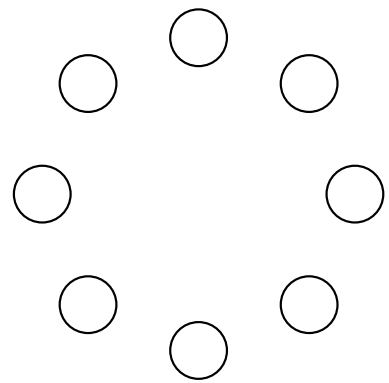
Décomposition de Graphes et Algorithmes

Édouard Bonnet

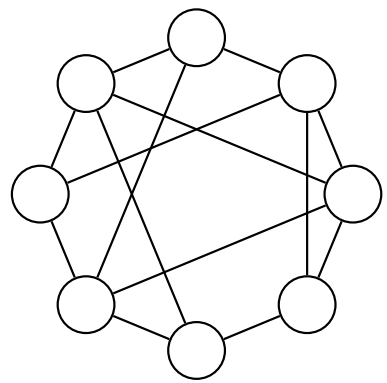
CNRS, ENS de Lyon, LIP

12 mai 2026, Académie des Sciences

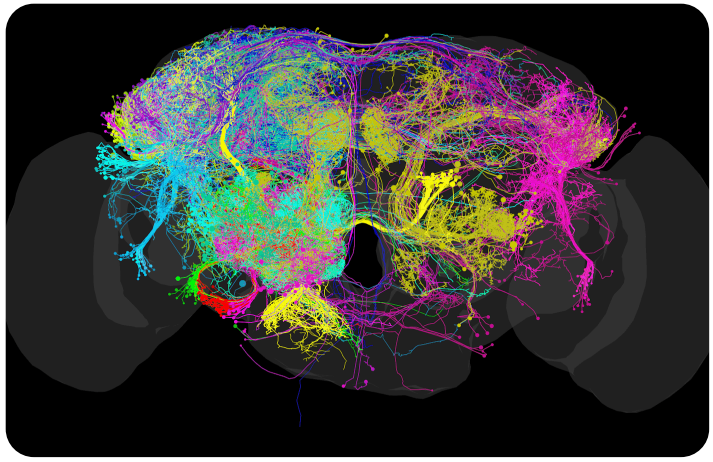
Les graphes sont partout

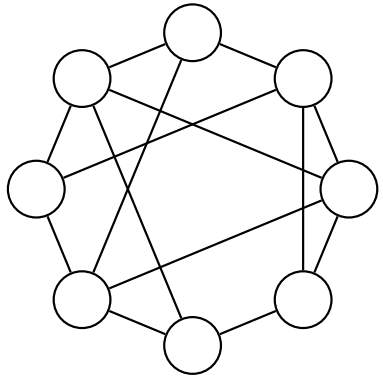


Les graphes sont partout

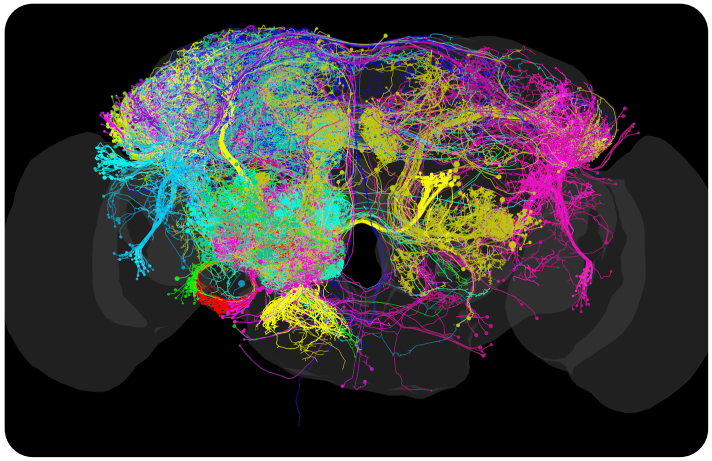


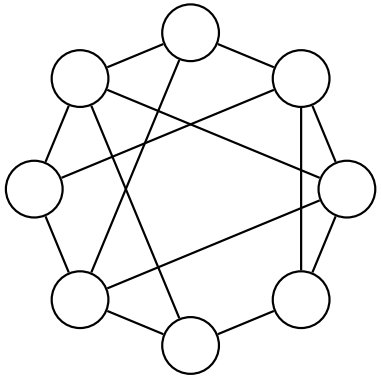
Les graphes sont partout

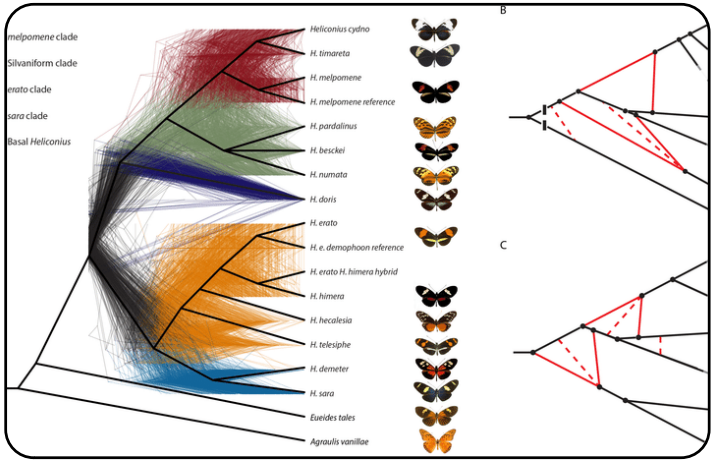




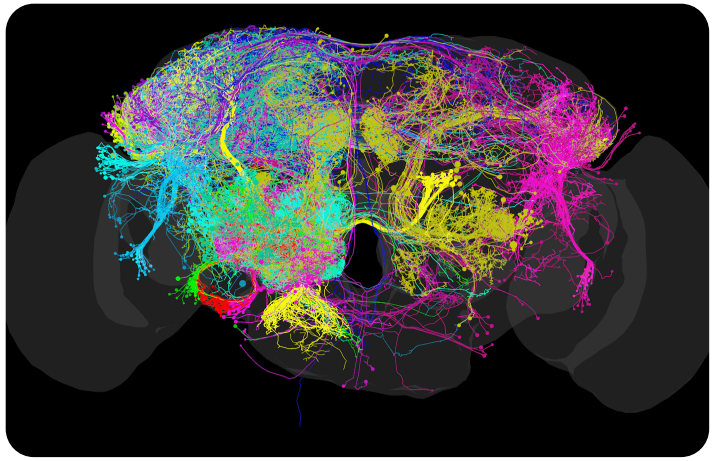
Les graphes sont partout

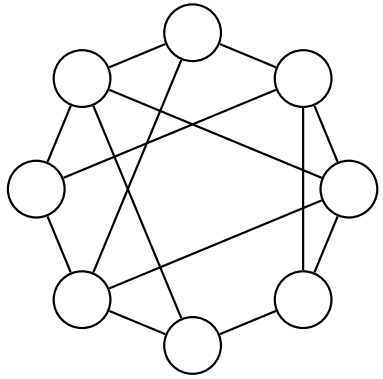


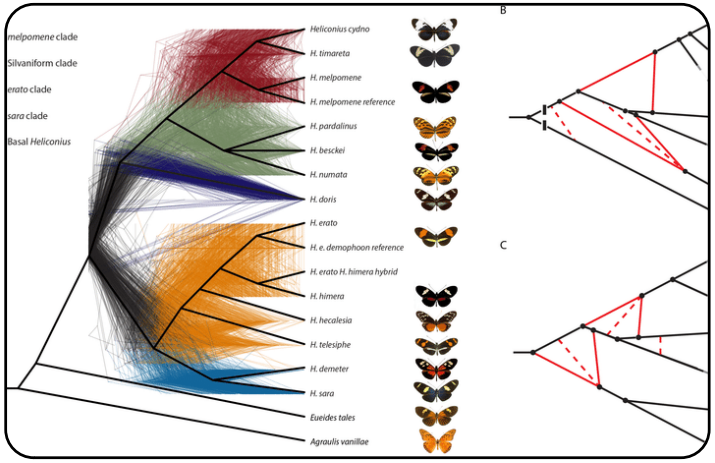




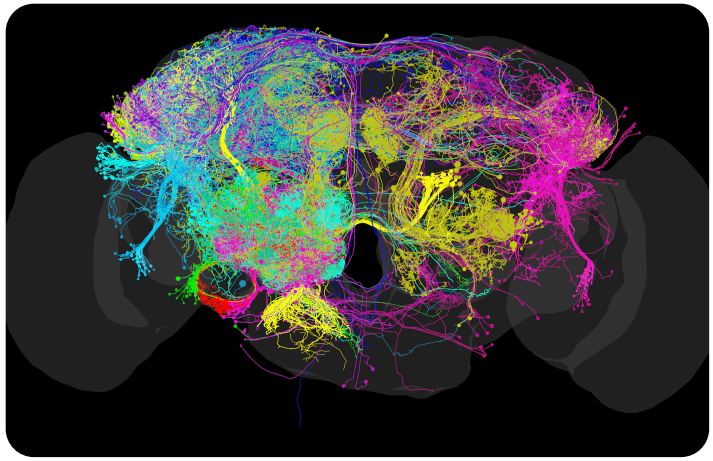
Les graphes sont partout

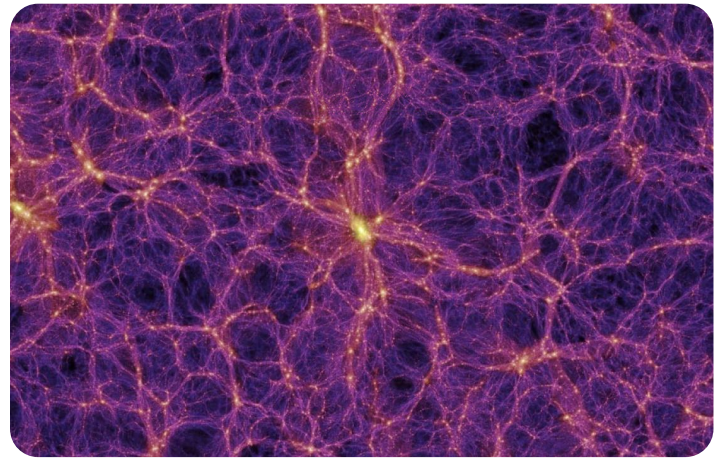


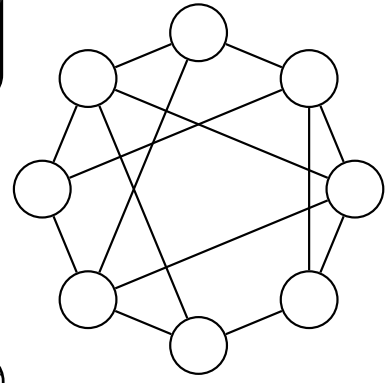


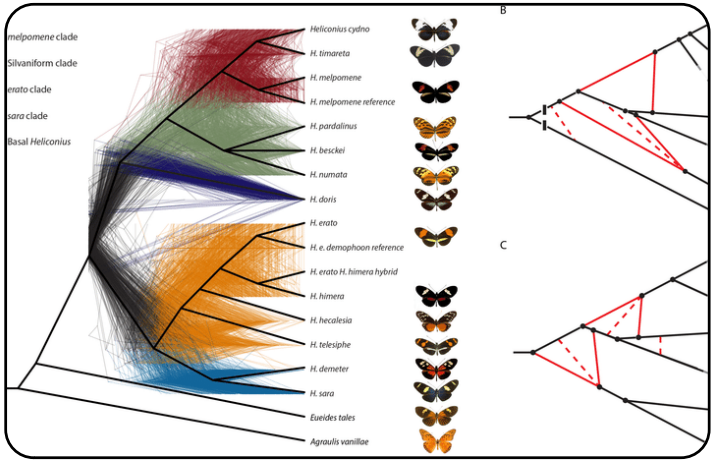


Les graphes sont partout



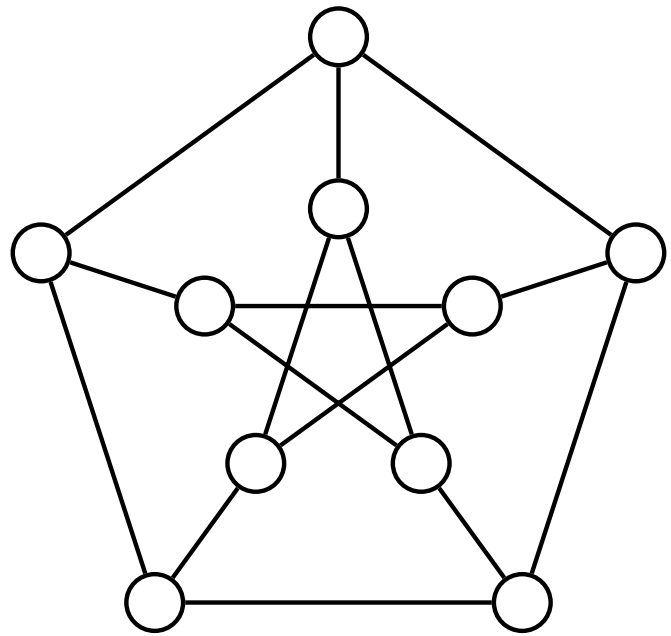






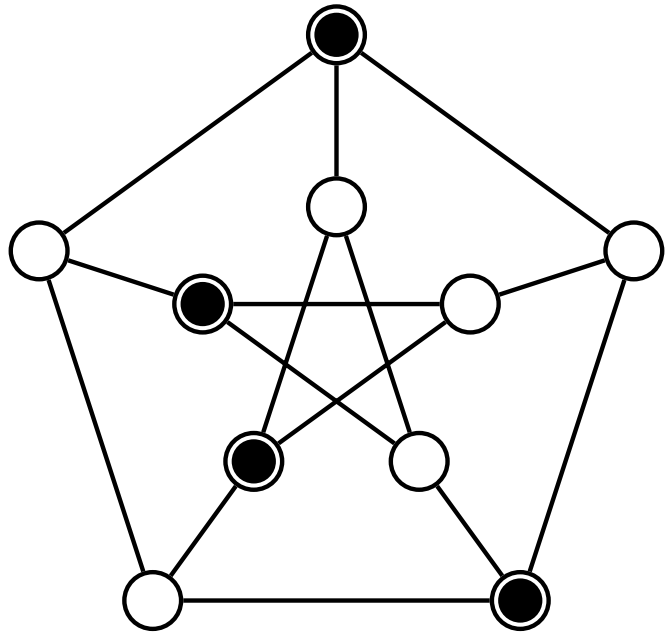
Problème du stable maximum

Trouver un plus grand ensemble de sommets 2 à 2 non adjacents



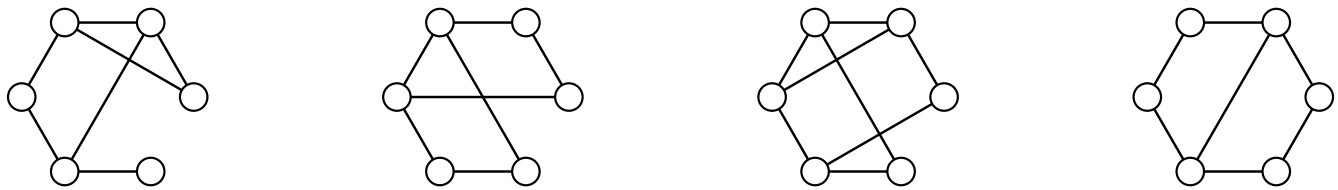
Problème du stable maximum

Trouver un plus grand ensemble de sommets 2 à 2 non adjacents



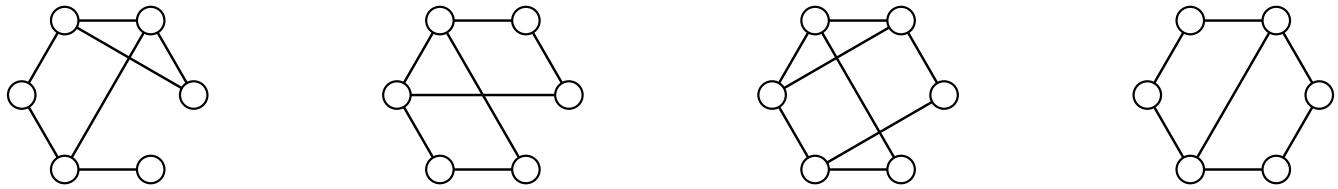
Stable maximum dans des graphes peu connectés

Supposons que l'entrée a $\frac{n}{s}$ parties indépendantes de taille s



Stable maximum dans des graphes peu connectés

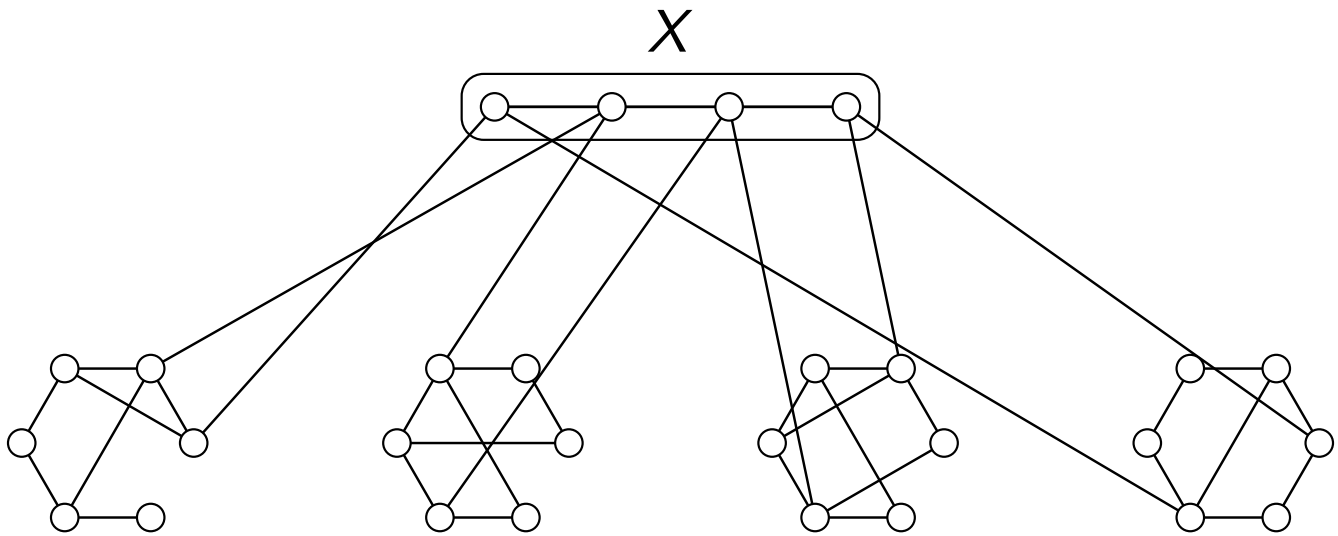
Supposons que l'entrée a $\frac{n}{s}$ parties indépendantes de taille s



Algorithme en temps $\frac{n}{s} \cdot T(s)$

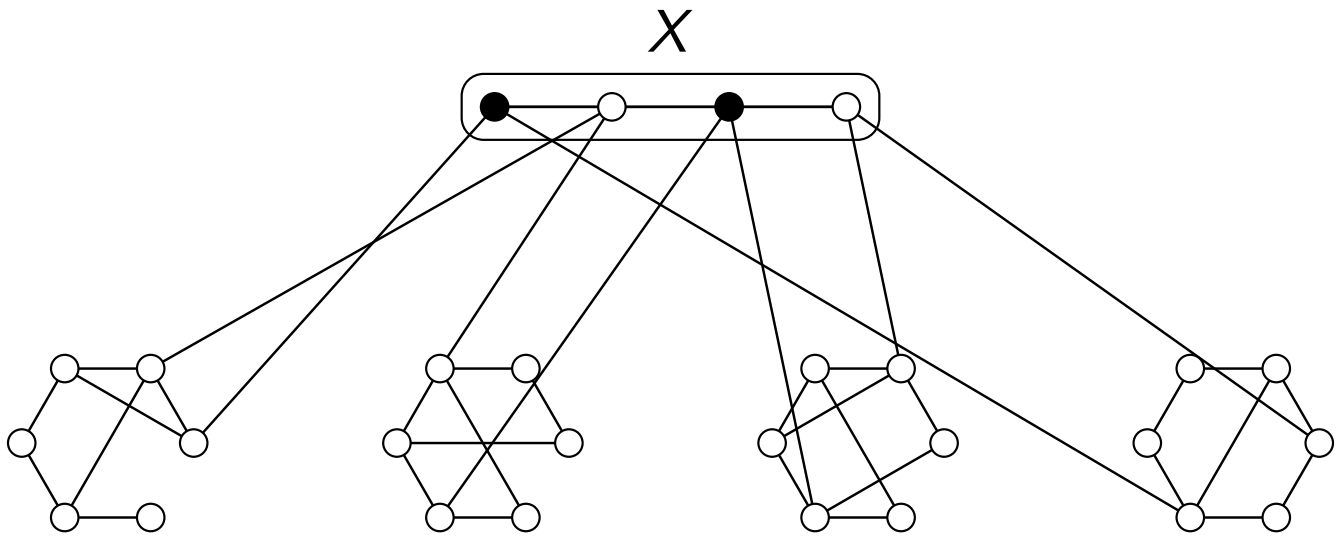
Stable maximum dans des graphes peu connectés

Supposons que l'entrée a $\frac{n}{s}$ parties indépendantes de taille s
$+$ un petit ensemble X de sommets



Stable maximum dans des graphes peu connectés

Supposons que l'entrée a $\frac{n}{s}$ parties indépendantes de taille s
$+$ un petit ensemble X de sommets



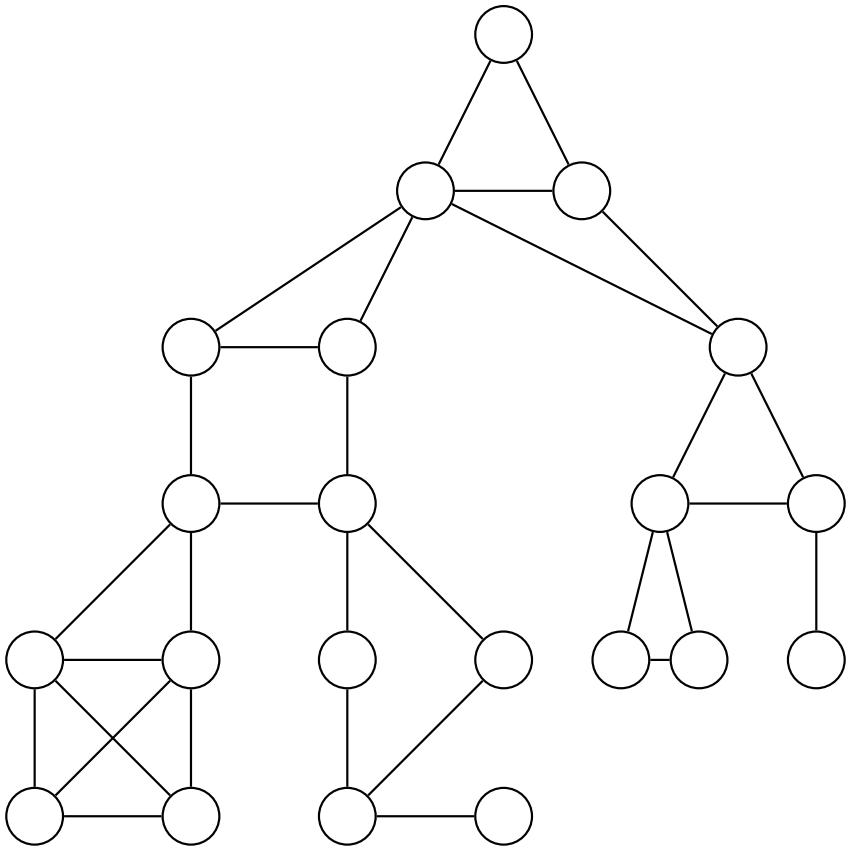
Algorithme en temps $2^{|X|} \cdot \frac{n}{s} \cdot T(s) = O_{|X|,s}(n)$

Stable maximum dans des graphes peu connectés

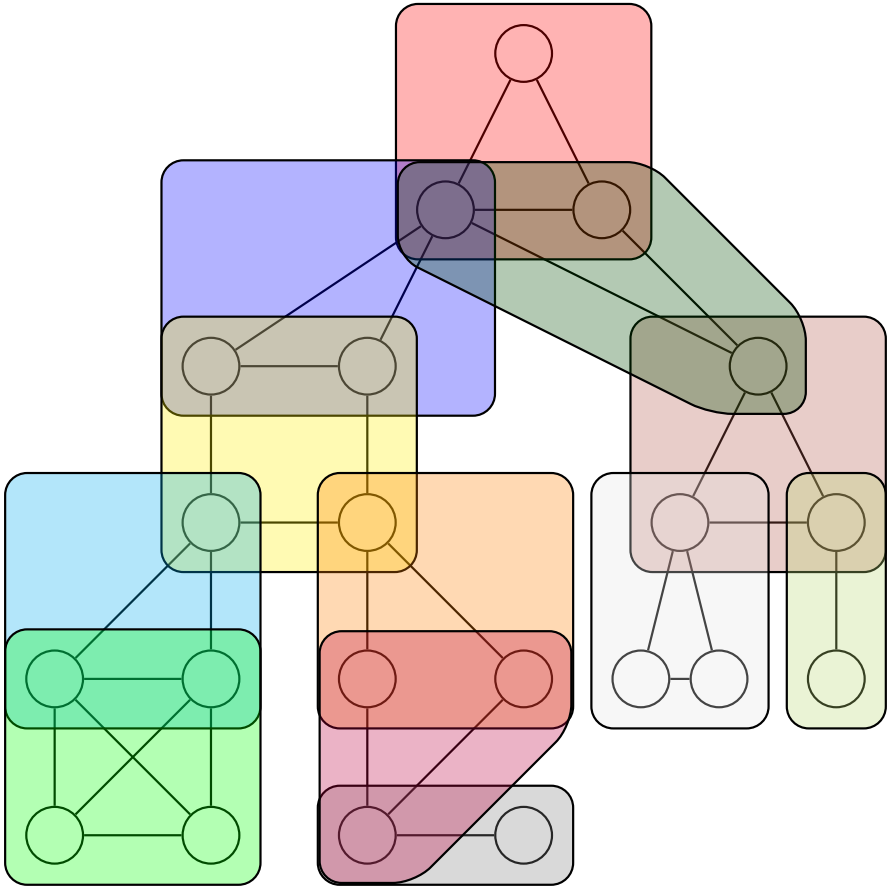
Supposons que l'entrée a $\frac{n}{s}$ parties indépendantes de taille s
$+$ un petit ensemble X de sommets

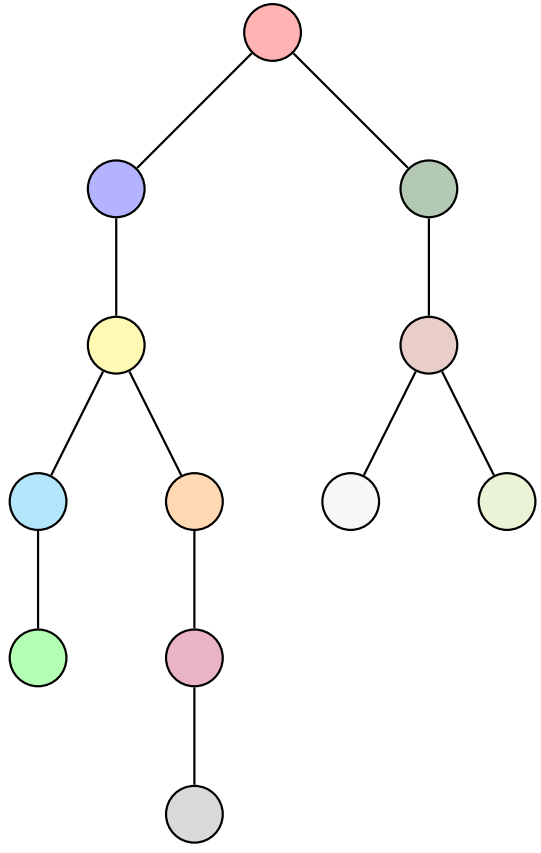
Algorithme en temps $2^{|X|} \cdot \frac{n}{s} \cdot T(s) = O_{|X|,s}(n)$

Décomposition arborescente

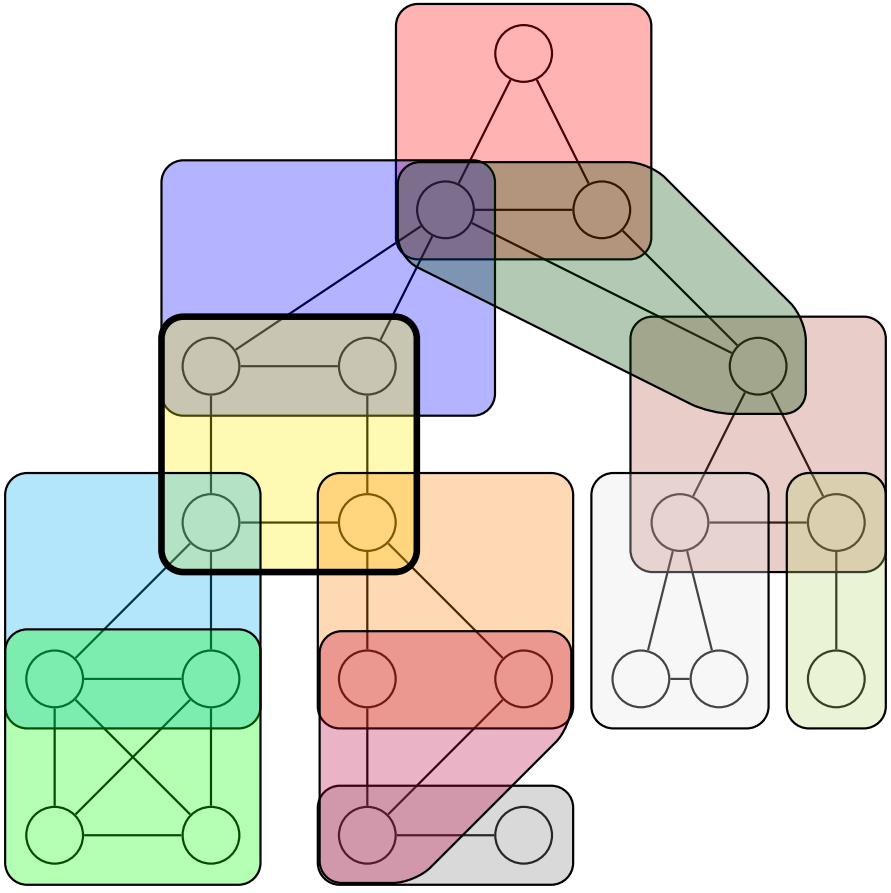


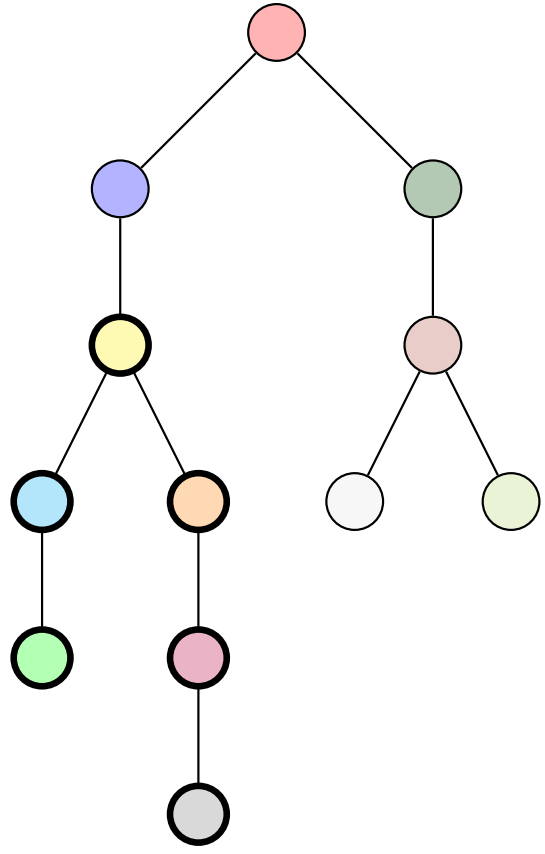
Décomposition arborescente

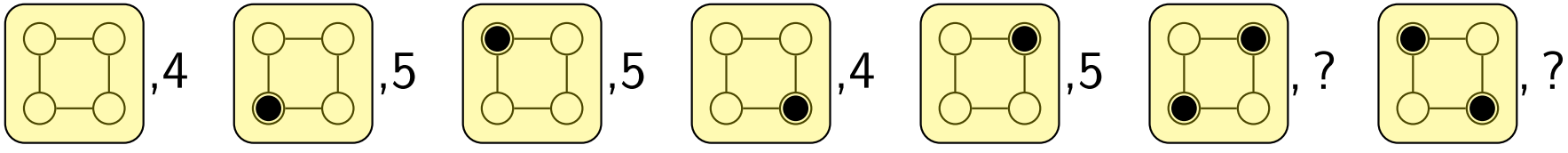




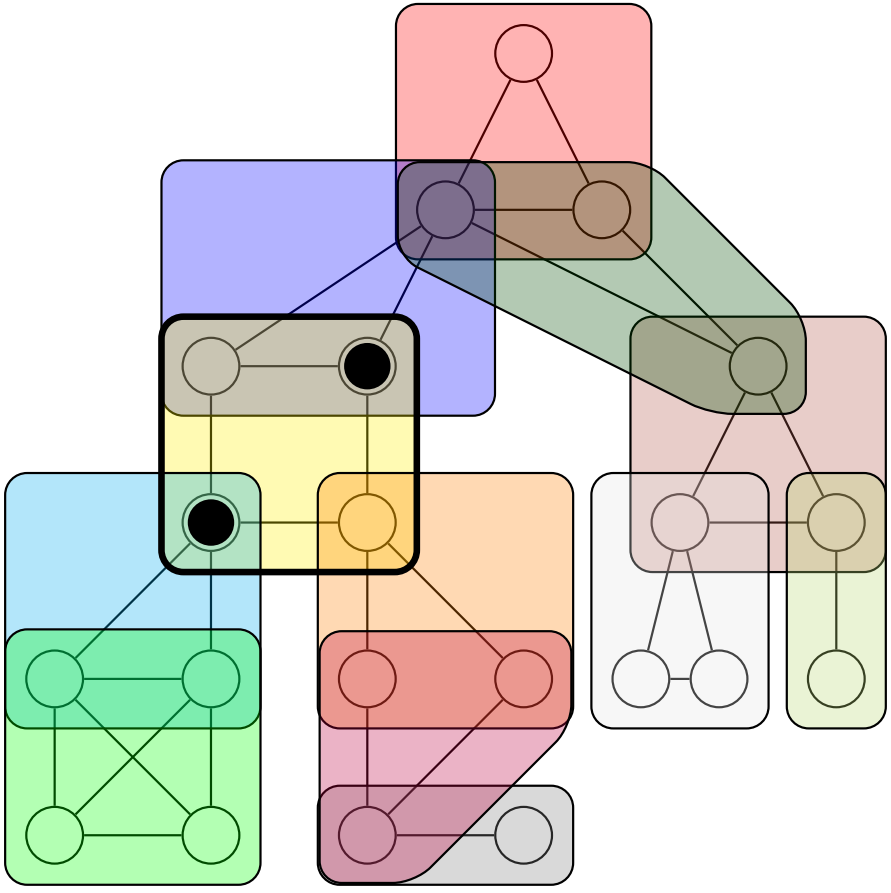
Décomposition arborescente : résoudre le stable maximum

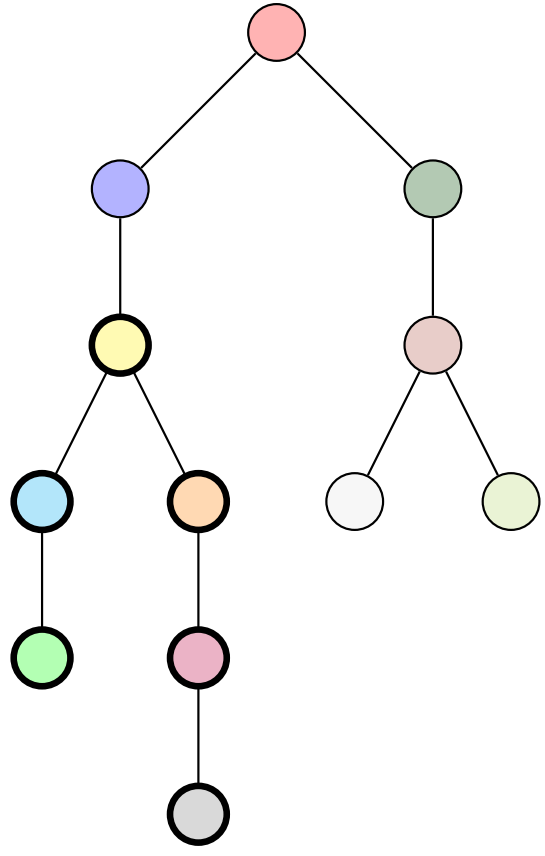


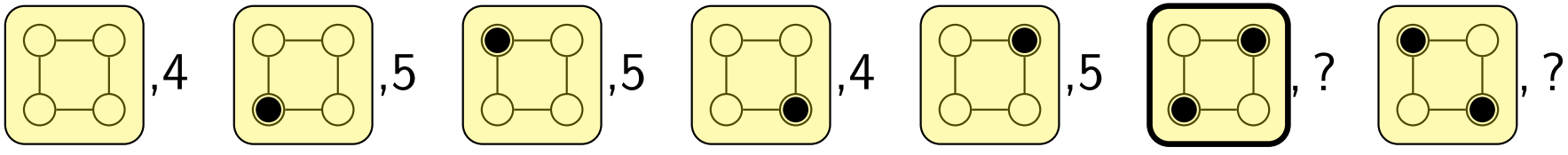




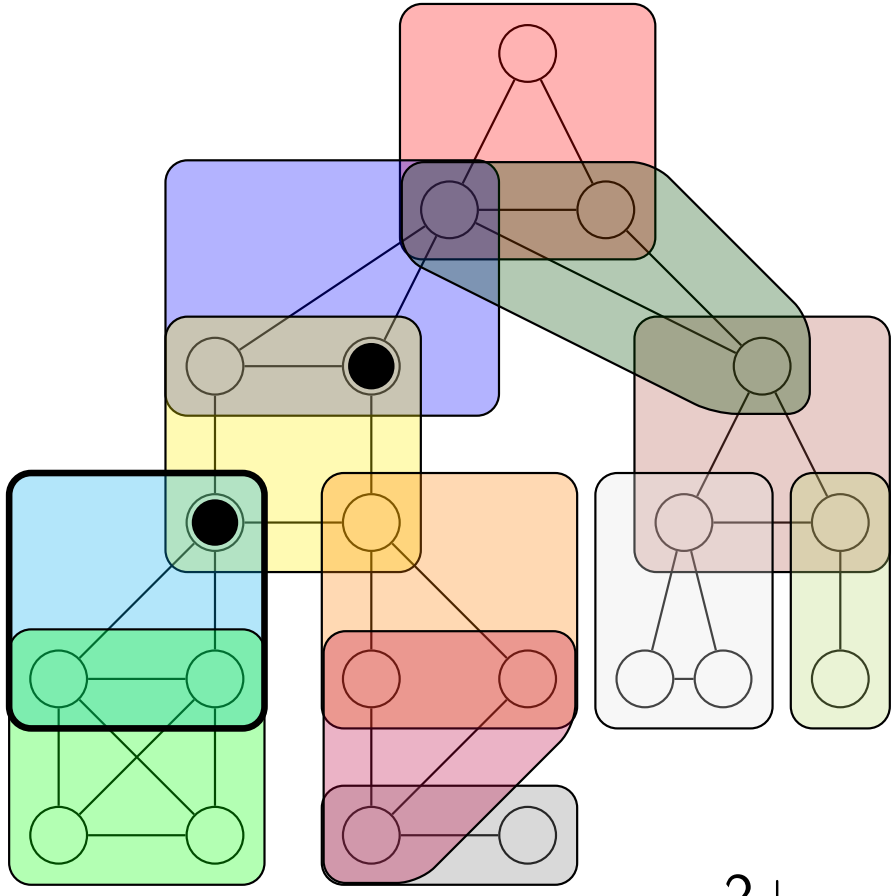
Décomposition arborescente : résoudre le stable maximum

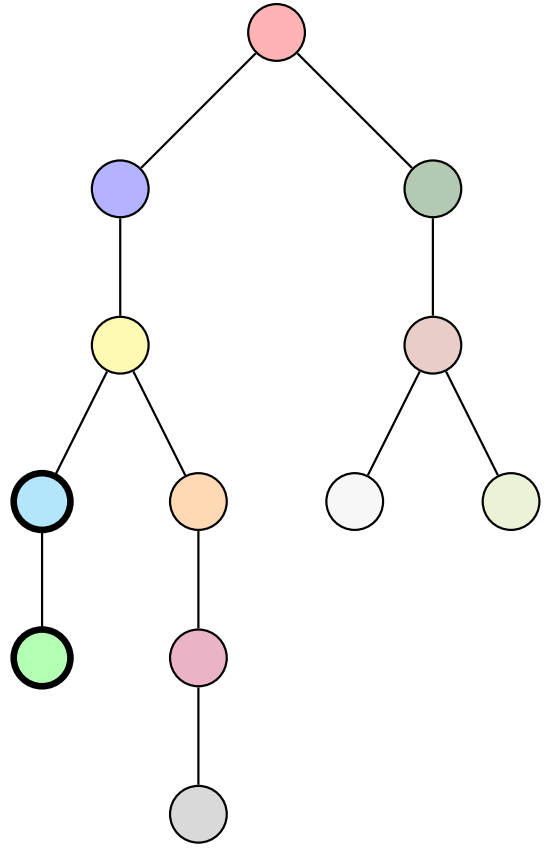


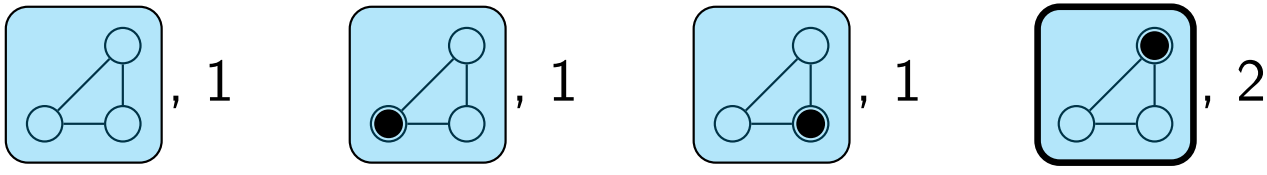




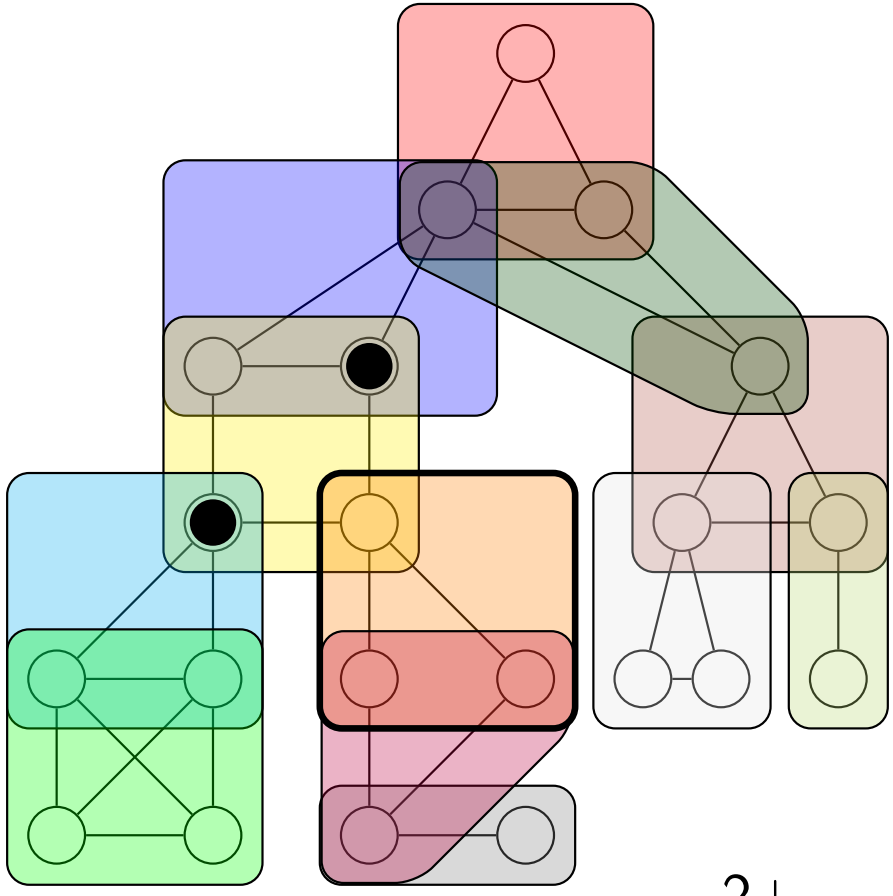
Décomposition arborescente : résoudre le stable maximum

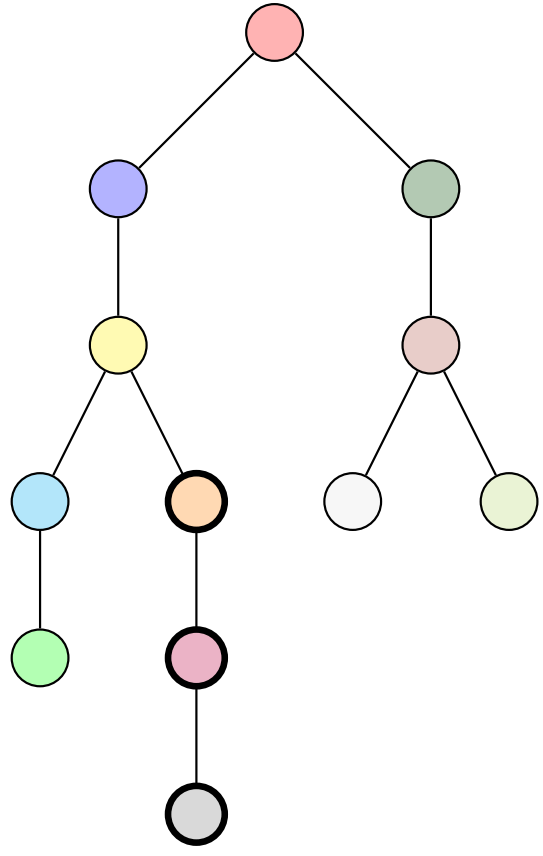




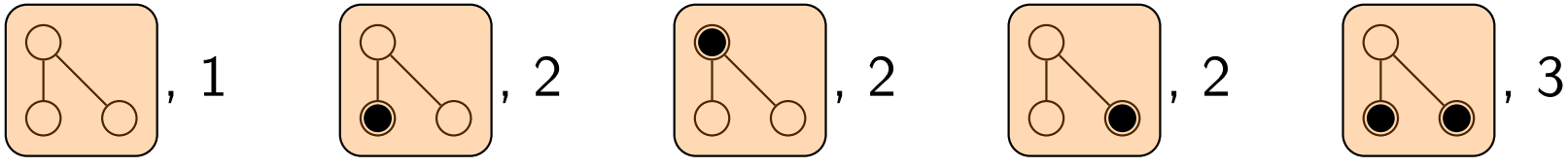


Décomposition arborescente : résoudre le stable maximum

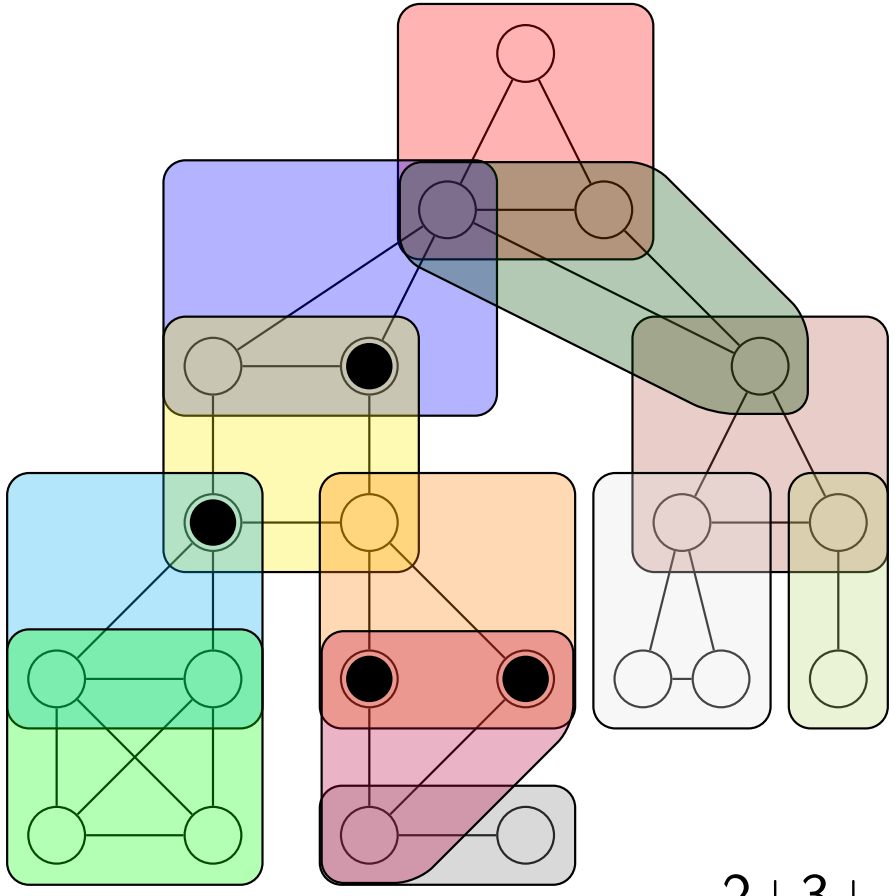


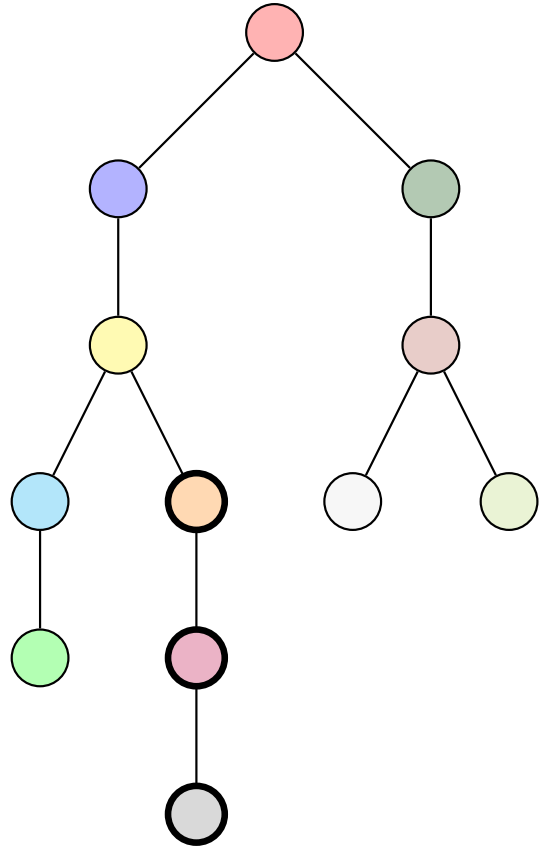


2+

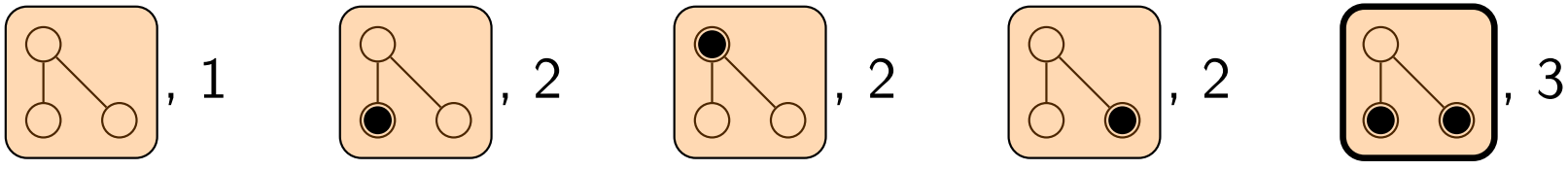


Décomposition arborescente : résoudre le stable maximum

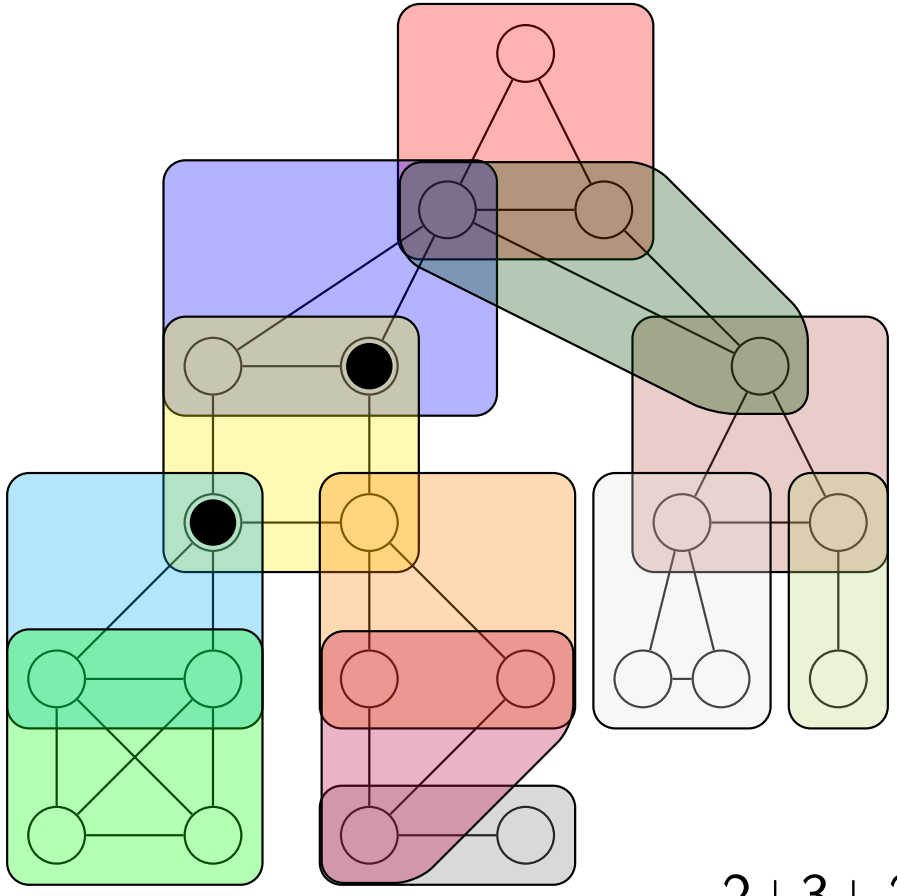


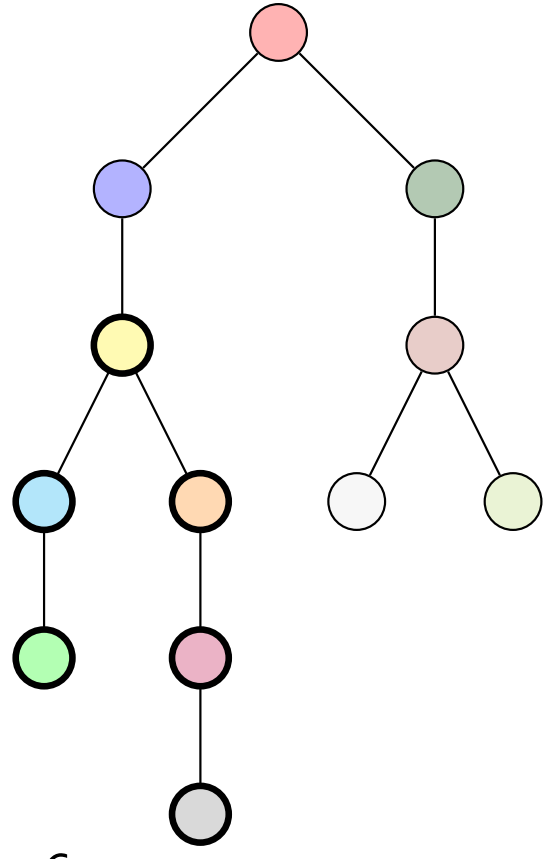


2+3+

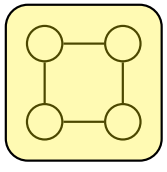
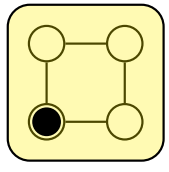
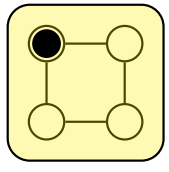
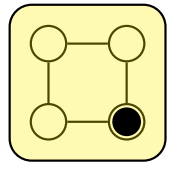
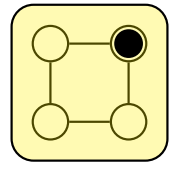
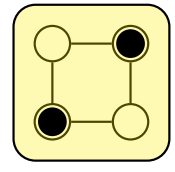
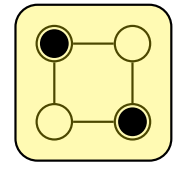


Décomposition arborescente : résoudre le stable maximum





$2+3+2-1=6$


,4

,5

,5

,4

,5

,**6**

, ?

Logiques sur les graphes

Premier ordre : relation d'adjacence $E(\cdot, \cdot)$ et $=$,
connecteurs booléens, quantification sur les sommets

Exemple :
$\varphi_{\triangle} := \exists x \exists y \exists z \; x \neq y \wedge x \neq z \wedge y \neq z \wedge E(x, y) \wedge E(x, z) \wedge E(y, z)$

Logiques sur les graphes

Premier ordre : relation d'adjacence $E(\cdot, \cdot)$ et $=$,
connecteurs booléens, quantification sur les sommets

Exemple :
$\varphi_{\triangle} := \exists x \exists y \exists z \; x \neq y \wedge x \neq z \wedge y \neq z \wedge E(x, y) \wedge E(x, z) \wedge E(y, z)$

Monadique du second ordre (MSO) : Premier ordre $+$
quantification sur des ensembles de sommets et appartenance

Logiques sur les graphes

Premier ordre : relation d'adjacence $E(\cdot, \cdot)$ et $=$,
connecteurs booléens, quantification sur les sommets

Exemple :
$\varphi_{\triangle} := \exists x \exists y \exists z \; x \neq y \wedge x \neq z \wedge y \neq z \wedge E(x, y) \wedge E(x, z) \wedge E(y, z)$

Monadique du second ordre (MSO) : Premier ordre $+$
quantification sur des ensembles de sommets et appartenance

Exemple : $\varphi_{\text{3-col}} := \exists X \exists Y \exists Z \; \forall x \; (x \in X \vee x \in Y \vee x \in Z)$

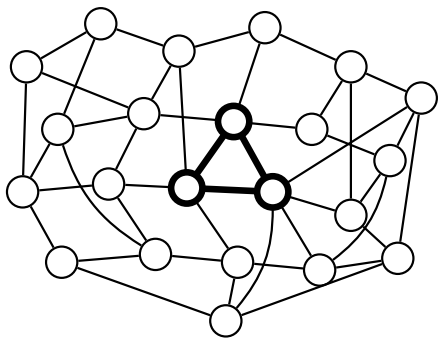
$$\wedge \; \text{stable}(X) \wedge \text{stable}(Y) \wedge \text{stable}(Z)$$

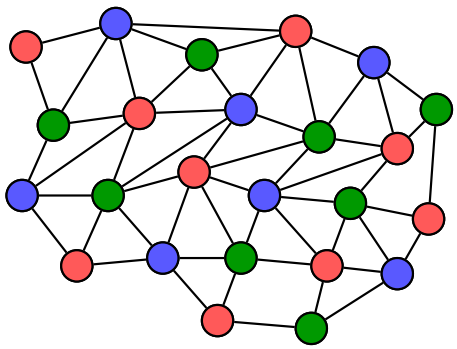
avec $\text{stable}(X) := \forall x \forall y \; (x \in X \wedge y \in X) \Rightarrow \neg E(x, y)$

Problèmes définis par des logiques

Une formule close exprime le problème

Exemple : existence d'un triangle ($\varphi_{\triangle}$), 3-coloration ($\varphi_{\text{3-col}}$), etc.

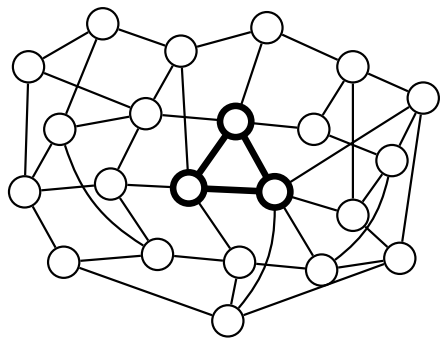


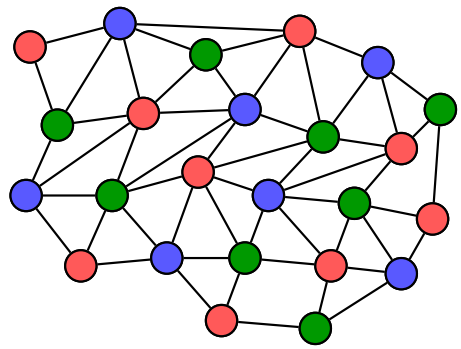


Problèmes définis par des logiques

Une formule close exprime le problème

Exemple : existence d'un triangle ($\varphi_{\triangle}$), 3-coloration ($\varphi_{\text{3-col}}$), etc.





Mais aussi : maximiser/minimiser la taille de X tel que $\varphi(X)$

Exemple : stable maximum (maximiser X tel que stable(X))

Méta-algorithmes

Théorème (Courcelle '90)

Tout problème exprimable en logique MSO a un algorithme linéaire sur les instances de largeur arborescente fixée.

Méta-algorithmes

Théorème (Courcelle '90)

Tout problème exprimable en logique MSO a un algorithme linéaire sur les instances de largeur arborescente fixée.

Théorème (Courcelle–Makowsky–Rotics '00)

Tout problème exprimable en logique MSO a un algorithme polynomial sur les instances de largeur de clique fixée.

On ne peut pas aller plus loin que la largeur de clique bornée

Méta-algorithmes

Théorème (Courcelle '90)

Tout problème exprimable en logique MSO a un algorithme linéaire sur les instances de largeur arborescente fixée.

Théorème (Courcelle–Makowsky–Rotics '00)

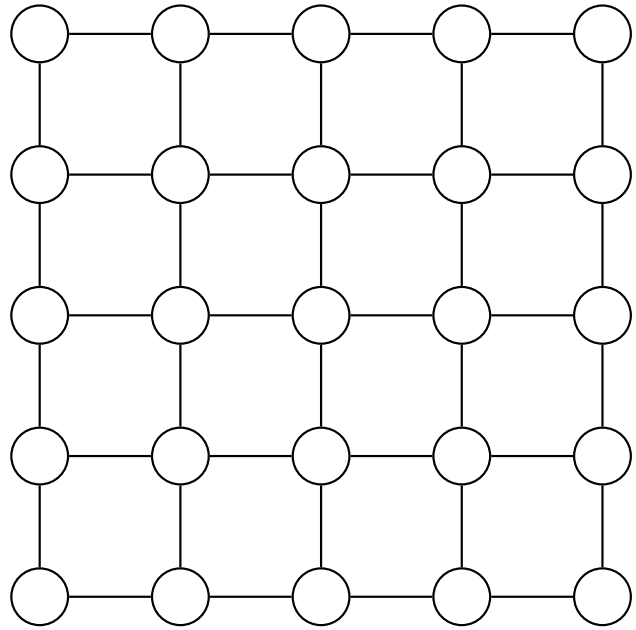
Tout problème exprimable en logique MSO a un algorithme polynomial sur les instances de largeur de clique fixée.

On ne peut pas aller plus loin que la largeur de clique bornée

Et la logique du premier ordre ?

La twin-width : une décomposition dynamique

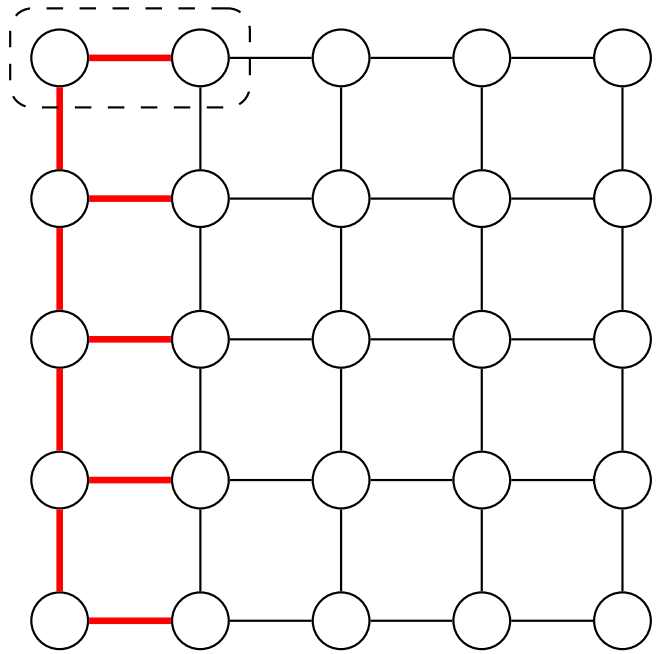
trigraphe : graphe avec deux types d'arêtes, noires et rouges



La twin-width : une décomposition dynamique

trigraphe : graphe avec deux types d'arêtes, noires et rouges

suite de contractions :
suite de trigraphes où tour à tour deux sommets sont unis

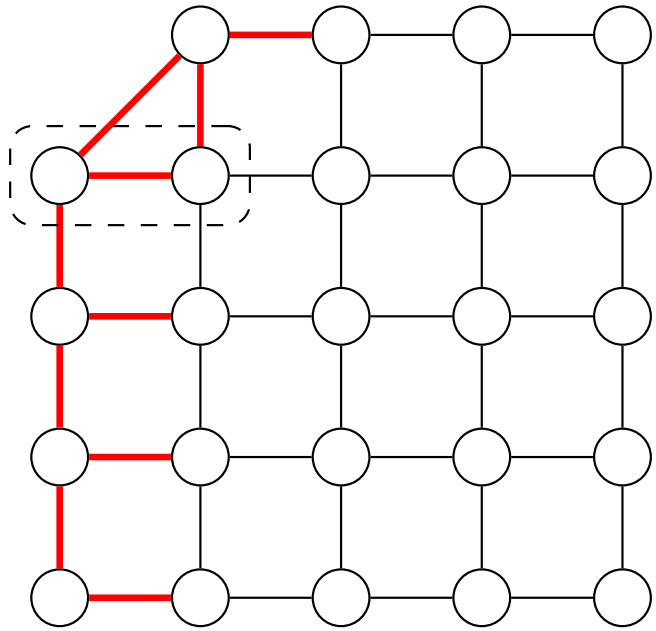


La twin-width : une décomposition dynamique

trigraphe : graphe avec deux types d'arêtes, noires et rouges

suite de contractions :
suite de trigraphes où tour à tour deux sommets sont unis

twin-width :
plus petit d tel que le (tri)graphe a une suite de contractions avec degré rouge au plus d

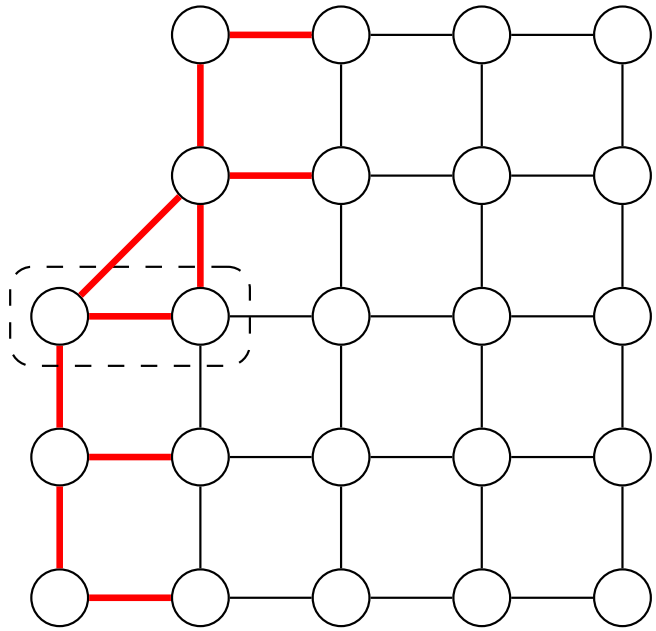


La twin-width : une décomposition dynamique

trigraphe : graphe avec deux types d'arêtes, noires et rouges

suite de contractions :
suite de trigraphes où tour à tour deux sommets sont unis

twin-width :
plus petit d tel que le (tri)graphe a une suite de contractions avec degré rouge au plus d

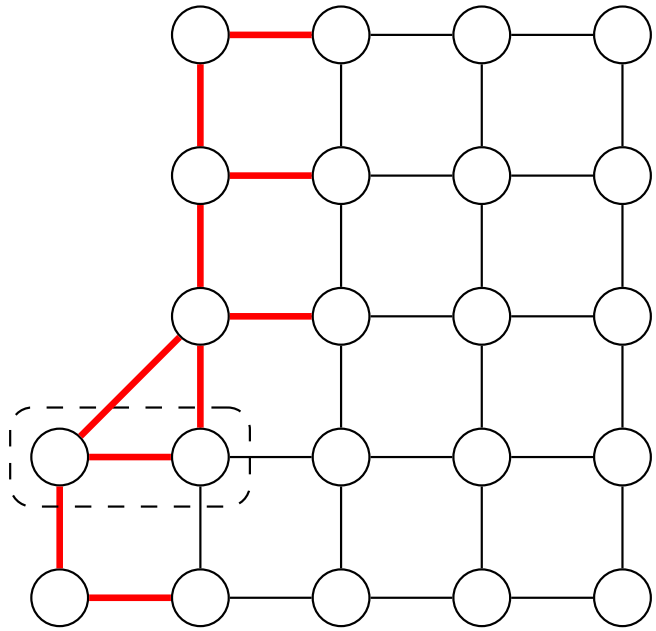


La twin-width : une décomposition dynamique

trigraphe : graphe avec deux types d'arêtes, noires et rouges

suite de contractions :
suite de trigraphes où tour à tour deux sommets sont unis

twin-width :
plus petit d tel que le (tri)graphe a une suite de contractions avec degré rouge au plus d

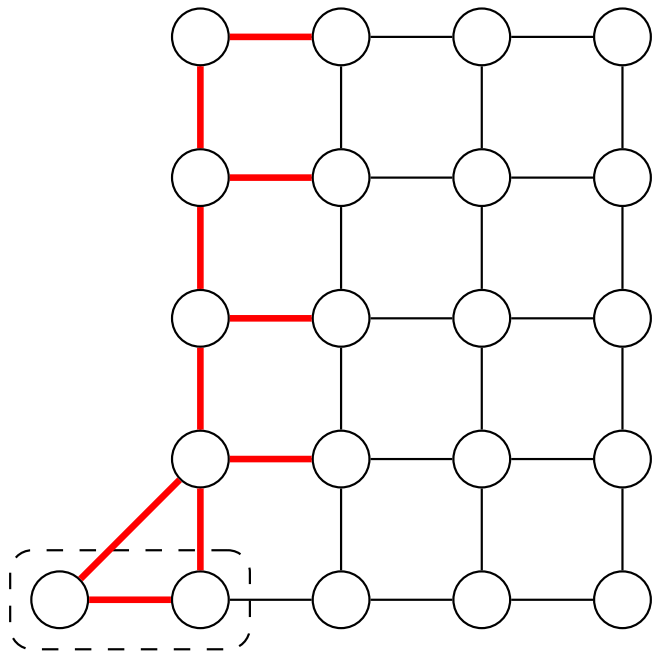


La twin-width : une décomposition dynamique

trigraphe : graphe avec deux types d'arêtes, noires et rouges

suite de contractions :
suite de trigraphes où tour à tour deux sommets sont unis

twin-width :
plus petit d tel que le (tri)graphe a une suite de contractions avec degré rouge au plus d

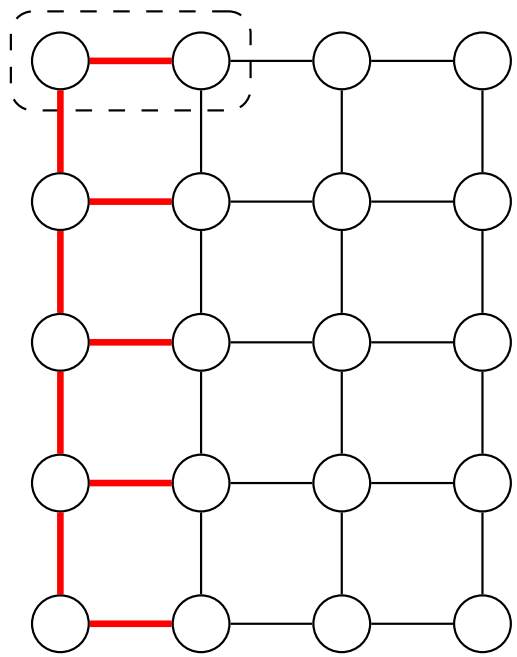


La twin-width : une décomposition dynamique

trigraphe : graphe avec deux types d'arêtes, noires et rouges

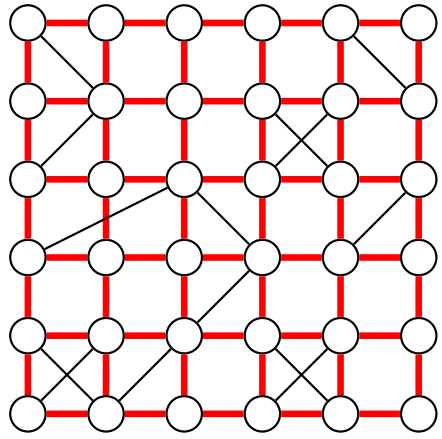
suite de contractions :
suite de trigraphes où tour à tour deux sommets sont unis

twin-width :
plus petit d tel que le (tri)graphe a une suite de contractions avec degré rouge au plus d

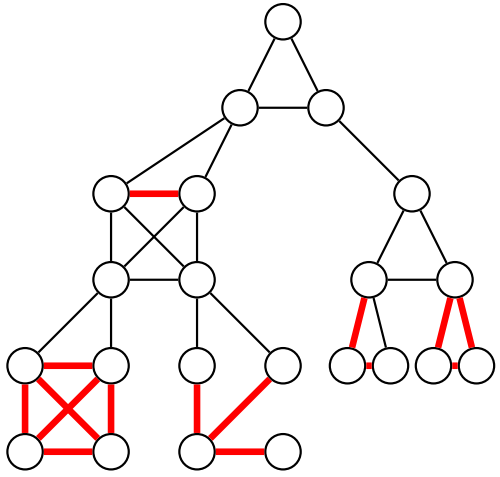


Paramètres réduits

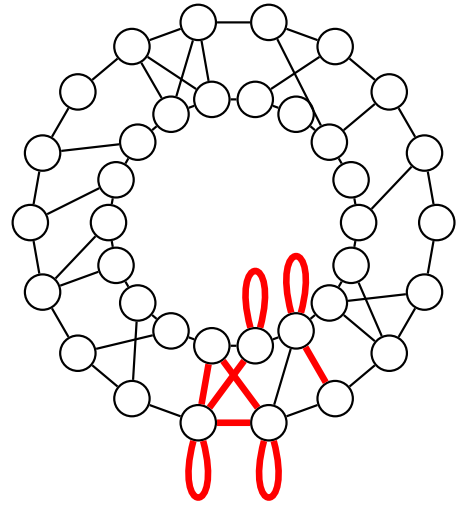
Autres conditions sur les graphes rouges



degré maximum réduit = twin-width



taille de composante réduite ≡ largeur de clique

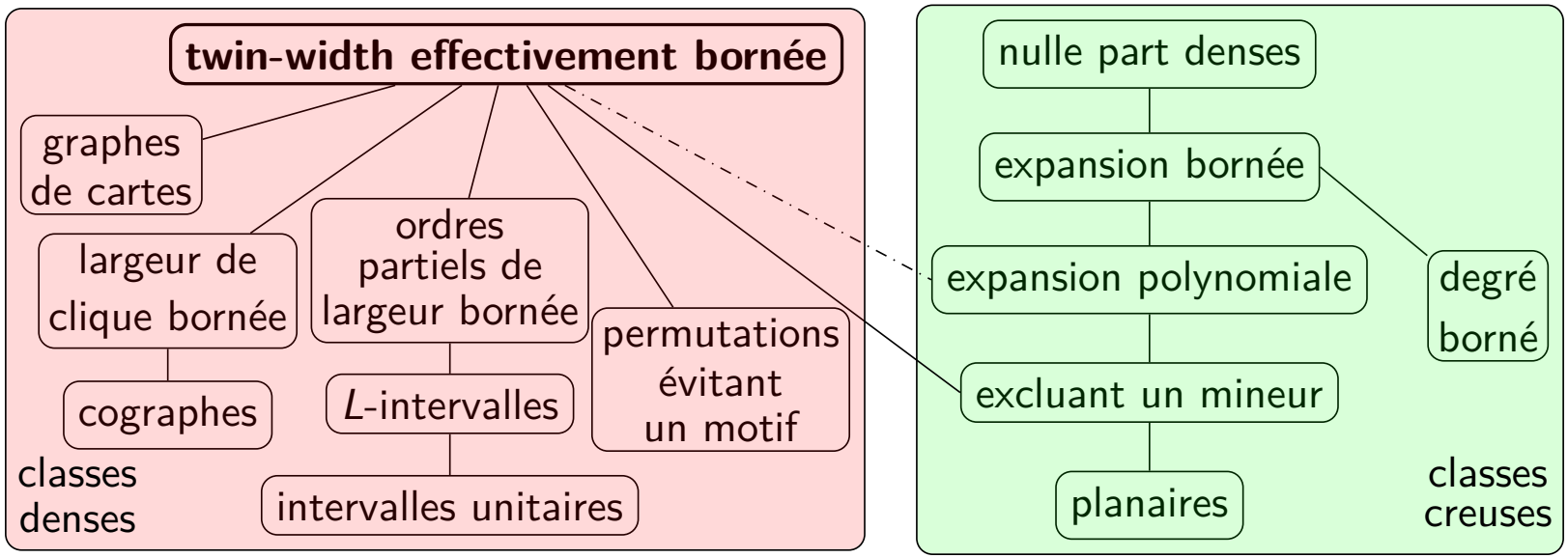


nombre d'arêtes réduit

Méta-algorithmes

Théorème (B., Kim, Thomassé, Watrigant '20)

Tout problème en logique du premier ordre a un algorithme linéaire sur les graphes de twin-width effectivement bornée.



Méta-algorithmes

Théorème (B., Kim, Thomassé, Watrigant '20)

Tout problème en logique du premier ordre a un algorithme linéaire sur les graphes de twin-width effectivement bornée.

Reformulation du théorème de Courcelle–Makowsky–Rotics :

Théorème (B., Kim, Reinald, Thomassé '22)

Tout problème en logique MSO a un algorithme polynomial sur les graphes de taille de composante réduite bornée.

D'autres applications algorithmiques

La twin-width s'étend aux matrices

Théorème (B., Geniet, Kim, Moon '26)

On peut calculer le produit d'une matrice $n \times n$ de twin-width bornée avec une autre matrice en temps $O(n^2 \log n)$.

Théorème (B., Duron, Pilipczuk, Sokołowski, Toruńczyk '26+)

La distance entre chaque paire de sommets peut être calculée en temps $O(n^2)$ dans un graphe de twin-width bornée à n sommets.

D'autres applications algorithmiques

La twin-width s'étend aux matrices

Théorème (B., Geniet, Kim, Moon '26)

On peut calculer le produit d'une matrice $n \times n$ de twin-width bornée avec une autre matrice en temps $O(n^2 \log n)$.

Théorème (B., Duron, Pilipczuk, Sokołowski, Toruńczyk '26+)

La distance entre chaque paire de sommets peut être calculée en temps $O(n^2)$ dans un graphe de twin-width bornée à n sommets.

Avec une suite de contractions, on peut aussi :
approcher le stable maximum et la coloration minimum avec de meilleurs ratios, lister les triangles en temps linéaire, etc.

Perspectives

- Calcul efficace des suites de contractions
- Trouver le paramètre le plus général pour la logique du premier ordre (flip-width, merge-width)
- Applications en théorie des graphes
- Étendre la twin-width à des relations d'ordre supérieur

Perspectives

- Calcul efficace des suites de contractions
- Trouver le paramètre le plus général pour la logique du premier ordre (flip-width, merge-width)
- Applications en théorie des graphes
- Étendre la twin-width à des relations d'ordre supérieur

Merci pour votre attention !