Structural and algorithmic aspects of identification problems in graphs

Thèse présentée à l'Université Clermont Auvergne, France
École Doctorale des Sciences Pour l'Ingénieur, Unité de recherche: LIMOS

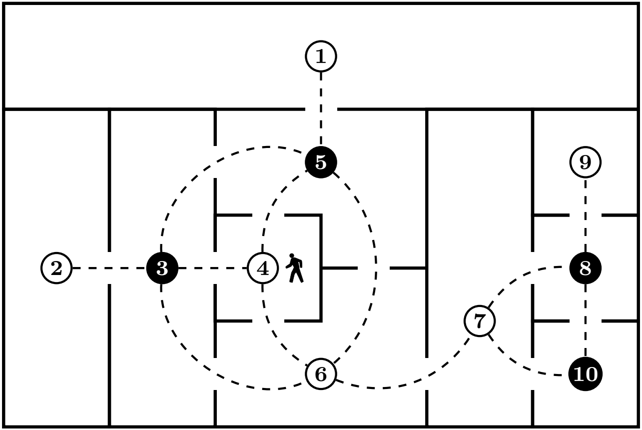
Auteur :	Dipayan CHAKRABORTY
Directrice de thèse :	Annegret WAGLER
Co-encadrant :	Florent FOUCAUD
Co-encadrant :	Michael HENNING

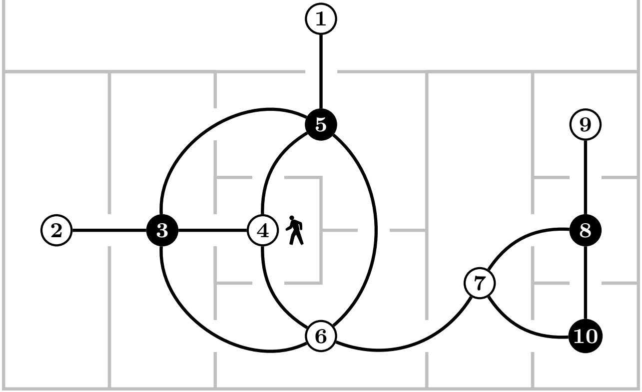
Date de soutenance : 09 decembre 2024
pour obtenir le grade de Docteur (spécialité : informatique)

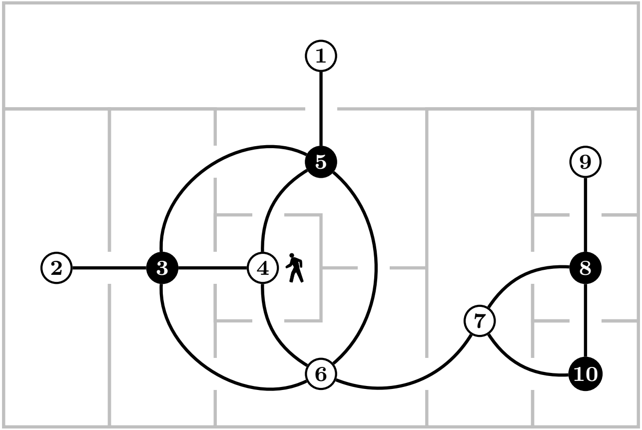
Devant le jury composé de :

Annegret WAGLER	Professeure, Université Clermont Auvergne	Directrice
Florent FOUCAUD	Maître de conférences, Université Clermont Auvergne	Co-encadrant
Michael HENNING	Professeur, University of Johannesburg	Co-encadrant
Paul DORBEC	Professeur, Université de Caen-Normandie	Examinateur
Ralf KLASING	Directeur de Recherche, CNRS, Université de Bordeaux	Examinateur
Tero LAIHONEN	Professeur, University of Turku	Rapporteur
Arnaud PÊCHER	Professeur, Université de Bordeaux	Rapporteur

Part I. Introduction: Identification problems in graphs


1
5
9
2
3
4
8
7
6
10


1
5
9
2
3
4
8
7
6
10



Graph $G = (V, E)$

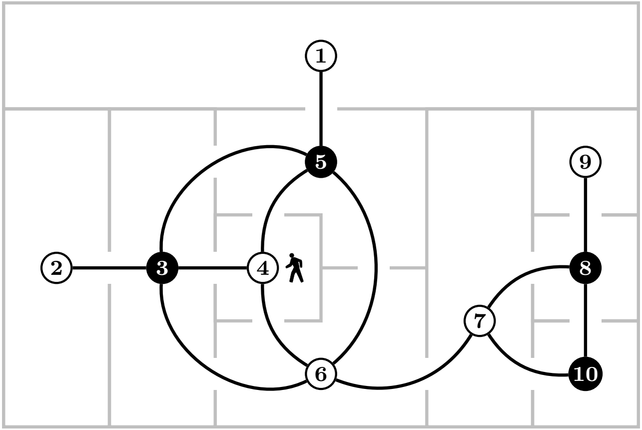
Open neighborhood:
$N(v) = \{u : uv \in E\}$

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

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

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

"**Code**" C = set of black vertices



Graph $G = (V, E)$

Open neighborhood:

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

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

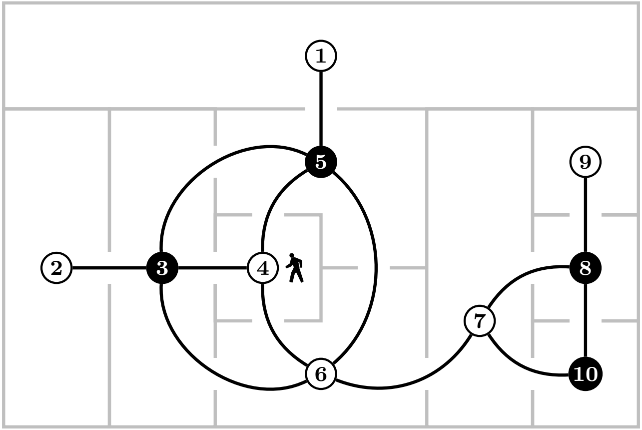
Closed neighborhood:

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

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

"**Code**" C = set of black vertices

(1) A detector can monitor upto distance 1



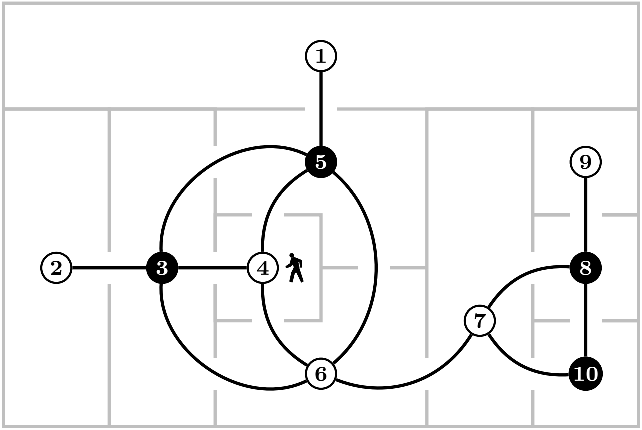
Graph $G = (V, E)$

Open neighborhood:
$N(v) = \{u : uv \in E\}$
$N(7) = \{6, 8, 10\}$

Closed neighborhood:
$N[v] = N(v) \cup \{v\}$
$N[7] = \{6, 8, 10, 7\}$

"**Code**" C = set of black vertices

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



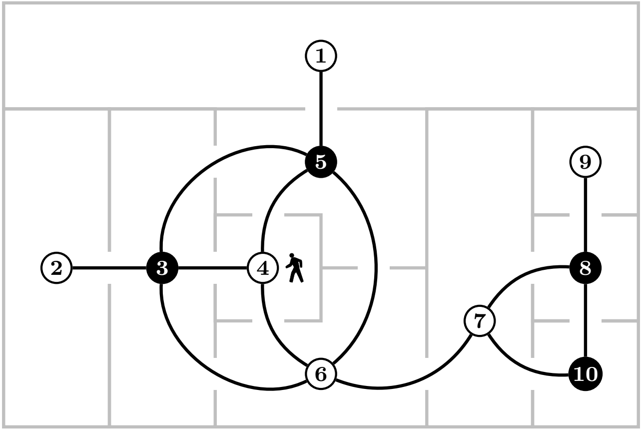
Graph $G = (V, E)$

Open neighborhood:
$N(v) = \{u : uv \in E\}$
$N(7) = \{6, 8, 10\}$

Closed neighborhood:
$N[v] = N(v) \cup \{v\}$
$N[7] = \{6, 8, 10, 7\}$

"**Code**" C = set of black vertices

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



Graph $G = (V, E)$

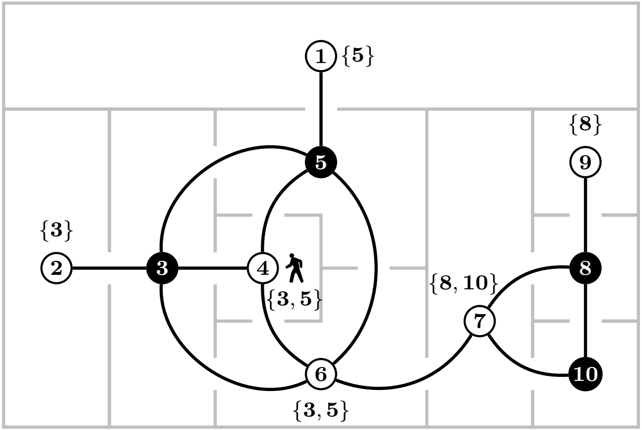
Open neighborhood:
$N(v) = \{u : uv \in E\}$
$N(7) = \{6, 8, 10\}$

Closed neighborhood:
$N[v] = N(v) \cup \{v\}$
$N[7] = \{6, 8, 10, 7\}$

"**Code**" C = set of black vertices

Dominating set: A set $C \subseteq V$ such that $N[v] \cap C \neq \emptyset$ for all $v \in V$
Total-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 $G = (V, E)$

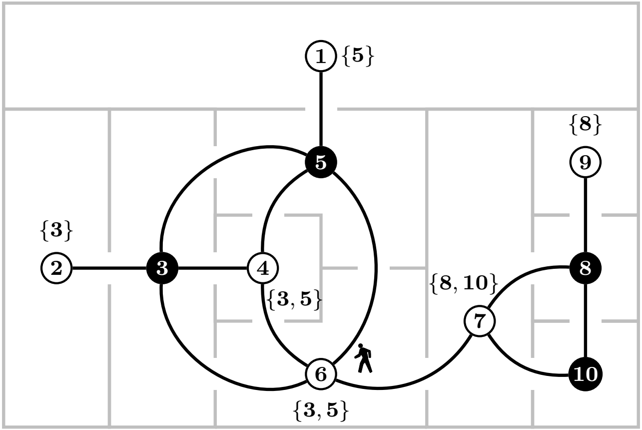
Open neighborhood:
$N(v) = \{u : uv \in E\}$
$N(7) = \{6, 8, 10\}$

Closed neighborhood:
$N[v] = N(v) \cup \{v\}$
$N[7] = \{6, 8, 10, 7\}$

"**Code**" C = set of black vertices

Dominating set: A set $C \subseteq V$ such that $N[v] \cap C \neq \emptyset$ for all $v \in V$
Total-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 $G = (V, E)$

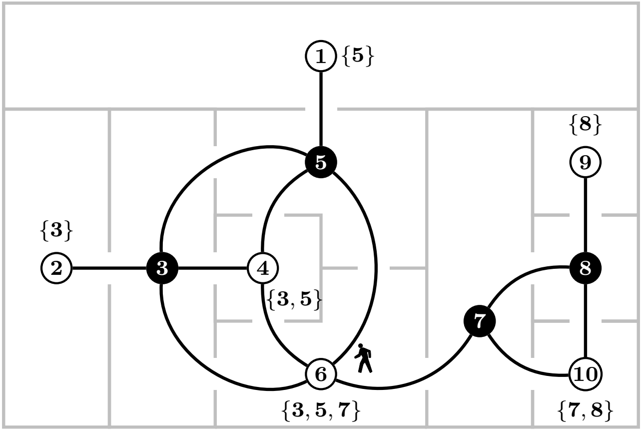
Open neighborhood:
$N(v) = \{u : uv \in E\}$
$N(7) = \{6, 8, 10\}$

Closed neighborhood:
$N[v] = N(v) \cup \{v\}$
$N[7] = \{6, 8, 10, 7\}$

"**Code**" C = set of black vertices

Dominating set: A set $C \subseteq V$ such that $N[v] \cap C \neq \emptyset$ for all $v \in V$
Total-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 $G = (V, E)$

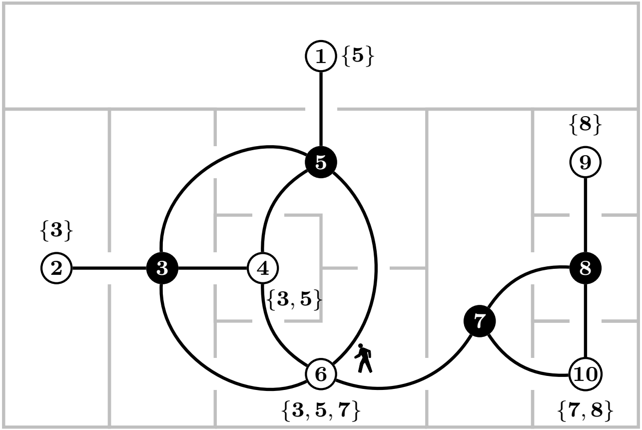
Open neighborhood:
$N(v) = \{u : uv \in E\}$
$N(7) = \{6, 8, 10\}$

Closed neighborhood:
$N[v] = N(v) \cup \{v\}$
$N[7] = \{6, 8, 10, 7\}$

"**Code**" C = set of black vertices

Dominating set: A set $C \subseteq V$ such that $N[v] \cap C \neq \emptyset$ for all $v \in V$
Total-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:
$N[v] = N(v) \cup \{v\}$
$N[7] = \{6, 8, 10, 7\}$

"**Code**" C = set of black vertices

Dominating set: A set $C \subseteq V$ such that $N[v] \cap C \neq \emptyset$ for all $v \in V$
Total-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
$N(u) \cap C \neq N(v) \cap C \iff \big(N(u) \triangle N(v)\big) \cap C \neq \emptyset$ for all $u, v \in V \setminus C$

Open-separator

A vertex subset C of a graph G is called...

Locating set: if $\big(N(u) \triangle N(v)\big) \cap C \neq \emptyset$ for all $u, v \in V \setminus C$

No faults in detectors: Detectors can distinguish between its vertex and its neighbor

Closed-separator

Closed-separating set: if $\big(N[u] \triangle N[v]\big) \cap C \neq \emptyset$ for all $u, v \in V$

Detector fault type 1: Detector cannot distinguish between itself and its neighbors

Open-separating set: if $\big(N(u) \triangle N(v)\big) \cap C \neq \emptyset$ for all $u, v \in V$

Detector fault type 2: Detector is completely disabled / destroyed

Full-separating set: if $\big(N[u] \triangle N[v]\big) \cap C = \big(N(u) \triangle N(v)\big) \cap C \neq \emptyset$ for all $u, v \in V$

Detector fault type 1 and detector fault type 2

A code C must intersect the following sets...

Sep	L-Sep	C-Sep	O-Sep	F-Sep
adj	$N(u) \triangle N(v)$	$N[u] \triangle N[v]$	$N(u) \triangle N(v)$	$N[u] \triangle N[v]$
non-adj	$N[u] \triangle N[v]$	$N[u] \triangle N[v]$	$N(u) \triangle N(v)$	$N(u) \triangle N(v)$

A code C must intersect the following sets...

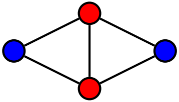
Sep	L-Sep		C-Sep		O-Sep		F-Sep	
Code	LD	LTD	CD	CTD	OD	OTD	FD	FTD
adj	$N(u) \triangle N(v)$		$N[u] \triangle N[v]$		$N(u) \triangle N(v)$		$N[u] \triangle N[v]$	
non-adj	$N[u] \triangle N[v]$						$N(u) \triangle N(v)$	
D/TD	$N[u]$	$N(u)$	$N[u]$	$N(u)$	$N[u]$	$N(u)$	$N[u]$	$N(u)$
	Locating-dominating	Locating total-dominating	Closed-separating dominating	Closed-separating total-dominating	Open-separating dominating	Open-separating total-dominating	Full-separating dominating	Full-separating total-dominating

A code C must intersect the following sets...

Sep	L-Sep		C-Sep		O-Sep		F-Sep	
Code	LD	LTD	CD	CTD	OD	OTD	FD	FTD
adj	$N(u) \triangle N(v)$		$N[u] \triangle N[v]$		$N(u) \triangle N(v)$		$N[u] \triangle N[v]$	
non-adj	$N[u] \triangle N[v]$						$N(u) \triangle N(v)$	
D/TD	$N[u]$	$N(u)$	$N[u]$	$N(u)$	$N[u]$	$N(u)$	$N[u]$	$N(u)$

Existence of codes...

Codes	LD	LTD	OD	OTD	CD	CTD	FD	FTD
no isolated vertices		x		x		x		x
no open twins			x	x			x	x
no closed twins					x	x	x	x



open twins $N(u) = N(v) \iff N(u) \triangle N(v) = \emptyset$

closed twins $N[u] = N[v] \iff N[u] \triangle N[v] = \emptyset$

Year	Code	Authors
1988:	LD-code	P. Slater
1998:	CD-code	M. Karpovsky, K. Chakrabarty & L. Levitin
2002: **2010:**	OTD-code	I. Honkala, T. Laihonen, S Ranto S. Seo & P. Slater
2006:	LTD-code ITD-code	T. Haynes, M. Henning & J. Howard
2024:	OD-code FD-code FTD-code	D. Chakraborty & A. Wagler

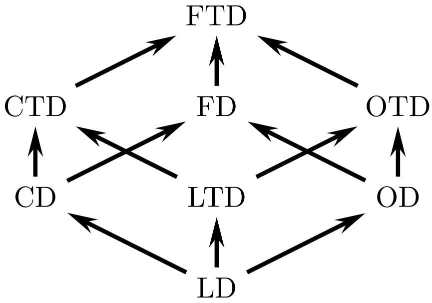
$\mathrm{X} \in \textsc{Codes} = \{\mathrm{LD}, \mathrm{LTD}, \mathrm{CD}, \mathrm{CTD}, \mathrm{OD}, \mathrm{OTD}, \mathrm{FD}, \mathrm{FTD}\}$

X-number: $\gamma^{\mathrm{X}}(G) = \min\{|C| : C \text{ is an X-code of } G\}$

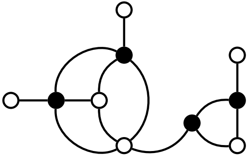
Domination number: $\gamma(G) = \min\{|C| : C \text{ is a dominating set of } G\}$

Total-domination number:
$\gamma_t(G) = \min\{|C| : C \text{ is a total-dominating set of } G\}$

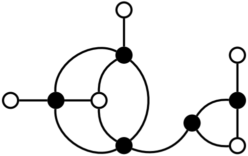
$$\begin{aligned} \gamma(G) &\leq \gamma^{\mathrm{X}}(G) \quad \text{if X is based on domination} \\ \gamma_t(G) &\leq \gamma^{\mathrm{X}}(G) \quad \text{if X is based on total-domination} \end{aligned}$$



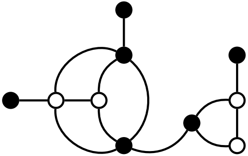
$X' \longrightarrow X$ stands for
$\gamma^{\mathrm{X}'}(G) \leq \gamma^{\mathrm{X}}(G)$



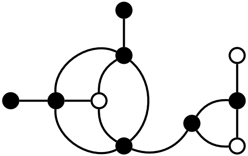
$\gamma^{\mathrm{LD}}(G) = \gamma^{\mathrm{LTD}}(G) = 4.$



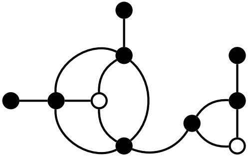
$\gamma^{\mathrm{OD}}(G) = \gamma^{\mathrm{OTD}}(G) = 5.$



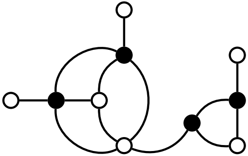
$\gamma^{\mathrm{ID}}(G) = 6.$



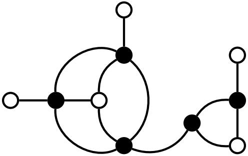
$\gamma^{\mathrm{FD}}(G) = 7.$



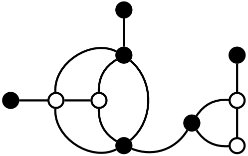
$\gamma^{\mathrm{ITD}}(G) = \gamma^{\mathrm{FTD}}(G) = 8.$



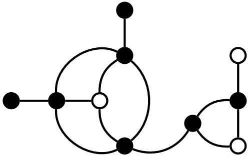
$\gamma^{\mathrm{LD}}(G) = \gamma^{\mathrm{LTD}}(G) = 4.$



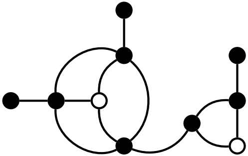
$\gamma^{\mathrm{OD}}(G) = \gamma^{\mathrm{OTD}}(G) = 5.$



$\gamma^{\mathrm{ID}}(G) = 6.$



$\gamma^{\mathrm{FD}}(G) = 7.$



$\gamma^{\mathrm{ITD}}(G) = \gamma^{\mathrm{FTD}}(G) = 8.$

$\gamma^{\mathrm{LD}}(G) = 2.$

$\gamma^{\mathrm{LTD}}(G) = \gamma^{\mathrm{ID}}(G) = \gamma^{\mathrm{ITD}}(G) = 3.$

$\gamma^{\mathrm{OD}}(G) = 4.$

$\gamma^{\mathrm{OTD}}(G) = \gamma^{\mathrm{FD}}(G) = \gamma^{\mathrm{FTD}}(G) = 4.$

Part II. Structural aspects of identification problems in graphs

Locating dominating codes on subcubic graphs...

— joint work with Anni Hakanen and Tuomo Lehtilä

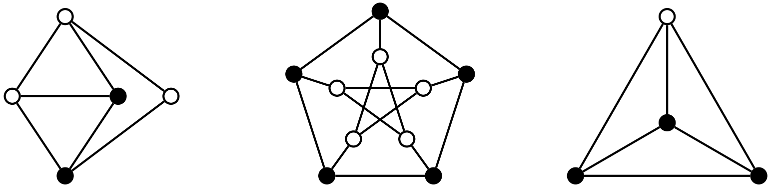
(University of Turku, Finland)

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
$N(u) \cap C \neq N(v) \cap C \iff \big(N(u) \triangle N(v)\big) \cap C \neq \emptyset$ for all $u, v \in V \setminus C$.

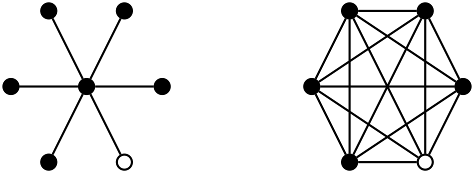
Subcubic graph: A graph in which each vertex is of degree at most 3.

Cubic graph: A graph in which each vertex is of degree exactly 3.



Conjecture (Garijo, González & Márques, 2014)

If a connected graph G on n vertices is twin-free, then $\gamma^{\text{LD}}(G) \leq \frac{n}{2}$

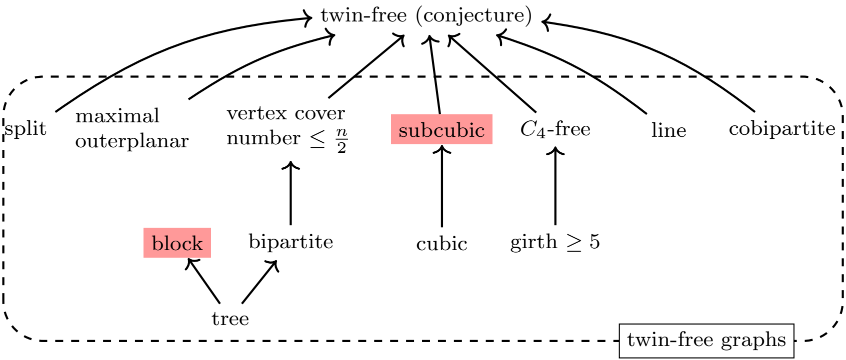


$$\gamma^{\text{LD}}(G) = n - 1,\ \gamma(G) = 1$$

Theorem (Ore, 1962)

If G is a connected graph on n vertices, then $\gamma(G) \leq \frac{n}{2}$.

The n-half conjecture is true for...



Theorem (Bousquet, Chuet, Falgas-Ravry, Jacques, Morelle, 2024)

If G is a connected twin-free graph on n vertices, then $\gamma^{\mathrm{LD}}(G) \leq \lceil \frac{5}{8}n \rceil$.

Theorem (Foucaud and Henning, 2016)

If G is a twin-free, cubic graph of order n, then $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Question (Foucaud and Henning, 2016)

Can we allow twins for cubic graphs? except $G \ncong K_4, K_{3,3} \ldots$???

Question (Foucaud and Henning, 2016)

Is the conjecture true for subcubic graphs?

Theorem (Foucaud and Henning, 2016)

If G is a twin-free, cubic graph of order n, then $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Question (Foucaud and Henning, 2016)

Can we allow twins for cubic graphs? except $G \ncong K_4, K_{3,3} \ldots$.???

ANSWER: **YES**; except $G \ncong K_4, K_{3,3}$. [C., Hakanen & Lehtilä, 2024]

Question (Foucaud and Henning, 2016)

Is the conjecture true for subcubic graphs?

ANSWER: **YES**; also with closed twins and degree 3 open twins.

[C., Hakanen & Lehtilä, 2024]

Theorem (C., Hakanen and Lehtilä, 2024)

Let $G \ncong K_{3,3}$ be a connected subcubic graph on n vertices, with at least 7 edges, and without open twins of degree 1 or 2. Then, $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Theorem (C., Hakanen and Lehtilä, 2024)

If a connected subcubic graph G on n vertices is twin-free, then $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Theorem (C., Hakanen and Lehtilä, 2024)

Let $G \ncong K_{3,3}$ be a connected subcubic graph on n vertices, with at least 7 edges, and without open twins of degree 1 or 2. Then, $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Theorem (C., Hakanen and Lehtilä, 2024)

If a connected subcubic graph G on n vertices is twin-free, then $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Theorem (C., Hakanen and Lehtilä, 2024)

If G is a connected, open-twin-free, subcubic graph on n vertices other than K_3 or K_4, then $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Theorem (C., Hakanen and Lehtilä, 2024)

Let $G \ncong K_{3,3}$ be a connected subcubic graph on n vertices, with at least 7 edges, and without open twins of degree 1 or 2. Then, $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Theorem (C., Hakanen and Lehtilä, 2024)

If a connected subcubic graph G on n vertices is twin-free, then $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Proof technique.

- Proof by induction on $n + m$; $m = |E|$. Determine that result is true for some “smallest” $n + m$.

Theorem (C., Hakanen and Lehtilä, 2024)

If a connected subcubic graph G on n vertices is twin-free, then $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Proof technique.

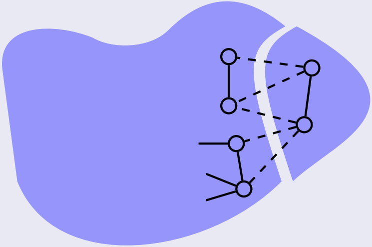
- Proof by induction on $n + m$; $m = |E|$. Determine that result is true for some "smallest" $n + m$.
- Cut a convenient part of the graph.

□

Theorem (C., Hakanen and Lehtilä, 2024)

If a connected subcubic graph G on n vertices is twin-free, then $\gamma^{\mathrm{LD}}(G) \le \frac{n}{2}$.

Proof technique.



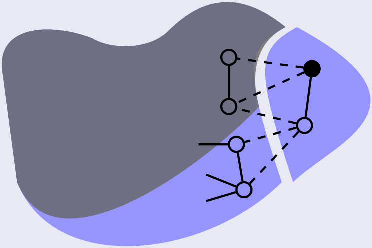
- Proof by induction on $n + m$; $m = |E|$. Determine that result is true for some "smallest" $n + m$.
- Cut a convenient part of the graph.

□

Theorem (C., Hakanen and Lehtilä, 2024)

If a connected subcubic graph G on n vertices is twin-free, then $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Proof technique.



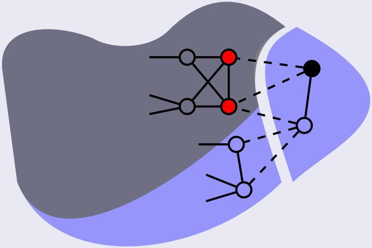
- Proof by induction on $n + m$; $m = |E|$. Determine that result is true for some "smallest" $n + m$.
- Cut a convenient part of the graph.
- If rest is twin-free, choose a desired solution.

$\square$

Theorem (C., Hakanen and Lehtilä, 2024)

If a connected subcubic graph G on n vertices is twin-free, then $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Proof technique.



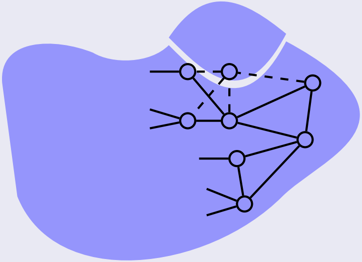
- Proof by induction on $n+m$; $m = |E|$. Determine that result is true for some "smallest" $n+m$.
- Cut a convenient part of the graph.
- If rest is twin-free, choose a desired solution.

□

Theorem (C., Hakanen and Lehtilä, 2024)

If a connected subcubic graph G on n vertices is twin-free, then $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Proof technique.



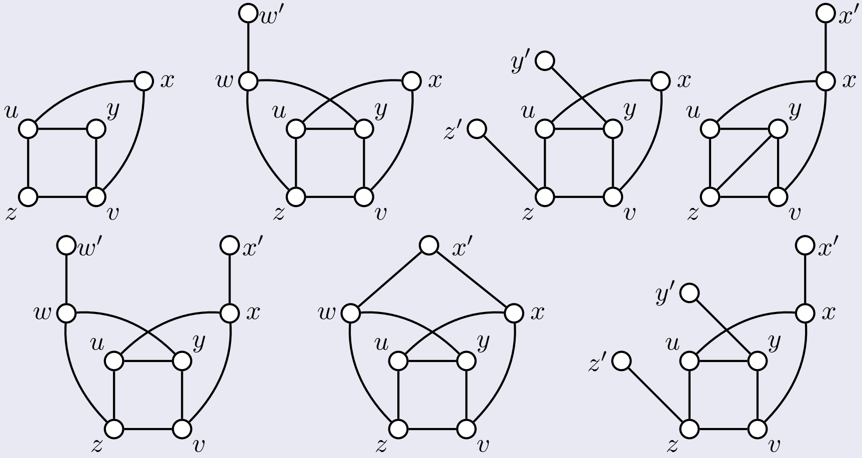
- Proof by induction on $n+m$; $m = |E|$. Determine that result is true for some "smallest" $n+m$.
- Cut a convenient part of the graph.
- If rest is twin-free, choose a desired solution.
- If not twin-free, choose another part to cut and proceed.

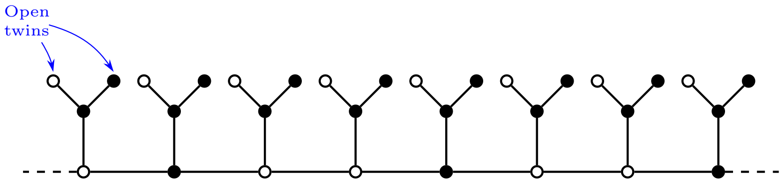
□

Theorem (C., Hakanen and Lehtilä, 2024)

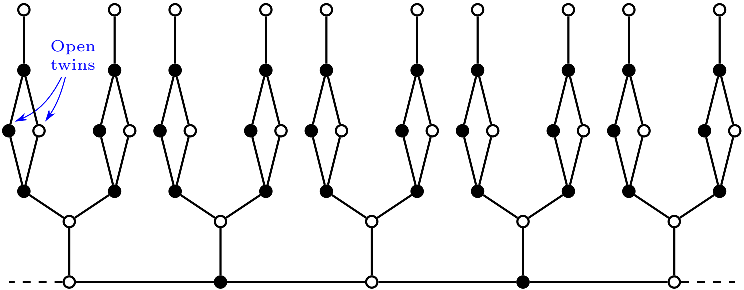
Let $G \ncong K_{3,3}$ be a connected subcubic graph on n vertices, with at least 7 edges, and without open twins of degree 1 or 2. Then, $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Proof idea.





Subcubic graph with degree 1 open twins not satisfying the conjecture.



Subcubic graph with degree 2 open twins not satisfying the conjecture.

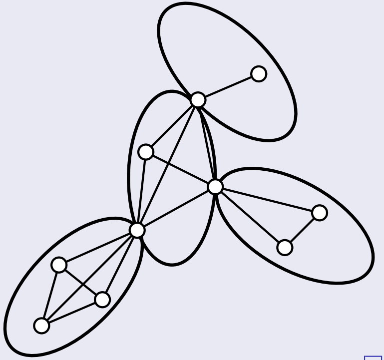
Locating dominating codes on block graphs...
— joint work with Florent Foucaud, Aline Parreau and Annegret Wagler

(Université Clermont Auvergne / Université Lyon 1)

Theorem (C., Foucaud, Parreau and Wagler, 2024)

For a connected twin-free block graph G on n vertices, $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Proof sketch.



Locating dominating codes on block graphs...

— joint work with Florent Foucaud, Aline Parreau and Annegret Wagler

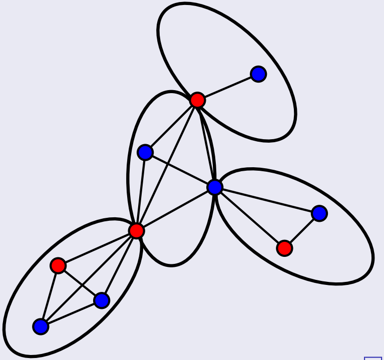
(Université Clermont Auvergne / Université Lyon 1)

Theorem (C., Foucaud, Parreau and Wagler, 2024)

For a connected twin-free block graph G on n vertices, $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Proof sketch.

- Partition V into two parts **R** and **B**.

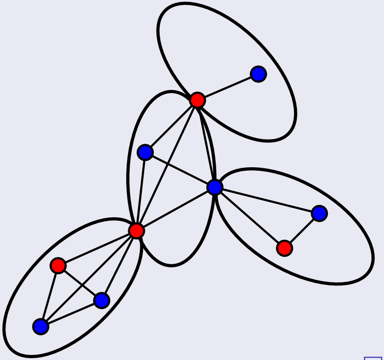


Theorem (C., Foucaud, Parreau and Wagler, 2024)

For a connected twin-free block graph G on n vertices, $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Proof sketch.

- Partition V into two parts **R** and **B**.
- Both **R** and **B** are LD codes of G.



Locating dominating codes on block graphs...

— joint work with Florent Foucaud, Aline Parreau and Annegret Wagler

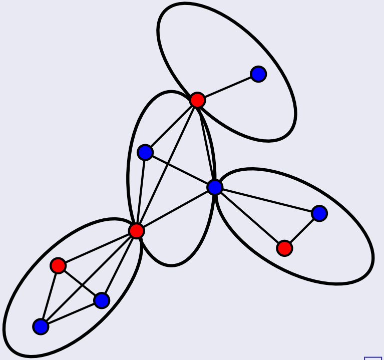
(Université Clermont Auvergne / Université Lyon 1)

Theorem (C., Foucaud, Parreau and Wagler, 2024)

For a connected twin-free block graph G on n vertices, $\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$.

Proof sketch.

- Partition V into two parts **R** and **B**.
- Both **R** and **B** are LD codes of G.
- Either one of $|\mathbf{R}|$ or $|\mathbf{B}| \leq \frac{1}{2}n$.



A similar conjecture for LTD-codes...

Theorem (Cockayne, Dawes & Hedetniemi, 1980)

If G is a connected graph on n vertices, then $\gamma_t(G) \leq \frac{2}{3}n$.

Conjecture (Foucaud & Henning, 2016)

If a connected graph G on n vertices is twin-free, then $\gamma^{\mathrm{LTD}}(G) \leq \frac{2}{3}n$

$\frac{2}{3}$-bound for locating total-dominating codes...
— joint work with F. Foucaud, A. Hakanen, M. Henning & A. Wagler

(Université Clermont Auvergne / Univ. of Johannesburg/ Univ. of Turku)

Theorem (C., Foucaud, Hakanen, Henning & Wagler, 2024)

If G is a connected subcubic graph on n vertices such that $G \ncong K_1, K_2, K_4, K_{1,3}$, then $\gamma^{\mathrm{LTD}}(G) \le \frac{2}{3}n$.

Theorem (C., Foucaud, Hakanen, Henning & Wagler, 2024)

If G is a connected, twin-free block graph on n vertices, then $\gamma^{\mathrm{LTD}}(G) \le \frac{2}{3}n$.

Part III. Algorithmic aspects of identification problems in graphs

Decision version of finding the minimum X-code in a graph:

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

X-CODE is NP-hard for all X $\in$ CODES!

NP-hardness related to FD- and FTD-codes
— joint work with Annegret Wagler

(Université Clermont Auvergne)

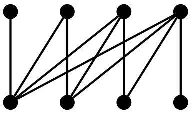
Dominating set: A set $C \subseteq V$ such that $N[v] \cap C \neq \emptyset$ for all $v \in V$.

Total-dominating set: A set $C \subseteq V$ such that $N(v) \cap C \neq \emptyset$ for all $v \in V$.

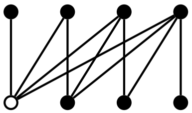
Full-separating set: if $\big(N[u] \triangle N[v]\big) \cap C = \big(N(u) \triangle N(v)\big) \cap C \neq \emptyset$ for all $u, v \in V$.

Theorem (C. and Wagler, 2024)

If a graph G admits an FTD-code, then $\gamma^{\mathrm{FTD}}(G) - 1 \leq \gamma^{\mathrm{FD}}(G) \leq \gamma^{\mathrm{FTD}}(G)$.



(a) $\gamma^{\mathrm{FTD}}(G) = n$



(b) $\gamma^{\mathrm{FD}}(G) = n - 1$

FD-Code

Input: (G, k): A graph G and a positive integer k.

Question: Does there exist an FD-code C of G such that $|C| \leq k$?

FTD-Code

Input: (G, k): A graph G and a positive integer k.

Question: Does there exist an FTD-code C of G such that $|C| \leq k$?

FD = FTD − 1

Input: A graph G and an integer k.

Question: Is $\gamma^{\text{FTD}}(G) = k$ and $\gamma^{\text{FD}}(G) = k - 1$?

FD-Code **NP-hard!**

Input: (G, k): A graph G and a positive integer k.

Question: Does there exist an FD-code C of G such that $|C| \leq k$?

FTD-Code **NP-hard!**

Input: (G, k): A graph G and a positive integer k.

Question: Does there exist an FTD-code C of G such that $|C| \leq k$?

FD = FTD − 1 **NP-hard!**

Input: A graph G and an integer k.

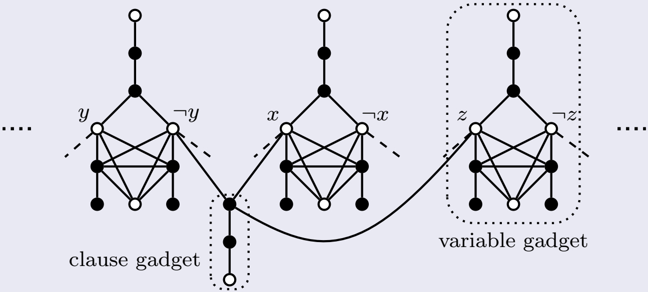
Question: Is $\gamma^{\text{FTD}}(G) = k$ and $\gamma^{\text{FD}}(G) = k - 1$?

Theorem (C. and Wagler, 2024)

FTD-Code *is NP-complete.*

Proof sketch.

Reduction from 3-SAT with formula ψ on n variables and m clauses.
E.g. $\psi = (x \vee \neg y \vee z) \wedge (\neg x \vee \neg z \vee w) \wedge (\neg y \vee z \vee \neg w)$.



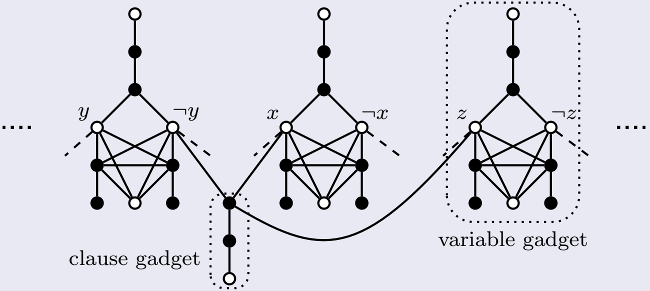
Theorem (C. and Wagler, 2024)

FTD-CODE *is NP-complete.*

Proof sketch.

Reduction from 3-SAT with formula ψ on n variables and m clauses.
E.g. $\psi = (x \vee \neg y \vee z) \wedge (\neg x \vee \neg z \vee w) \wedge (\neg y \vee z \vee \neg w)$.

ψ satisfiable $\iff$ $(G^{\psi}, k = 7n + 2m)$ is YES-instance of FTD-CODE



Theorem (C. and Wagler, 2024)

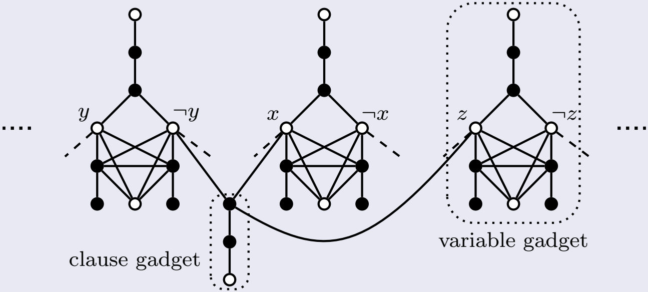
FTD-Code *is NP-complete.*

Proof sketch.

Reduction from 3-SAT with formula ψ on n variables and m clauses.
E.g. $\psi = (x \vee \neg y \vee z) \wedge (\neg x \vee \neg z \vee w) \wedge (\neg y \vee z \vee \neg w)$.

ψ satisfiable $\iff$ $(G^{\psi}, k = 7n + 2m)$ is YES-instance of FTD-Code

ψ satisfiable $\Longrightarrow$ $\exists C$ such that $|C| = \gamma^{\text{FTD}}(G^{\psi}) = k$



Theorem (C. and Wagler, 2024)

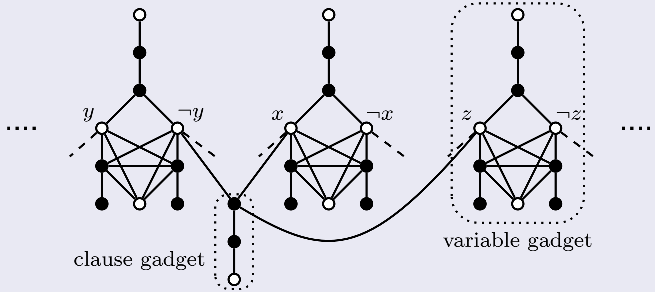
FTD-CODE *is NP-complete.*

Proof sketch.

Reduction from 3-SAT with formula ψ on n variables and m clauses.
E.g. $\psi = (x \vee \neg y \vee z) \wedge (\neg x \vee \neg z \vee w) \wedge (\neg y \vee z \vee \neg w)$.

ψ satisfiable $\iff$ $(G^{\psi}, k = 7n + 2m)$ is YES-instance of FTD-CODE

ψ satisfiable $\Longleftarrow$ $\exists C$ such that $|C| = \gamma^{\mathrm{FTD}}(G^{\psi}) = k$



Theorem (C. and Wagler, 2024)

FTD-CODE *is NP-complete.*

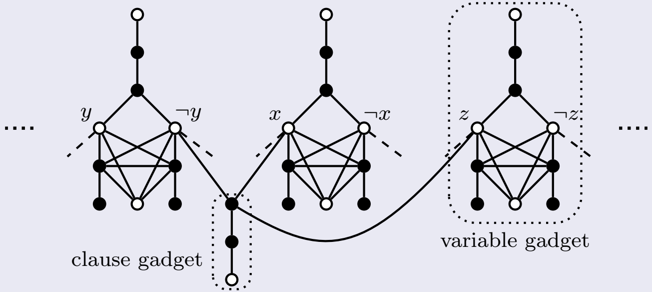
Proof sketch.

Reduction from 3-SAT with formula ψ on n variables and m clauses.
E.g. $\psi = (x \vee \neg y \vee z) \wedge (\neg x \vee \neg z \vee w) \wedge (\neg y \vee z \vee \neg w)$.

ψ satisfiable $\iff$ $(G^{\psi}, k = 7n + 2m)$ is YES-instance of FTD-CODE

ψ satisfiable $\Longleftarrow$ $\exists C$ such that $|C| = \gamma^{\mathrm{FTD}}(G^{\psi}) = k$

ψ satisfiable $\iff$ $(G^{\psi}, 7n + 2m - 1)$ is YES-instance of FD-CODE



Decision version of finding the minimum X-code in a graph:

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

X-CODE is NP-hard for all X $\in$ CODES!

Decision version of finding the minimum X-code in a graph:

X-CODE **Input:** (G, k): A graph G and a positive integer k. **Question:** Does there exist an X-code C of G such that $\|C\| \leq k$?

X-CODE is NP-hard for all X $\in$ CODES!

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 $f(k) \cdot n^{O(1)}$? e.g. $f(k) = 2^k$, 2^{k^2}, 2^{2^k} ...

Decision version of finding the minimum X-code in a graph:

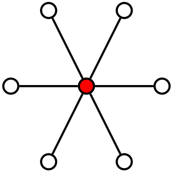
X-Code
Input: (G, k): A graph G and a positive integer k.
Question: Does there exist an X-code C of G such that $|C| \leq k$?

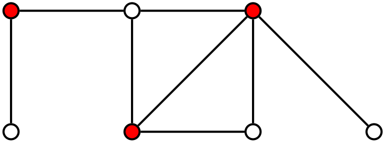
X-Code is NP-hard for all X $\in$ Codes!

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 $f(k) \cdot n^{O(1)}$? e.g. $f(k) = 2^k$, 2^{k^2}, 2^{2^k} ...

Note: X-Code is FPT when parameterized by solution size k.
Reason: $|V(G)| = O(2^k)$. Thus brute force gives $2^{O(k^2)} \cdot n^{O(1)}$ runtime.

Vertex cover: A set $S \subset V$ such that $V \setminus S$ is an independent set.
Vertex cover number: $\mathsf{vc} = \min\{|S| : S \text{ is a vertex cover of } G\}$





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

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

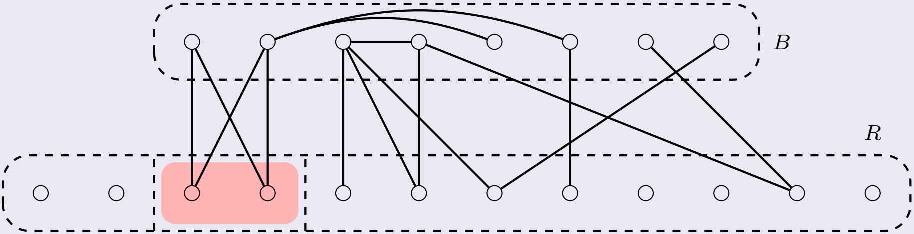
Algorithm (by dynamic programming):

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

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

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

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



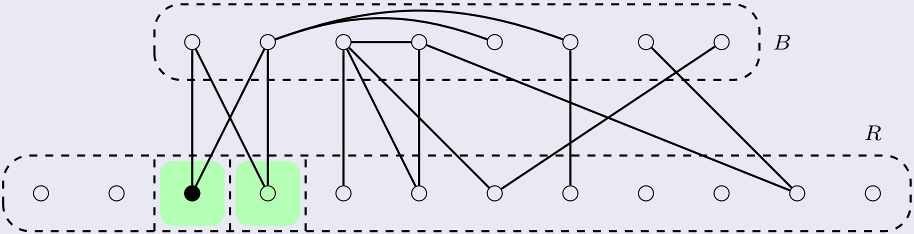
Algorithm (by dynamic programming):

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

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

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

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



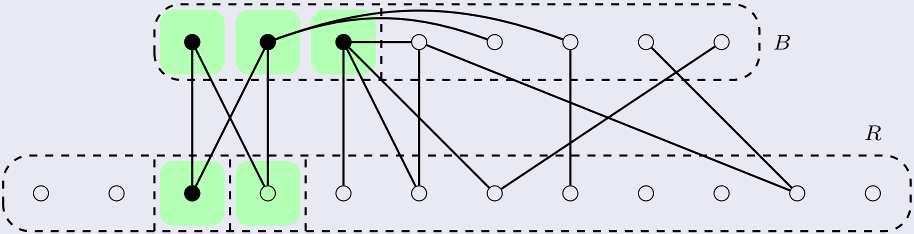
Algorithm (by dynamic programming):

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

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

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

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



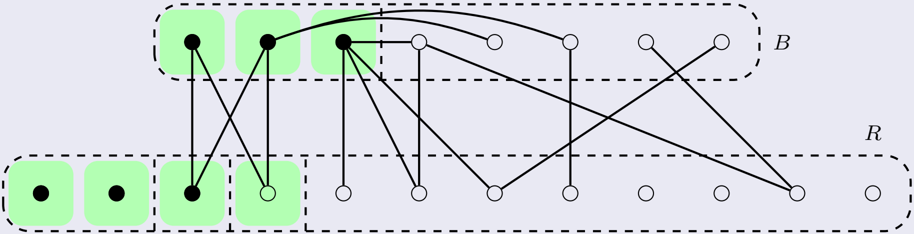
Algorithm (by dynamic programming):

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

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

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

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



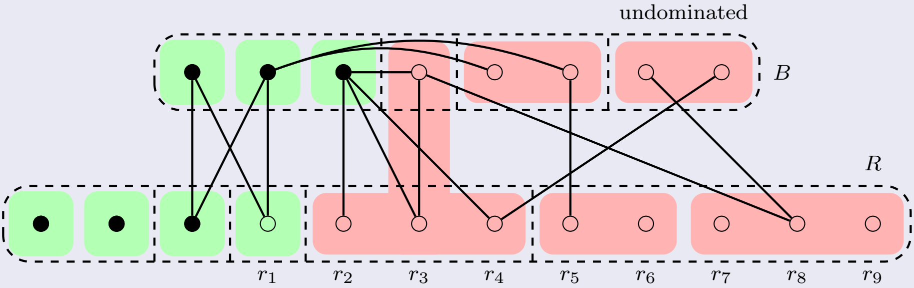
Algorithm (by dynamic programming):

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

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

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

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



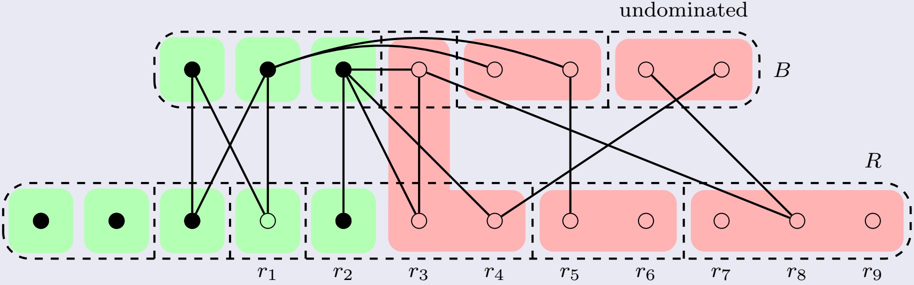
Algorithm (by dynamic programming):

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

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

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

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



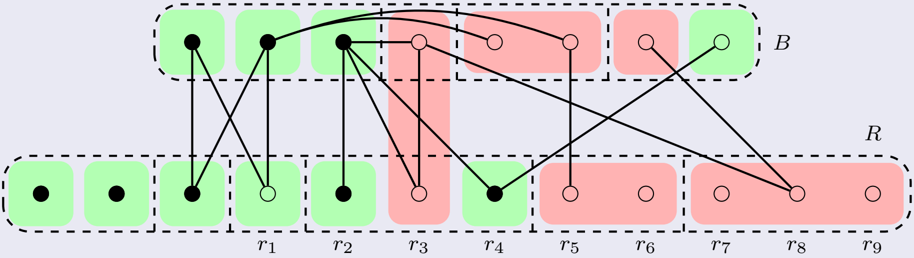
Algorithm (by dynamic programming):

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

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

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

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



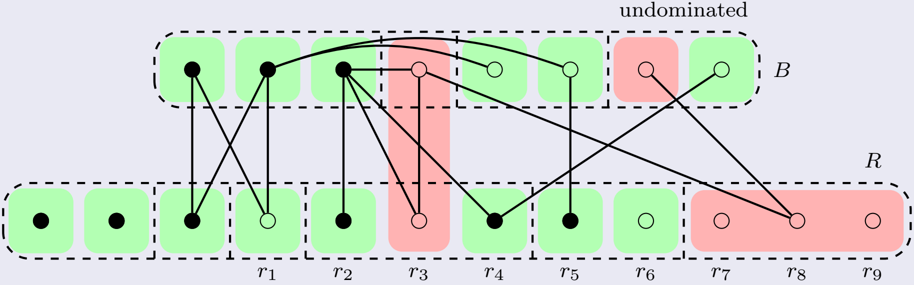
Algorithm (by dynamic programming):

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

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

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

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



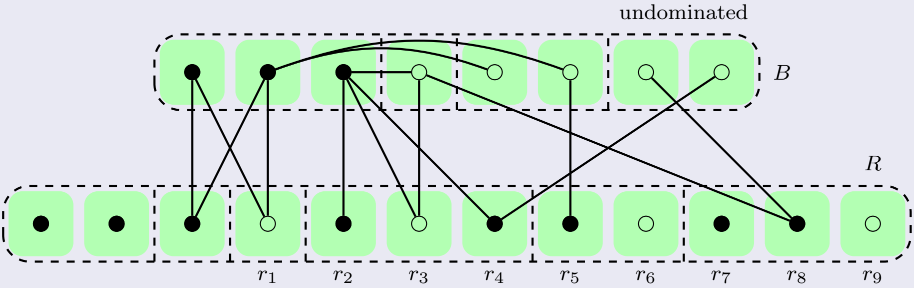
Algorithm (by dynamic programming):

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

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

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

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



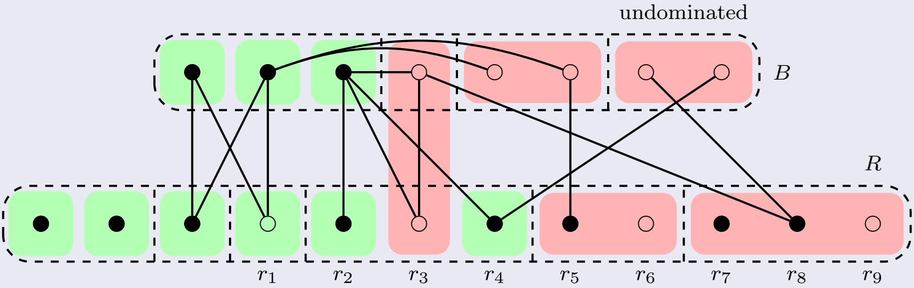
Algorithm (by dynamic programming):

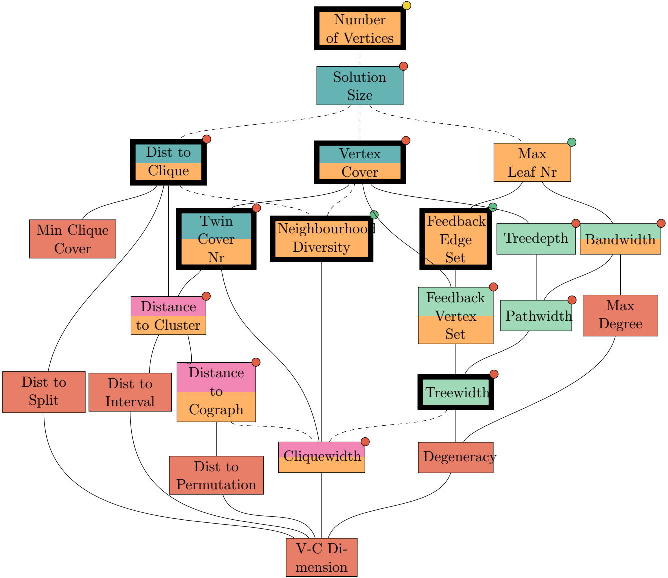
- Find a minimum vertex cover in time $1.2528^{\mathsf{vc}} \cdot n^{O(1)}$ [Harris & Narayanaswamy, STACS 2024].
- Build optimum solution by the **dynamic programming**:

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

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

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





single-exp FPT   slightly super-exp FPT   double-exp FPT   FPT   para-NP-h.

linear kernel   (tight) quadratic kernel   no polynomial kernel

Conclusion

Summary...

- Introduction of new codes: **OD**, **FD** and **FTD**.
- Proving several **combinatorial conjectures** and **results on bounds** of all eight code numbers of graphs.
- Such results have been proven on several graph classes like **subcubic**, **block**, **split**, **cobipartite**, **trees**, **triangle-free** etc.
- **NP-hardness** related results for the new codes introduced.
- **FPT-algorithms** for LD-Code with respect to several graph parameters.
- **Tight lower bounds** for running times of algorithms of LD-Code under well-accepted hardness hypothesis.

Some questions...

Question (Conjecture: Garijo, González & Márques, 2014)

Can the n-half conjecture ($\gamma^{\mathrm{LD}}(G) \leq \frac{n}{2}$) *be proven in general?*

Question (Conjecture: Foucaud & Henning, 2016)

Can the n-two-thirds conjecture ($\gamma^{\mathrm{LTD}}(G) \leq \frac{2}{3}n$) *be proven in general?*

Question

What is the characterization of (twin-free) (sub)cubic graphs for which the above conjectures are tight?

Some questions...

Question

Can the $\text{FD} = \text{FTD} - 1$ *problem be polynomial-time solvable on some graph classes? For example, for trees?*

Question

Can the $\text{OD} = \text{OTD} - 1$ *problem be polynomial-time solvable on some graph classes?*

Question

FPT-algorithms for other codes (especially, the newer ones) in terms of graph parameters like vertex cover number, treewidth, etc?

A code must intersect the following sets...

Sep	L-Sep		C-Sep		O-Sep		F-Sep	
Code	LD	LTD	CD	CTD	OD	OTD	FD	FTD
adj	$N(u) \triangle N(v)$		$N[u] \triangle N[v]$		$N(u) \triangle N(v)$		$N[u] \triangle N[v]$	
non-adj	$N[u] \triangle N[v]$						$N(u) \triangle N(v)$	
D/TD	$N[u]$	$N(u)$	$N[u]$	$N(u)$	$N[u]$	$N(u)$	$N[u]$	$N(u)$

A code must intersect the following sets...

<table>
<tr><td>Sep</td><td colspan="2">L-Sep</td><td colspan="2">C-Sep</td><td colspan="2">O-Sep</td><td colspan="2">F-Sep</td></tr>
<tr><td>Code</td><td>LLD</td><td>LLTD</td><td>CD</td><td>CTD</td><td>OD</td><td>OTD</td><td>FD</td><td>FTD</td></tr>
<tr><td>adj</td><td colspan="2">$N(u) \triangle N(v)$</td><td colspan="2" rowspan="2">$N[u] \triangle N[v]$</td><td colspan="2" rowspan="2">$N(u) \triangle N(v)$</td><td colspan="2">$N[u] \triangle N[v]$</td></tr>
<tr><td>non-adj</td><td colspan="2">V</td><td colspan="2">$N(u) \triangle N(v)$</td></tr>
<tr><td>D/TD</td><td>$N[u]$</td><td>$N(u)$</td><td>$N[u]$</td><td>$N(u)$</td><td>$N[u]$</td><td>$N(u)$</td><td>$N[u]$</td><td>$N(u)$</td></tr>
</table>

subset	**Abbvr.**	**dist-1 separators**	**dist-2+ separators**	**nbhd**
1	V. subset	$\{V(G)\}$	$\{V(G)\}$	$\{V(G)\}$
2	D-set	$\{V(G)\}$	$\{V(G)\}$	$N[u]$
3	TD-set	$\{V(G)\}$	$\{V(G)\}$	$N(u)$
4	CS-set	$N[u] \triangle N[v]$, u, v adj	$N[u] \triangle N[v]$, u, v non-adj	$\{V(G)\}$
5	OS-set	$N(u) \triangle N(v)$, u, v adj	$N(u) \triangle N(v)$, u, v non-adj	$\{V(G)\}$
6	LS-set	$N(u) \triangle N(v)$, u, v adj	$N[u] \triangle N[v]$, u, v non-adj	$\{V(G)\}$
7	FS-set	$N[u] \triangle N[v]$, u, v adj	$N(u) \triangle N(v)$, u, v non-adj	$\{V(G)\}$
8	LCS-set	$N[u] \triangle N[v]$, u, v adj	$\{V(G)\}$	$\{V(G)\}$
9	LOS-set	$N(u) \triangle N(v)$, u, v adj	$\{V(G)\}$	$\{V(G)\}$
10	NLCS-set	$\{V(G)\}$	$N[u] \triangle N[v]$, u, v non-adj	$\{V(G)\}$
11	NLOS-set	$\{V(G)\}$	$N(u) \triangle N(v)$, u, v non-adj	$\{V(G)\}$
12	CD-code	$N[u] \triangle N[v]$, u, v adj	$N[u] \triangle N[v]$, u, v non-adj	$N(u)$
13	OD-code	$N(u) \triangle N(v)$, u, v adj	$N(u) \triangle N(v)$, u, v non-adj	$N[u]$
14	LD-code	$N(u) \triangle N(v)$, u, v adj	$N[u] \triangle N[v]$, u, v non-adj	$N[u]$
15	FD-code	$N[u] \triangle N[v]$, u, v adj	$N(u) \triangle N(v)$, u, v non-adj	$N[u]$

	Abbvr.	dist-1 separators	dist-2+ separators	nbhd
16	CTD-code	$N[u] \triangle N[v]$, u, v adj	$N[u] \triangle N[v]$, u, v non-adj	$N(u)$
17	OTD-code	$N(u) \triangle N(v)$, u, v adj	$N(u) \triangle N(v)$, u, v non-adj	$N(u)$
18	LTD-code	$N(u) \triangle N(v)$, u, v adj	$N[u] \triangle N[v]$, u, v non-adj	$N(u)$
19	FTD-code	$N[u] \triangle N[v]$, u, v adj	$N(u) \triangle N(v)$, u, v non-adj	$N(u)$
20	LCD-code	$N[u] \triangle N[v]$, u, v adj	$\{V(G)\}$	$N[u]$
21	LLD-code	$N(u) \triangle N(v)$, u, v adj	$\{V(G)\}$	$N[u]$
22	LCTD-code	$N[u] \triangle N[v]$, u, v adj	$\{V(G)\}$	$N(u)$
23	LLTD-code	$N(u) \triangle N(v)$, u, v adj	$\{V(G)\}$	$N(u)$
24	NLCD-code	$\{V(G)\}$	$N[u] \triangle N[v]$, u, v non-adj	$N[u]$
25	NLOD-code	$\{V(G)\}$	$N(u) \triangle N(v)$, u, v non-adj	$N[u]$
26	NLCTD-code	$\{V(G)\}$	$N[u] \triangle N[v]$, u, v non-adj	$N(u)$
27	NLOTD-code	$\{V(G)\}$	$N(u) \triangle N(v)$, u, v non-adj	$N(u)$

Thank You!