Coloration de graphes parfaits sans Balanced Skew Partition

Théophile Trunck

JGA 2012

Novembre 2012

Co-Auteurs

Travail avec :

- Maria Chudnovsky, Columbia University, New York
- Nicolas Trotignon, CNRS, LIP, ENS Lyon
- Kristina Vušković, Union University, Belgrade and Leeds University

Définitions

- Un graphe G est *de Berge* si dans G et dans $\overline{G}$, tous les trous (i.e. les cycles sans cordes de taille au moins 4) sont impairs.
- Un graphe G est *parfait* si tous les sous graphes induit H de G vérifient $\chi(H) = \omega(H)$.

Théorème (Chudnovsky, Robertson, Seymour and Thomas, 2002)

Les graphes de Berge sont parfaits.

Théorème (Gröstchel, Lovász, and Schrijver, 1980's)

Il existe un algorithme polynomial pour colorier les graphes parfaits

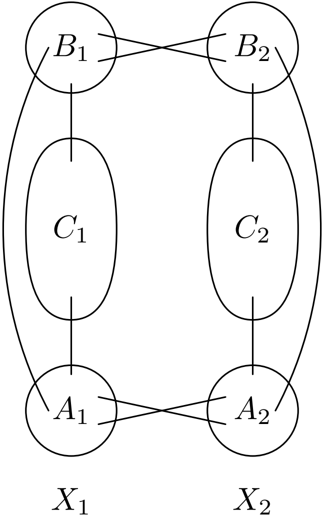
Décompositions

Théorème (Chudnovsky, Robertson, Seymour and Thomas, 2002)

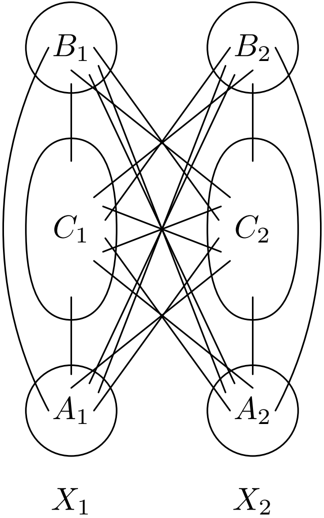
Tout graphe de Berge est basique ou admet une décomposition

Décompositions :
2-join, $\overline{\text{2-join}}$, Paire Homogène, Balanced Skew Partition

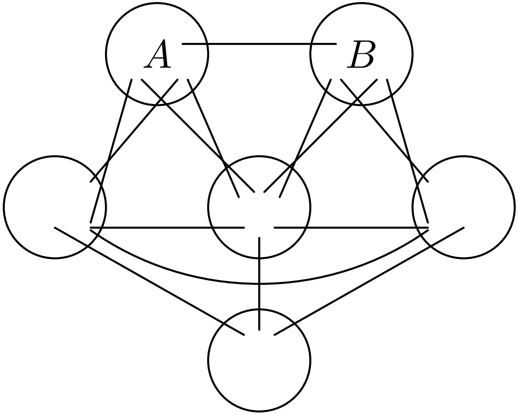
2-join



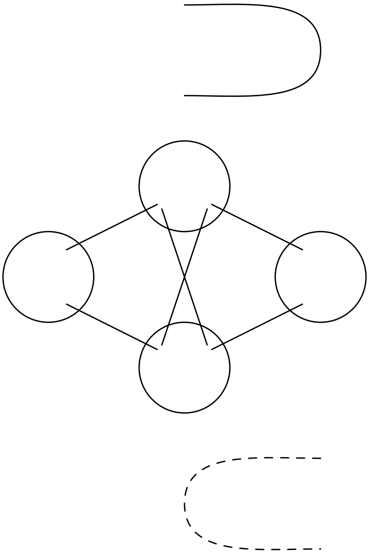
2-join



Paire Homogène



Balanced Skew Partition



Résultat principal

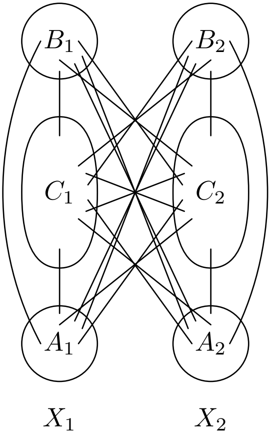
Théorème (Chudnovsky, Trotignon, T.T., Vušković, 2011)

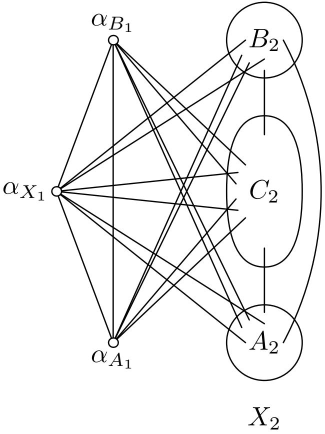
Il existe un algorithme en $\mathcal{O}(n^7)$ pour colorier les graphes de Berge sans balanced skew partition.

Remarques

- (Grötschel, Lovász and Schrijver) Si on sait calculer le stable max pondéré et la clique max pondérée on sait colorier.
- Notre classe est auto-complementaire.
- On cherche donc seulement à calculer le stable max pondéré.

Utiliser les décompositions





Décompositions extrêmes

- On cherche a construire un bloc sur lequel tous les calculs sont polynomiaux.
- Un bloc basique est un bon candidat

Théorème

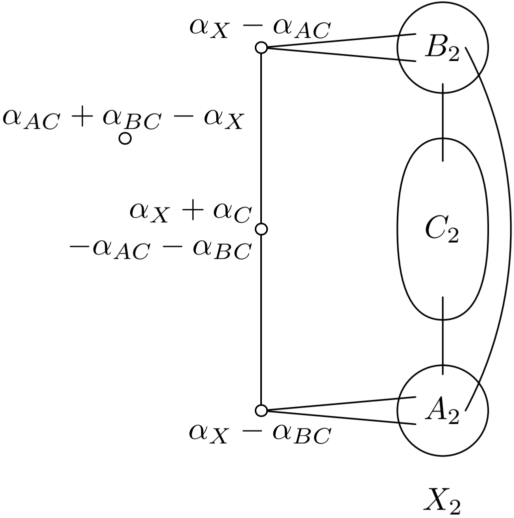
Un bloc minimal (qui ne contient pas d'autre bloc dans G) est basique.

- Remarquez que ce résultat est faux en général.
- On sait trouver un bloc minimal en $\mathcal{O}(n^5)$

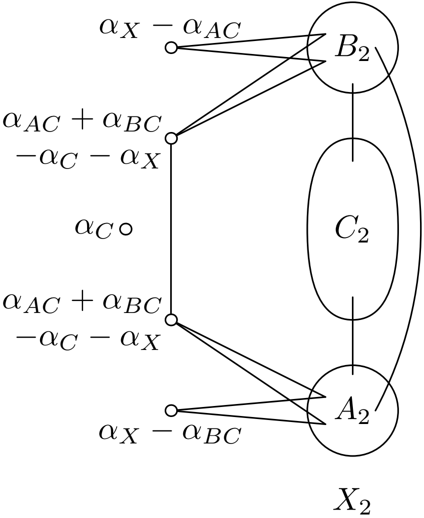
Résultats sur les classes de bases

Classe	Reconnaissance	Stable max
Bipartite	$\mathcal{O}(n)$	$\mathcal{O}(n^3)$
$\overline{\text{Bipartite}}$	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$
Line BIP	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$
$\overline{\text{Line BIP}}$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$
Double Split	$\mathcal{O}(n^5)$	$\mathcal{O}(n^5)$

Calculer α avec les 2-joins pairs



Calculer α avec les 2-joins impairs



Magic Lemma

Lemme

Soit (X_1, X_2) *un 2-join.*

- *si* (X_1, X_2) *est pair alors* $\alpha_X + \alpha_C \geq \alpha_{AC} + \alpha_{BC}$
- *si* (X_1, X_2) *est impair alors* $\alpha_X + \alpha_C \leq \alpha_{AC} + \alpha_{BC}$

2-join impair, $\alpha_C + \alpha_X \leq \alpha_{AC} + \alpha_{BC}$

Démonstration.

- D (C) stable max de X_1 (C_1)

□

2-join impair, $\alpha_C + \alpha_X \leq \alpha_{AC} + \alpha_{BC}$

Démonstration.

- D (C) stable max de X_1 (C_1)
- $C \cup D$ est biparti

$\square$

2-join impair, $\alpha_C + \alpha_X \leq \alpha_{AC} + \alpha_{BC}$

Démonstration.

- D (C) stable max de X_1 (C_1)
- $C \cup D$ est biparti
- Y_A (Y_B) les sommets connexes à A (B) dans $C \cup D$

$\square$

2-join impair, $\alpha_C + \alpha_X \leq \alpha_{AC} + \alpha_{BC}$

Démonstration.

- D (C) stable max de X_1 (C_1)
- $C \cup D$ est biparti
- Y_A (Y_B) les sommets connexes à A (B) dans $C \cup D$
- P plus court chemin de A à B dans $C \cup D$

□

2-join impair, $\alpha_C + \alpha_X \leq \alpha_{AC} + \alpha_{BC}$

Démonstration.

- D (C) stable max de X_1 (C_1)
- $C \cup D$ est biparti
- Y_A (Y_B) les sommets connexes à A (B) dans $C \cup D$
- P plus court chemin de A à B dans $C \cup D$
- P est pair absurde

$\square$

2-join impair, $\alpha_C + \alpha_X \leq \alpha_{AC} + \alpha_{BC}$

Démonstration.

- D (C) stable max de X_1 (C_1)
- $C \cup D$ est biparti
- Y_A (Y_B) les sommets connexes à A (B) dans $C \cup D$
- P plus court chemin de A à B dans $C \cup D$
- P est pair absurde
- $Y_A \cap Y_B = \emptyset$ et Y_A anticomplet à Y_B

□

2-join impair, $\alpha_C + \alpha_X \leq \alpha_{AC} + \alpha_{BC}$

Démonstration.

- D (C) stable max de X_1 (C_1)
- $C \cup D$ est biparti
- Y_A (Y_B) les sommets connexes à A (B) dans $C \cup D$
- P plus court chemin de A à B dans $C \cup D$
- P est pair absurde
- $Y_A \cap Y_B = \emptyset$ et Y_A anticomplet à Y_B
- $Z_A = (D \cap Y_A) \cup (C \cap Y_B) \cup (C \setminus (Y_A \cup Y_B))$

□

2-join impair, $\alpha_C + \alpha_X \leq \alpha_{AC} + \alpha_{BC}$

Démonstration.

- D (C) stable max de X_1 (C_1)
- $C \cup D$ est biparti
- Y_A (Y_B) les sommets connexes à A (B) dans $C \cup D$
- P plus court chemin de A à B dans $C \cup D$
- P est pair absurde
- $Y_A \cap Y_B = \emptyset$ et Y_A anticomplet à Y_B
- $Z_A = (D \cap Y_A) \cup (C \cap Y_B) \cup (C \setminus (Y_A \cup Y_B))$
- $Z_B = (D \cap Y_B) \cup (C \cap Y_A) \cup (D \setminus (Y_A \cup Y_B))$

□

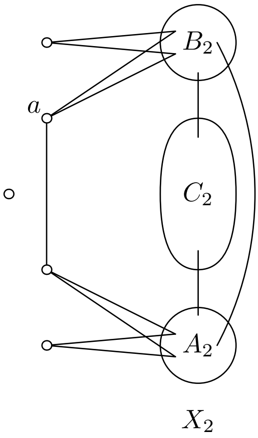
2-join impair, $\alpha_C + \alpha_X \leq \alpha_{AC} + \alpha_{BC}$

Démonstration.

- D (C) stable max de X_1 (C_1)
- $C \cup D$ est biparti
- Y_A (Y_B) les sommets connexes à A (B) dans $C \cup D$
- P plus court chemin de A à B dans $C \cup D$
- P est pair absurde
- $Y_A \cap Y_B = \emptyset$ et Y_A anticomplet à Y_B
- $Z_A = (D \cap Y_A) \cup (C \cap Y_B) \cup (C \setminus (Y_A \cup Y_B))$
- $Z_B = (D \cap Y_B) \cup (C \cap Y_A) \cup (D \setminus (Y_A \cup Y_B))$
- Z_A stable de $A_1 \cup C_1$, Z_B stable de $B_1 \cup C_1$

$\square$

Rester dans la classe



- Si un graphe de Berge admet un star cutset, il admet une BSP
- *a* est le centre d'un star cutset
- On ne reste pas dans la classe

Switchable Pairs

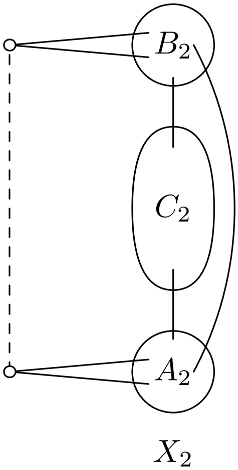
On procède en deux temps

- on utilise des blocs de décompositions simples qui préservent la classe.
- quand on a complètement décomposé on étend les blocs avec les informations de construction pour faire le calcul.

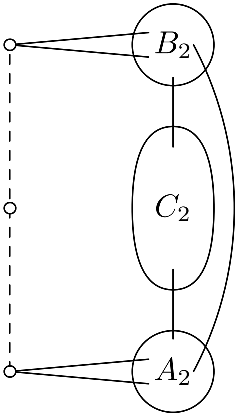
Attention à ne pas casser les blocs de décomposition. Pour cela on utilise des *switchable pairs* qui ont deux gros avantages

- on encode bien les informations d'adjacences floues
- les décompositions utilisent des vraies arêtes et ne risquent pas de décomposer les blocs.

Blocs de décomposition, 2-join impair

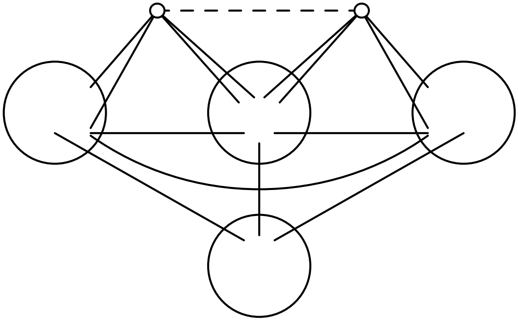


Blocs de décomposition, 2-join pair

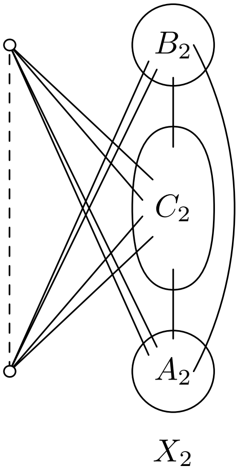


X_2

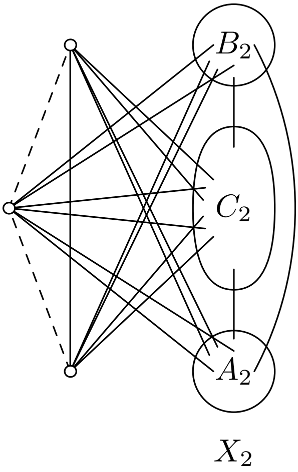
Blocs de décomposition, paire homogène



Blocs de décomposition, $\overline{\text{2-join}}$ impair



Blocs de décomposition, $\overline{\text{2-join}}$ pair



Résumé

- Adapter le théorème de décomposition aux trigraphes obtenus via nos blocs de décomposition

Résumé

- Adapter le théorème de décomposition aux trigraphes obtenus via nos blocs de décomposition
- Montrer que nos blocs d'introduisent pas de BSP

Résumé

- Adapter le théorème de décomposition aux trigraphes obtenus via nos blocs de décomposition
- Montrer que nos blocs d'introduisent pas de BSP
- Montrer qu'un bloc minimal est extrême

Résumé

- Adapter le théorème de décomposition aux trigraphes obtenus via nos blocs de décomposition
- Montrer que nos blocs d'introduisent pas de BSP
- Montrer qu'un bloc minimal est extrême
- Trouver algorithmiquement un bloc minimal

Résumé

- Adapter le théorème de décomposition aux trigraphes obtenus via nos blocs de décomposition
- Montrer que nos blocs d'introduisent pas de BSP
- Montrer qu'un bloc minimal est extrême
- Trouver algorithmiquement un bloc minimal
- Gérer les graphes basiques

Résumé

- Adapter le théorème de décomposition aux trigraphes obtenus via nos blocs de décomposition
- Montrer que nos blocs d'introduisent pas de BSP
- Montrer qu'un bloc minimal est extrême
- Trouver algorithmiquement un bloc minimal
- Gérer les graphes basiques
- Décomposer en regardant comment sont placés les stables

Résumé

- Adapter le théorème de décomposition aux trigraphes obtenus via nos blocs de décomposition
- Montrer que nos blocs d'introduisent pas de BSP
- Montrer qu'un bloc minimal est extrême
- Trouver algorithmiquement un bloc minimal
- Gérer les graphes basiques
- Décomposer en regardant comment sont placés les stables
- Pour chaque bloc de décomposition et chaque classe de base trouver un bloc de calcul qui preserve la classe

Résumé

- Adapter le théorème de décomposition aux trigraphes obtenus via nos blocs de décomposition
- Montrer que nos blocs d'introduisent pas de BSP
- Montrer qu'un bloc minimal est extrême
- Trouver algorithmiquement un bloc minimal
- Gérer les graphes basiques
- Décomposer en regardant comment sont placés les stables
- Pour chaque bloc de décomposition et chaque classe de base trouver un bloc de calcul qui preserve la classe
- Étendre les blocs de décomposition en blocs de calcul et trouver α sur un graphe basique

Résumé

- Adapter le théorème de décomposition aux trigraphes obtenus via nos blocs de décomposition
- Montrer que nos blocs d'introduisent pas de BSP
- Montrer qu'un bloc minimal est extrême
- Trouver algorithmiquement un bloc minimal
- Gérer les graphes basiques
- Décomposer en regardant comment sont placés les stables
- Pour chaque bloc de décomposition et chaque classe de base trouver un bloc de calcul qui preserve la classe
- Étendre les blocs de décomposition en blocs de calcul et trouver α sur un graphe basique
- Utiliser l'astuce de Gröstchel, Lovász et Schrijver pour obtenir la coloration.

Conclusion

Notre méthode utilise des choses spécifiques aux graphes de Berge, mais :

- l'idée des switchable pairs simplifie énormément l'encodage des adjacences floues et préserve naturellement les blocs
- avoir une décomposition extrême donne de la liberté sur un coté
- ajouter des décompositions peut donner des décompositions extrêmes

Conclusion

Notre méthode utilise des choses spécifiques aux graphes de Berge, mais :

- l'idée des switchable pairs simplifie énormément l'encodage des adjacences floues et préserve naturellement les blocs
- avoir une décomposition extrême donne de la liberté sur un coté
- ajouter des décompositions peut donner des décompositions extrêmes

Perspectives :

- Connaissez-vous des classes de graphes naturelles décomposables par le même type de décomposition ?
- Comment gérer les balanced skew partitions ?