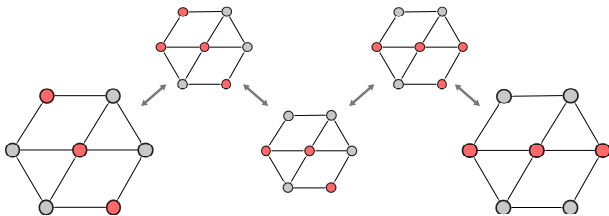


Optimisation décrementale de la reconfiguration d'ensembles dominants

Alexandre Blanché¹ Haruka Mizuta² Paul Ouvrard¹
Akira Suzuki²

¹LaBRI, Bordeaux

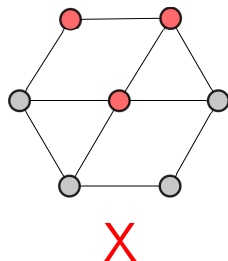
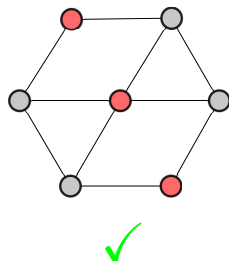
²Tohoku University, Sendai, Japon



16 Novembre 2018

Définition

Un **ensemble dominant** est un sous-ensemble de sommets dont le voisinage contient tous les sommets.



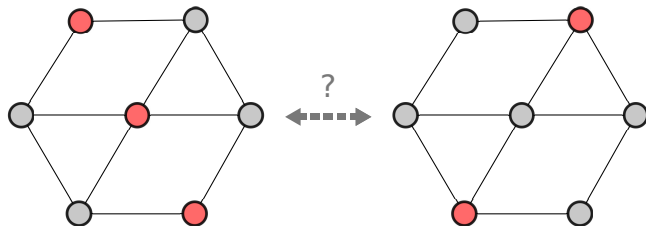
Problème de l'ensemble dominant

- **Instance** : Un graphe G , un entier s
- **Question** : Existe-t-il un dominant de G de taille au plus s ?

Ce problème est NP-complet.

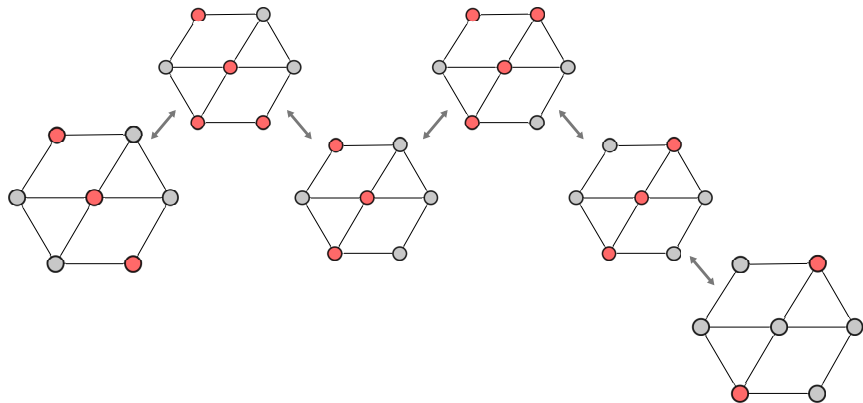
Reconfiguration d'ensembles dominants

Modèle : ajouts ou suppressions successives de sommets



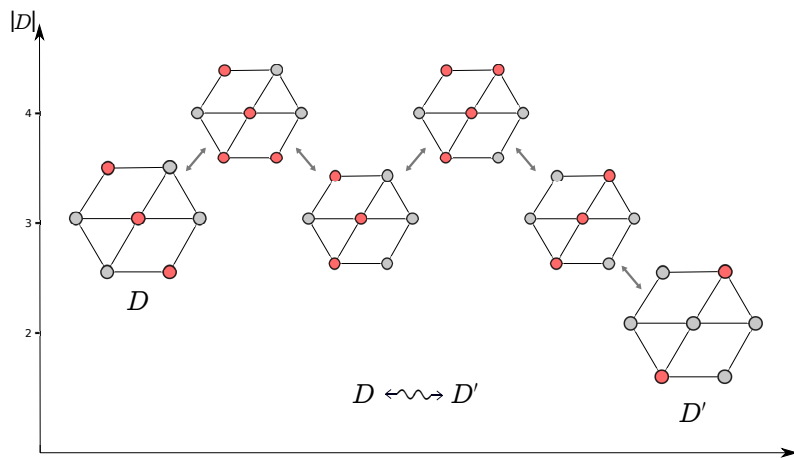
Reconfiguration d'ensembles dominants

Modèle : ajouts ou suppressions successives de sommets

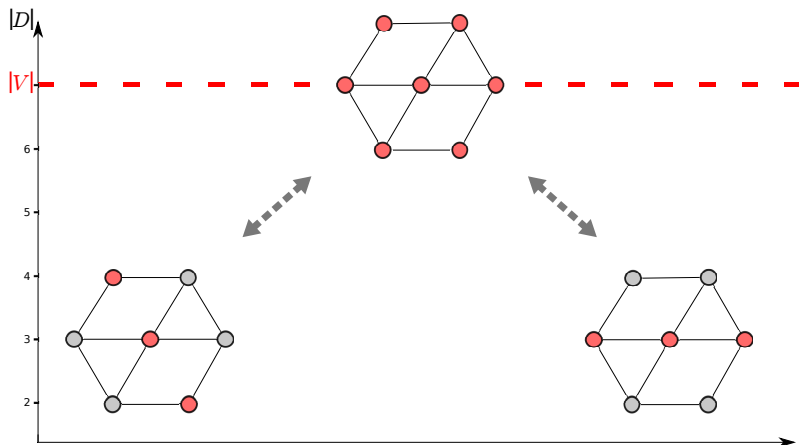


Reconfiguration d'ensembles dominants

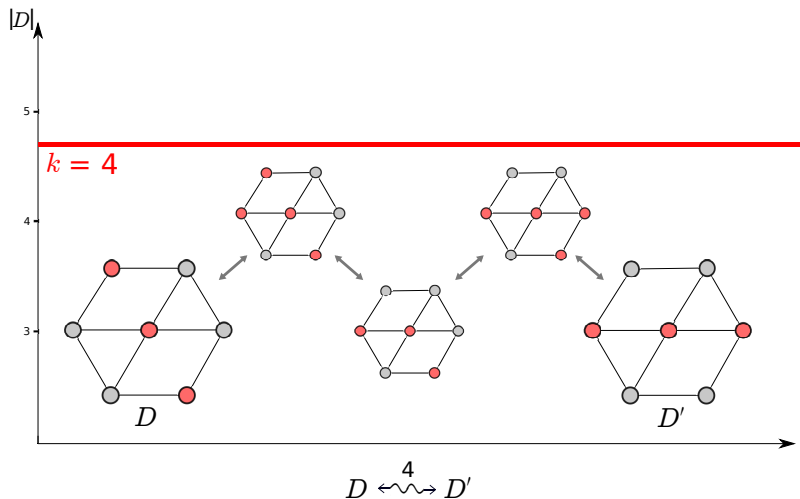
Modèle : ajouts ou suppressions successives de sommets



Sans borne sur la taille, tous les dominants sont joignables par ajouts ou suppressions successives :

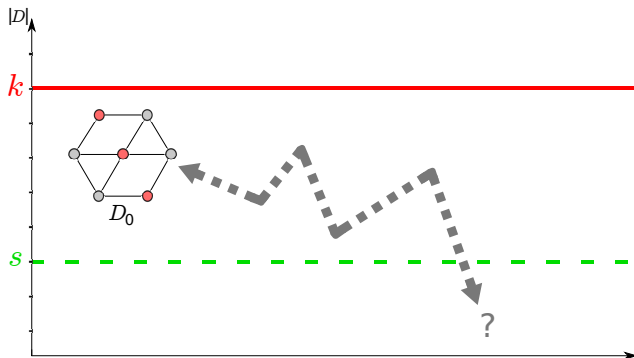


On limite donc la taille des dominants considérés par une borne k :



OPT-DSR (OPTimization variant of Dominating Set Reconfiguration)

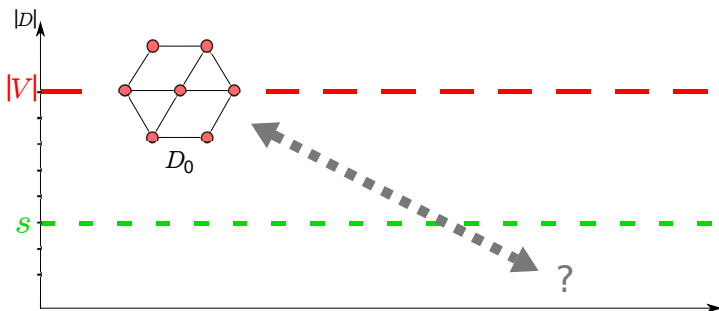
- **Instance** : Un graphe G , deux entiers k, s , un dominant D_0 de taille $|D_0| \leq k$.
- **Question** : Existe-t-il un dominant D_s de taille $|D_s| \leq s$, tel que $D_0 \overset{k}{\rightsquigarrow} D_s$?



Observation

OPT-DSR généralise le problème de l'ensemble dominant.

Un graphe $G = (V, E)$ a un dominant de taille $\leq s$
 \Leftrightarrow
l'instance $(G, k = |V|, s, D = V)$ est solution de OPT-DSR.



Observation

OPT-DSR généralise le problème de l'ensemble dominant.

Un graphe $G = (V, E)$ a un dominant de taille $\leq s$
 \Leftrightarrow
l'instance $(G, k = |V|, s, D = V)$ est solution de OPT-DSR.

Corollaire

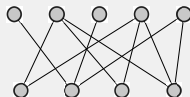
OPT-DSR est NP-dur.

$(NP \subseteq PSPACE \subseteq EXPTIME)$

Théorème [B., Mizuta, Ouvrard, Suzuki, (2018+)]

OPT-DSR est PSPACE-complet, y compris lorsque le graphe d'entrée :

- est un graphe biparti ;

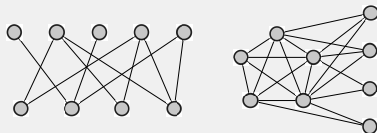


$(NP \subseteq PSPACE \subseteq EXPTIME)$

Théorème [B., Mizuta, Ouvrard, Suzuki, (2018+)]

OPT-DSR est PSPACE-complet, y compris lorsque le graphe d'entrée :

- est un graphe biparti ;
- est un graphe split ;

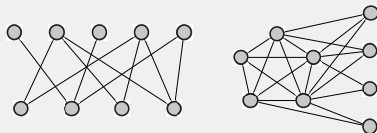


($NP \subseteq PSPACE \subseteq EXPTIME$)

Théorème [B., Mizuta, Ouvrard, Suzuki, (2018+)]

OPT-DSR est PSPACE-complet, y compris lorsque le graphe d'entrée :

- est un graphe biparti ;
- est un graphe split ;
- a une *pathwidth* bornée.

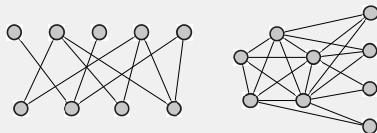


($NP \subseteq PSPACE \subseteq EXPTIME$)

Théorème [B., Mizuta, Ouvrard, Suzuki, (2018+)]

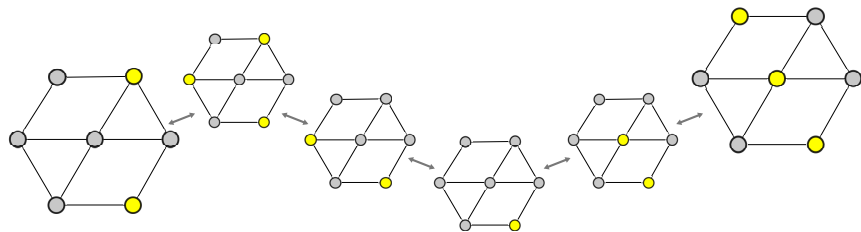
OPT-DSR est PSPACE-complet, y compris lorsque le graphe d'entrée :

- est un graphe biparti ;
- est un graphe split ;
- a une *pathwidth* bornée.



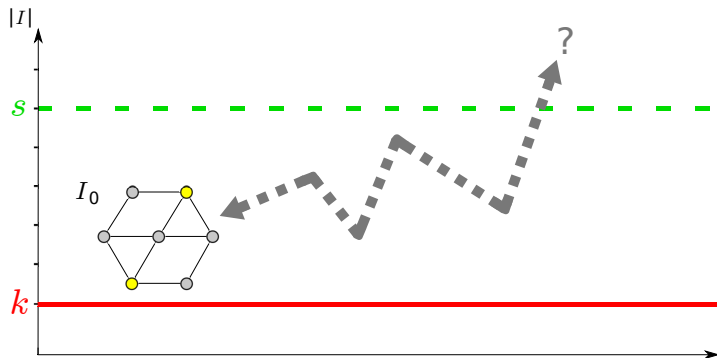
Preuve : Adaptation d'un résultat sur les **stables** et le problème **OPT-ISR**, analogue à OPT-DSR.

OPT-ISR concerne la reconfiguration de **stables**.



OPT-ISR

- **Instance** : Un graphe G , $k, s \in \mathbb{N}$, un stable I_0 avec $|I_0| \geq k$.
- **Question** : Existe-t-il un stable I_s avec $|I_s| \geq s$ et $I_0 \overset{\geq k}{\rightsquigarrow} I_s$?



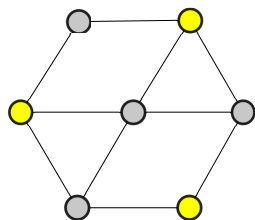
OPT-ISR

- **Instance** : Un graphe G , $k, s \in \mathbb{N}$, un stable I_0 avec $|I_0| \geq k$.
- **Question** : Existe-t-il un stable I_s avec $|I_s| \geq s$ et $I_0 \overset{\geq k}{\rightsquigarrow} I_s$?

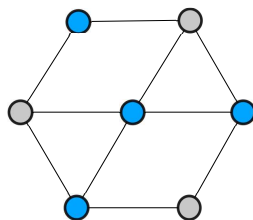
Théorème [Ito, Mizuta, Nishimura, Suzuki (2018)]

OPT-ISR est PSPACE-complet, y compris lorsque le graphe d'entrée est un graphe split ou biparti ou a une pathwidth bornée.

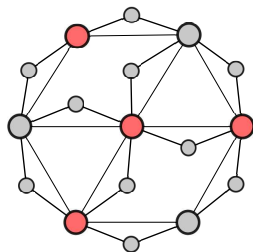
Idée de la réduction



Un stable



Un vertex cover
(son complémentaire)



Un dominant équivalent

Corollaire

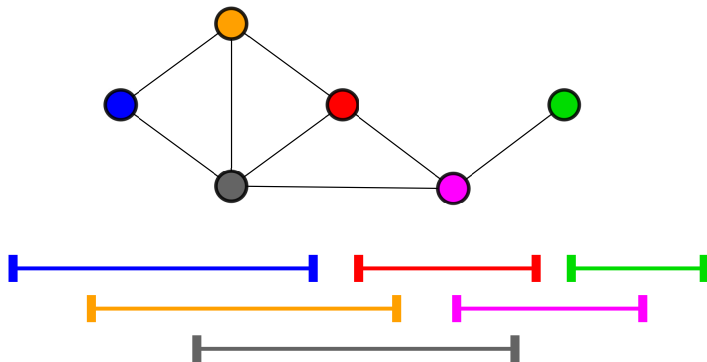
OPT-ISR est PSPACE-dur \Rightarrow **OPT-VCR** est PSPACE-dur

Théorème [B., Mizuta, Ouvrard, Suzuki, (2018+)]

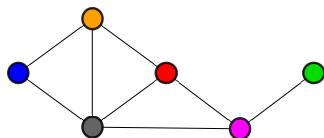
OPT-DSR est PSPACE-dur.

Théorème [B., Mizuta, Ouvrard, Suzuki, (2018+)]

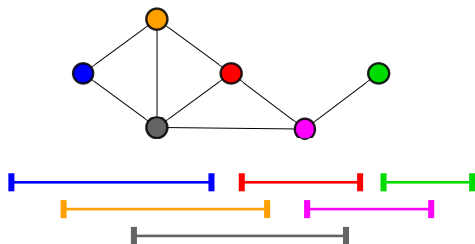
OPT-DSR peut être résolu en temps polynomial sur les graphes d'intervalles.



Preuve : (G, k, s, D) , avec G un graphe d'intervalles

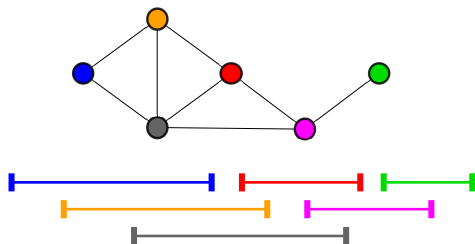


Preuve : (G, k, s, D) , avec G un graphe d'intervalles



- On construit une représentation possible de G sous forme d'intervalles, en temps linéaire en $|G|$.

Preuve : (G, k, s, D) , avec G un graphe d'intervalles

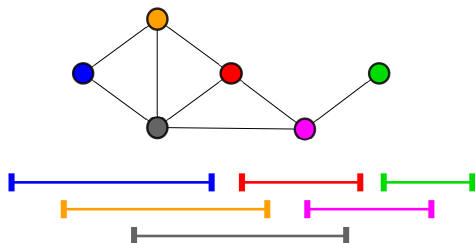


- On construit un **dominant minimum** D_m de G , en temps linéaire en $|G|$.

Lemme [Bonamy, Dorbec, Ouvrard (2017+)]

On peut reconfigurer D en D' t.q. $D_m \subseteq D'$, sous la borne $|D| + 1$, en temps linéaire en $|G|$.

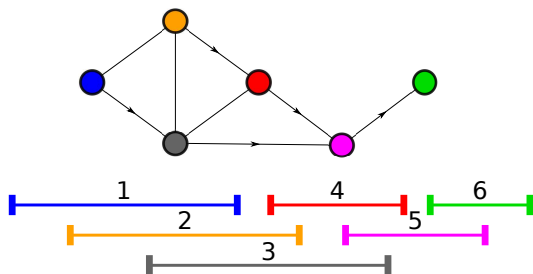
Preuve : (G, k, s, D) , avec G un graphe d'intervalles



- On construit un **dominant minimum** D_m de G , en temps linéaire en $|G|$.
- On a donc $D \stackrel{k}{\leftrightarrow} D' \stackrel{k}{\leftrightarrow} D_m$.
On peut répondre oui si $|D_m| \leq s$ et produire la séquence correspondante en temps linéaire en $|G|$.

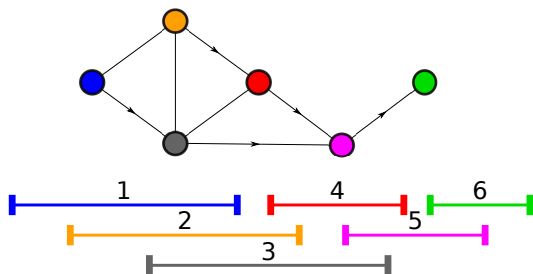
Preuve : On construit un dominant minimum de G .

- Ordre : par date de fin
- *Voisin privilégié* de $v_i :=$ voisin max de v_i dans l'ordre



Preuve : On construit un dominant minimum de G .

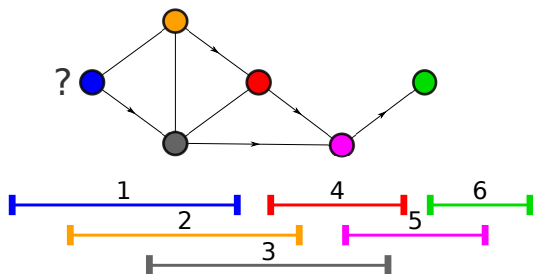
- Ordre : par date de fin
- *Voisin privilégié* de v_i := voisin max de v_i dans l'ordre



- **Algorithme** Parcourir les sommets dans l'ordre.
 - Si v_i est dominé, passer au suivant.
 - Sinon, ajouter son voisin privilégié au dominant.

Preuve : On construit un dominant minimum de G .

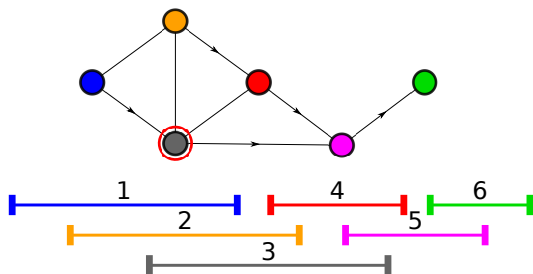
- Ordre : par date de fin
- *Voisin privilégié* de v_i := voisin max de v_i dans l'ordre



- **Algorithme** Parcourir les sommets dans l'ordre.
 - Si v_i est dominé, passer au suivant.
 - Sinon, ajouter son voisin privilégié au dominant.

Preuve : On construit un dominant minimum de G .

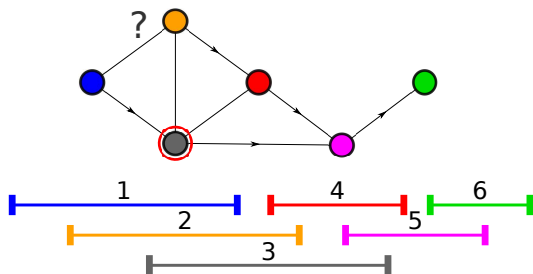
- Ordre : par date de fin
- *Voisin privilégié* de v_i := voisin max de v_i dans l'ordre



- **Algorithme** Parcourir les sommets dans l'ordre.
 - Si v_i est dominé, passer au suivant.
 - Sinon, ajouter son voisin privilégié au dominant.

Preuve : On construit un dominant minimum de G .

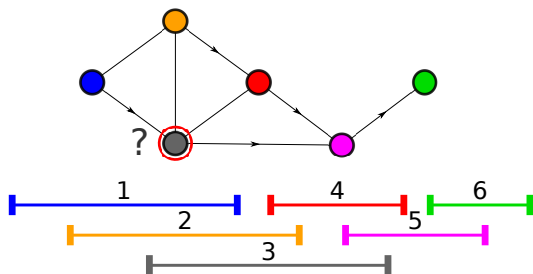
- Ordre : par date de fin
- *Voisin privilégié* de v_i := voisin max de v_i dans l'ordre



- **Algorithme** Parcourir les sommets dans l'ordre.
 - Si v_i est dominé, passer au suivant.
 - Sinon, ajouter son voisin privilégié au dominant.

Preuve : On construit un dominant minimum de G .

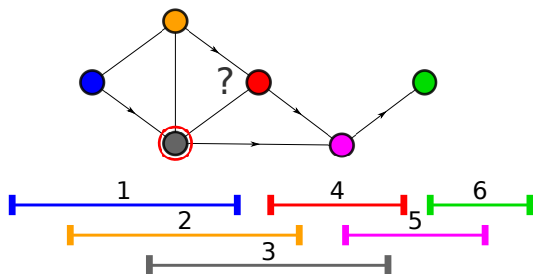
- Ordre : par date de fin
- *Voisin privilégié* de v_i := voisin max de v_i dans l'ordre



- **Algorithme** Parcourir les sommets dans l'ordre.
 - Si v_i est dominé, passer au suivant.
 - Sinon, ajouter son voisin privilégié au dominant.

Preuve : On construit un dominant minimum de G .

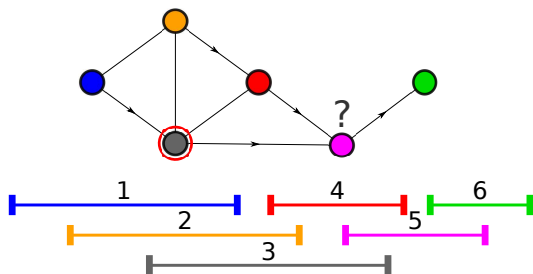
- Ordre : par date de fin
- *Voisin privilégié* de v_i := voisin max de v_i dans l'ordre



- **Algorithme** Parcourir les sommets dans l'ordre.
 - Si v_i est dominé, passer au suivant.
 - Sinon, ajouter son voisin privilégié au dominant.

Preuve : On construit un dominant minimum de G .

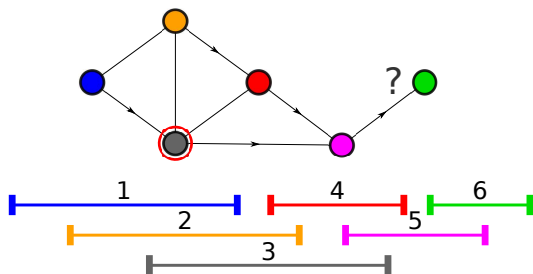
- Ordre : par date de fin
- *Voisin privilégié* de v_i := voisin max de v_i dans l'ordre



- **Algorithme** Parcourir les sommets dans l'ordre.
 - Si v_i est dominé, passer au suivant.
 - Sinon, ajouter son voisin privilégié au dominant.

Preuve : On construit un dominant minimum de G .

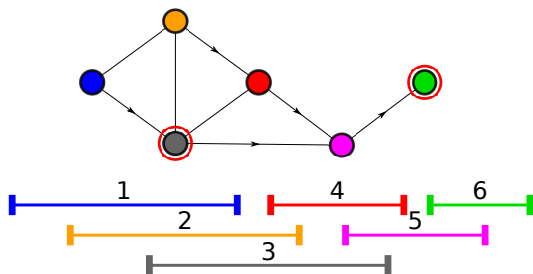
- Ordre : par date de fin
- *Voisin privilégié* de v_i := voisin max de v_i dans l'ordre



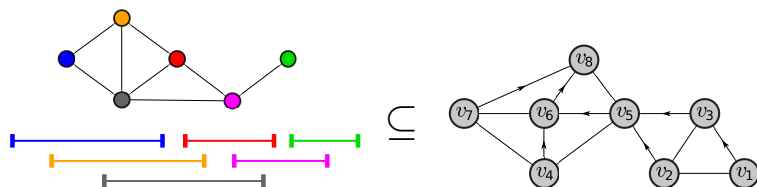
- **Algorithme** Parcourir les sommets dans l'ordre.
 - Si v_i est dominé, passer au suivant.
 - Sinon, ajouter son voisin privilégié au dominant.

Preuve : On construit un dominant minimum de G .

- Ordre : par date de fin
- *Voisin privilégié* de v_i := voisin max de v_i dans l'ordre



- **Algorithme** Parcourir les sommets dans l'ordre.
 - Si v_i est dominé, passer au suivant.
 - Sinon, ajouter son voisin privilégié au dominant.

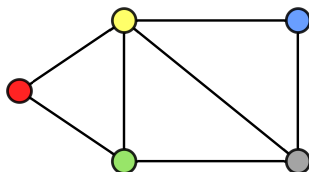


Graphes d'intervalles \subseteq Dually-chordal graphs

Théorème [B., Mizuta, Ouvrard, Suzuki, (2018+)]

OPT-DSR peut être résolu en temps polynomial sur les Dually-chordal graphs.

Un graphe G est d -dégénéré s'il possède un sommet v de degré $\leq d$ et $G - v$ est également d -dégénéré.

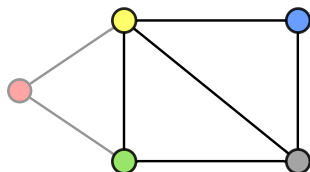


Un graphe 2-dégénéré

Théorème [B., Mizuta, Ouvrard, Suzuki, (2018+)]

OPT-DSR peut être résolu en temps $\text{FPT}(d + s)$,
i.e. en temps $f(d + s) \times n^{O(1)}$ si $|G| = n$;
où d est la dégénérescence du graphe
et s est la taille de la solution recherchée.

Un graphe G est d -dégénéré s'il possède un sommet v de degré $\leq d$ et $G - v$ est également d -dégénéré.

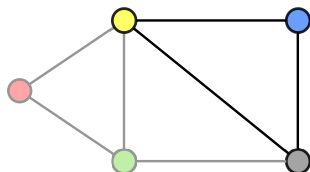


Un graphe 2-dégénéré

Théorème [B., Mizuta, Ouvrard, Suzuki, (2018+)]

OPT-DSR peut être résolu en temps $\text{FPT}(d + s)$,
i.e. en temps $f(d + s) \times n^{O(1)}$ si $|G| = n$;
où d est la dégénérescence du graphe
et s est la taille de la solution recherchée.

Un graphe G est d -dégénéré s'il possède un sommet v de degré $\leq d$ et $G - v$ est également d -dégénéré.

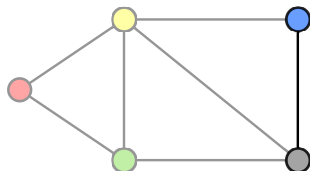


Un graphe 2-dégénéré

Théorème [B., Mizuta, Ouvrard, Suzuki, (2018+)]

OPT-DSR peut être résolu en temps $\text{FPT}(d + s)$,
i.e. en temps $f(d + s) \times n^{O(1)}$ si $|G| = n$;
où d est la dégénérescence du graphe
et s est la taille de la solution recherchée.

Un graphe G est d -dégénéré s'il possède un sommet v de degré $\leq d$ et $G - v$ est également d -dégénéré.

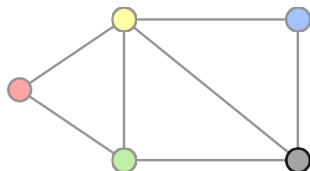


Un graphe 2-dégénéré

Théorème [B., Mizuta, Ouvrard, Suzuki, (2018+)]

OPT-DSR peut être résolu en temps $\text{FPT}(d + s)$,
i.e. en temps $f(d + s) \times n^{O(1)}$ si $|G| = n$;
où d est la dégénérescence du graphe
et s est la taille de la solution recherchée.

Un graphe G est d -dégénéré s'il possède un sommet v de degré $\leq d$ et $G - v$ est également d -dégénéré.



Un graphe 2-dégénéré

Théorème [B., Mizuta, Ouvrard, Suzuki, (2018+)]

OPT-DSR peut être résolu en temps $\text{FPT}(d + s)$,
i.e. en temps $f(d + s) \times n^{O(1)}$ si $|G| = n$;
où d est la dégénérescence du graphe
et s est la taille de la solution recherchée.

Théorème [B., Mizuta, Ouvrard, Suzuki, (2018+)]

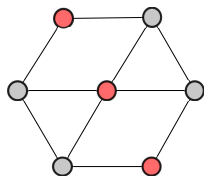
OPT-DSR peut être résolu en temps $FPT(\tau)$,
où τ est la taille d'un vertex cover minimum du graphe.

Théorème [B., Mizuta, Ouvrard, Suzuki, (2018+)]

OPT-DSR peut être résolu en temps $FPT(\tau)$,
où τ est la taille d'un vertex cover minimum du graphe.

Preuve :

- **Cas trivial** : Si $|D| = k$ et D est **minimal**, alors D est *figé*.
On ne peut ni retirer ni ajouter de sommet,
donc l'instance (G, k, s, D) est positive ssi $|D| \leq s$.
→ On peut tester cette condition en temps $O(|G|)$ ✓



Exemple pour $k = 3$

Théorème [B., Mizuta, Ouvrard, Suzuki, (2018+)]

OPT-DSR peut être résolu en temps $FPT(\tau)$,
où τ est la taille d'un vertex cover minimum du graphe.

Preuve :

- **Cas trivial** : Si $|D| = k$ et D est **minimal**, alors D est *figé*.
On ne peut ni retirer ni ajouter de sommet,
donc l'instance (G, k, s, D) est positive ssi $|D| \leq s$.
→ On peut tester cette condition en temps $O(|G|)$ ✓
- Si $|D| = k$ et D **non minimal**, alors on peut retirer un sommet à D et se ramener au dernier cas, $|D| < k$.

Désormais on suppose que $|D| < k$.

→ On peut ajouter au moins 1 sommet à D sans dépasser k

Théorème [B., Mizuta, Ouvrard, Suzuki, (2018+)]

OPT-DSR peut être résolu en temps $FPT(\tau)$,
où τ est la taille d'un vertex cover minimum du graphe.

Preuve :

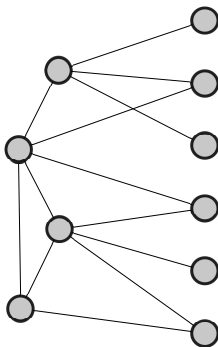
- On calcule τ en temps $FPT(\tau)$.
- 2 possibilités :
 - Soit $\tau > s$: Comme $d \leq \tau$, on a $d + s \leq 2\tau$.
→ on utilise l'algorithme en temps $FPT(d + s)$ ✓

Théorème [B., Mizuta, Ouvrard, Suzuki, (2018+)]

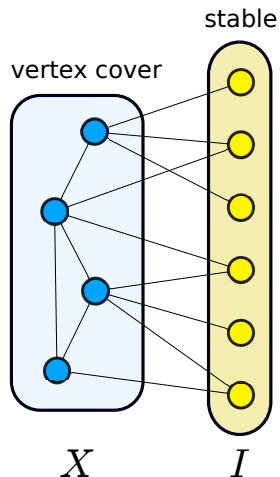
OPT-DSR peut être résolu en temps $FPT(\tau)$,
où τ est la taille d'un vertex cover minimum du graphe.

Preuve :

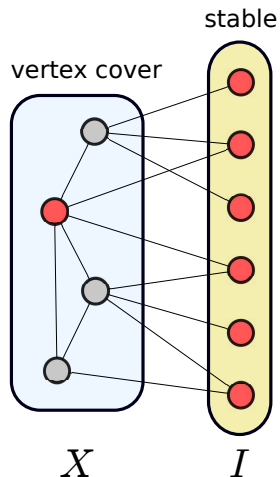
- On calcule τ en temps $FPT(\tau)$.
- 2 possibilités :
 - Soit $\tau > s$: Comme $d \leq \tau$, on a $d + s \leq 2\tau$.
→ on utilise l'algorithme en temps $FPT(d + s)$ ✓
 - Soit $\tau \leq s$: L'instance est dans ce cas une **instance positive**.
Pour le montrer, on va reconfigurer D en un dominant D' qui vérifie $|D'| \leq \tau$.



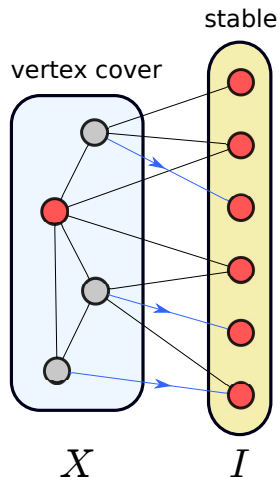
On se donne un graphe G .



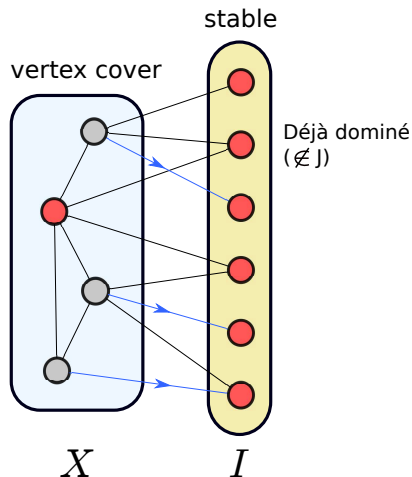
On construit un vertex cover minimum en temps $\text{FPT}(\tau)$.

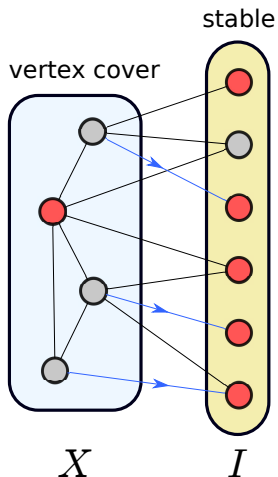


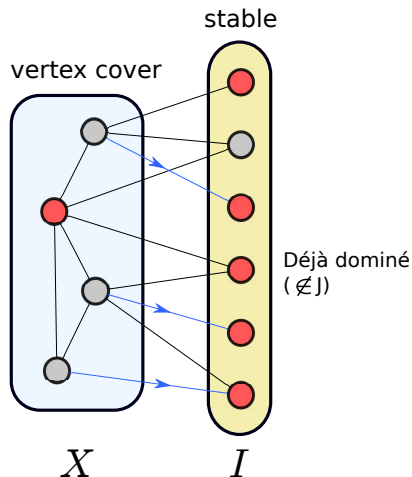
On se donne un dominant de départ D .

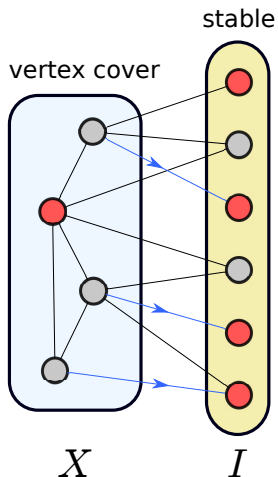


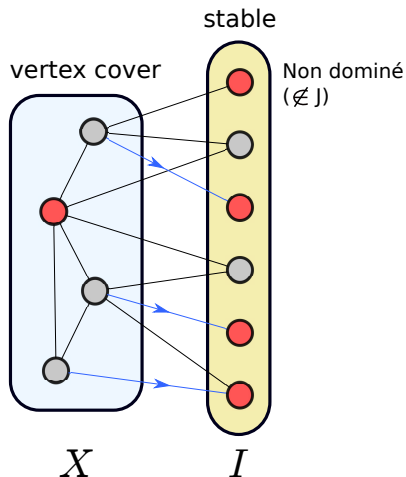
On associe un voisin $j(v)$ à chaque $v \in X \setminus D$.
 Soit $J = \{j(v) \mid v \in X \setminus D\}$.

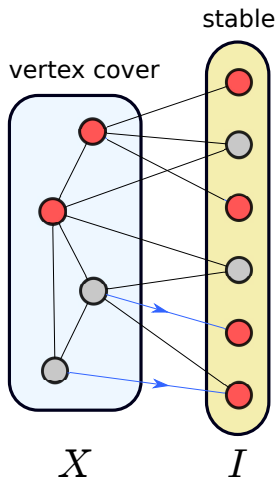


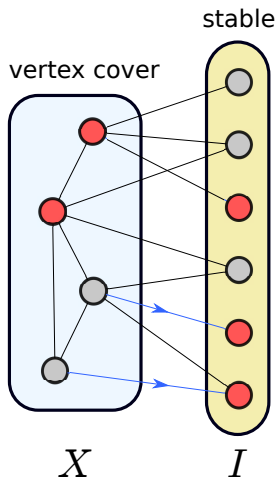


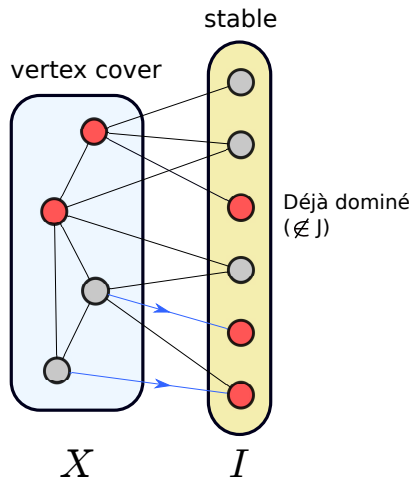


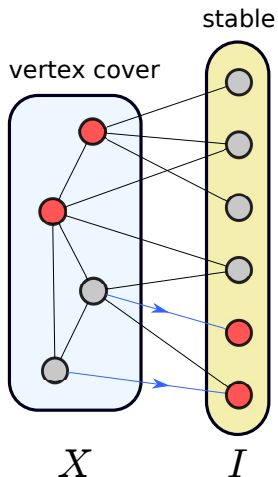


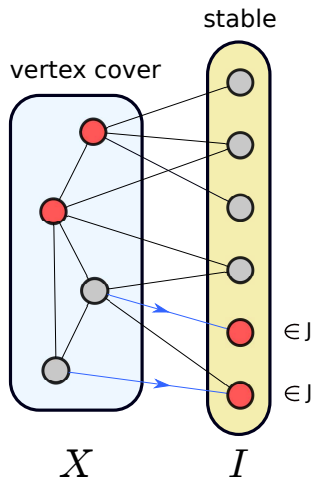






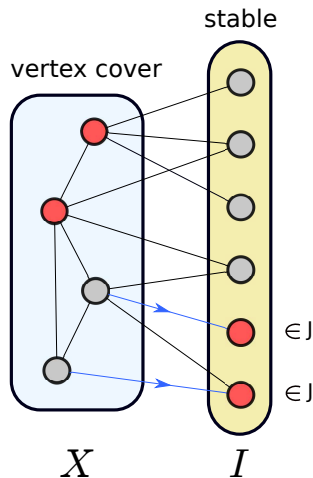






Lorsque $(I \cap D') \subseteq J$, on a :

$$\begin{aligned} |I \cap D'| &\leq |j^{-1}(I \cap D')| \\ &\leq |X \setminus D'| \\ &= \tau - |X \cap D'| \end{aligned}$$



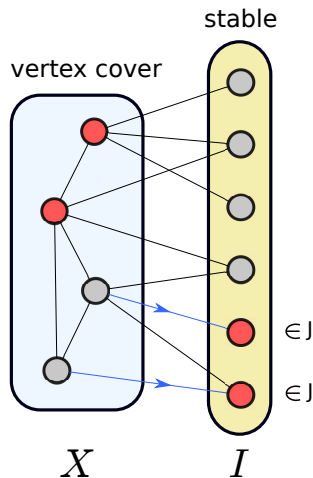
Lorsque $(I \cap D') \subseteq J$, on a :

$$\begin{aligned} |I \cap D'| &\leq |j^{-1}(I \cap D')| \\ &\leq |X \setminus D'| \\ &= \tau - |X \cap D'| \end{aligned}$$

On a donc :

$$\begin{aligned} |D'| &= |I \cap D'| + |X \cap D'| \\ &\leq \tau \\ &\leq s \end{aligned}$$

Donc D' est bien une solution :
l'instance est positive.



Complexité de OPT-DSR

- PSPACE-complet
(même sur les graphes bipartis, split, ou à pathwidth bornée)
- Polynomial sur les graphes Dually-chordal
(donc sur les graphes d'intervalles)
- $FPT(d + s)$ ($d =$ dégénérescence, $s =$ taille de la solution)
- $FPT(\tau)$ ($\tau =$ taille d'un vertex cover minimum)

Complexité de OPT-DSR

- PSPACE-complet
(même sur les graphes bipartis, split, ou à pathwidth bornée)
- Polynomial sur les graphes Dually-chordal
(donc sur les graphes d'intervalles)
- $FPT(d + s)$ ($d =$ dégénérescence, $s =$ taille de la solution)
- $FPT(\tau)$ ($\tau =$ taille d'un vertex cover minimum)

Merci de votre attention.

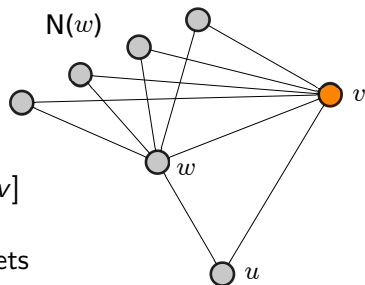
Graphes dually chordal

$G = (V, E)$ avec $V = \{v_1, v_2, \dots, v_n\}$.

$G_i := G[\{v_i, \dots, v_n\}]$

v est un *voisin maximum* de u si :
pour tout voisin w de u , $N[w] \subseteq N[v]$

(i.e. v est adjacent à tous les sommets
à distance ≤ 2 de u)



Graphes dually chordal

Un *ordre de voisinage maximum* est un ordre sur les sommets t.q. v_i possède un voisin maximum dans G_i , pour tout $i \in \llbracket 1, n \rrbracket$.

Un graphe est *dually chordal* s'il possède un ordre de voisinage maximum.

