

RECONFIGURATION D'ENSEMBLES DOMINANTS

PAR TOKEN SLIDING

Marthe BONAMY, Paul DORBEC, Paul OUVRARD

16 novembre 2017

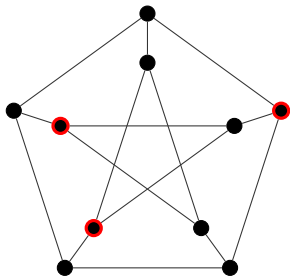
JGA 2017



université
de **BORDEAUX**

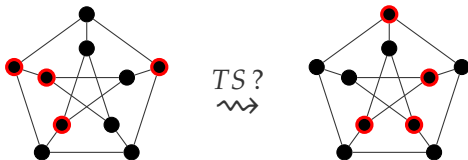
Définition

Sous-ensemble $D \subseteq V$ tel que chaque sommet qui n'est pas dans D est voisin d'un sommet de D .



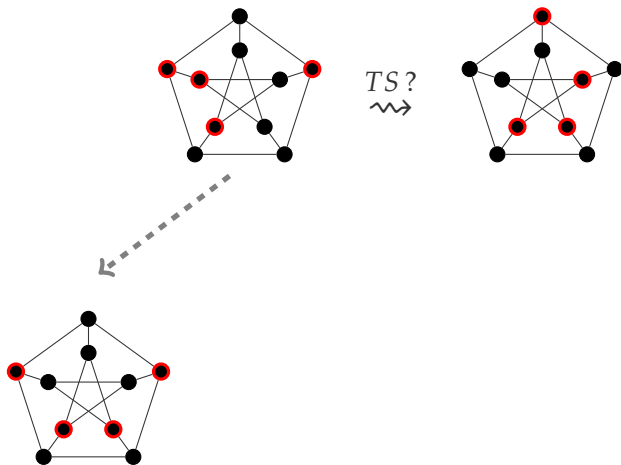
Reconfiguration dans les graphes

Règle : on glisse un sommet le long d'une arête et chaque solution intermédiaire est un dominant.



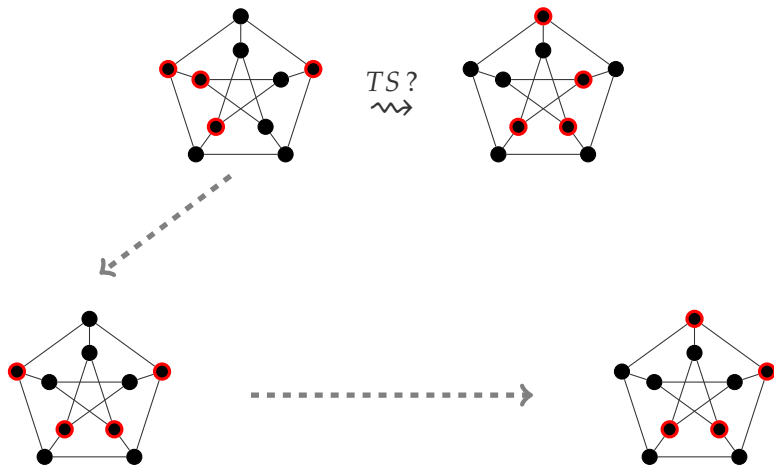
Reconfiguration dans les graphes

Règle : on glisse un sommet le long d'une arête et chaque solution intermédiaire est un dominant.



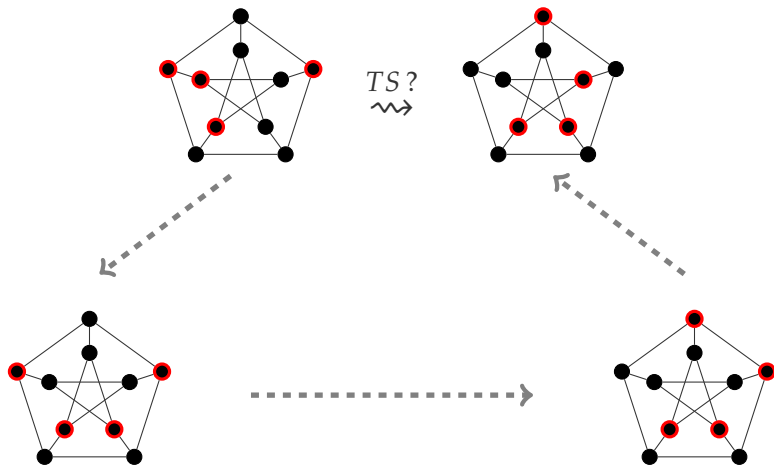
Reconfiguration dans les graphes

Règle : on glisse un sommet le long d'une arête et chaque solution intermédiaire est un dominant.



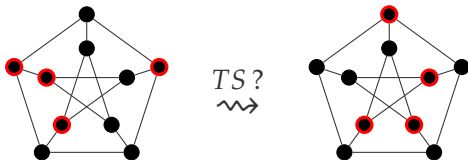
Reconfiguration dans les graphes

Règle : on glisse un sommet le long d'une arête et chaque solution intermédiaire est un dominant.



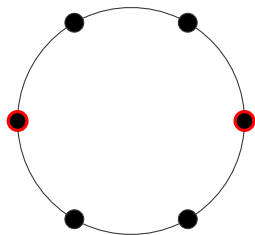
Reconfiguration dans les graphes

Règle : on glisse un sommet le long d'une arête et chaque solution intermédiaire est un dominant.

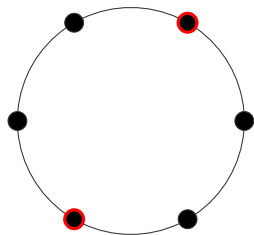


Exemple pour C_6

⚠ pas toujours possible !



TS?



Reconfiguration d'ensembles dominants

Entrée : un graphe G , deux ensembles dominants A et B de G
On veut : transformer étape par étape A en B ; chaque solution intermédiaire doit être un dominant de G .

Opérations élémentaires

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

Notations

Soit G un graphe et A, B deux ensembles dominants de taille k de G .

- $A \overset{TAR}{\rightsquigarrow} B$: on peut reconfigurer A en B avec le modèle TAR et chaque solution intermédiaire est de taille au plus $k + 1$.
- $A \overset{TJ}{\rightsquigarrow} B$: on peut reconfigurer A en B avec le modèle TJ.

Lemme

Soit G un graphe et A et B deux ensembles dominants de G de taille k . On a alors $A \overset{TAR}{\rightsquigarrow} B$ ssi $A \overset{TJ}{\rightsquigarrow} B$.

Preuve : adaptée de ¹ (Théorème 1)

- $TJ \Rightarrow TAR$: facile. Remplacer chaque étape $u \rightsquigarrow v$ par :
 - d'abord ajouter v ;
 - puis supprimer u .
 - \Rightarrow chaque solution a bien une taille au plus $k + 1$
- On double la longueur de la séquence

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

- $TAR \Rightarrow TJ$
 - TAR-séquence de longueur $2n$ avec des configurations de taille k ou $k + 1$
 - Alternance d'addition d'un sommet v suivi par la suppression d'un sommet u
 - Remplacer ces deux opérations par $u \overset{TJ}{\rightsquigarrow} v$
 - \Rightarrow on obtient une TJ-séquence de longueur n
 - TAR-séquence de longueur $2n$ avec des configurations de taille quelconques ($\leq k + 1$)
 - Il existe une sous-séquence qui consiste en la suppression d'un sommet u suivie de l'addition d'un sommet v
 - Si $u = v \rightarrow$ supprimer cette sous-séquence
 - Sinon, inverser l'ordre (ajouter v puis supprimer u)
 - (Éventuellement réitérer ce procédé)

Théorème [Haddadan *et al.*]

La reconfiguration d'ensemble dominants sous $TAR(k + 1)$ est PSPACE-complet.

Le problème est dans PSPACE² (Théorème 1).

Lemme

La reconfiguration d'ensembles dominants sous $TS(k)$ est PSPACE-complet dans les graphes scindés.

1. 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)

Preuve (idée) : réduction du problème sous TJ dans un graphe quelconque $G \rightarrow$ problème sous TS dans le graphe scindé correspondant.

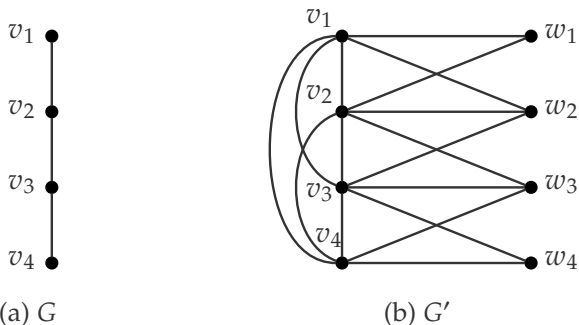
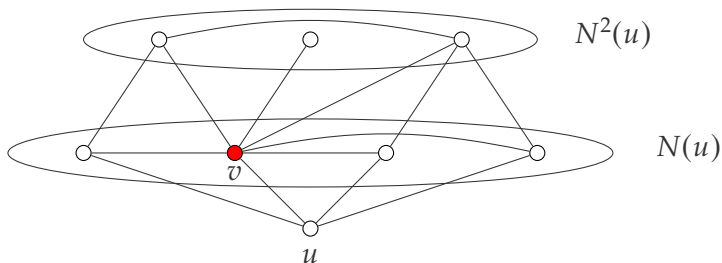


FIGURE – Graphe G et le graphe scindé G' correspondant

Définitions

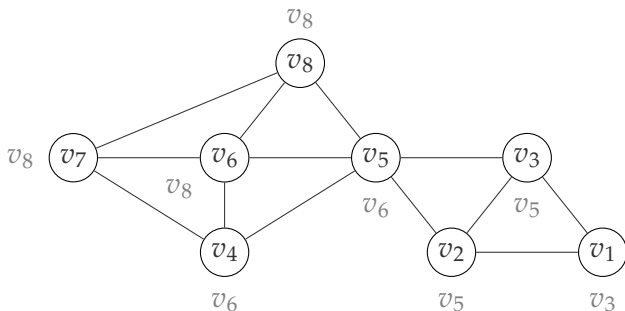
- $G = (V, E)$ avec $V = \{v_1, v_2, \dots, v_n\}$
- $G_i = G[\{v_i, v_{i+1}, \dots, v_n\}]$
- v *maximum neighbor* de u :
 - $v \in N[u]$
 - v est adjacent à tous les sommets à distance ≤ 2 de u
 - $\Rightarrow \forall w \in N[u] : N[w] \subseteq N[v]$



Définitions

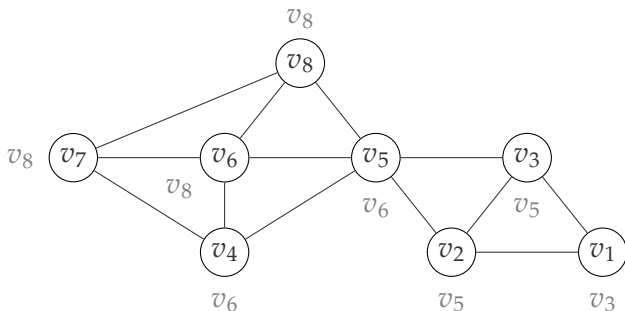
- *maximum neighborhood ordering (mno)* : ordre sur les sommets de telle sorte que v_i possède un *maximum neighbor* dans G_i

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*

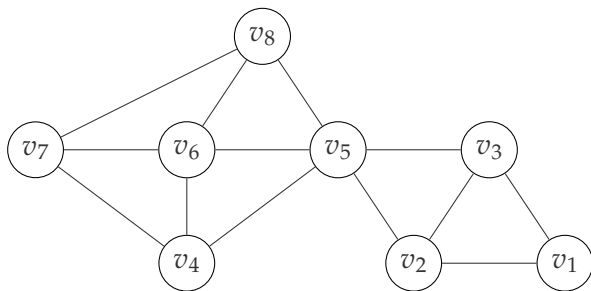
Entrée : un graphe dually chordal G , un *mno* propre

Sortie : le dominant canonique de G

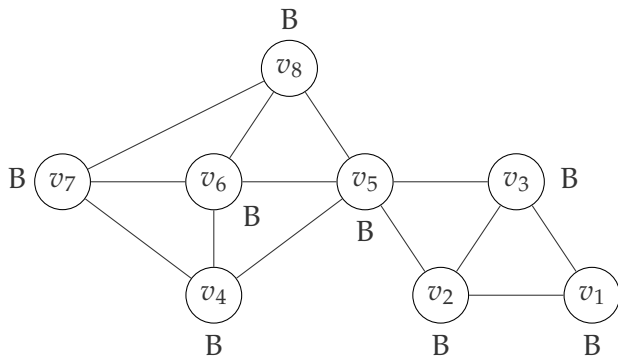
1. Étiqueter tous les sommets avec BOUNDED
2. Pour i allant de 1 à n
 - Si v_i est étiqueté BOUNDED
 - Marquer le sommet v_i
 - Étiqueter $mn(v_i)$ avec REQUIRED
 - Pour chaque sommet $v \in N(mn(v_i))$ qui n'est pas étiqueté REQUIRED, l'étiqueter avec FREE

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

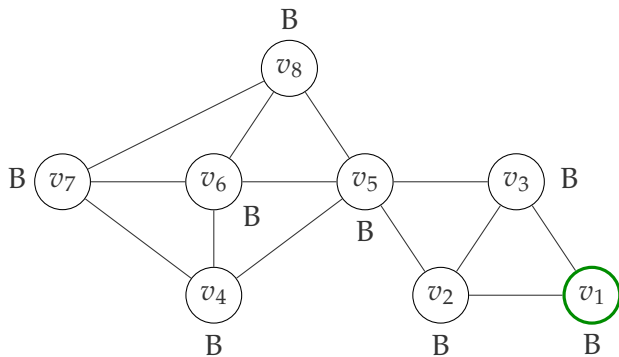
Exemple



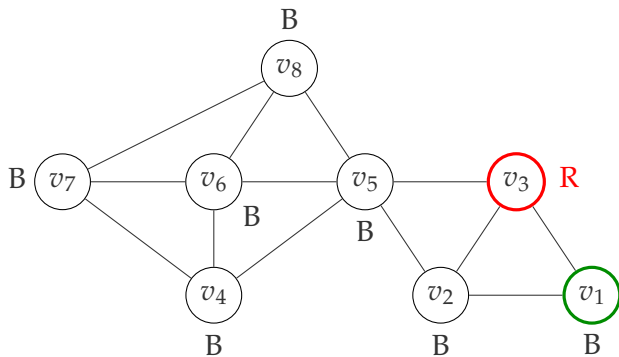
Exemple



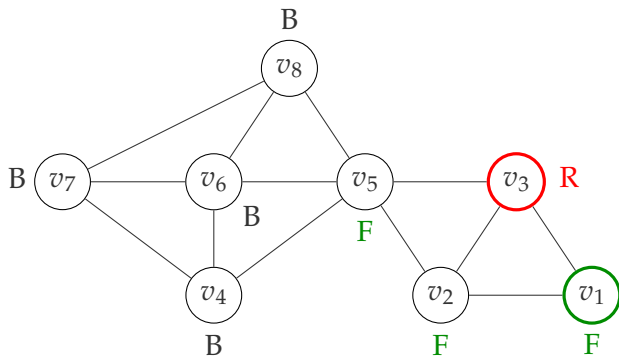
Exemple



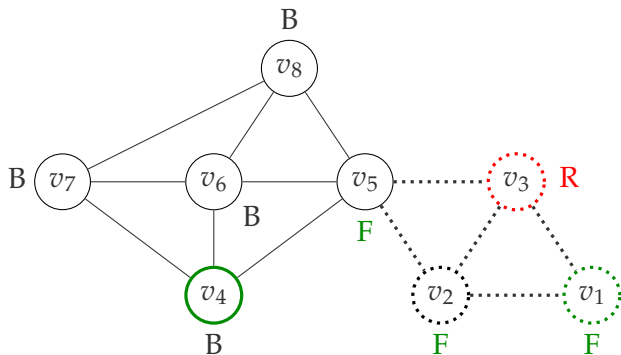
Exemple



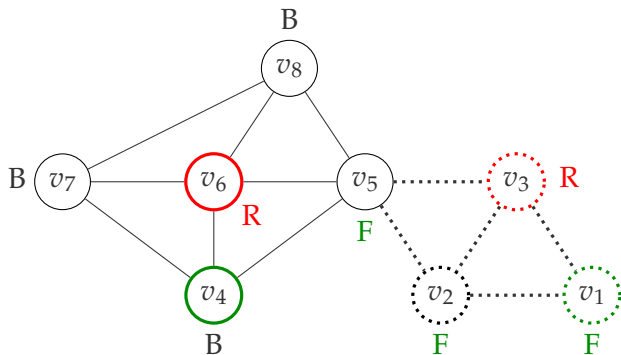
Exemple



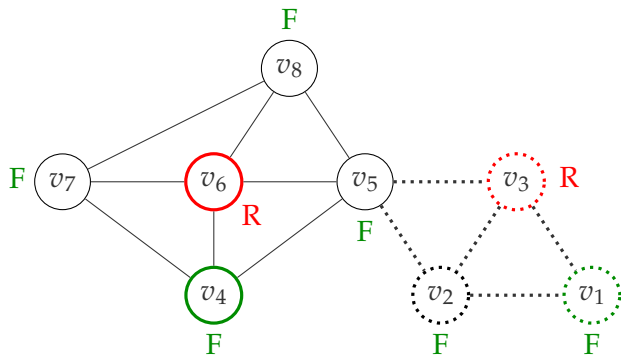
Exemple



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
 - $(x_1, c_1) \in E(G)$

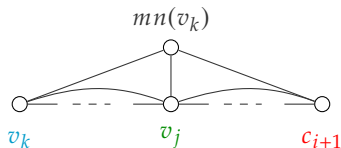
Algorithme de reconfiguration

- $i \Rightarrow i + 1$
- 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
 - Cographes

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 !

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

