Tree-breadth d'un graphe : propriétés structurelles et complexité

Guillaume Ducoffe [1,2] Sylvain Legay [3] Nicolas Nisse [1,2]

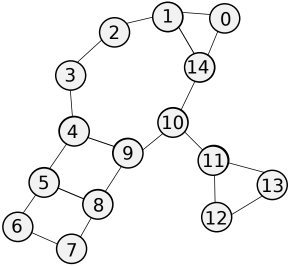
[1] Inria, France
[2] Univ. Nice Sophia Antipolis, CNRS, I3S, UMR 7271, 06900 Sophia Antipolis, France
[3] LRI, Laboratoire de Recherche en Informatique (Université Paris-Sud 11)

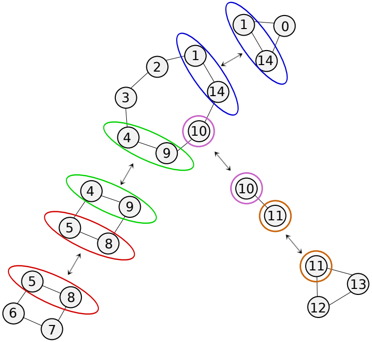
COATI

Décompositions de graphes (intuition)

séparations (répétées) d'un graphe

décomposition ∼ famille des sous-graphes obtenus (“sacs”)





décomposition <u>arborescente</u> = les sacs “forment un arbre”.

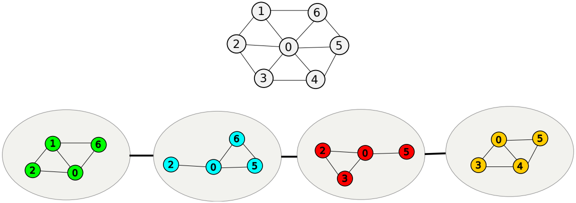
Définitions

décomposition arborescente $(T, \mathcal{X})$

- T est un arbre;
- $\mathcal{X}$ est une famille de sous-graphes (appelés “sacs”).

quelques contraintes:

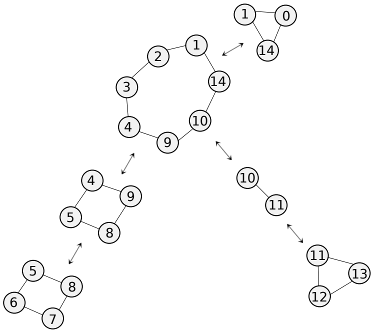
- chaque sommet est dans un sac;
- chaque arête est dans un sac;
- pour tout sommet v, $T_v = \{B \in \mathcal{X} \mid v \in B\}$ est un sous-arbre.

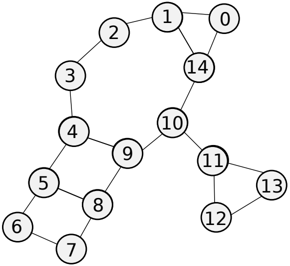


Intérêts des décompositions arborescentes

- vue "locale" du graphe
- algorithmes "diviser pour régner"

  ex: coloration du graphe

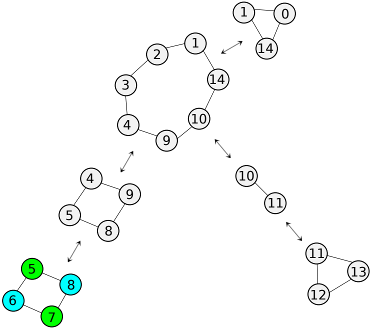


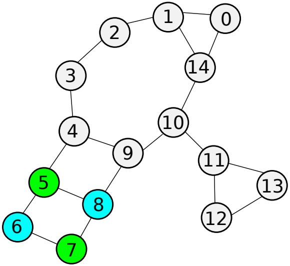


Intérêts des décompositions arborescentes

- vue "locale" du graphe
- algorithmes "diviser pour régner"

ex: coloration du graphe

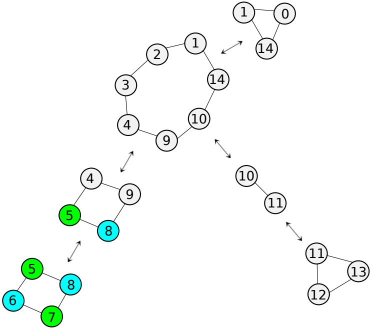


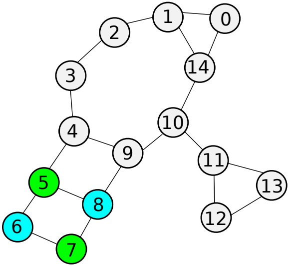


Intérêts des décompositions arborescentes

- vue "locale" du graphe
- algorithmes "diviser pour régner"

  ex: coloration du graphe

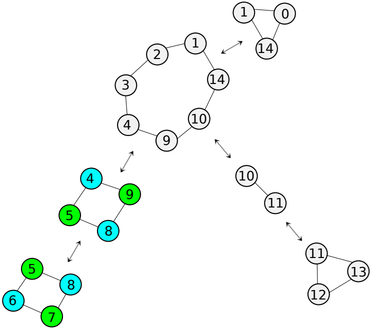


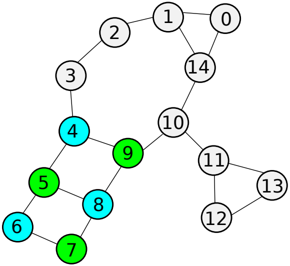


Intérêts des décompositions arborescentes

- vue "locale" du graphe
- algorithmes "diviser pour régner"

ex: coloration du graphe

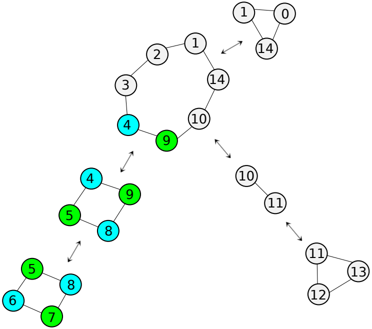


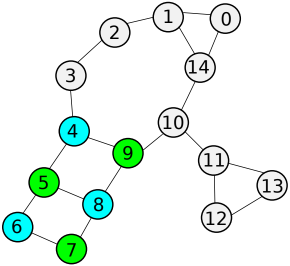


Intérêts des décompositions arborescentes

- vue "locale" du graphe
- algorithmes "diviser pour régner"

ex: coloration du graphe

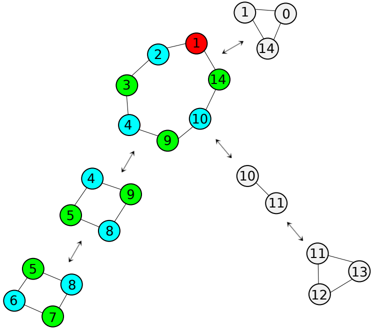


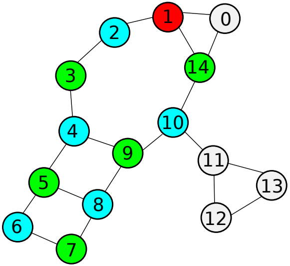


Intérêts des décompositions arborescentes

- vue “locale” du graphe
- algorithmes “diviser pour régner”

ex: coloration du graphe

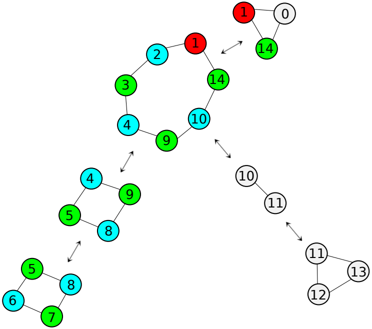


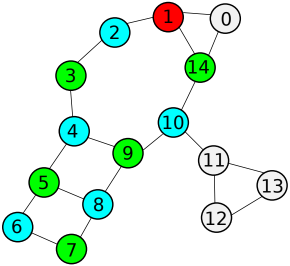


Intérêts des décompositions arborescentes

- vue "locale" du graphe
- algorithmes "diviser pour régner"

ex: coloration du graphe

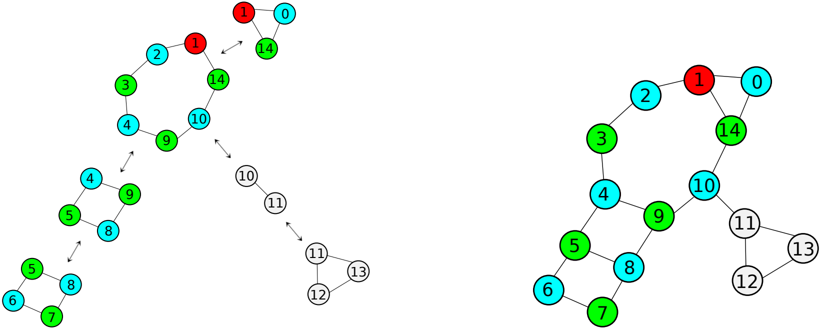




Intérêts des décompositions arborescentes

- vue "locale" du graphe
- algorithmes "diviser pour régner"

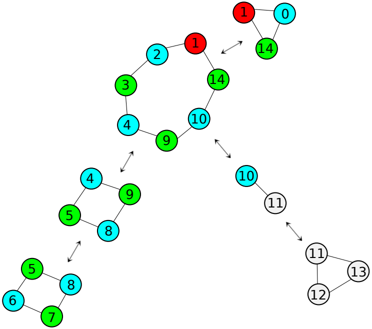
  ex: coloration du graphe

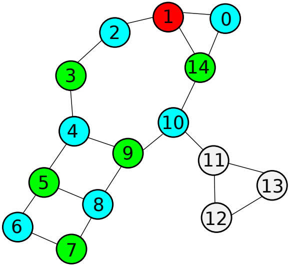


Intérêts des décompositions arborescentes

- vue "locale" du graphe
- algorithmes "diviser pour régner"

  ex: coloration du graphe

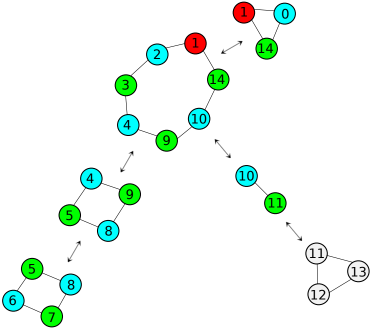


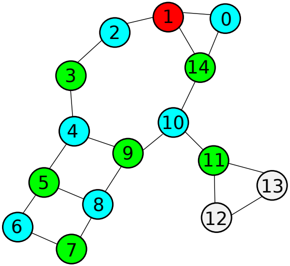


Intérêts des décompositions arborescentes

- vue "locale" du graphe
- algorithmes "diviser pour régner"

  ex: coloration du graphe

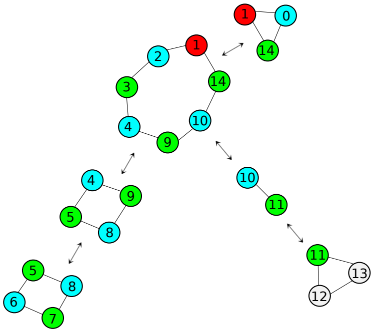


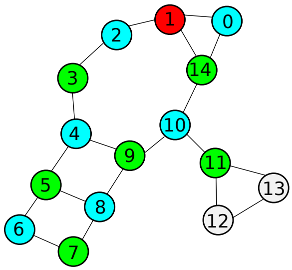


Intérêts des décompositions arborescentes

- vue “locale” du graphe
- algorithmes “diviser pour régner”

  ex: coloration du graphe

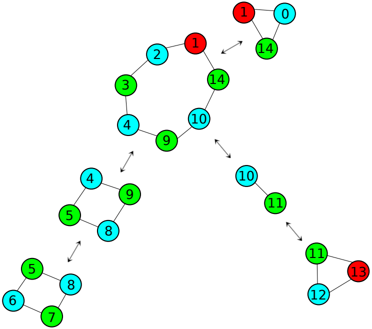


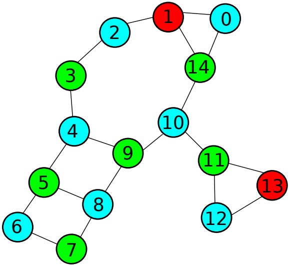


Intérêts des décompositions arborescentes

- vue "locale" du graphe
- algorithmes "diviser pour régner"

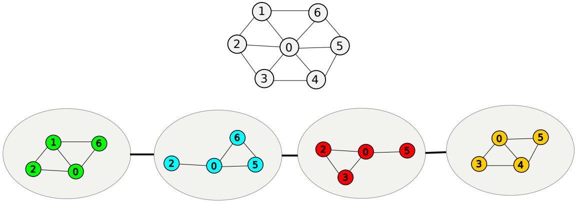
    ex: coloration du graphe





Propriétés des sacs pour les algorithmes ?

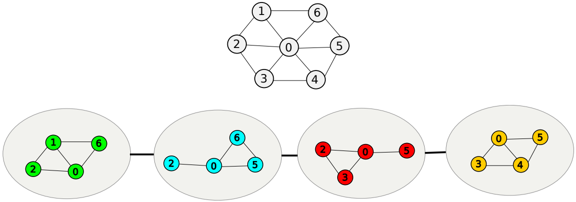
- sacs de petite taille;
  - sacs de taille $\leq k+1 \overset{\text{def}}{\Longrightarrow}$ *treewidth* $\leq k$.
  - résolution de problèmes par "*brute-force*": complexité $f(k) \cdot n^{\mathcal{O}(1)}$.



Propriétés des sacs pour les algorithmes ?

- sacs de petite taille;

  - sacs de taille $\leq k+1 \overset{\text{def}}{\Longrightarrow}$ *treewidth* $\leq k$.
  - résolution de problèmes par "*brute-force*": complexité $f(k) \cdot n^{\mathcal{O}(1)}$.

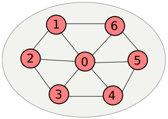


  - calcul d'une décomposition "optimale" :

- avec sacs de taille $\leq k$: en temps $k^{\mathcal{O}(k^3)} \cdot n$ **[Bodlaender, '96]**;
- avec sacs de taille $\leq 5k$: en temps $2^{\mathcal{O}(k)} \cdot n$ **[Bodlaender et al., '13]**;
- $\sqrt{\log OPT}$-approximation en temps polynomial **[Feige et al., '08]**.

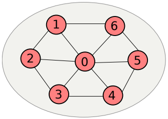
Propriétés des sacs pour les algorithmes ?

- sacs de petit diamètre;
  - sacs de diamètre $\leq k \overset{\text{def}}{\Longrightarrow}$ *treelength* $\leq k$.
  - suffisant pour des problèmes de distance:
    - $*$ *routage avec distortion* $\mathcal{O}(k)$ *et* $\mathcal{O}(k \cdot \log n)$ *entrées par table,*
    - $*$ *(*$+k$*)-approximation du diamètre et du rayon en temps linéaire,*
    - $*$ `PTAS` *pour le problème du voyageur de commerce.*



Propriétés des sacs pour les algorithmes ?

- sacs de petit diamètre;
  - sacs de diamètre $\leq k \overset{\text{def}}{\Longrightarrow}$ *treelength* $\leq k$.
  - suffisant pour des problèmes de distance:
    - *routage avec distortion $\mathcal{O}(k)$ et $\mathcal{O}(k \cdot \log n)$ entrées par table,*
    - *$(+k)$-approximation du diamètre et du rayon en temps linéaire,*
    - PTAS *pour le problème du voyageur de commerce.*

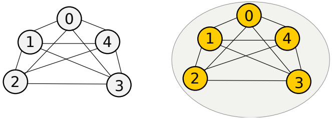


  - calcul d'une décomposition "optimale" :
    - NP-complet pour tout $k \geq 2$ **[Lokshtanov, '10]**;
    - avec sacs de diamètre $\leq 3k$: en temps $\mathcal{O}(n+m)$ **[Dourisboure et al., '07]**.

Treewidth vs. treelength

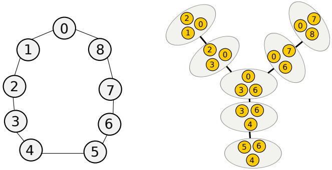
- **cas où treewidth $\gg$ treelength.**

  Graphe complet K_n: treewidth $n-1$, treelength 1.



- **cas où treewidth $\ll$ treelength.**

  Cycle C_n: treewidth 2, treelength $\left\lceil \frac{n}{3} \right\rceil$.

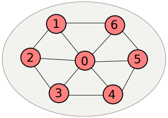


Propriétés des sacs pour les algorithmes ?

- [dans cet exposé] sacs de petit rayon;

  - sacs de rayon $\leq k \overset{\text{def}}{\Longrightarrow}$ *treebreadth* $\leq k$.

  - introduites indépendamment par **[Dragan et al., '15]**, et **[Li et al., '15]** pour:
    - *routage compact;*
    - *arbres couvrants avec faible distortion des distances;*
    - *plongement d'un graphe sur la ligne.*

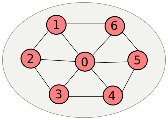


Propriétés des sacs pour les algorithmes ?

- [dans cet exposé] sacs de petit rayon;

  - sacs de rayon $\leq k \overset{\text{def}}{\Longrightarrow}$ *treebreadth* $\leq k$.

  - introduites indépendamment par **[Dragan et al., '15]**, et **[Li et al., '15]** pour:
    - *routage compact;*
    - *arbres couvrants avec faible distortion des distances;*
    - *plongement d'un graphe sur la ligne.*

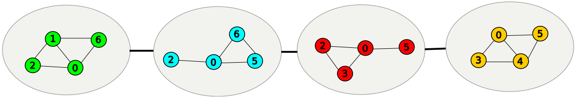


  - calcul d'une décomposition "optimale" : *complexité ouverte*

Nos contributions

Étude du calcul de la treebreadth.

1. **Variante NP-complet:** calcul de la *pathbreadth*;



2. **Équivalence:** calculer la treebreadth $\iff$ reconnaître si treebreadth ≤ 1.

   - treebreadth $\leq 1 \iff$ sacs dominés par un sommet.

3. Décider si treebreadth ≤ 1 **polynomial** pour les graphes planaires et les graphes bipartis.

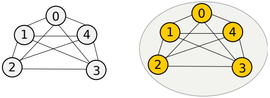
Premiers exemples

- Existence (treebreadth ≤ 1):

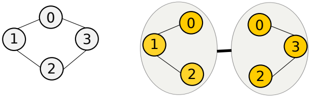
- Non-existence (treebreadth > 1):

Premiers exemples

- Existence (treebreadth ≤ 1):



  - graphe complet;

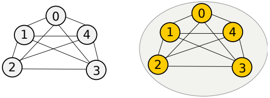


  - cycle à quatre sommets.

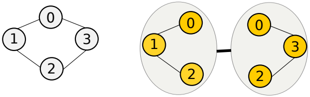
- Non-existence (treebreadth > 1):

Premiers exemples

- Existence (treebreadth ≤ 1):

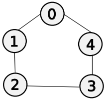


  - graphe complet;



  - cycle à quatre sommets.

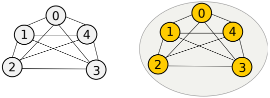
- Non-existence (treebreadth > 1):



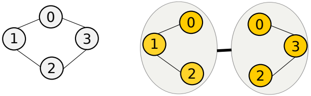
  - cycle à cinq sommets.

Premiers exemples

- Existence (treebreadth ≤ 1):

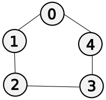


  - graphe complet;



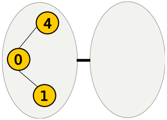
  - cycle à quatre sommets.

- Non-existence (treebreadth > 1):



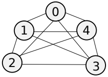
  - cycle à cinq sommets.

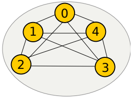
  par l'absurde . . .



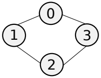
Premiers exemples

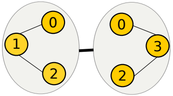
- Existence (treebreadth ≤ 1):





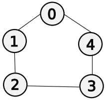
  - graphe complet;





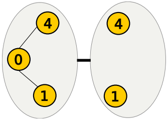
  - cycle à quatre sommets.

- Non-existence (treebreadth > 1):



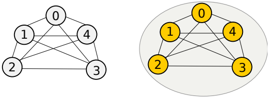
  - cycle à cinq sommets.

par l'absurde . . .

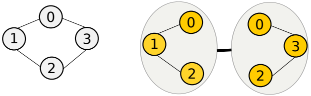


Premiers exemples

- Existence (treebreadth ≤ 1):

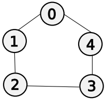


  - graphe complet;



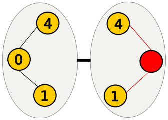
  - cycle à quatre sommets.

- Non-existence (treebreadth > 1):



  - cycle à cinq sommets.

  par l'absurde . . .



Propriétés structurelles des graphes de treebreadth ≤ 1

- ordre de domination: $V = (v_1, v_2, \ldots, v_n)$ et pour tout $i < n$, $\exists j > i$ t.q.

$$N(v_i) \setminus (v_1, \ldots, v_{i-1}) \subseteq N[v_j].$$

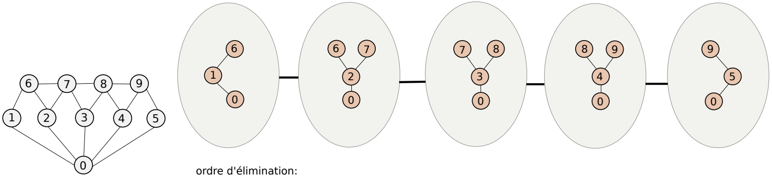
G de treebreadth $\leq 1 \implies G$ a un ordre de domination.

Propriétés structurelles des graphes de treebreadth ≤ 1

- ordre de domination: $V = (v_1, v_2, \ldots, v_n)$ et pour tout $i < n$, $\exists j > i$ t.q.

$$N(v_i) \setminus (v_1, \ldots, v_{i-1}) \subseteq N[v_j].$$

G de treebreadth $\leq 1 \implies G$ a un ordre de domination.

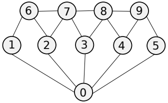


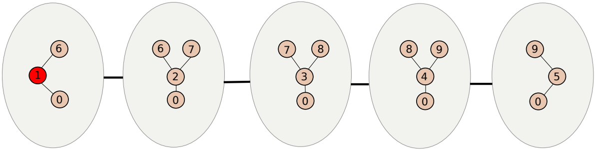
Propriétés structurelles des graphes de treebreadth ≤ 1

- <u>ordre de domination</u>: $V = (v_1, v_2, \ldots, v_n)$ et pour tout $i < n$, $\exists j > i$ t.q.

$$N(v_i) \setminus (v_1, \ldots, v_{i-1}) \subseteq N[v_j].$$

G de treebreadth $\leq 1 \implies G$ a un ordre de domination.





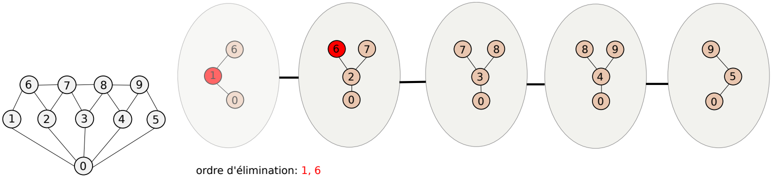
ordre d'élimination: 1

Propriétés structurelles des graphes de treebreadth ≤ 1

- ordre de domination: $V = (v_1, v_2, \ldots, v_n)$ et pour tout $i < n$, $\exists j > i$ t.q.

$$N(v_i) \setminus (v_1, \ldots, v_{i-1}) \subseteq N[v_j].$$

G de treebreadth $\leq 1 \implies G$ a un ordre de domination.

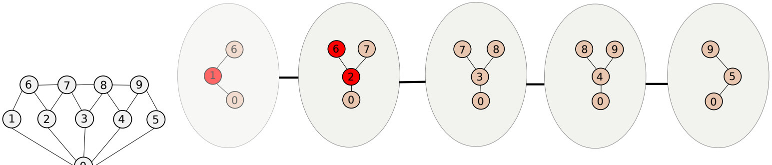


Propriétés structurelles des graphes de treebreadth ≤ 1

- <u>ordre de domination</u>: $V = (v_1, v_2, \ldots, v_n)$ et pour tout $i < n$, $\exists j > i$ t.q.

$$N(v_i) \setminus (v_1, \ldots, v_{i-1}) \subseteq N[v_j].$$

G de treebreadth $\leq 1 \implies G$ a un ordre de domination.



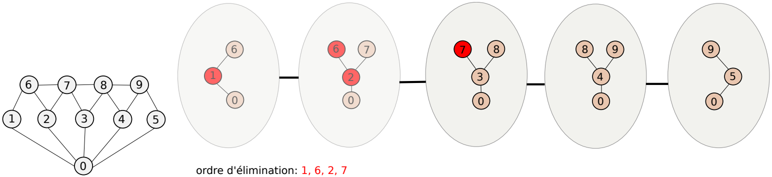
ordre d'élimination: 1, 6, 2

Propriétés structurelles des graphes de treebreadth ≤ 1

- ordre de domination: $V = (v_1, v_2, \ldots, v_n)$ et pour tout $i < n$, $\exists j > i$ t.q.

$$N(v_i) \setminus (v_1, \ldots, v_{i-1}) \subseteq N[v_j].$$

G de treebreadth $\leq 1 \implies G$ a un ordre de domination.



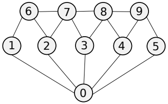
ordre d'élimination: 1, 6, 2, 7

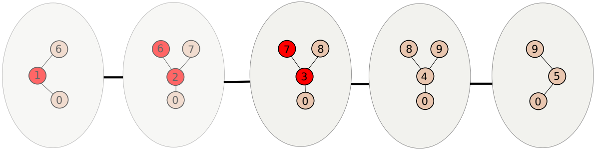
Propriétés structurelles des graphes de treebreadth ≤ 1

- ordre de domination: $V = (v_1, v_2, \ldots, v_n)$ et pour tout $i < n$, $\exists j > i$ t.q.

$$N(v_i) \setminus (v_1, \ldots, v_{i-1}) \subseteq N[v_j].$$

G de treebreadth $\leq 1 \implies G$ a un ordre de domination.





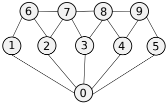
ordre d'élimination: 1, 6, 2, 7, 3

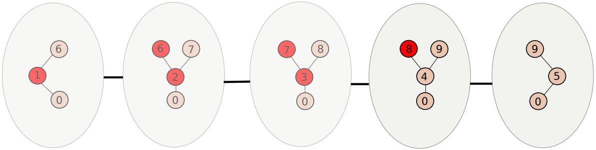
Propriétés structurelles des graphes de treebreadth ≤ 1

- ordre de domination: $V = (v_1, v_2, \ldots, v_n)$ et pour tout $i < n$, $\exists j > i$ t.q.

$$N(v_i) \setminus (v_1, \ldots, v_{i-1}) \subseteq N[v_j].$$

G de treebreadth $\leq 1 \implies G$ a un ordre de domination.





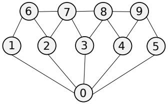
ordre d'élimination: 1, 6, 2, 7, 3, 8

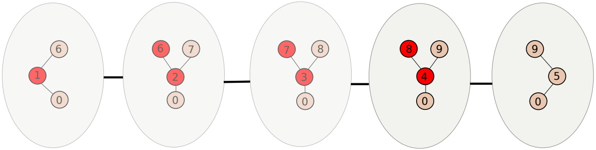
Propriétés structurelles des graphes de treebreadth ≤ 1

- ordre de domination: $V = (v_1, v_2, \ldots, v_n)$ et pour tout $i < n$, $\exists j > i$ t.q.

$$N(v_i) \setminus (v_1, \ldots, v_{i-1}) \subseteq N[v_j].$$

G de treebreadth $\leq 1 \implies G$ a un ordre de domination.





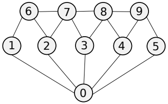
ordre d'élimination: 1, 6, 2, 7, 3, 8, 4

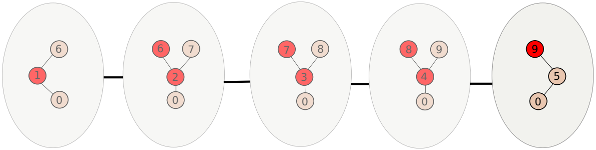
Propriétés structurelles des graphes de treebreadth ≤ 1

- <u>ordre de domination</u>: $V = (v_1, v_2, \ldots, v_n)$ et pour tout $i < n$, $\exists j > i$ t.q.

$$N(v_i) \setminus (v_1, \ldots, v_{i-1}) \subseteq N[v_j].$$

G de treebreadth $\leq 1 \implies G$ a un ordre de domination.





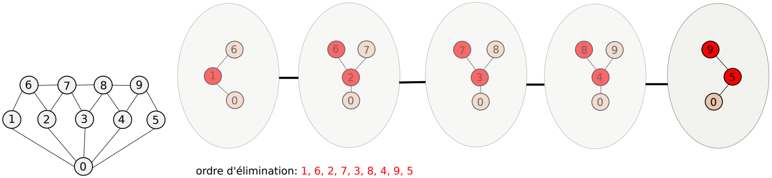
ordre d'élimination: 1, 6, 2, 7, 3, 8, 4, 9

Propriétés structurelles des graphes de treebreadth ≤ 1

- <u>ordre de domination</u>: $V = (v_1, v_2, \ldots, v_n)$ et pour tout $i < n$, $\exists j > i$ t.q.

$$N(v_i) \setminus (v_1, \ldots, v_{i-1}) \subseteq N[v_j].$$

G de treebreadth $\leq 1 \implies G$ a un ordre de domination.



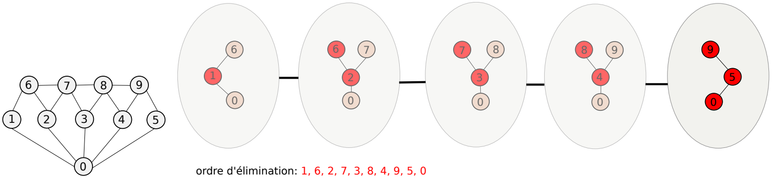
ordre d'élimination: 1, 6, 2, 7, 3, 8, 4, 9, 5

Propriétés structurelles des graphes de treebreadth ≤ 1

- ordre de domination: $V = (v_1, v_2, \ldots, v_n)$ et pour tout $i < n$, $\exists j > i$ t.q.

$$N(v_i) \setminus (v_1, \ldots, v_{i-1}) \subseteq N[v_j].$$

G de treebreadth $\leq 1 \implies G$ a un ordre de domination.



ordre d'élimination: 1, 6, 2, 7, 3, 8, 4, 9, 5, 0

Propriétés structurelles des graphes de treebreadth ≤ 1

- séparabilité: soient $G = (V, E)$, S un séparateur minimal de G, et C une *full component* de $G \setminus S$.

  G **de treebreadth** $\leq 1 \implies G[C \cup S]$ **et** $G[V \setminus C]$ **de treebreadth** ≤ 1**.**

  - **équivalence** si S est une clique.

Reconnaissance des graphes de treebreadth ≤ 1

quelques cas polynomiaux

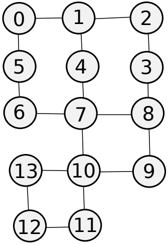
- graphes bipartis .

- graphes planaires.

Reconnaissance des graphes de treebreadth ≤ 1

quelques cas polynomiaux

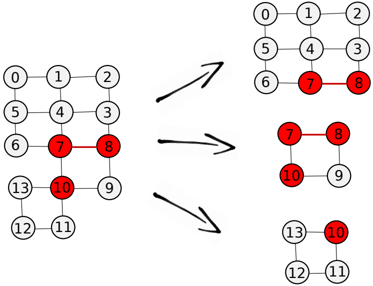
- graphes bipartis .



Reconnaissance des graphes de treebreadth ≤ 1

quelques cas polynomiaux

- graphes bipartis (sans clique-séparatrice).

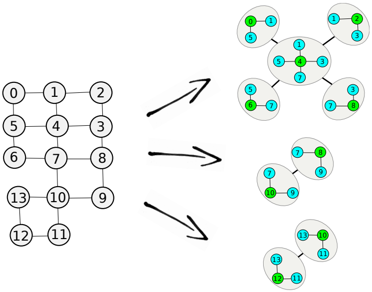


Reconnaissance des graphes de treebreadth ≤ 1

quelques cas polynomiaux

- graphes bipartis (sans clique-séparatrice).

  - treebreadth $\leq 1 \implies$ ensemble des sacs $=$ tous les voisinages d'un côté de la bipartition.

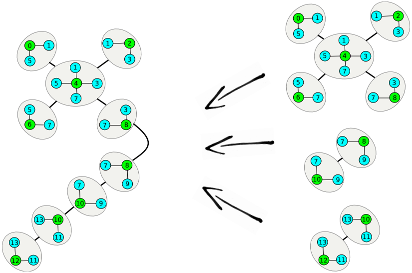


Reconnaissance des graphes de treebreadth ≤ 1

quelques cas polynomiaux

- graphes bipartis (sans clique-séparatrice).

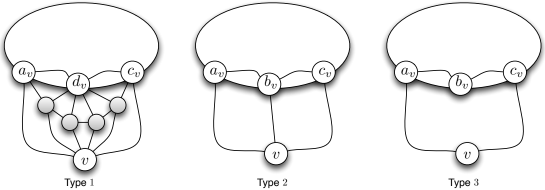
  - treebreadth $\leq 1 \implies$ ensemble des sacs $=$ tous les voisinages d'un côté de la bipartition.



Reconnaissance des graphes de treebreadth ≤ 1

quelques cas polynomiaux

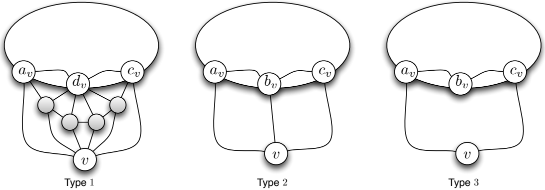
- graphes planaires (sans clique-séparatrice)
  - règles de réduction
  - trois cas de base:



Reconnaissance des graphes de treebreadth ≤ 1

quelques cas polynomiaux

- graphes planaires (sans clique-séparatrice)
  - règles de réduction
  - trois cas de base:



- **Propriété:** G planaire de treebreadth $\leq 1 \Longrightarrow G$ de treewidth ≤ 4.

Conclusion

- calcul de la treebreadth $\Longleftrightarrow$ reconnaissance des graphes de treebreadth ≤ 1;

- plusieurs propriétés algorithmiques des graphes de treebreadth ≤ 1;

- reconnaissance en temps polynomial des graphes planaires et des graphes bipartis de treebreadth ≤ 1.

Question ouverte: complexité pour les graphes généraux ?

$\longrightarrow$ calcul de la pathbreadth NP-complet.

$\longrightarrow$ complexité pour les graphes sans mineur $K_{3,p}$?

$\longrightarrow$ (in)approximabilité ? (il existe une 3-approximation, **[Dragan et al., '15]**)