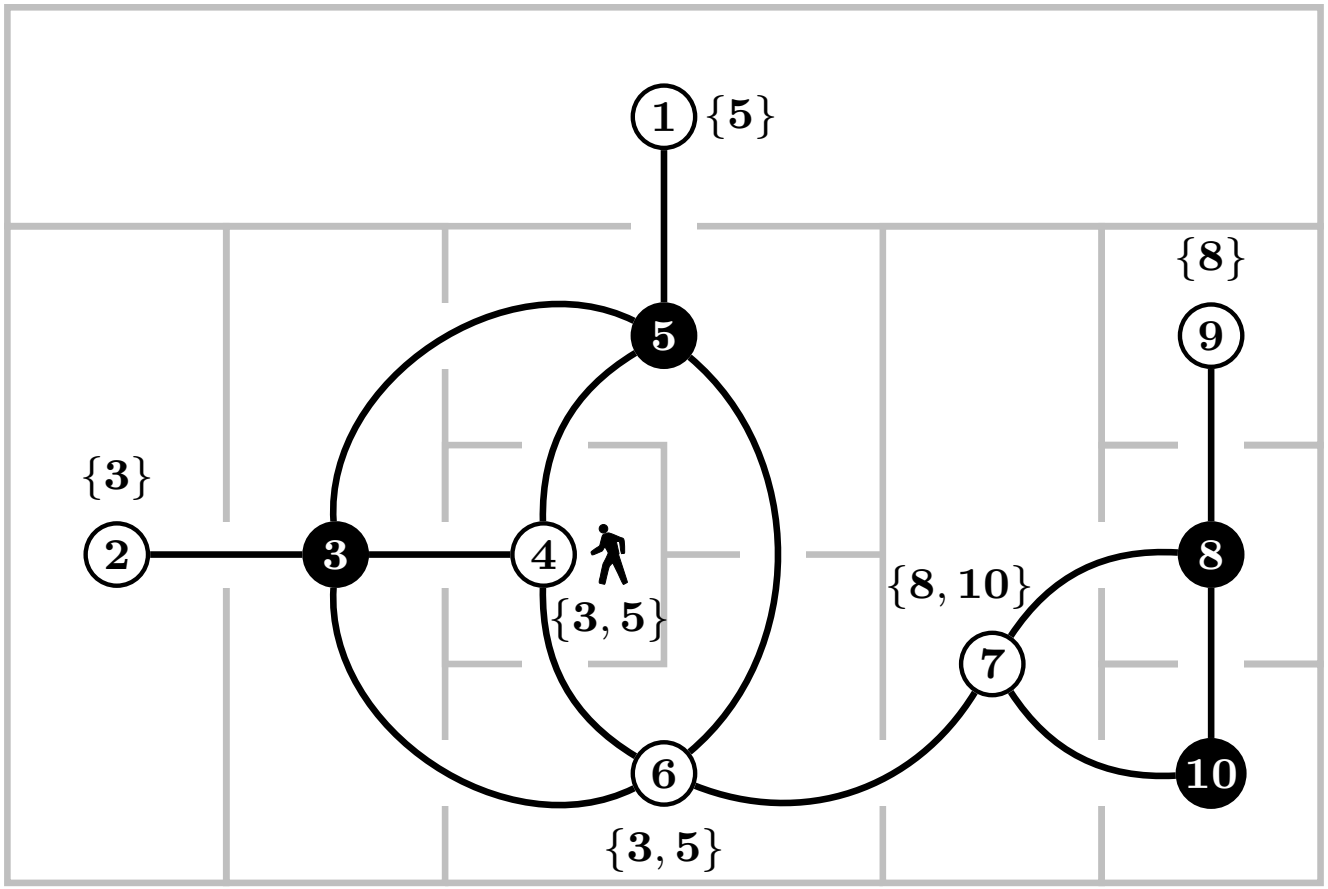
$N[v] = N(v) \cup \{v\}$

$N[7] = \{6, 8, 10, 7\}$

C: set of black vertices

Dominating set: A set $C \subseteq V$ such that $N[v] \cap C \neq \emptyset$ for all $v \in V$

(2) A detector can distinguish between itself and a neighbor



Graph $G = (V, E)$

Open neighborhood:

$N(v) = \{u : uv \in E\}$

$N(7) = \{6, 8, 10\}$

Closed neighborhood:

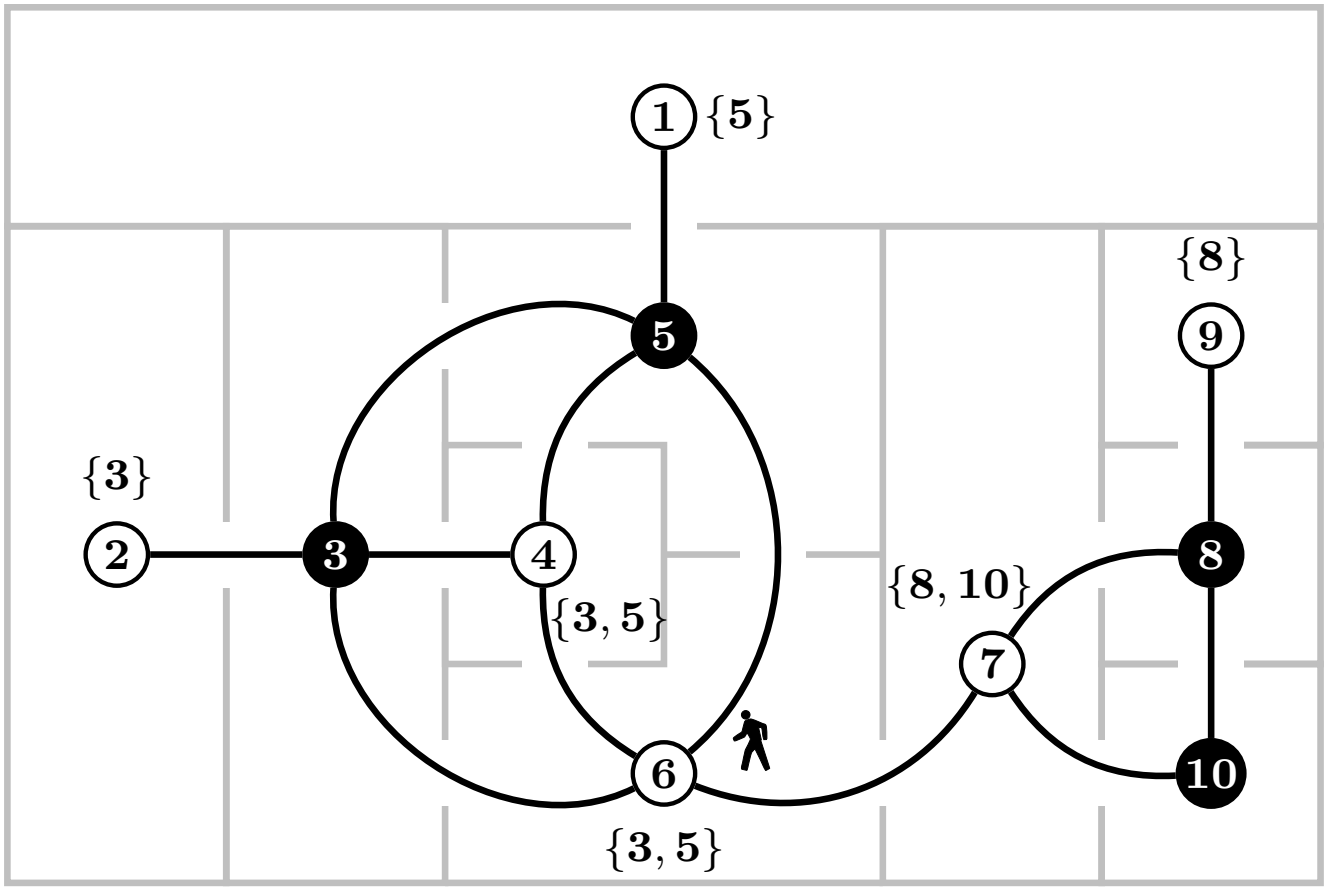
$N[v] = N(v) \cup \{v\}$

$N[7] = \{6, 8, 10, 7\}$

C: set of black vertices

Dominating set: A set $C \subseteq V$ such that $N[v] \cap C \neq \emptyset$ for all $v \in V$

(2) A detector can distinguish between itself and a neighbor



Graph $G = (V, E)$

Open neighborhood:

$N(v) = \{u : uv \in E\}$

$N(7) = \{6, 8, 10\}$

Closed neighborhood:

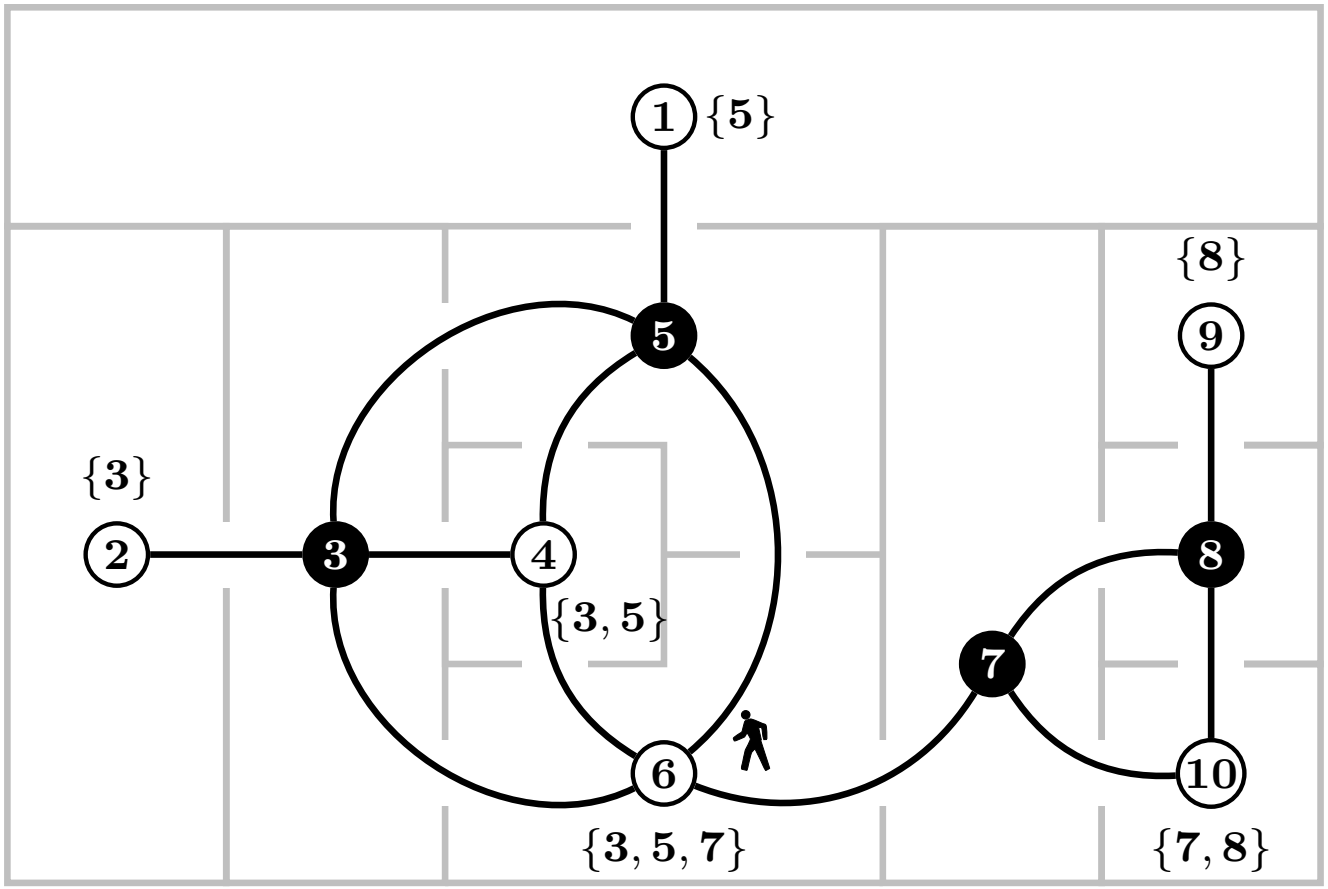
$N[v] = N(v) \cup \{v\}$

$N[7] = \{6, 8, 10, 7\}$

C: set of black vertices

Dominating set: A set $C \subseteq V$ such that $N[v] \cap C \neq \emptyset$ for all $v \in V$

(2) A detector can distinguish between itself and a neighbor



Graph $G = (V, E)$

Open neighborhood:

$N(v) = \{u : uv \in E\}$

$N(7) = \{6, 8, 10\}$

Closed neighborhood:

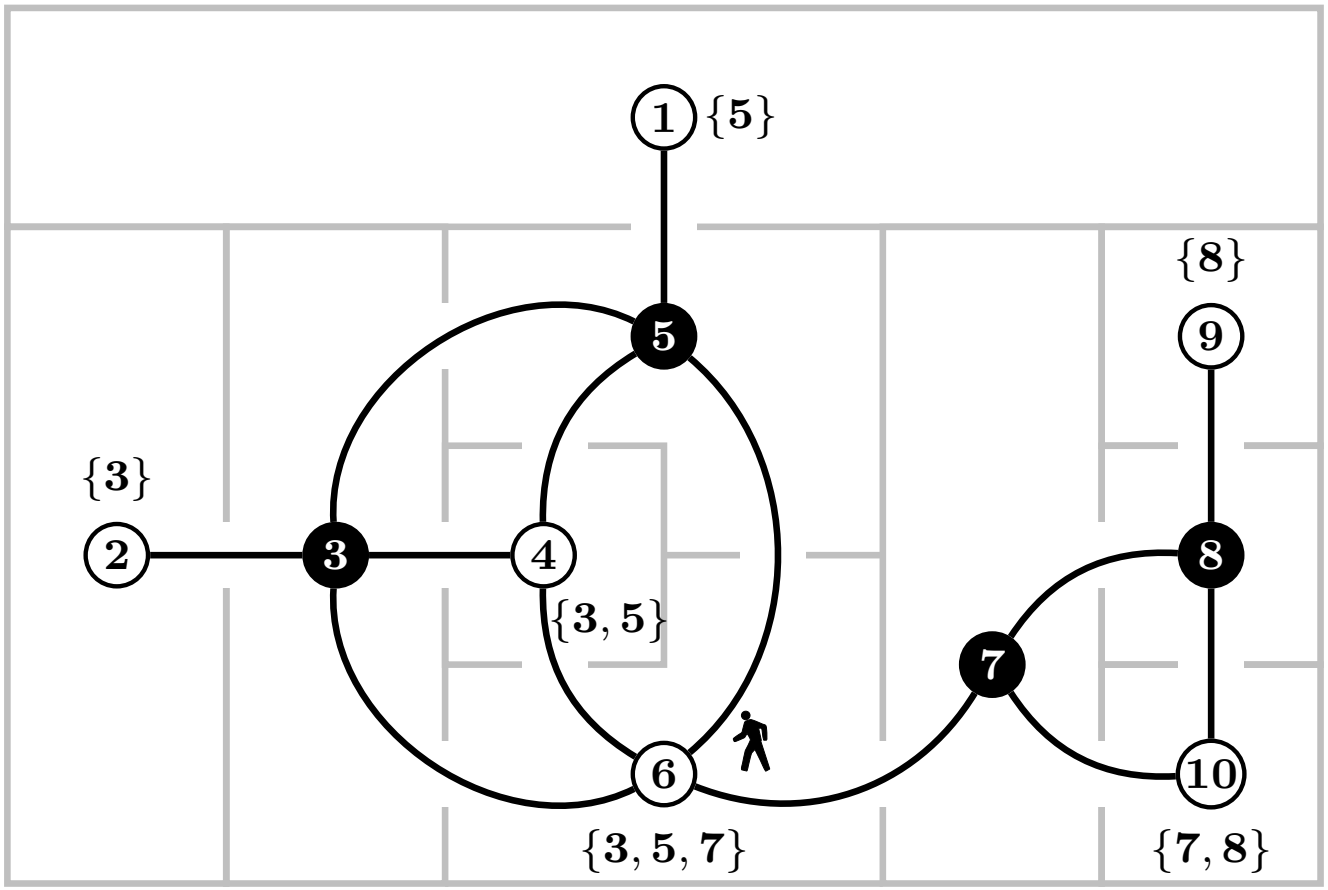
$N[v] = N(v) \cup \{v\}$

$N[7] = \{6, 8, 10, 7\}$

C: set of black vertices

Dominating set: A set $C \subseteq V$ such that $N[v] \cap C \neq \emptyset$ for all $v \in V$

(2) A detector can distinguish between itself and a neighbor



Graph $G = (V, E)$

Open neighborhood:

$N(v) = \{u : uv \in E\}$

$N(7) = \{6, 8, 10\}$

Closed neighborhood:

$N[v] = N(v) \cup \{v\}$

$N[7] = \{6, 8, 10, 7\}$

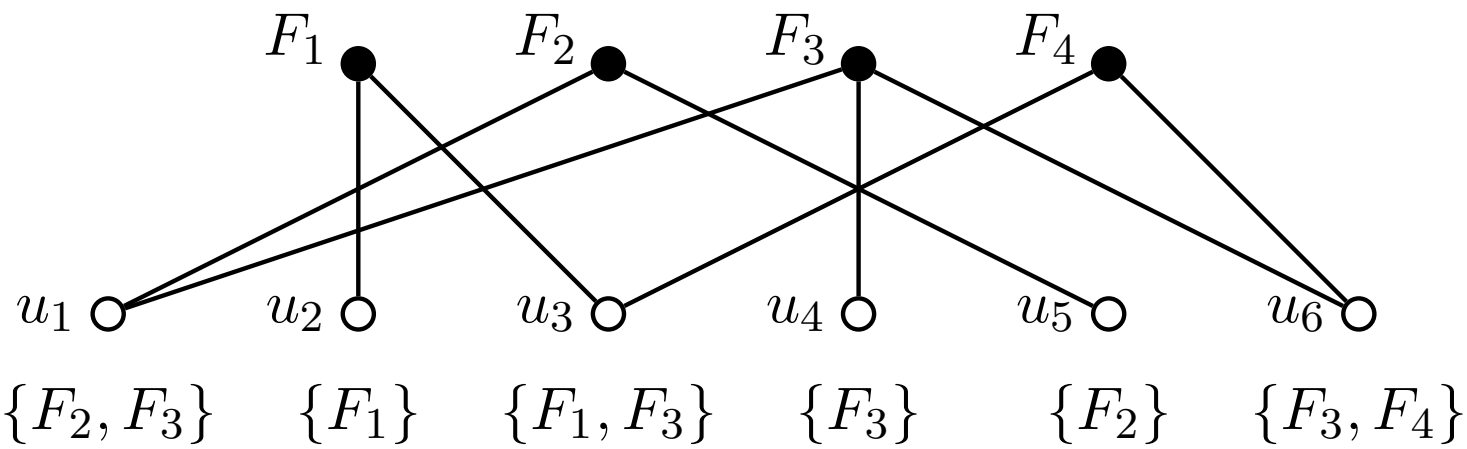
C: set of black vertices

Dominating set: A set $C \subseteq V$ such that $N[v] \cap C \neq \emptyset$ for all $v \in V$

Locating set: A set $C \subseteq V$ such that $N(u) \cap C \neq N(v) \cap C$ for all $u, v \in V \setminus C$

Definition

Locating dominating (LD) set [Slater, 1988]: is a set $C \subseteq V$ of a graph $G = (V, E)$ such that C is
(i) a dominating set of G; and
(ii) $N(u) \cap C \neq N(v) \cap C$ for all $u, v \in V \setminus C$.



Decision version of finding the minimum LD-set in a graph:

MINIMUM LD-SET
Input: (G, k): A graph G and a positive integer k.
Question: Does there exist an LD-set C of G such that $|C| \leq k$?

MINIMUM LD-SET is NP-hard!

Decision version of finding the minimum LD-set in a graph:

> Minimum LD-Set
> **Input:** (G, k): A graph G and a positive integer k.
> **Question:** Does there exist an LD-set C of G such that $|C| \leq k$?

Minimum LD-Set is NP-hard!

What about *Fixed Parameter Tractable* (FPT) algorithms?
i.e. given a graph parameter k, can we find an algorithm to find a minimum code in time $f(k) \cdot n^{\mathcal{O}(1)}$? e.g. $f(k) = 2^k, 2^{k^2}, 2^{2^k} \ldots$

Decision version of finding the minimum LD-set in a graph:

MINIMUM LD-SET
Input: (G, k): A graph G and a positive integer k.
Question: Does there exist an LD-set C of G such that $|C| \leq k$?

MINIMUM LD-SET is NP-hard!

What about *Fixed Parameter Tractable* (FPT) algorithms?
i.e. given a graph parameter k, can we find an algorithm to find a minimum code in time $f(k) \cdot n^{\mathcal{O}(1)}$? e.g. $f(k) = 2^k, 2^{k^2}, 2^{2^k} \ldots$

Note [Slater, 1988]: MINIMUM LD-SET is FPT when parameterized by solution size k.

Reason: $|V(G)| = \mathcal{O}(2^k)$. Thus brute force gives $2^{\mathcal{O}(k^2)} \cdot n^{\mathcal{O}(1)}$ runtime.

Our contribution (Part I) : Minimum LD-Set wrt solution size (k)

Theorem (C., Foucaud, Majumdar, Tale, ISAAC 2024)

Unless the ETH fails, Minimum LD-Set *does not admit an algorithm running in time* $2^{o(k^2)} \cdot n^{\mathcal{O}(1)}$.

Our contribution (Part I) : Minimum LD-Set wrt solution size (k)

Theorem (C., Foucaud, Majumdar, Tale, ISAAC 2024)

Unless the ETH fails, Minimum LD-Set *does not admit an algorithm running in time* $2^{o(k^2)} \cdot n^{\mathcal{O}(1)}$.

Theorem (Slater, 1988)

Minimum LD-Set *has an algorithm with running time* $2^{\mathcal{O}(k^2)}$.

Our contribution (Part I) : Minimum LD-Set wrt treewidth (tw)

Theorem (C., Foucaud, Majumdar, Tale, ISAAC 2024)

Unless the ETH fails, Minimum LD-Set *does not admit an algorithm running in time* $2^{2^{o(\mathsf{tw})}} \cdot n^{\mathcal{O}(1)}$.

Our contribution (Part I) : Minimum LD-Set wrt treewidth (tw)

Theorem (C., Foucaud, Majumdar, Tale, ISAAC 2024)

Unless the ETH fails, Minimum LD-Set *does not admit an algorithm running in time* $2^{2^{o(\mathsf{tw})}} \cdot n^{\mathcal{O}(1)}$.

Theorem (C., Foucaud, Majumdar, Tale, ISAAC 2024)

Minimum LD-Set *admits an algorithm with running time* $2^{2^{\mathcal{O}(\mathsf{tw})}} \cdot n$.

Examples of problems with at least double-exponential FPT algorithms with respect to treewidth (tw)

- Choosability problem: $\mathcal{O}(2^{2^{o(\mathsf{tw})}}) \cdot n^{\mathcal{O}(1)}$ ... unless ETH fails [D. Marx & V. Mitsou (2016); I. Bliznets & M. Hecher (2024)]

- Projective Model Counting: $\mathcal{O}(2^{2^{\mathsf{tw}+4}} \cdot n^2)$ ... unless ETH fails [J.K. Fichte et al. (2018)]

- QSat: $\mathrm{tow}(\ell, \mathsf{tw}) \cdot n^{\mathcal{O}(1)} = 2^{2^{\cdot^{\cdot^{\cdot^{2^{\mathsf{tw}}}}}}} \cdot n^{\mathcal{O}(1)}$ ... unless ETH fails [J.K. Fichte et al. (2020)]

- Etc...

⚠ All the above problems are either #NP-complete or Σ_2^p-complete or Π_2^p-complete.

Foucaud et al. [ICALP 2024]: Problems in class NP$=\Sigma_1^p$ with double-exponential lower bounds in tw.

Metric Dimension : $2^{f(\text{diam})^{o(\text{tw})}} \cdot n^{\mathcal{O}(1)}$ ... unless ETH fails

Input: A graph G and a positive interger k.
Question: Does there exist $S \subseteq V(G)$ such that $|S| \leq k$ and, for any pair of vertices $u, v \in V(G)$, there exists a vertex $w \in S$ with $d(w, u) \neq d(w, v)$?

Strong Metric Dimension : $2^{2^{o(\text{tw})}} \cdot n^{\mathcal{O}(1)}$ ... unless ETH fails

Input: A graph G and a positive interger k.
Question: Does there exist $S \subseteq V(G)$ such that $|S| \leq k$ and, for any pair of vertices $u, v \in V(G)$, there exists a vertex $w \in S$ with either u lies on some shortest path between v and w, or v lies on some shortest path between u and w?

Geodetic Set : $2^{f(\text{diam})^{o(\text{tw})}} \cdot n^{\mathcal{O}(1)}$ ... unless ETH fails

Input: A graph G and a positive interger k.
Question: Does there exist $S \subseteq V(G)$ such that $|S| \leq k$ and, for any pair of vertices $u, v \in V(G)$, there are two vertices $s_1, s_2 \in S$ such that a shortest path from s_1 to s_2 contains u?

Our contribution (Part I) : Minimum LD-Set wrt treewidth (tw)

Theorem (C., Foucaud, Majumdar, Tale, ISAAC 2024)

Unless the ETH fails, Minimum LD-Set *does not admit an algorithm running in time* $2^{2^{o(\mathsf{tw})}} \cdot n^{\mathcal{O}(1)}$.

... motivated by results on metric dimension by Foucaud et al. [ICALP 2024] since

identification problems = metric dimension problems at "distance 1".

Our contribution (Part II) : Minimum LD-Set wrt vertex cover number (vc), distance to clique (dc) and twin cover number (tc)

Theorem (C., Foucaud, Majumdar, Tale, CIAC 2025)

Minimum LD-Set *admits an algorithm with running time* $2^{\mathcal{O}(\mathsf{vc} \log \mathsf{vc})} \cdot n^{\mathcal{O}(1)}$.

Our contribution (Part II) : Minimum LD-Set wrt vertex cover number (vc), distance to clique (dc) and twin cover number (tc)

Theorem (C., Foucaud, Majumdar, Tale, CIAC 2025)

Minimum LD-Set *admits an algorithm with running time* $2^{\mathcal{O}(\mathsf{vc}\log\mathsf{vc})} \cdot n^{\mathcal{O}(1)}$.

... the above result is extended to:

Theorem (C., Foucaud, Majumdar, Tale, 2025)

Minimum LD-Set *admits an algorithm with running time* $2^{\mathcal{O}(\mathsf{dc}\log\mathsf{dc})} \cdot n^{\mathcal{O}(1)}$.

Theorem (C., Foucaud, Majumdar, Tale, 2025)

Minimum LD-Set *admits an algorithm with running time* $2^{\mathcal{O}(\mathsf{tc}\log\mathsf{tc})} \cdot n^{\mathcal{O}(1)}$.

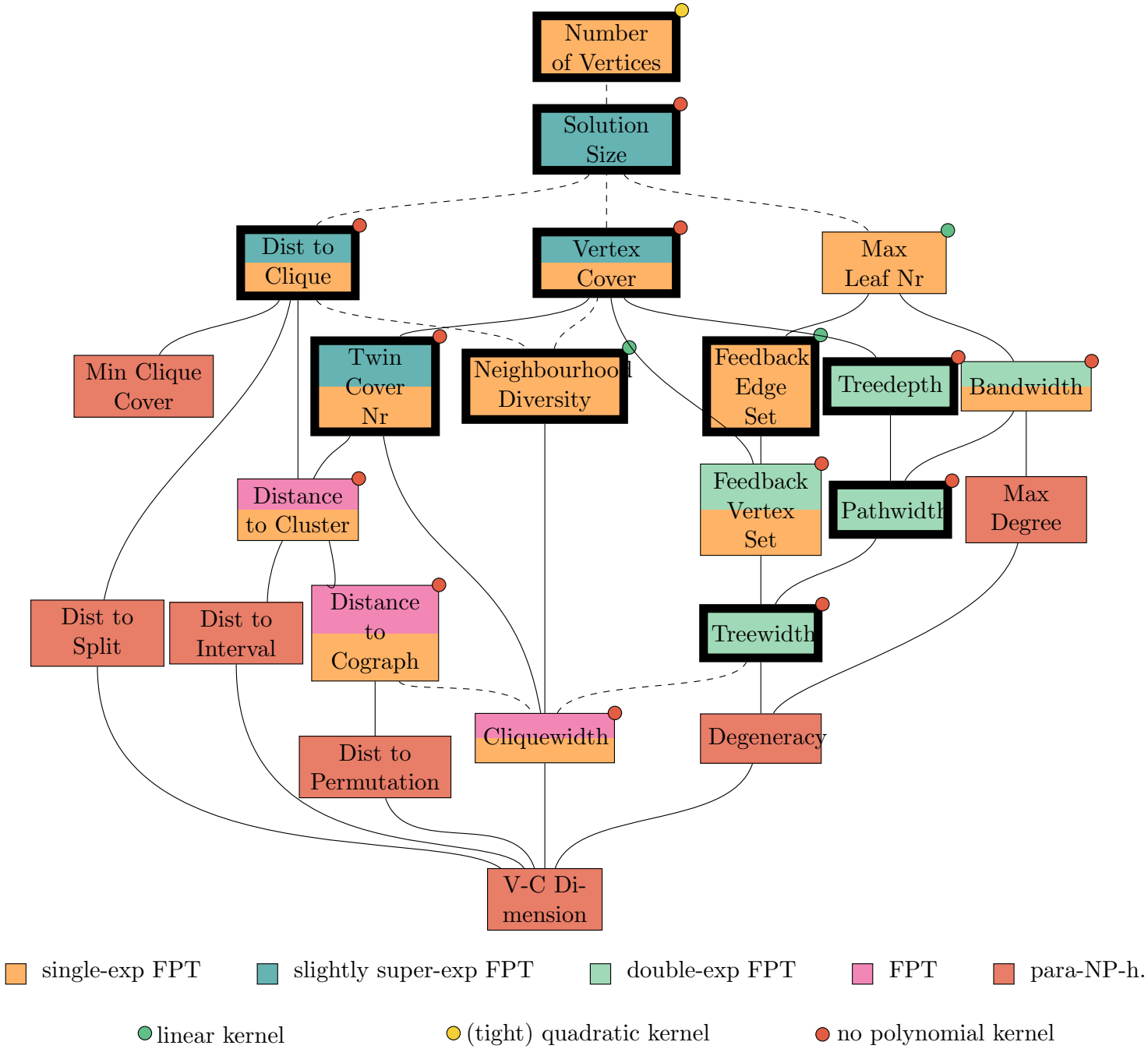
Our contribution (Part II) : Minimum LD-Set wrt neighborhood diversity (nd) and feedback edge set number (fes)

Theorem (C., Foucaud, Majumdar, Tale, CIAC 2025)

Minimum LD-Set *admits a kernel with* $\mathcal{O}(\mathsf{nd})$ *vertices in time* $\mathcal{O}(mn)$.

Theorem (C., Foucaud, Majumdar, Tale, CIAC 2025)

Minimum LD-Set *admits a kernel with* $\mathcal{O}(\mathsf{fes})$ *vertices and edges.*


Number of Vertices
Solution Size
Dist to Clique
Vertex Cover
Max Leaf Nr
Min Clique Cover
Twin Cover Nr
Neighbourhood Diversity
Feedback Edge Set
Treedepth
Bandwidth
Distance to Cluster
Feedback Vertex Set
Pathwidth
Max Degree
Dist to Split
Dist to Interval
Distance to Cograph
Treewidth
Cliquewidth
Degeneracy
Dist to Permutation
V-C Di-mension
single-exp FPT
slightly super-exp FPT
double-exp FPT
FPT
para-NP-h.
linear kernel
(tight) quadratic kernel
no polynomial kernel

Theorem (C., Foucaud, Majumdar, Tale, ISAAC 2024)

Unless the ETH fails, Minimum LD-Set *does not admit an algorithm running in time* $2^{2^{o(\mathsf{tw})}} \cdot n^{\mathcal{O}(1)}$.

(3,3)-Sat [Tovey, 1984]

- Boolean formula ψ in 3-CNF
- Each variable x appears at most 3 times
- (Assume) both any x, both x and $\neg x$ exists in ψ
- Unless ETH fails, (3,3)-Sat does not admit a $2^{o(m+n)}$-time algorithm

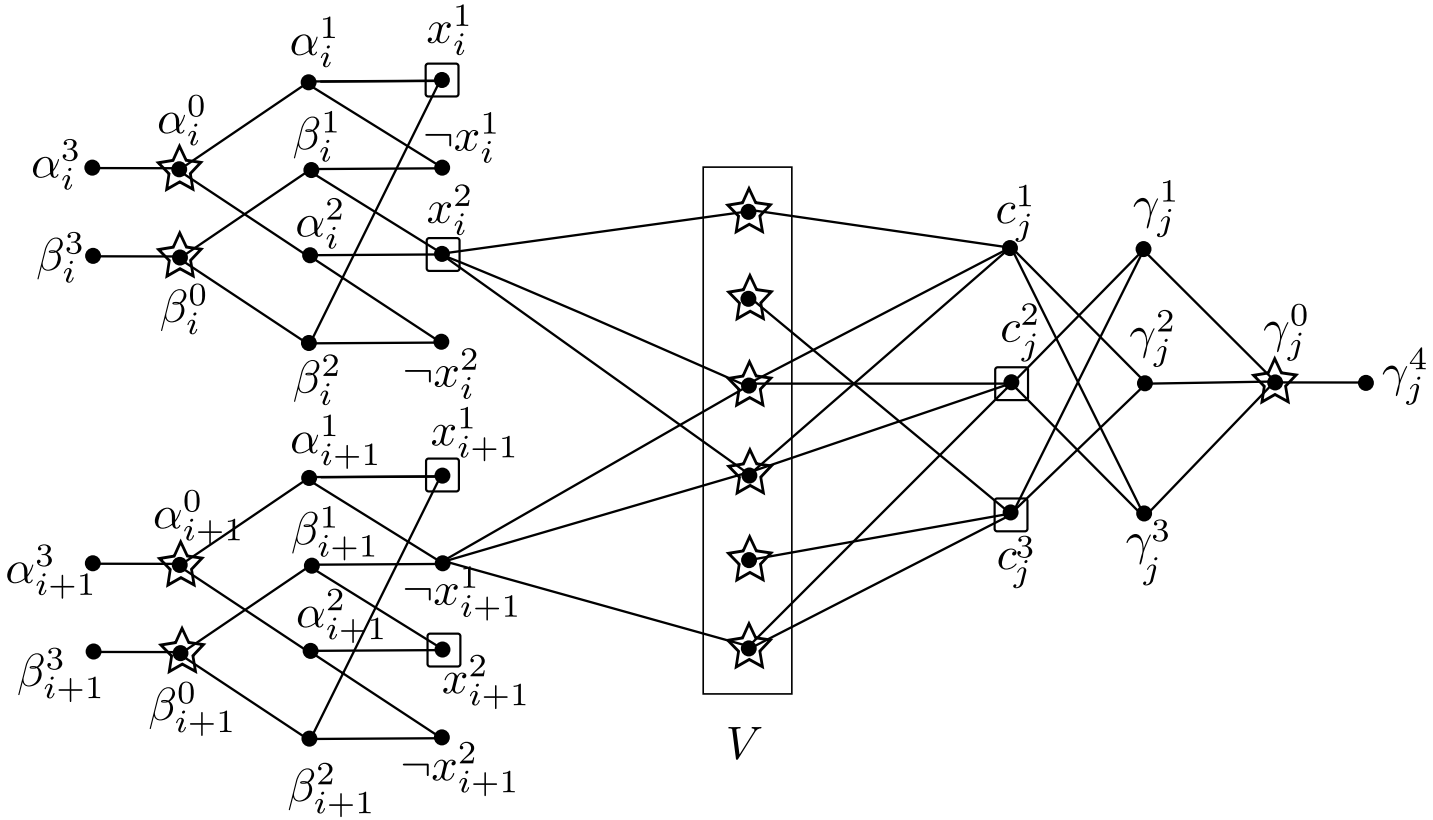
$\Longleftrightarrow$

Minimum LD-Set

- (Assume) every support vertex is in an LD-set
- A dominating set C is an LD-set if $N(u) \cap C \neq N(v) \cap C$ for all $u, v \in V \setminus C$ and $d(u, v) \leq 2$

Theorem (C., Foucaud, Majumdar, Tale, ISAAC 2024)

Unless the ETH fails, Minimum LD-Set *does not admit an algorithm running in time* $2^{2^{o(\mathsf{tw})}} \cdot n^{\mathcal{O}(1)}$.



$k = 4n + 3m + 4|V|$; where $|V| = 2p$ with p smallest interger with $4n \leq \binom{2p}{p} \sim \frac{4^p}{\sqrt{\pi \cdot p}} \leq \frac{4^p}{2^p} = 2^p \implies p = \mathcal{O}(\log n) \implies |V| = \mathcal{O}(\log n)$
$\implies \mathsf{tw}(G) = \mathcal{O}(\log n)$

Theorem (C., Foucaud, Majumdar, Tale, CIAC 2025)

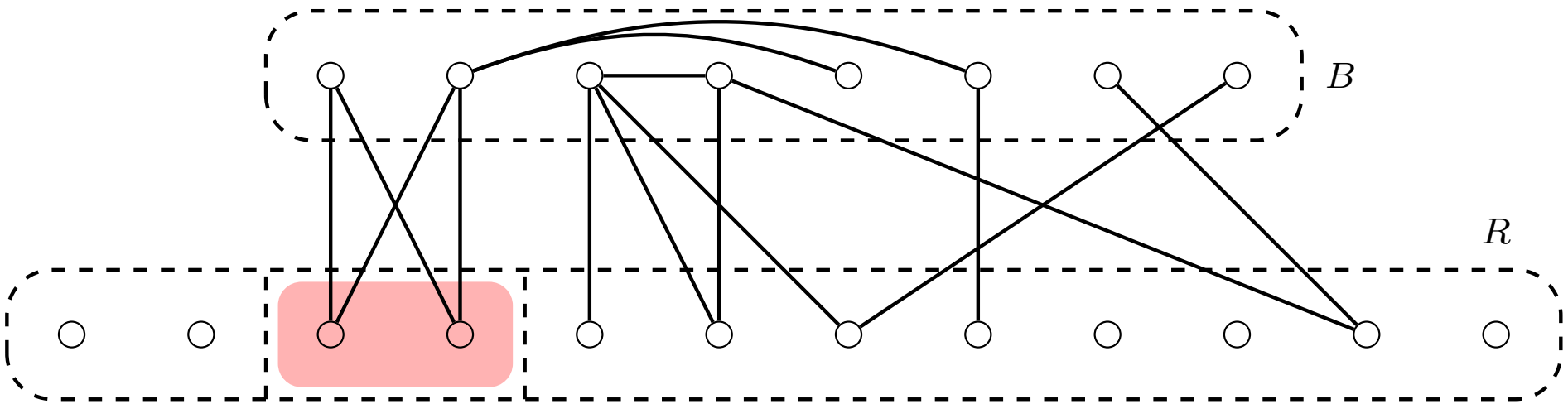
MINIMUM LD-SET *admits an algorithm with running time* $2^{\mathcal{O}(\mathsf{vc}\log\mathsf{vc})} \cdot n^{\mathcal{O}(1)}$.

- Find a minimum vertex cover in time $1.2528^{\mathsf{vc}} \cdot n^{O(1)}$ [Harris et al., 2024].
- Build optimum solution by the **dynamic programming**:

$$\mathtt{opt}[i,\mathcal{P},S] = \min \begin{cases} \mathtt{opt}[i-1,\mathcal{P},S], \\ 1 + \min\limits_{\substack{\mathcal{P}' \Cap \mathcal{P}(r_i)=\mathcal{P}, \\ S' \cup N(r_i)=S}} \mathtt{opt}[i-1,\mathcal{P}',S']. \end{cases}$$

$$\begin{array}{l} \mathtt{opt}[i,\mathcal{P},S] = \min |C|, \\ C \subset \{r_1, r_2, \ldots, r_i\}, \\ C \rightsquigarrow (\mathcal{P},S) \end{array}$$

- Algorithm **brute forces** all partitions of vertex cover.



Theorem (C., Foucaud, Majumdar, Tale, CIAC 2025)

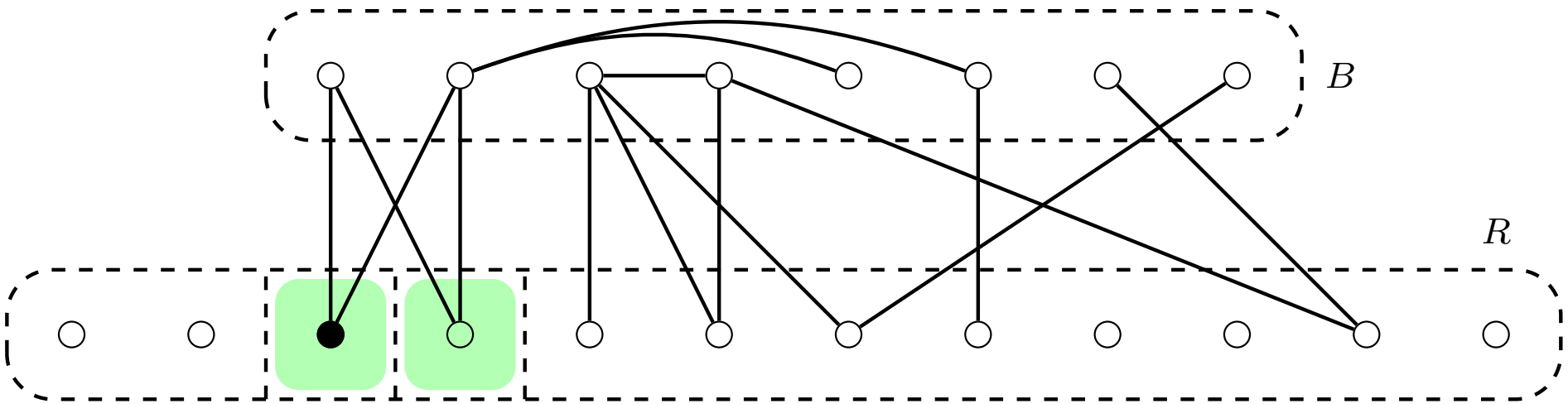
MINIMUM LD-SET *admits an algorithm with running time* $2^{\mathcal{O}(\mathsf{vc}\log\mathsf{vc})} \cdot n^{\mathcal{O}(1)}$.

- Find a minimum vertex cover in time $1.2528^{\mathsf{vc}} \cdot n^{O(1)}$ [Harris et al., 2024].
- Build optimum solution by the **dynamic programming**:

$$\mathtt{opt}[i,\mathcal{P},S] = \min \begin{cases} \mathtt{opt}[i-1,\mathcal{P},S], \\ 1 + \min\limits_{\substack{\mathcal{P}' \Cap \mathcal{P}(r_i)=\mathcal{P}, \\ S' \cup N(r_i)=S}} \mathtt{opt}[i-1,\mathcal{P}',S']. \end{cases}$$

$$\begin{array}{l} \mathtt{opt}[i,\mathcal{P},S] = \min |C|, \\ C \subset \{r_1, r_2, \ldots, r_i\}, \\ C \rightsquigarrow (\mathcal{P},S) \end{array}$$

- Algorithm **brute forces** all partitions of vertex cover.



Theorem (C., Foucaud, Majumdar, Tale, CIAC 2025)

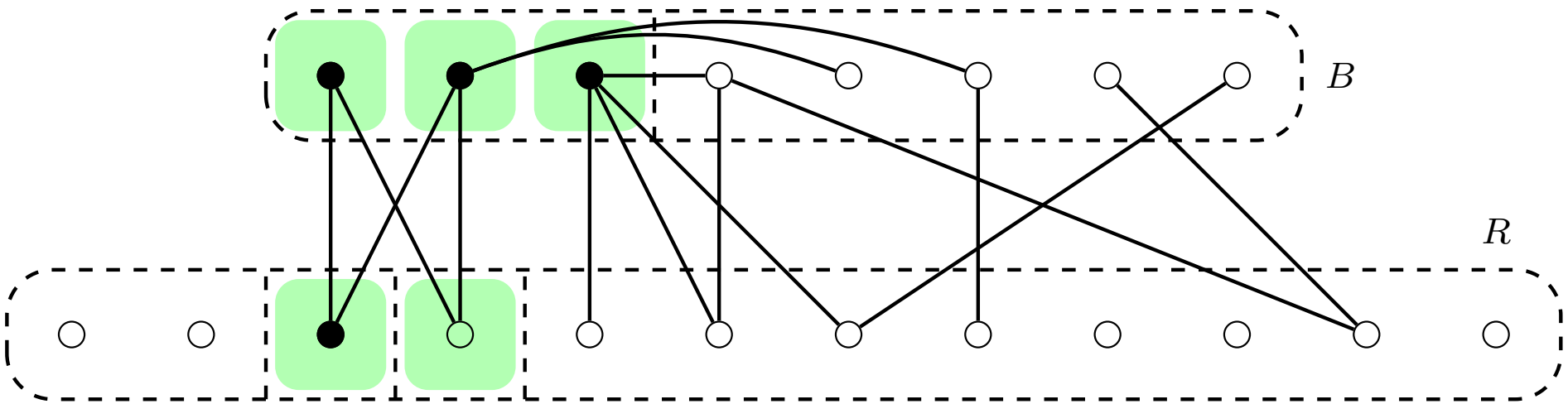
MINIMUM LD-SET *admits an algorithm with running time* $2^{\mathcal{O}(\mathsf{vc}\log\mathsf{vc})} \cdot n^{\mathcal{O}(1)}$.

- Find a minimum vertex cover in time $1.2528^{\mathsf{vc}} \cdot n^{O(1)}$ [Harris et al., 2024].
- Build optimum solution by the **dynamic programming**:

$$\mathtt{opt}[i,\mathcal{P},S] = \min \begin{cases} \mathtt{opt}[i-1,\mathcal{P},S], \\ 1 + \min\limits_{\substack{\mathcal{P}' \Cap \mathcal{P}(r_i)=\mathcal{P}, \\ S' \cup N(r_i)=S}} \mathtt{opt}[i-1,\mathcal{P}',S']. \end{cases}$$

$$\begin{array}{l} \mathtt{opt}[i,\mathcal{P},S] = \min |C|, \\ C \subset \{r_1, r_2, \ldots, r_i\}, \\ C \rightsquigarrow (\mathcal{P},S) \end{array}$$

- Algorithm **brute forces** all partitions of vertex cover.



Theorem (C., Foucaud, Majumdar, Tale, CIAC 2025)

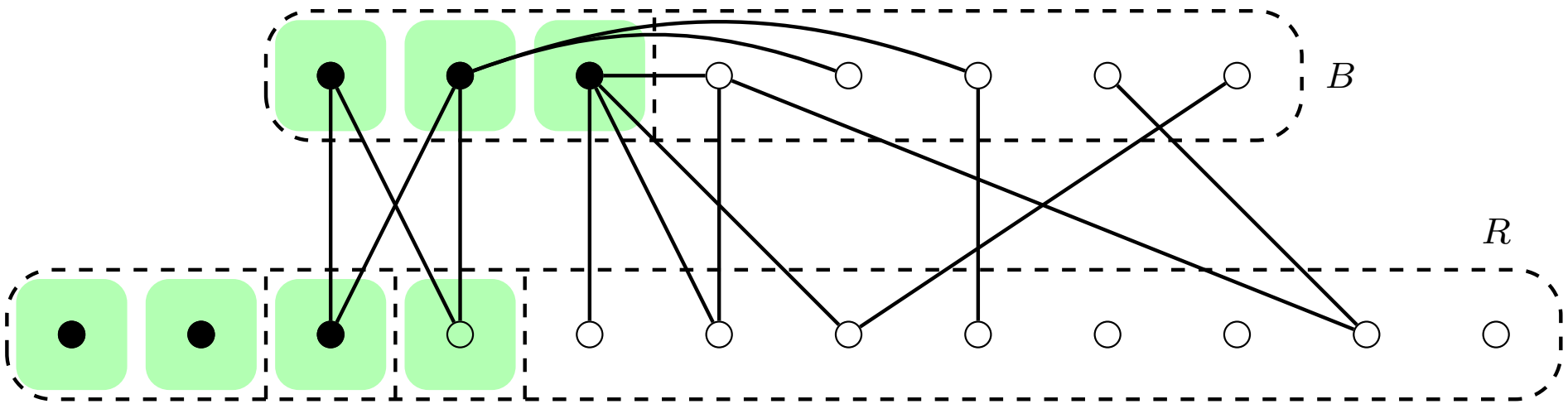
MINIMUM LD-SET *admits an algorithm with running time* $2^{\mathcal{O}(\mathsf{vc}\log\mathsf{vc})} \cdot n^{\mathcal{O}(1)}$.

- Find a minimum vertex cover in time $1.2528^{\mathsf{vc}} \cdot n^{O(1)}$ [Harris et al., 2024].
- Build optimum solution by the **dynamic programming**:

$$\mathtt{opt}[i,\mathcal{P},S] = \min \begin{cases} \mathtt{opt}[i-1,\mathcal{P},S], \\ 1 + \min\limits_{\substack{\mathcal{P}' \Cap \mathcal{P}(r_i)=\mathcal{P}, \\ S' \cup N(r_i)=S}} \mathtt{opt}[i-1,\mathcal{P}',S']. \end{cases}$$

$$\begin{array}{l} \mathtt{opt}[i,\mathcal{P},S] = \min |C|, \\ C \subset \{r_1, r_2, \ldots, r_i\}, \\ C \rightsquigarrow (\mathcal{P},S) \end{array}$$

- Algorithm **brute forces** all partitions of vertex cover.



Theorem (C., Foucaud, Majumdar, Tale, CIAC 2025)

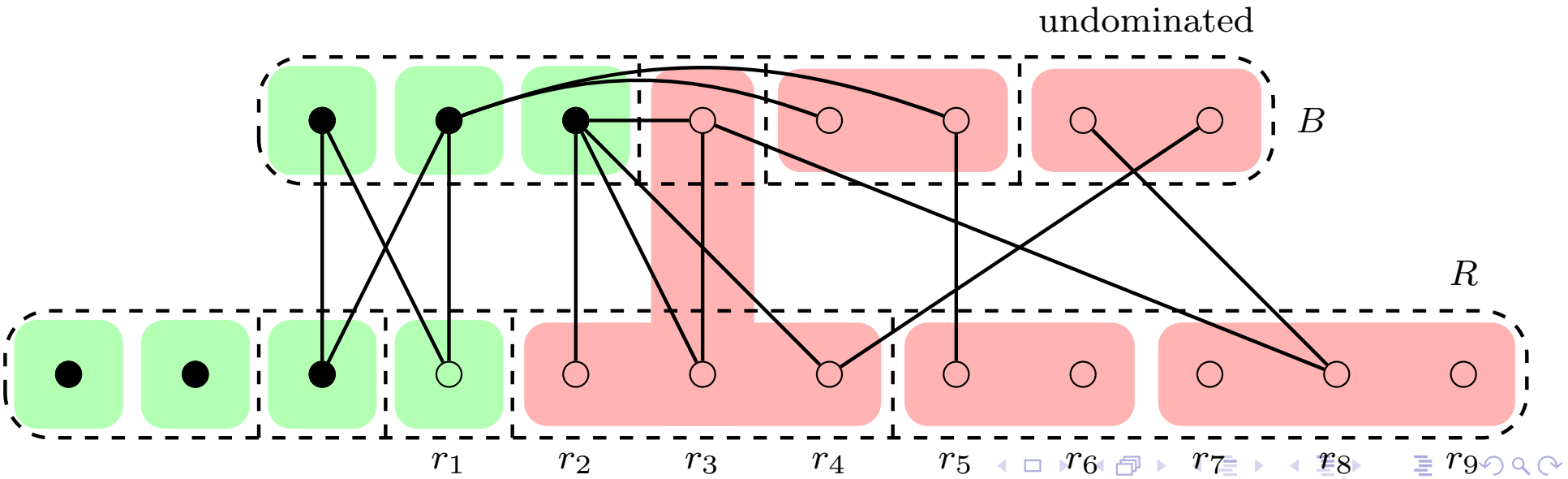
MINIMUM LD-SET *admits an algorithm with running time* $2^{\mathcal{O}(\mathsf{vc}\log\mathsf{vc})} \cdot n^{\mathcal{O}(1)}$.

- Find a minimum vertex cover in time $1.2528^{\mathsf{vc}} \cdot n^{O(1)}$ [Harris et al., 2024].
- Build optimum solution by the **dynamic programming**:

$$\mathtt{opt}[i,\mathcal{P},S] = \min \begin{cases} \mathtt{opt}[i-1,\mathcal{P},S], \\ 1 + \min\limits_{\substack{\mathcal{P}' \Cap \mathcal{P}(r_i)=\mathcal{P}, \\ S' \cup N(r_i)=S}} \mathtt{opt}[i-1,\mathcal{P}',S']. \end{cases}$$

$$\begin{aligned} &\mathtt{opt}[i,\mathcal{P},S] = \min |C|, \\ &C \subset \{r_1, r_2, \ldots, r_i\}, \\ &C \rightsquigarrow (\mathcal{P},S) \end{aligned}$$

- Algorithm **brute forces** all partitions of vertex cover.



Theorem (C., Foucaud, Majumdar, Tale, CIAC 2025)

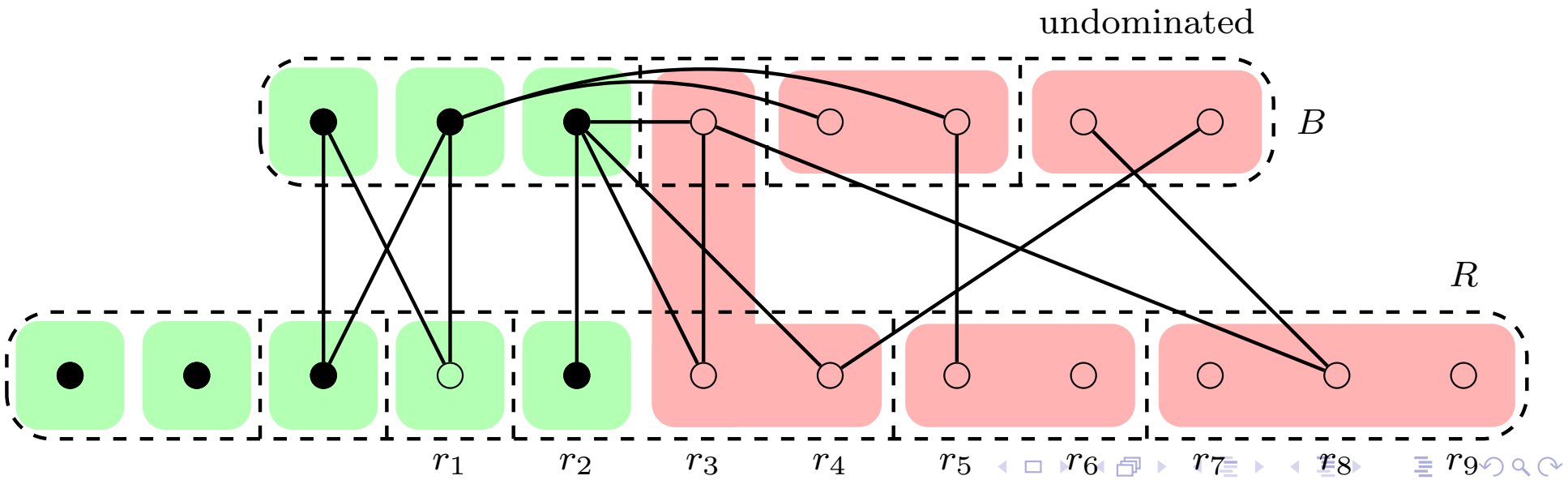
MINIMUM LD-SET *admits an algorithm with running time* $2^{\mathcal{O}(\mathsf{vc}\log\mathsf{vc})} \cdot n^{\mathcal{O}(1)}$.

- Find a minimum vertex cover in time $1.2528^{\mathsf{vc}} \cdot n^{O(1)}$ [Harris et al., 2024].
- Build optimum solution by the **dynamic programming**:

$$\mathtt{opt}[i,\mathcal{P},S] = \min \begin{cases} \mathtt{opt}[i-1,\mathcal{P},S], \\ 1 + \min\limits_{\substack{\mathcal{P}' \Cap \mathcal{P}(r_i)=\mathcal{P}, \\ S' \cup N(r_i)=S}} \mathtt{opt}[i-1,\mathcal{P}',S']. \end{cases}$$

$$\begin{array}{l} \mathtt{opt}[i,\mathcal{P},S] = \min |C|, \\ C \subset \{r_1, r_2, \ldots, r_i\}, \\ C \rightsquigarrow (\mathcal{P},S) \end{array}$$

- Algorithm **brute forces** all partitions of vertex cover.



Theorem (C., Foucaud, Majumdar, Tale, CIAC 2025)

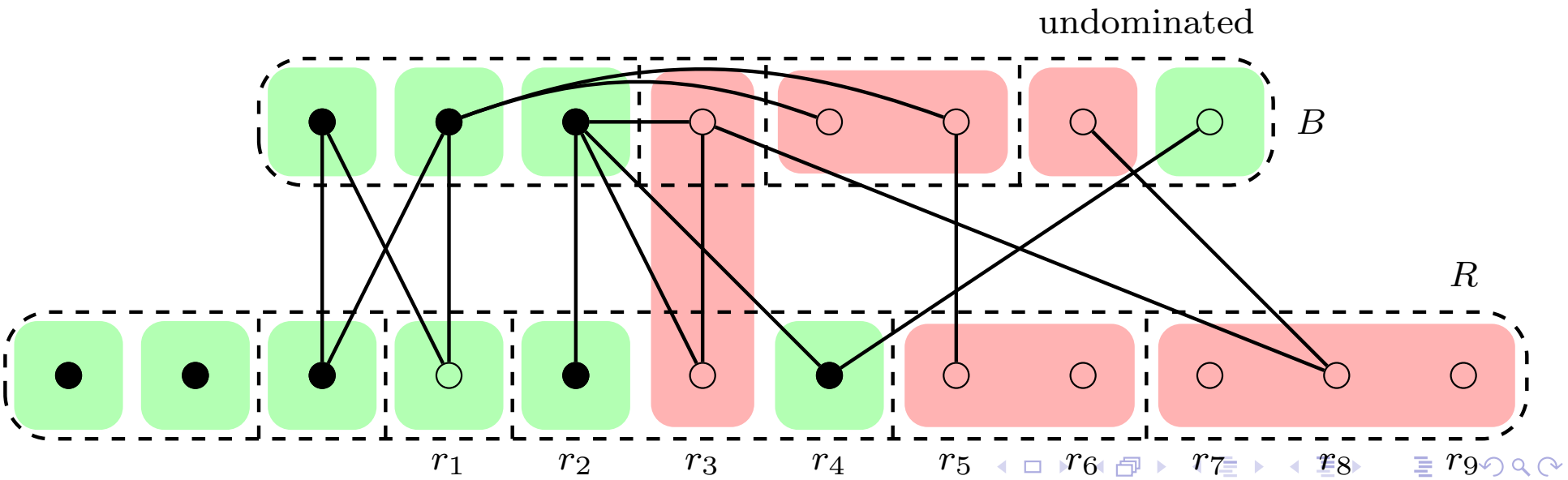
MINIMUM LD-SET *admits an algorithm with running time* $2^{\mathcal{O}(\mathsf{vc}\log\mathsf{vc})} \cdot n^{\mathcal{O}(1)}$.

- Find a minimum vertex cover in time $1.2528^{\mathsf{vc}} \cdot n^{O(1)}$ [Harris et al., 2024].
- Build optimum solution by the **dynamic programming**:

$$\mathtt{opt}[i,\mathcal{P},S] = \min \begin{cases} \mathtt{opt}[i-1,\mathcal{P},S], \\ 1 + \min\limits_{\substack{\mathcal{P}' \Cap \mathcal{P}(r_i)=\mathcal{P}, \\ S' \cup N(r_i)=S}} \mathtt{opt}[i-1,\mathcal{P}',S']. \end{cases}$$

$$\begin{aligned} &\mathtt{opt}[i,\mathcal{P},S] = \min |C|, \\ &C \subset \{r_1, r_2, \ldots, r_i\}, \\ &C \rightsquigarrow (\mathcal{P},S) \end{aligned}$$

- Algorithm **brute forces** all partitions of vertex cover.



Theorem (C., Foucaud, Majumdar, Tale, CIAC 2025)

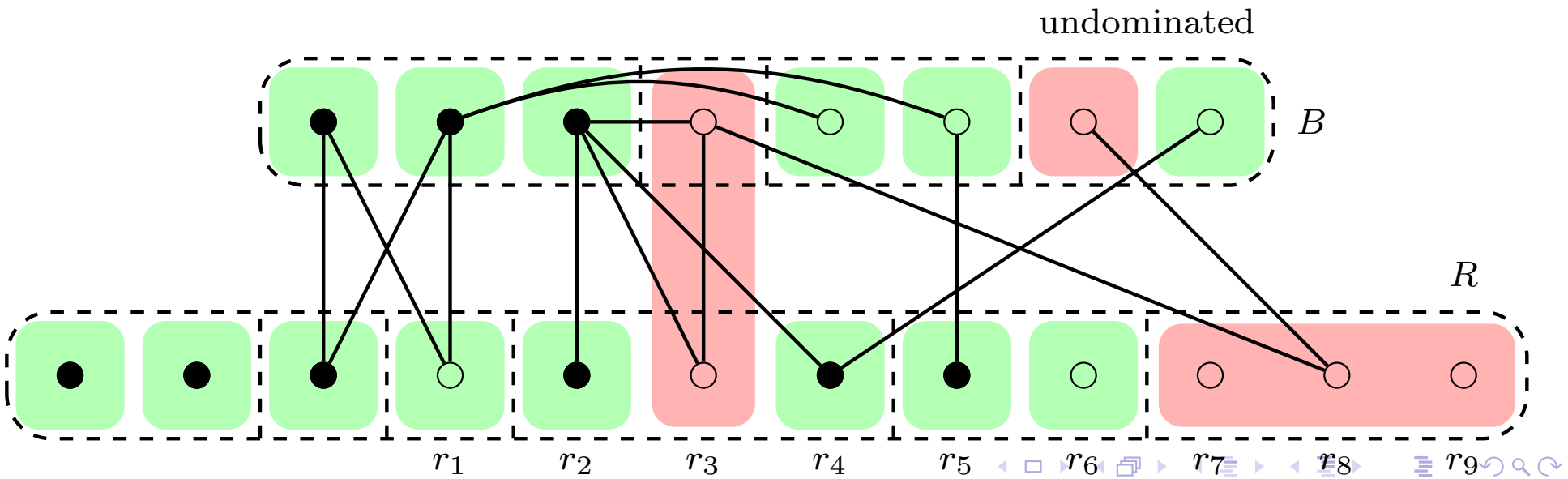
MINIMUM LD-SET *admits an algorithm with running time* $2^{\mathcal{O}(\mathsf{vc}\log\mathsf{vc})} \cdot n^{\mathcal{O}(1)}$.

- Find a minimum vertex cover in time $1.2528^{\mathsf{vc}} \cdot n^{O(1)}$ [Harris et al., 2024].
- Build optimum solution by the **dynamic programming**:

$$\mathtt{opt}[i,\mathcal{P},S] = \min \begin{cases} \mathtt{opt}[i-1,\mathcal{P},S], \\ 1 + \min\limits_{\substack{\mathcal{P}' \Cap \mathcal{P}(r_i)=\mathcal{P}, \\ S' \cup N(r_i)=S}} \mathtt{opt}[i-1,\mathcal{P}',S']. \end{cases}$$

$$\begin{aligned} &\mathtt{opt}[i,\mathcal{P},S] = \min |C|, \\ &C \subset \{r_1, r_2, \ldots, r_i\}, \\ &C \rightsquigarrow (\mathcal{P},S) \end{aligned}$$

- Algorithm **brute forces** all partitions of vertex cover.



Theorem (C., Foucaud, Majumdar, Tale, CIAC 2025)

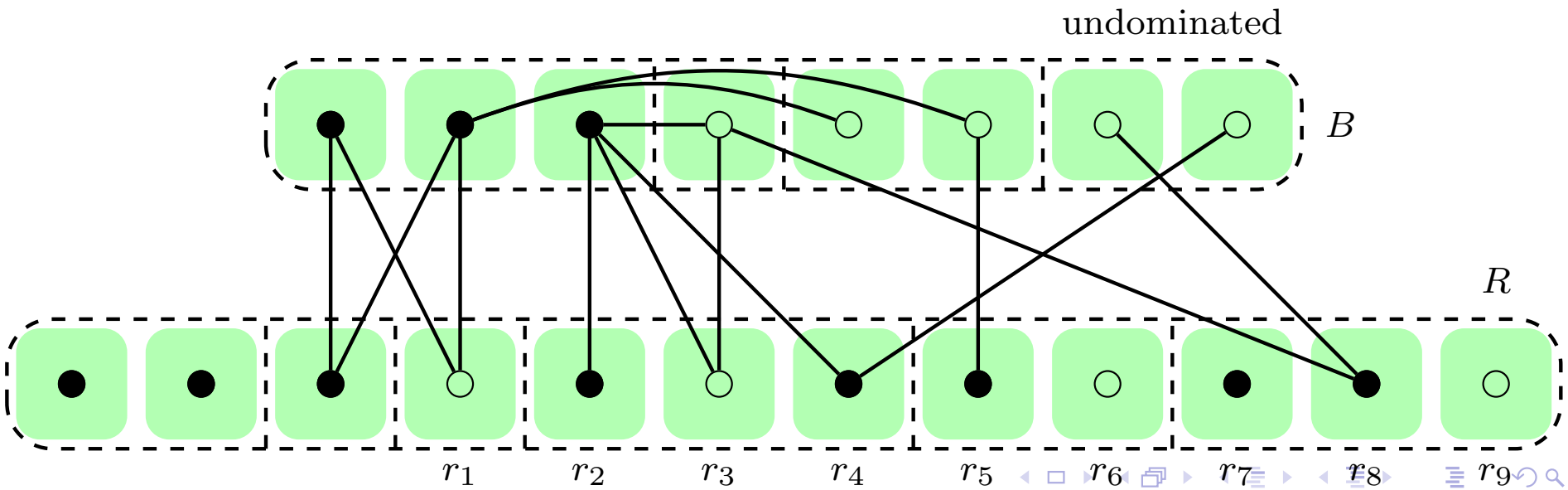
MINIMUM LD-SET *admits an algorithm with running time* $2^{\mathcal{O}(\mathsf{vc}\log\mathsf{vc})} \cdot n^{\mathcal{O}(1)}$.

- Find a minimum vertex cover in time $1.2528^{\mathsf{vc}} \cdot n^{O(1)}$ [Harris et al., 2024].
- Build optimum solution by the **dynamic programming**:

$$\mathtt{opt}[i,\mathcal{P},S] = \min \begin{cases} \mathtt{opt}[i-1,\mathcal{P},S], \\ 1 + \min\limits_{\substack{\mathcal{P}' \Cap \mathcal{P}(r_i)=\mathcal{P}, \\ S' \cup N(r_i)=S}} \mathtt{opt}[i-1,\mathcal{P}',S']. \end{cases}$$

$$\begin{array}{l} \mathtt{opt}[i,\mathcal{P},S] = \min |C|, \\ C \subset \{r_1, r_2, \ldots, r_i\}, \\ C \rightsquigarrow (\mathcal{P},S) \end{array}$$

- Algorithm **brute forces** all partitions of vertex cover.



Theorem (C., Foucaud, Majumdar, Tale, CIAC 2025)

MINIMUM LD-SET *admits an algorithm with running time* $2^{\mathcal{O}(\mathsf{vc}\log\mathsf{vc})} \cdot n^{\mathcal{O}(1)}$.

- Find a minimum vertex cover in time $1.2528^{\mathsf{vc}} \cdot n^{O(1)}$ [Harris et al., 2024].
- Build optimum solution by the **dynamic programming**:

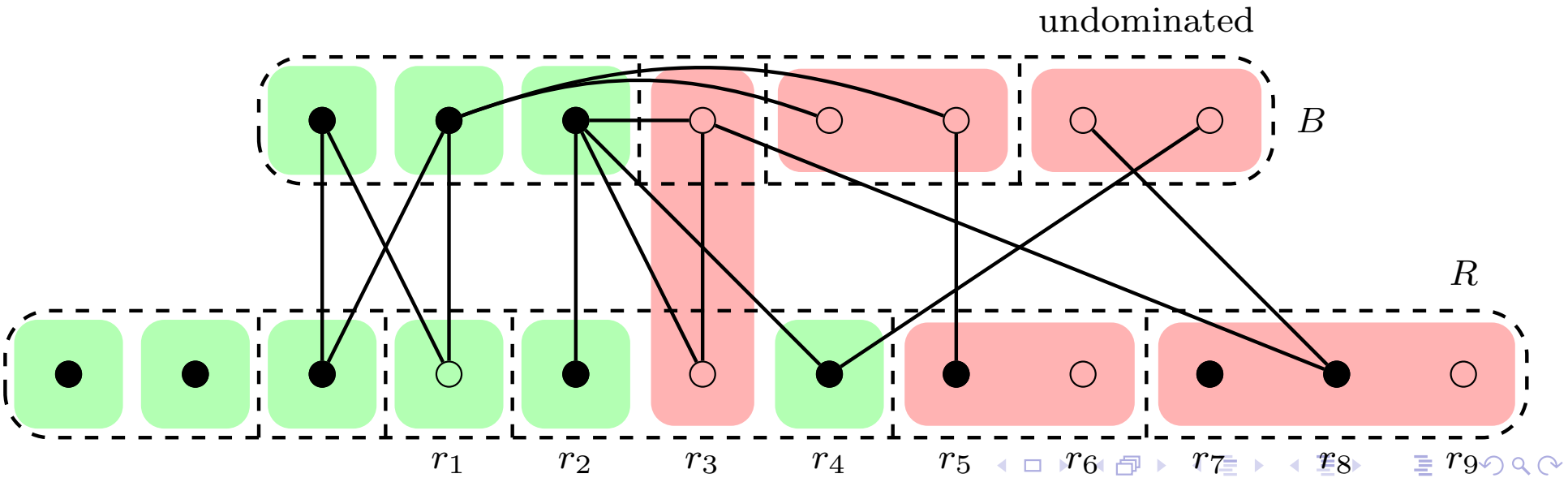
$$\mathtt{opt}[i,\mathcal{P},S] = \min \begin{cases} \mathtt{opt}[i-1,\mathcal{P},S], \\ 1 + \min\limits_{\substack{\mathcal{P}' \uplus \mathcal{P}(r_i)=\mathcal{P}, \\ S' \cup N(r_i)=S}} \mathtt{opt}[i-1,\mathcal{P}',S']. \end{cases}$$

Running time:

$\mathcal{P}$: $2^{\mathsf{vc}\log\mathsf{vc}} \cdot |R|$

S : 2^{vc}

- Algorithm **brute forces** all partitions of vertex cover.



Some questions...

- Can our double-exponential lower bound reduction with respect to treewidth be adapted to feedback vertex set?
- Can the running time of the slightly super-exponential FPT algorithm with respect to vertex cover number be reduced to a single exponential running time?
- Can the FPT algorithm with respect to vertex cover number be adapted for distance to cluster?

Thank You!