

# DOMINATING SETS RECONFIGURATION

UNDER TOKEN SLIDING

---

Marthe BONAMY, Paul DORBEC, Paul OUVRARD

July 16, 2018

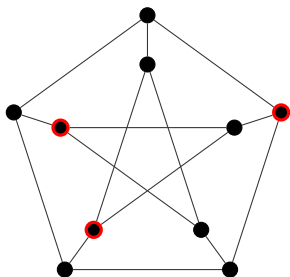


université  
de **BORDEAUX**

# Definition: dominating set

## Dominating set $D$

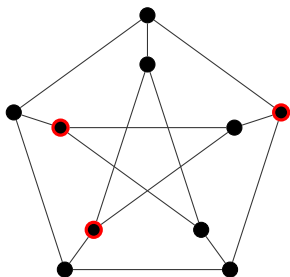
$D \subseteq V$  such that every vertex not in  $D$  has  $\geq 1$  neighbor in  $D$ .



# Definition: dominating set

## Dominating set $D$

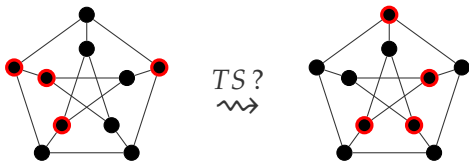
$D \subseteq V$  such that every vertex not in  $D$  has  $\geq 1$  neighbor in  $D$ .



$\gamma(G)$ : minimum size of a dominating set of  $G$ .

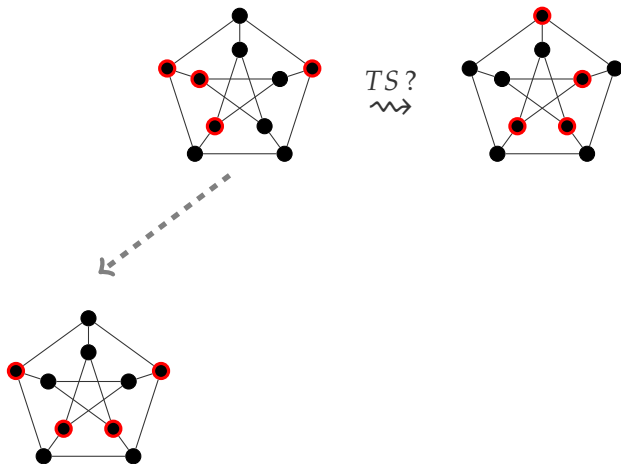
# Reconfiguration in graphs

**Rule:** each intermediate solution must be a dominating set, we are only allowed to slide a token along an edge!



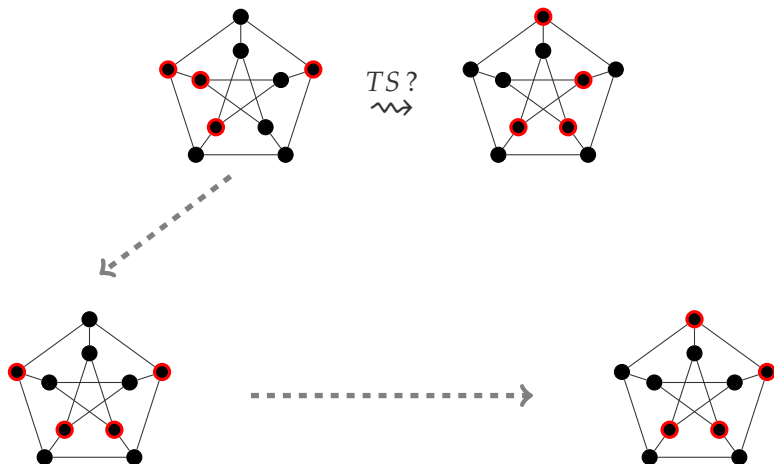
# Reconfiguration in graphs

**Rule:** each intermediate solution must be a dominating set, we are only allowed to slide a token along an edge!



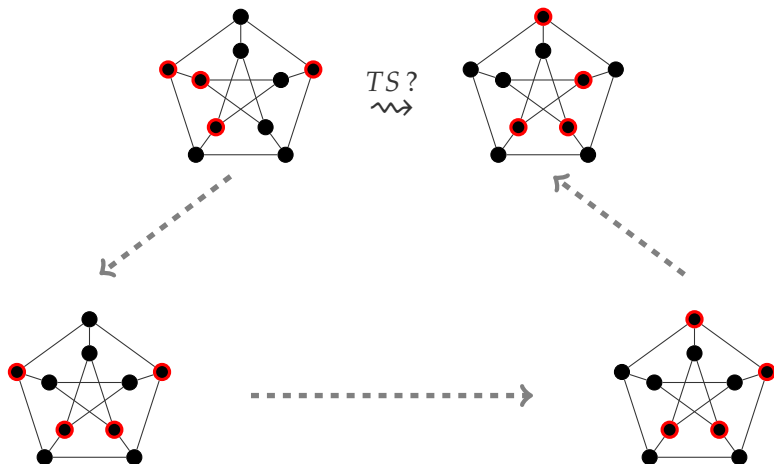
# Reconfiguration in graphs

**Rule:** each intermediate solution must be a dominating set, we are only allowed to slide a token along an edge!



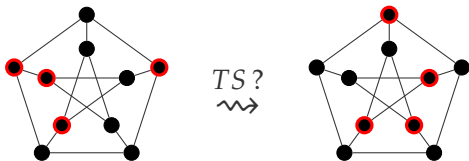
# Reconfiguration in graphs

**Rule:** each intermediate solution must be a dominating set, we are only allowed to slide a token along an edge!



# Reconfiguration in graphs

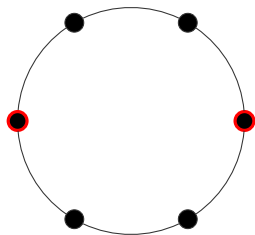
**Rule:** each intermediate solution must be a dominating set, we are only allowed to slide a token along an edge!



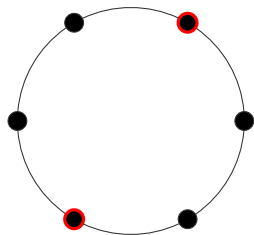


# Exemple pour $C_6$

⚠ not always possible!



*TS?*



## Dominating Set Reconfiguration

**Input:** a graph  $G$ , two dominating sets  $A$  and  $B$  of  $G$

**We want:** to transform step-by-step  $A$  into  $B$ ; each intermediate solution must be a dominating set.

## Elementary operations

- *token addition and removal* (TAR)
- *token jumping* (TJ)
- *token sliding* (TS)

## Notations

Let  $G$  be a graph and  $A, B$  be two dominating sets of size  $k$  of  $G$ .

- $A \overset{TAR}{\rightsquigarrow} B$ : we can reconfigure  $A$  into  $B$  with the TAR model and each solution is of size at most  $k + 1$ .
- $A \overset{TJ}{\rightsquigarrow} B$ : we can reconfigure  $A$  into  $B$  with the TJ model.

## Lemma

Let  $G$  be a graph and  $A$  and  $B$  be two dominating sets of  $G$  of size  $k$ . We have  $A \overset{TAR}{\rightsquigarrow} B$  iff  $A \overset{TJ}{\rightsquigarrow} B$ .

**Proof:** adapted from <sup>1</sup> (Theorem 1)

- $TJ \Rightarrow TAR$ : easy. Replace each move  $u \rightsquigarrow v$  by:
  - first add  $v$
  - remove  $u$
  - $\Rightarrow$  each solution has size at most  $k + 1$
- We double the length of the sequence

---

<sup>1</sup>M. Kaminski, P. Medvedev, and M. Milanic. Complexity of independent set reconfigurability problems. *Theoretical Computer Science*, 439:9 – 15, 2012.

- $TAR \Rightarrow TJ$ 
  - TAR-sequence of length  $2n$  with configurations of size  $k$  or  $k + 1$ 
    - Alternation of addition of  $v$  followed by a removal of  $u$
    - Replace by  $u \xrightarrow{TJ} v$
    - $\Rightarrow$  get a TJ-sequence of length  $n$
  - TAR-sequence of length  $2n$  with configurations of any size
    - There exists a subsequence which consists in the removal of a vertex  $u$  followed by the addition of a vertex  $v$
    - If  $u = v \rightarrow$  remove the subsequence
    - Otherwise, switch the order
    - (Possibly reiterate the process)

## Theorem [Haddadan *et al.*]

Reconfiguring dominating sets under  $\text{TAR}(k + 1)$  is PSPACE-complete.

**The problem is in PSPACE<sup>2</sup> (Theorem 1).**

## Lemma

Reconfiguring dominating sets under  $\text{TS}(k)$  is PSPACE-complete in split graphs.

---

<sup>1</sup>Ito, T., Demaine, E.D., Harvey, N.J.A., Papadimitriou, C.H., Sideri, M., Uehara, R., Uno, Y.: On the complexity of reconfiguration problems. *Theoretical Computer Science* 412, pp. 1054–1065 (2011)

# Complexity proof

**Proof:** reduction from TJ in some graph  $G \rightarrow$  TS in the corresponding split graph.

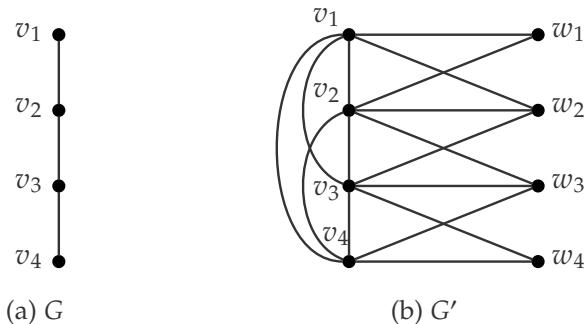
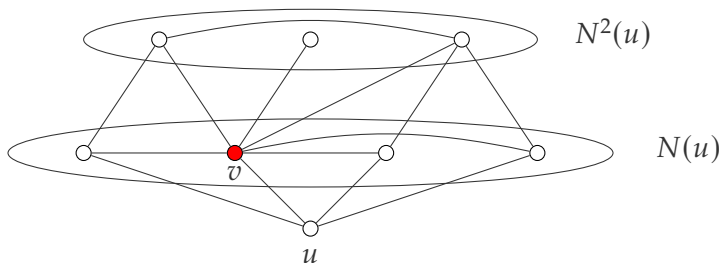


Figure: Graph  $G$  and the corresponding split graph  $G'$

# Définitions

- $G = (V, E)$  avec  $V = \{v_1, v_2, \dots, v_n\}$
- $G_i = G[\{v_i, v_{i+1}, \dots, v_n\}]$
- $v$  *maximum neighbor* de  $u$ :
  - $v \in N[u]$
  - $v$  est adjacent à tous les sommets à distance  $\leq 2$  de  $u$
  - $\Rightarrow \forall w \in N[u] : N[w] \subseteq N[v]$

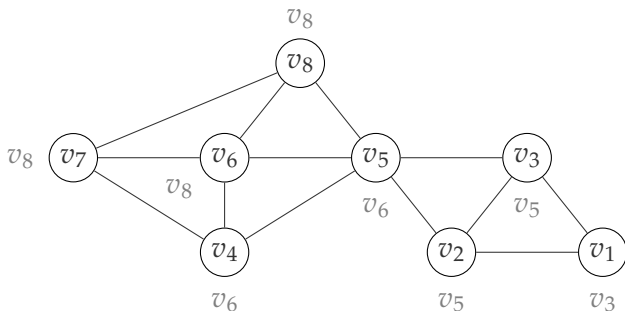




# Définitions

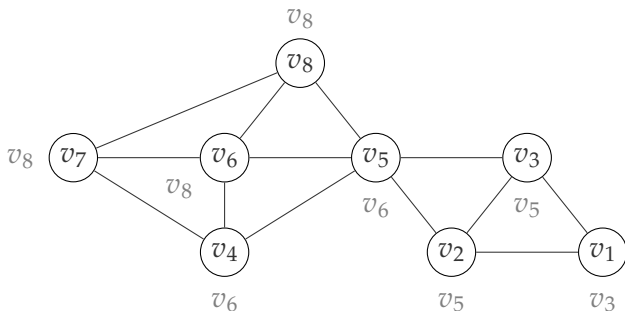
- *maximum neighborhood ordering (mno)*: ordre sur les sommets de telle sorte que  $v_i$  possède un *maximum neighbor* dans  $G_i$

Un graphe *dually chordal* est un graphe possédant un *maximum neighborhood ordering*.



## Lemme [Dorbec, Kosmrlj, Renault]

Soit  $G$  un graphe dually chordal. Il existe un *maximum neighborhood ordering*  $v_1, v_2, \dots, v_n$  de  $G$  tel que si le seul *maximum neighbor* de  $v_i$  dans  $G_i$  est lui-même, alors  $v_i$  est un sommet isolé dans  $G_i$ . On appelle un tel *mno* un *mno propre*.



Soit  $G$  un graphe dually chordal et  $A$  et  $B$  deux ensembles dominants minimum de  $G$ . On veut reconfigurer  $A$  en  $B$ .

**Preuve (idée):**

- On distingue un ensemble dominant minimum  
→ ensemble dominant *canonique* que l'on note  $C$
- On montre que  $A$  et  $B$  peuvent se reconfigurer en  $C$
- Opération réversible  $\Rightarrow$  on peut reconfigurer  $A$  en  $B$

# Algorithme de calcul du dominant *canonique*

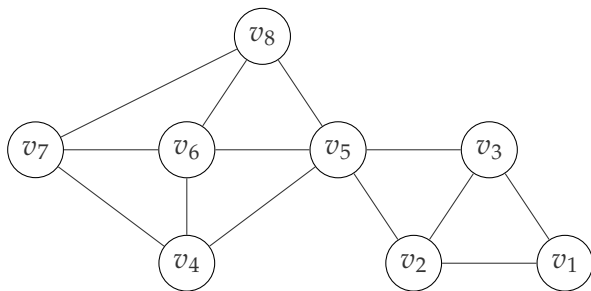
**Entrée:** un graphe dually chordal  $G$ , un *mno* propre

**Sortie:** le dominant canonique de  $G$

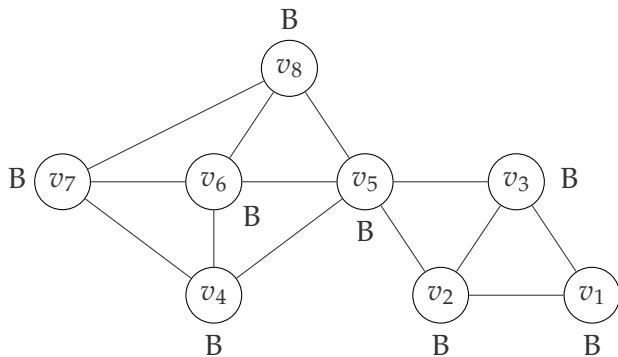
1. Étiqueter tous les sommets avec BOUNDED
2. Pour  $i$  allant de 1 à  $n$ 
  - Si  $v_i$  est étiqueté BOUNDED
    - Marquer le sommet  $v_i$
    - Étiqueter  $mn(v_i)$  avec REQUIRED
    - Pour chaque sommet  $v \in N(mn(v_i))$  qui n'est pas étiqueté REQUIRED, l'étiqueter avec FREE

⇒ Les sommets étiquetés avec REQUIRED forment un ensemble dominant minimum.

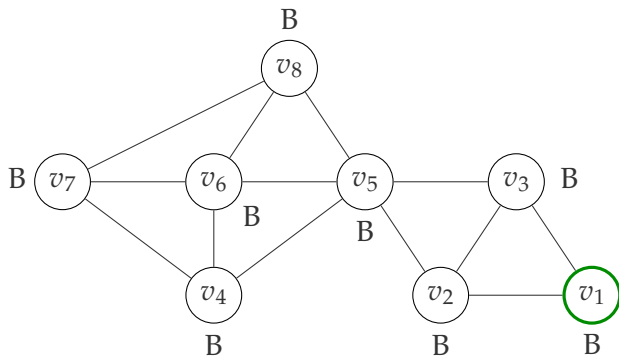
# Exemple



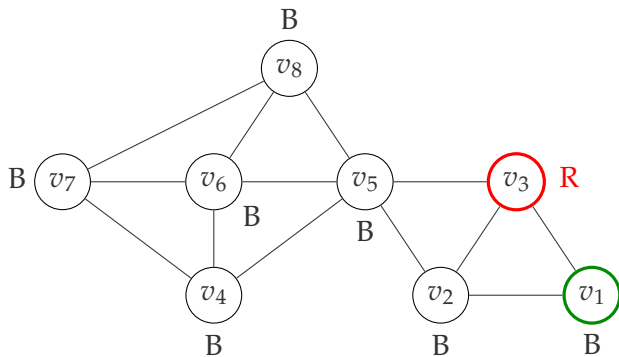
# Exemple



# Exemple

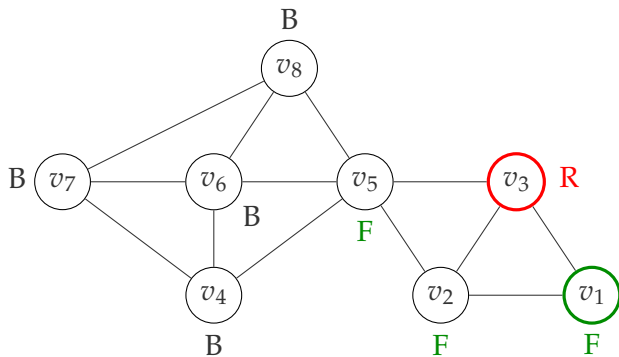


# Exemple

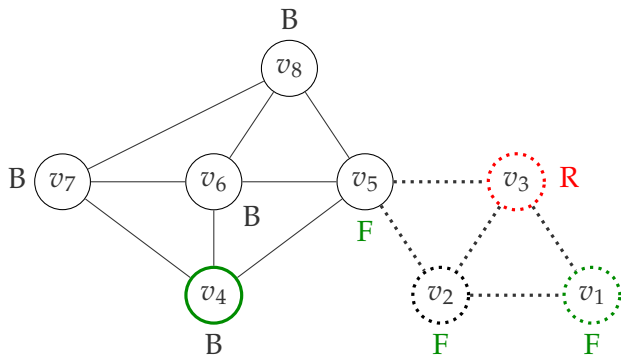




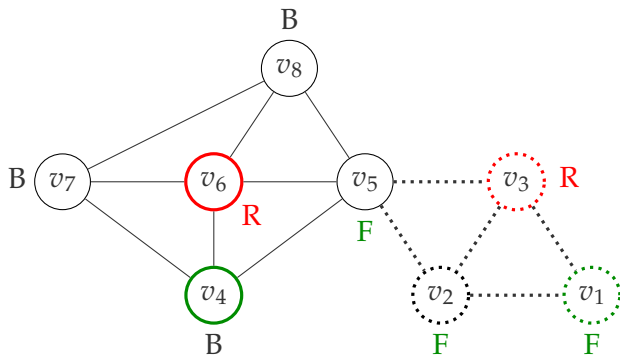
# Exemple



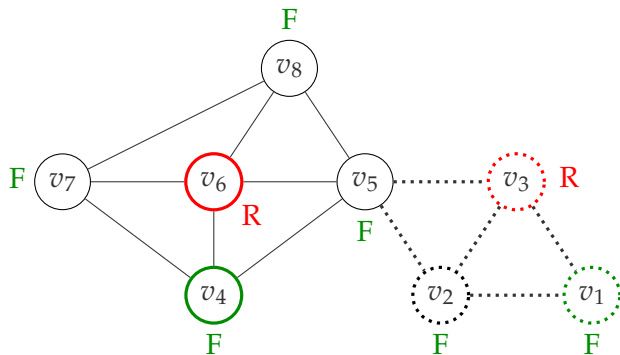
# Exemple



# Exemple



# Exemple



# Algorithme de reconfiguration

- $C = \{c_1, c_2, \dots, c_{\gamma(G)}\}$  le dominant canonique de  $G$
- $m_1, m_2, \dots, m_{\gamma(G)}$  les sommets marqués par l'algorithme précédent
- $D$ : dominant minimum de  $G$  que l'on souhaite reconfigurer en  $C$
- $x_i$  un sommet (quelconque) de  $N[m_i] \cap D$

## Preuve (idée):

- $i = 1$ :
  - $\forall w \in N[m_1], N[w] \subseteq N[c_1]$
  - En particulier:  $N[x_1] \subseteq N[c_1]$
  - $\Rightarrow (D \setminus \{x_1\}) \cup \{c_1\}$  est un dominant de  $G$
  - $(x_1, c_1) \in E(G)$

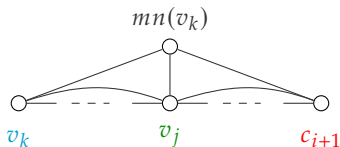
# Algorithme de reconfiguration

- $i \Rightarrow i + 1$
- Supposons que c'est vrai au rang  $i$  i.e.  
 $(D \setminus \{x_1, \dots, x_i\}) \cup \{c_1, \dots, c_i\}$  est un dominant de  $G$
- On note  $v_j = m_{i+1}$  et  $v_k = x_{i+1}$
- Nous distinguons deux cas:
  - $v_k \geq v_j$ :
    - tous les sommets avant  $v_j$  (dans l'ordre du  $mno$ ) sont déjà dominés
    - $N_j[x_{i+1}] \subseteq N_j[c_{i+1}]$
    - $\Rightarrow (D \setminus \{x_1, \dots, x_i, x_{i+1}\}) \cup \{c_1, \dots, c_i, c_{i+1}\}$  est également un dominant de  $G$
    - $\Rightarrow (x_{i+1}, c_{i+1}) \in E(G)$

# Algorithme de reconfiguration

○  $i \Rightarrow i + 1$

- Rappel:  $v_j = m_{i+1}$  et  $v_k = x_{i+1}$
- $v_k < v_j$ :
  - $v_k \notin \{c_1, c_2, \dots, c_i\}$  (car  $v_j$  est étiqueté BOUNDED)
  - tous les sommets avant  $v_j$  (toujours dans l'ordre du  $mno$ ) sont déjà dominés
  - Si  $(v_k, c_{i+1}) \in E(G)$ : fini
  - Sinon, la situation est la suivante:



- On peut glisser  $v_k$  vers  $c_{i+1}$  en deux étapes (via  $mn(v_k)$ ).

- Le problème est PSPACE-complet dans certaines classes de graphes...
  - Graphes scindés
  - Graphes planaires
  - Graphes de largeur arborescente bornée
  - Graphes bipartis
- ... mais polynomial en temps dans d'autres classes
  - Graphes d'intervalles
  - Graphes dually chordaux
  - Cographe



Une **question** intéressante possible:

- Existe-t-il une classe de graphes pour laquelle calculer un ensemble dominant minimum est difficile mais pour laquelle le problème de reconfiguration associé peut se résoudre en temps polynomial ?

**Merci de votre attention !**

$TS(k) \not\Rightarrow TS(k + 1)$

