Partition en graphes denses pour la détection de communautés

Julien Darlay

jdarlay@bouygues.com

Nadia Brauner (G-SCOP) - Julien Moncel (LAAS)

Bouygues e-lab

Équipe de R&D du groupe Bouygues (12 permanents, 5 docteurs)

- Nouvelles technologies
- Optimisation

Des applications dans tous les métiers

- Media planning TF1
- Planning de personnels Bouygues Télécom
- Dimensionnement de parking Bouygues Immobilier
- Plannification de chantiers Bouygues Construction

20% de notre temps dédié à la recherche

Diagnostic médical

Détection de pneumopathies interstitielles diffuses pour l'hôpital Avicenne

Id	diag	fatigue	chirurgie	douleur	antéc	âge
1	neg	0	1	0	0	76
2	neg	0	1	1	0	45
3	pos	1	0	1	1	61
4	pos	1	1	1	0	75

- Générer des "motifs" à partir d'un historique médical
  - $\mathit{fatigue} \Rightarrow \mathit{positif}$
  - $\mathit{age} > 50 \wedge \overline{\mathit{douleur}} \Rightarrow \mathit{positif}$
  - $\overline{\mathit{chirurgie}} \wedge \mathit{antecedent} \Rightarrow \mathit{negatif}$
  - $\overline{\mathit{douleur}} \Rightarrow \mathit{negatif}$
- Logical Analysis of Data [Boros *et al.*]

Partition de graphes pour le diagnostic médical

Méthode LAD

- Génération de plusieurs milliers de motifs
- Sélection d'un ensemble minimal de motifs (modèle)
- Évaluation du modèle

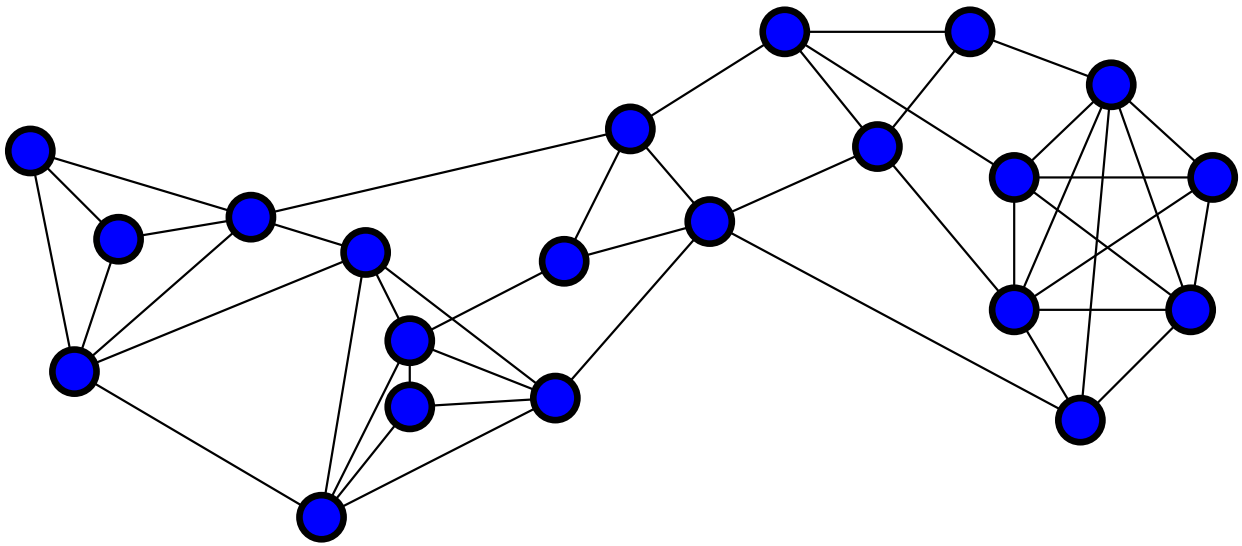
Partition de graphes pour le diagnostic médical

Méthode LAD

- Génération de plusieurs milliers de motifs
- Sélection d'un ensemble minimal de motifs (modèle)
- Évaluation du modèle

Nombreux motifs synonymes : nécessité de les regrouper

- Sommet : motif
- Arête entre deux motifs "similaires"



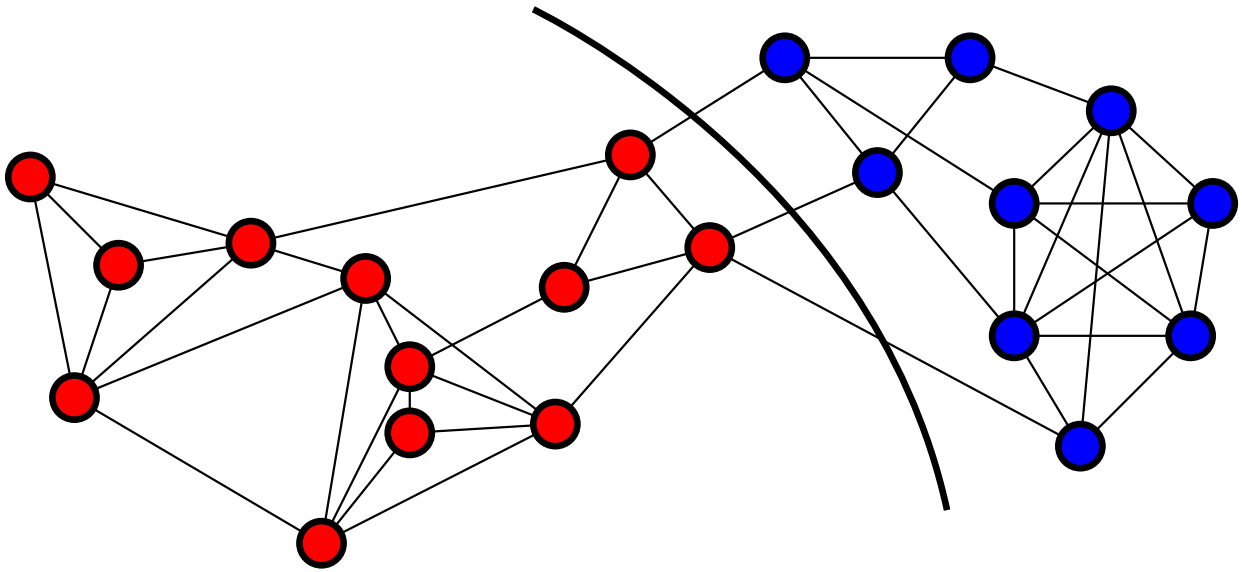
Partition de graphes pour le diagnostic médical

Méthode LAD

- Génération de plusieurs milliers de motifs
- Sélection d'un ensemble minimal de motifs (modèle)
- Évaluation du modèle

Nombreux motifs synonymes : nécessité de les regrouper

- Sommet : motif
- Arête entre deux motifs "similaires"



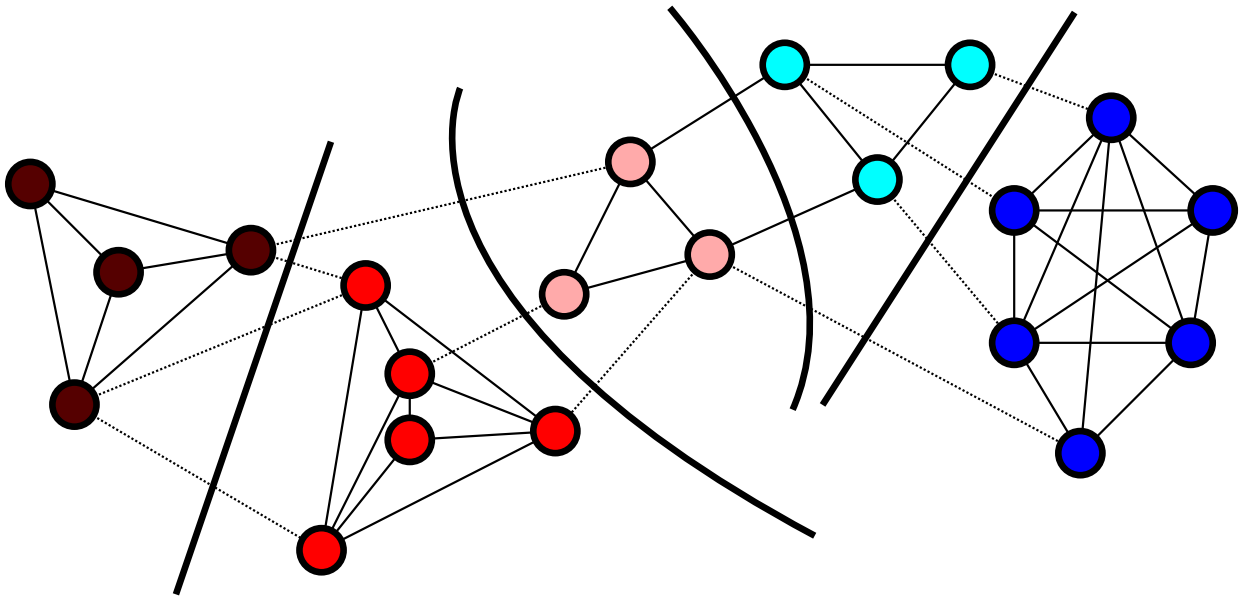
Partition de graphes pour le diagnostic médical

Méthode LAD

- Génération de plusieurs milliers de motifs
- Sélection d'un ensemble minimal de motifs (modèle)
- Évaluation du modèle

Nombreux motifs synonymes : nécessité de les regrouper

- Sommet : motif
- Arête entre deux motifs "similaires"

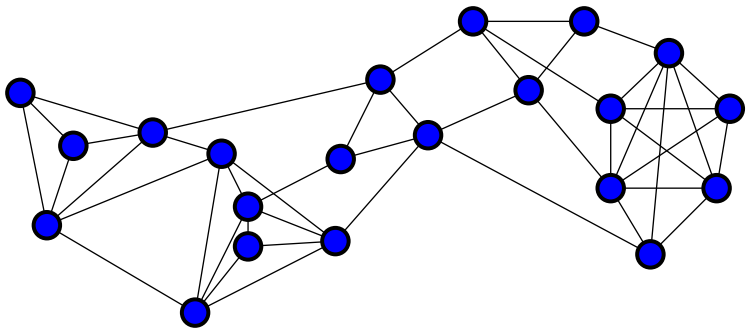


Définitions

On considère

- $G = (V, E)$ un graphe simple
- Π l'ensemble des partitions de V
- $f : \Pi \to \mathbb{R}$ une fonction d'évaluation d'une partition de V

On cherche une solution du problème de partition $\max_{\mathcal{P} \in \Pi} f(\mathcal{P})$



Littérature

Littérature

- coupe minimum [Newman 04]
- méthode spectrale [Biggs 93]
- Modularité [Newman, Girvan 04]
- regroupement hiérarchique [Clauset 08]

Littérature

Littérature

- coupe minimum [Newman 04]
- méthode spectrale [Biggs 93]
- Modularité [Newman, Girvan 04]
- regroupement hiérarchique [Clauset 08]

Applications

- réseaux sociaux [Ugander 11]
- traitement d'images
- maillage de structures
- conception de circuits

Littérature

Littérature

- coupe minimum [Newman 04]
- méthode spectrale [Biggs 93]
- Modularité [Newman, Girvan 04]
- regroupement hiérarchique [Clauset 08]

Applications

- réseaux sociaux [Ugander 11]
- traitement d'images
- maillage de structures
- conception de circuits

No free lunch theorem [Kleinberg 02]

Densité d'un graphe

La densité d'un graphe $G = (V, E)$ [Goldberg 84]

$$d(G) = \frac{|E|}{|V|}$$

Densité d'un graphe

La densité d'un graphe $G = (V, E)$ [Goldberg 84]

$$d(G) = \frac{|E|}{|V|}$$

- Un graphe dense est "presque" une clique
- Trouver un sous graphe de densité max. est polynomial [Goldberg 84]
- Lorsque la taille est fixée : NP-Difficile [Charikar 00]
- Nombreux algorithmes d'approximation [Billionnet, Roupin 2004]

Densité d'une partition

La densité d'une partition $\mathcal{P}$ des sommets d'un graphe G

$$d_G(\mathcal{P}) = \sum_{X \in \mathcal{P}} d(G[X])$$

Nos résultats [D., Brauner, Moncel 12]

- Trouver une partition de densité maximum est NP-Difficile
- Le problème ne semble pas approximable
- Polynomial sur les arbres

Un problème NP-difficile

Réduction depuis k-Coloration

- Étude du problème dans le complémentaire $\overline{G}$

$$d_G(\mathcal{P}) = \frac{|V|}{2} - \left\{ \frac{|\mathcal{P}|}{2} + d_{\overline{G}}(\mathcal{P}) \right\}$$

Un problème NP-difficile

Réduction depuis k-Coloration

- Étude du problème dans le complémentaire $\overline{G}$

$$\min_{\mathcal{P}} f(\mathcal{P}) = \left\{ \frac{|\mathcal{P}|}{2} + d_{\overline{G}}(\mathcal{P}) \right\}$$

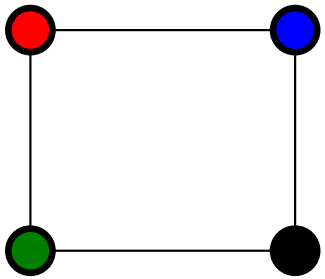
Un problème NP-difficile

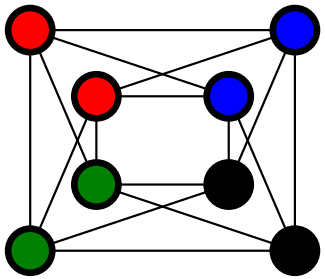
Réduction depuis k-Coloration

- Étude du problème dans le complémentaire $\overline{G}$

$$\min_{\mathcal{P}} f(\mathcal{P}) = \left\{ \frac{|\mathcal{P}|}{2} + d_{\overline{G}}(\mathcal{P}) \right\}$$

- Transformer $\overline{G}$ en $\overline{G}^q$





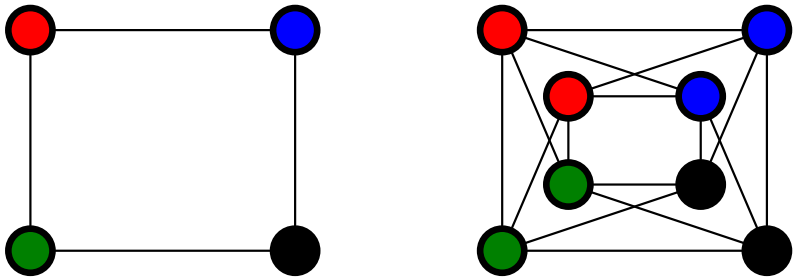
Un problème NP-difficile

Réduction depuis k-Coloration

- Étude du problème dans le complémentaire $\overline{G}$

$$\min_{\mathcal{P}} f(\mathcal{P}) = \left\{ \frac{|\mathcal{P}|}{2} + d_{\overline{G}}(\mathcal{P}) \right\}$$

- Transformer $\overline{G}$ en $\overline{G}^q$



- k coloration de $\overline{G} \Rightarrow k$ coloration de $\overline{G}^q \Rightarrow f(\mathcal{P}^*) \leq \frac{k}{2}$

Un problème NP-difficile

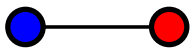
$$\min_{\mathcal{P}} f(\mathcal{P}) = \left\{ \frac{|\mathcal{P}|}{2} + d_{\overline{G}}(\mathcal{P}) \right\}$$

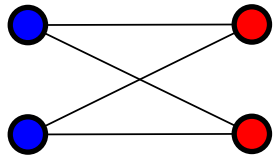
- Soit $\mathcal{P}^*$ la partition optimale de $\overline{G}^q$ telle que $f(\mathcal{P}^*) \leq \frac{k}{2}$

Un problème NP-difficile

$$\min_{\mathcal{P}} f(\mathcal{P}) = \left\{ \frac{|\mathcal{P}|}{2} + d_{\overline{G}}(\mathcal{P}) \right\}$$

- Soit $\mathcal{P}^*$ la partition optimale de $\overline{G}^q$ telle que $f(\mathcal{P}^*) \leq \frac{k}{2}$
- Soit u un sommet de $\overline{G}$ et $u^1 ... u^q$ les sommets de $\overline{G}^q$



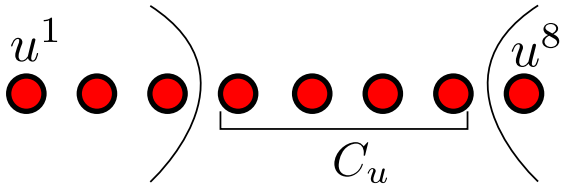


Un problème NP-difficile

$$\min_{\mathcal{P}} f(\mathcal{P}) = \left\{ \frac{|\mathcal{P}|}{2} + d_{\overline{G}}(\mathcal{P}) \right\}$$

- Soit $\mathcal{P}^*$ la partition optimale de $\overline{G}^q$ telle que $f(\mathcal{P}^*) \leq \frac{k}{2}$
- Soit u un sommet de $\overline{G}$ et $u^1 ... u^q$ les sommets de $\overline{G}^q$

- C_u la classe majoritaire de $u^1 ... u^q$ dans $\mathcal{P}^*$

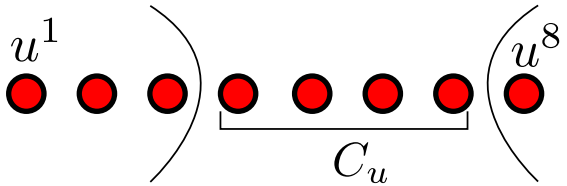


Un problème NP-difficile

$$\min_{\mathcal{P}} f(\mathcal{P}) = \left\{ \frac{|\mathcal{P}|}{2} + d_{\overline{G}}(\mathcal{P}) \right\}$$

- Soit $\mathcal{P}^*$ la partition optimale de $\overline{G}^q$ telle que $f(\mathcal{P}^*) \leq \frac{k}{2}$
- Soit u un sommet de $\overline{G}$ et $u^1 ... u^q$ les sommets de $\overline{G}^q$

- C_u la classe majoritaire de $u^1 ... u^q$ dans $\mathcal{P}^*$



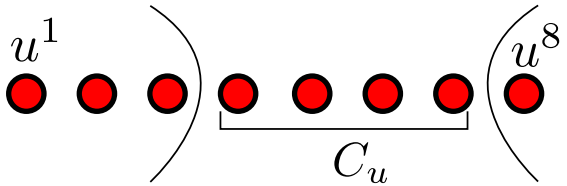
- Si $q = |V|^4$ et (u, v) une arête de $\overline{G} \Rightarrow C_u \neq C_v$ (comptage)

Un problème NP-difficile

$$\min_{\mathcal{P}} f(\mathcal{P}) = \left\{ \frac{|\mathcal{P}|}{2} + d_{\overline{G}}(\mathcal{P}) \right\}$$

- Soit $\mathcal{P}^*$ la partition optimale de $\overline{G}^q$ telle que $f(\mathcal{P}^*) \leq \frac{k}{2}$
- Soit u un sommet de $\overline{G}$ et $u^1 ... u^q$ les sommets de $\overline{G}^q$

- C_u la classe majoritaire de $u^1 ... u^q$ dans $\mathcal{P}^*$



- Si $q = |V|^4$ et (u, v) une arête de $\overline{G} \Rightarrow C_u \neq C_v$ (comptage)
- Coloration de $\overline{G}$ en k couleurs avec les C_u

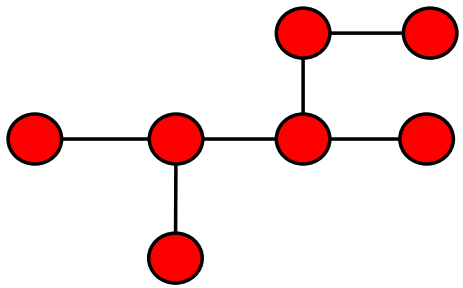
Le cas des arbres

Contexte

- $T = (V, E)$ un arbre
- Trouver une partition $\mathcal{P}$ qui maximise $d(\mathcal{P})$

Méthode

- Structure de la solution optimale
- Algorithme de construction

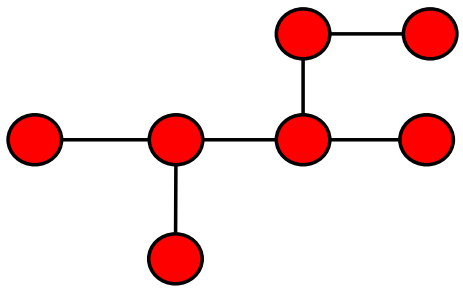


Le cas des arbres

Premières observations

$T = (V, E)$ un arbre et $\mathcal{P}^*$ une partition optimale de T

- Densité d'un arbre est $\frac{n-1}{n}$

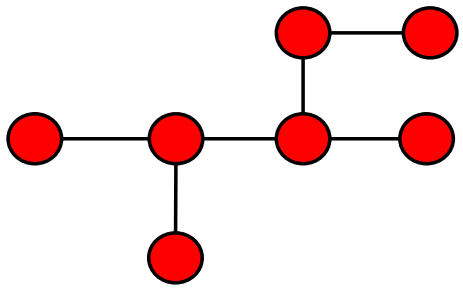


Le cas des arbres

Premières observations

$T = (V, E)$ un arbre et $\mathcal{P}^*$ une partition optimale de T

- Densité d'un arbre est $\frac{n-1}{n}$
- Chaque classe contient au moins deux sommets

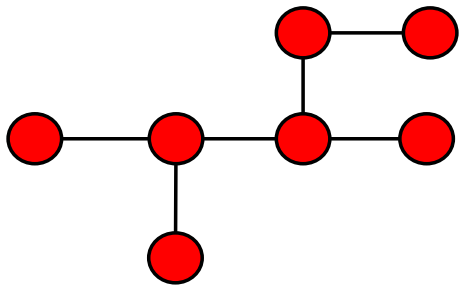


Le cas des arbres

Premières observations

$T = (V, E)$ un arbre et $\mathcal{P}^*$ une partition optimale de T

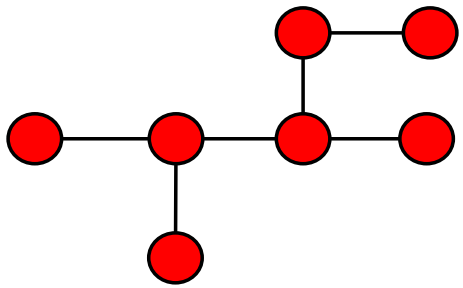
- Densité d'un arbre est $\frac{n-1}{n}$
- Chaque classe contient au moins deux sommets
- Le sous graphe induit par une classe est connexe



Premières observations

$T = (V, E)$ un arbre et $\mathcal{P}^*$ une partition optimale de T

- Densité d'un arbre est $\frac{n-1}{n}$
- Chaque classe contient au moins deux sommets
- Le sous graphe induit par une classe est connexe
- La densité d'une classe est dans $[\frac{1}{2}; 1[$



Le cas des arbres

Caractérisation des solutions optimales

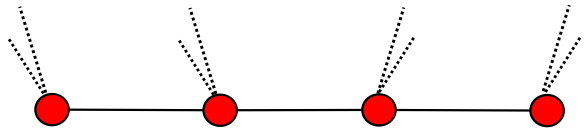
Propriété

Chaque classe d'une partition optimale est une étoile non triviale

Caractérisation des solutions optimales

Propriété

Chaque classe d'une partition optimale est une étoile non triviale

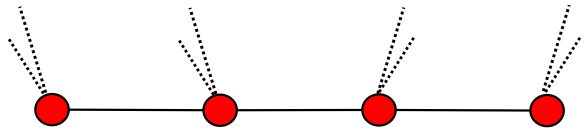


$$d(G[X]) < 1$$

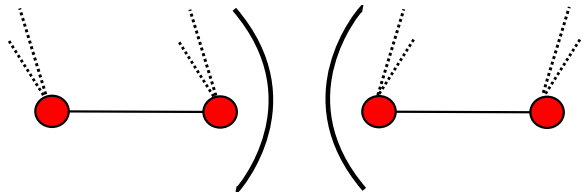
Caractérisation des solutions optimales

Propriété

Chaque classe d'une partition optimale est une étoile non triviale



$$d(G[X]) < 1$$



$$d(G[X_1]) + d(G[X_2]) \geq \frac{1}{2} + \frac{1}{2} = 1$$

Lien avec les couplages

Propriété

Soit $\mathcal{M}^*$ un couplage maximum et $\mathcal{P}^*$ une partition de densité maximale alors $|\mathcal{M}^*| = |\mathcal{P}^*|$

Démonstration

Lien avec les couplages

Propriété

Soit $\mathcal{M}^*$ un couplage maximum et $\mathcal{P}^*$ une partition de densité maximale alors $|\mathcal{M}^*| = |\mathcal{P}^*|$

Démonstration

- En prenant une arête dans chaque classe : $|\mathcal{P}^*| = |\mathcal{M}| \leq |\mathcal{M}^*|$

Lien avec les couplages

Propriété

Soit $\mathcal{M}^*$ un couplage maximum et $\mathcal{P}^*$ une partition de densité maximale alors $|\mathcal{M}^*| = |\mathcal{P}^*|$

Démonstration

- En prenant une arête dans chaque classe : $|\mathcal{P}^*| = |\mathcal{M}| \leq |\mathcal{M}^*|$
- Si $|\mathcal{M}| < |\mathcal{M}^*|$, il existe une chaîne augmentante [Petersen 1891]

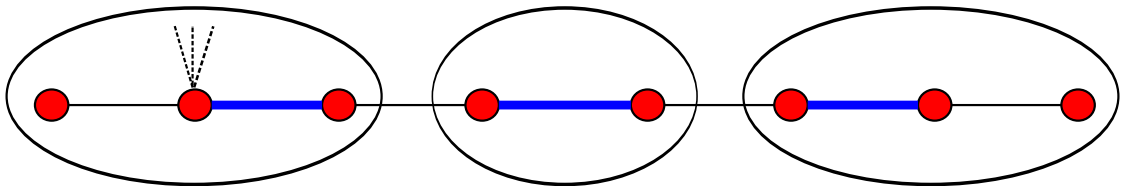
Lien avec les couplages

Propriété

Soit $\mathcal{M}^*$ un couplage maximum et $\mathcal{P}^*$ une partition de densité maximale alors $|\mathcal{M}^*| = |\mathcal{P}^*|$

Démonstration

- En prenant une arête dans chaque classe : $|\mathcal{P}^*| = |\mathcal{M}| \leq |\mathcal{M}^*|$
- Si $|\mathcal{M}| < |\mathcal{M}^*|$, il existe une chaîne augmentante [Petersen 1891]



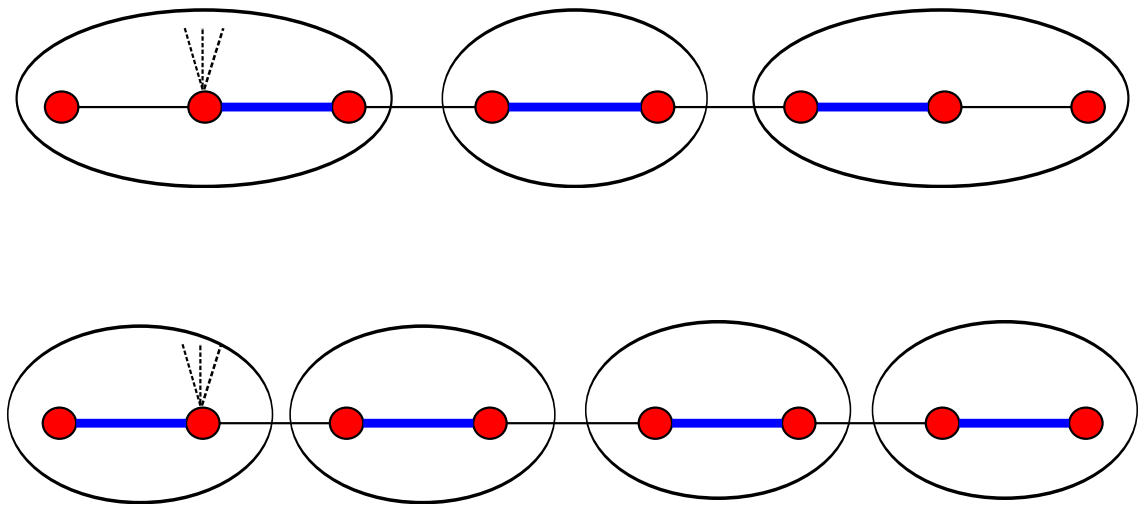
Lien avec les couplages

Propriété

Soit $\mathcal{M}^*$ un couplage maximum et $\mathcal{P}^*$ une partition de densité maximale alors $|\mathcal{M}^*| = |\mathcal{P}^*|$

Démonstration

- En prenant une arête dans chaque classe : $|\mathcal{P}^*| = |\mathcal{M}| \leq |\mathcal{M}^*|$
- Si $|\mathcal{M}| < |\mathcal{M}^*|$, il existe une chaîne augmentante [Petersen 1891]



Lien avec les transversaux

Propriété

Soit $\mathcal{T}^*$ un transversal minimum et $\mathcal{P}^*$ une partition de densité maximale alors les sommets de $\mathcal{T}^*$ sont les centres des étoiles de $\mathcal{P}^*$

Démonstration

- Par le théorème de [König 31], si $\mathcal{T}^*$ est un transversal minimum : $|\mathcal{T}^*| = |\mathcal{M}^*| = |\mathcal{P}^*|$

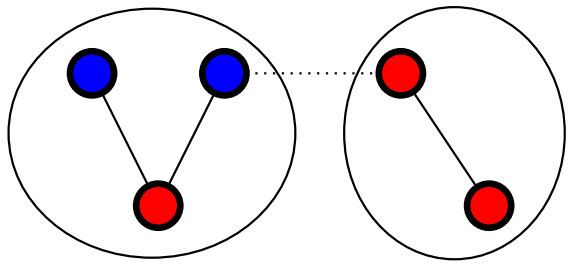
Lien avec les transversaux

Propriété

Soit $\mathcal{T}^*$ un transversal minimum et $\mathcal{P}^*$ une partition de densité maximale alors les sommets de $\mathcal{T}^*$ sont les centres des étoiles de $\mathcal{P}^*$

Démonstration

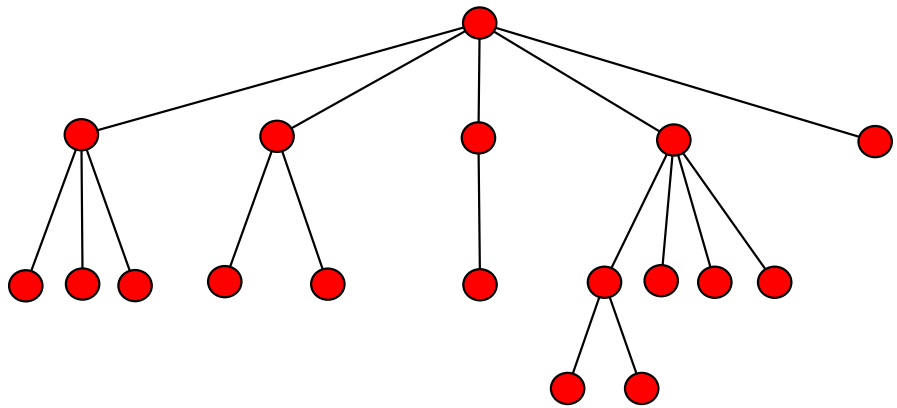
- Par le théorème de [König 31], si $\mathcal{T}^*$ est un transversal minimum : $|\mathcal{T}^*| = |\mathcal{M}^*| = |\mathcal{P}^*|$
- Si deux sommets de $\mathcal{T}^*$ sont dans la même classe alors il existe une classe X sans sommet de $\mathcal{T}^*$



Algorithme de partition

Programmation dynamique

- Enraciner l'arbre

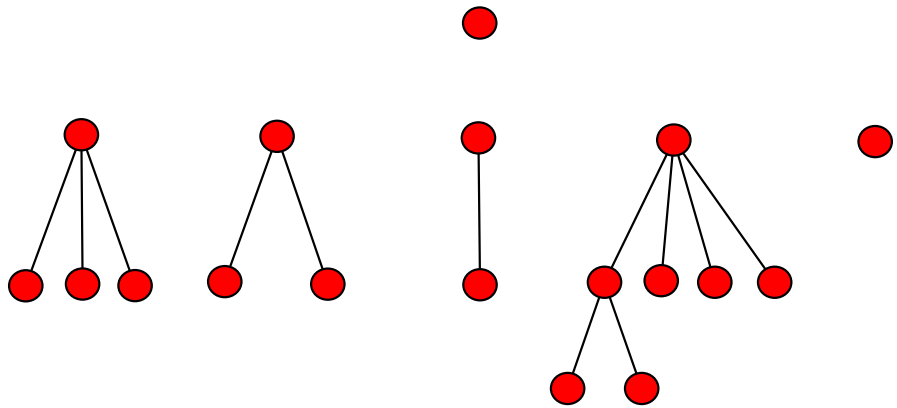


Le cas des arbres

Algorithme de partition

Programmation dynamique

- Enraciner l'arbre
- Calculer la partition optimale de chaque sous-arbre

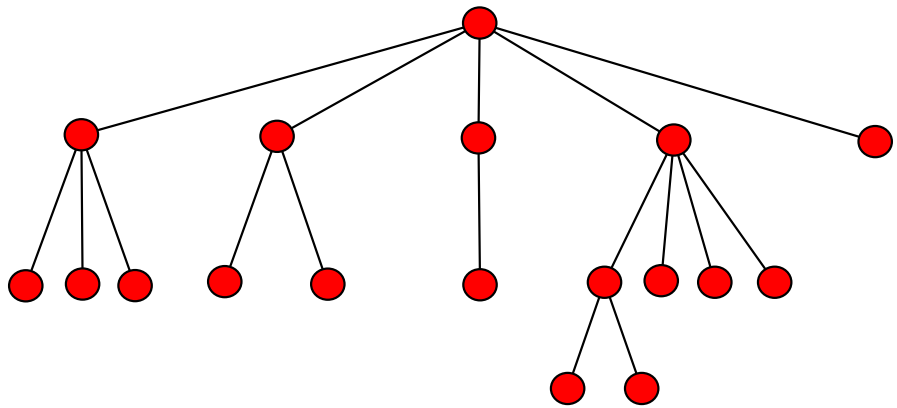


Le cas des arbres

Algorithme de partition

Programmation dynamique

- Enraciner l'arbre
- Calculer la partition optimale de chaque sous-arbre
- Calculer un transversal minimum

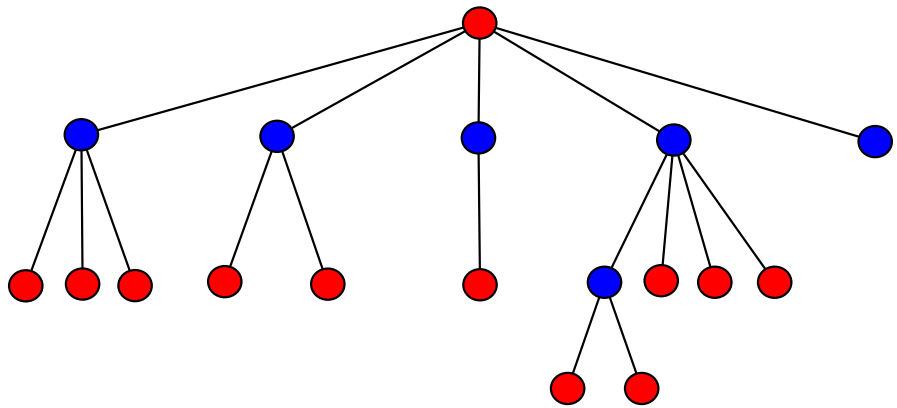


Le cas des arbres

Algorithme de partition

Programmation dynamique

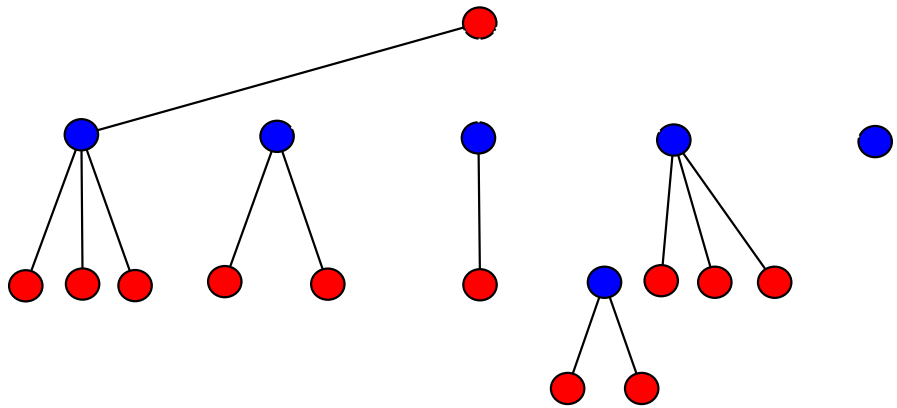
- Enraciner l'arbre
- Calculer la partition optimale de chaque sous-arbre
- Calculer un transversal minimum
  1. La racine n'est pas dans le transversal



Algorithme de partition

Programmation dynamique

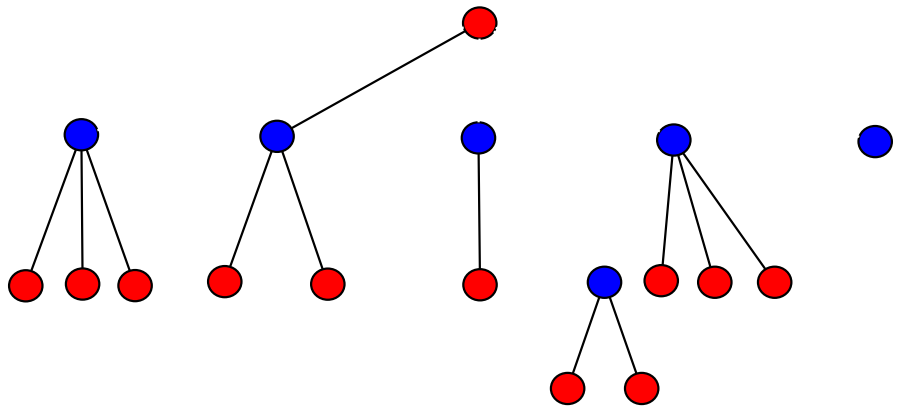
- Enraciner l'arbre
- Calculer la partition optimale de chaque sous-arbre
- Calculer un transversal minimum
  1. La racine n'est pas dans le transversal



Algorithme de partition

Programmation dynamique

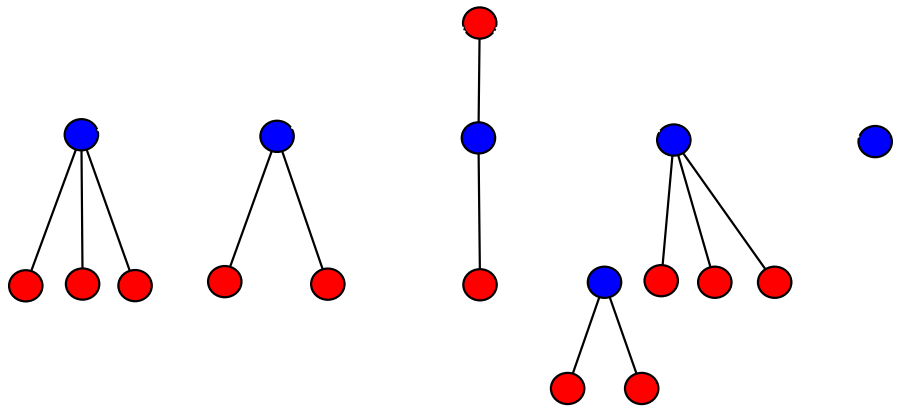
- Enraciner l'arbre
- Calculer la partition optimale de chaque sous-arbre
- Calculer un transversal minimum
  1. La racine n'est pas dans le transversal



Algorithme de partition

Programmation dynamique

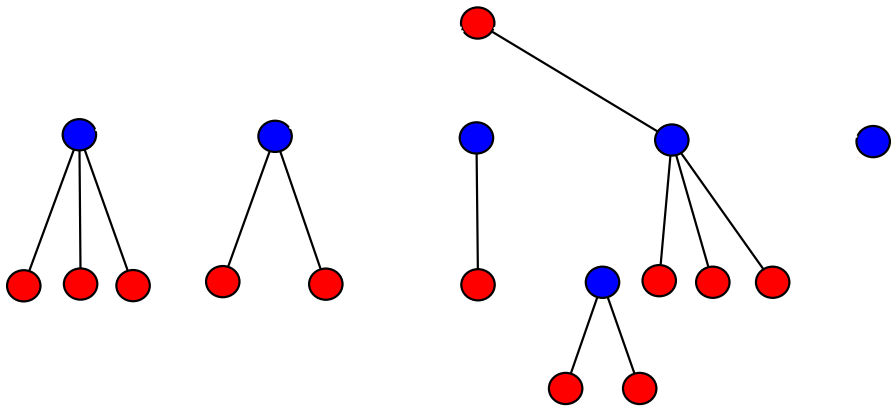
- Enraciner l'arbre
- Calculer la partition optimale de chaque sous-arbre
- Calculer un transversal minimum
  1. La racine n'est pas dans le transversal



Algorithme de partition

Programmation dynamique

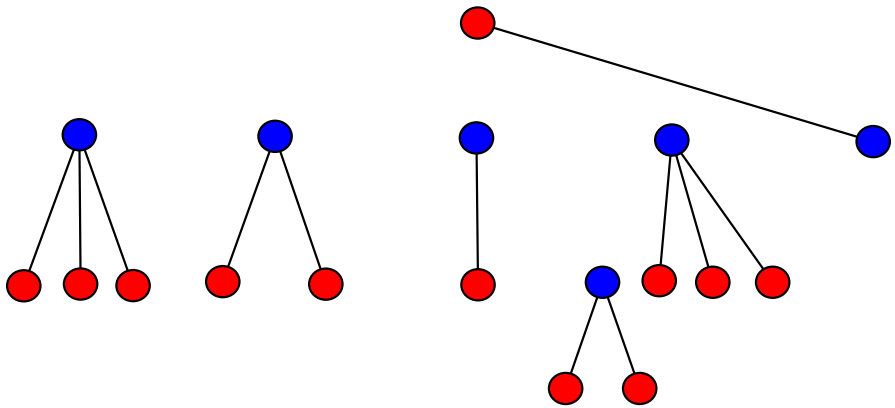
- Enraciner l'arbre
- Calculer la partition optimale de chaque sous-arbre
- Calculer un transversal minimum
  1. La racine n'est pas dans le transversal



Algorithme de partition

Programmation dynamique

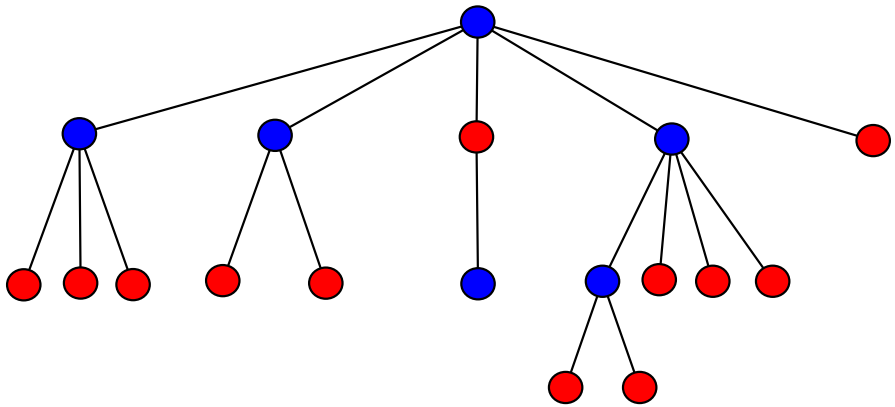
- Enraciner l'arbre
- Calculer la partition optimale de chaque sous-arbre
- Calculer un transversal minimum
  1. La racine n'est pas dans le transversal



Algorithme de partition

Programmation dynamique

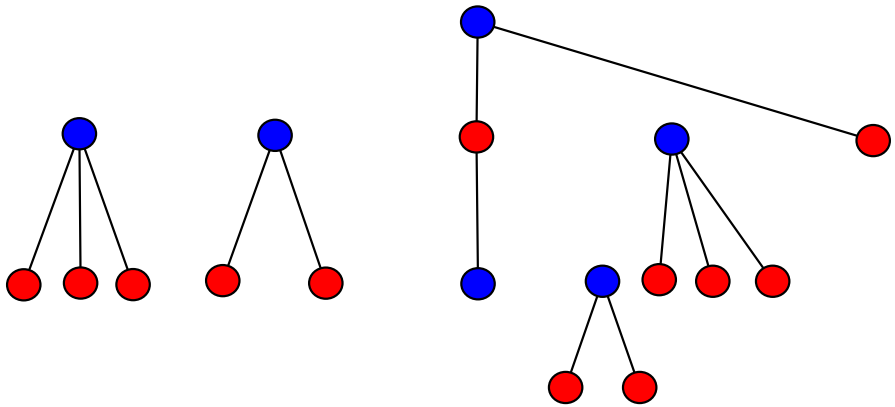
- Enraciner l'arbre
- Calculer la partition optimale de chaque sous-arbre
- Calculer un transversal minimum
  1. La racine n'est pas dans le transversal
  2. La racine est dans le transversal



Algorithme de partition

Programmation dynamique

- Enraciner l'arbre
- Calculer la partition optimale de chaque sous-arbre
- Calculer un transversal minimum
  1. La racine n'est pas dans le transversal
  2. La racine est dans le transversal

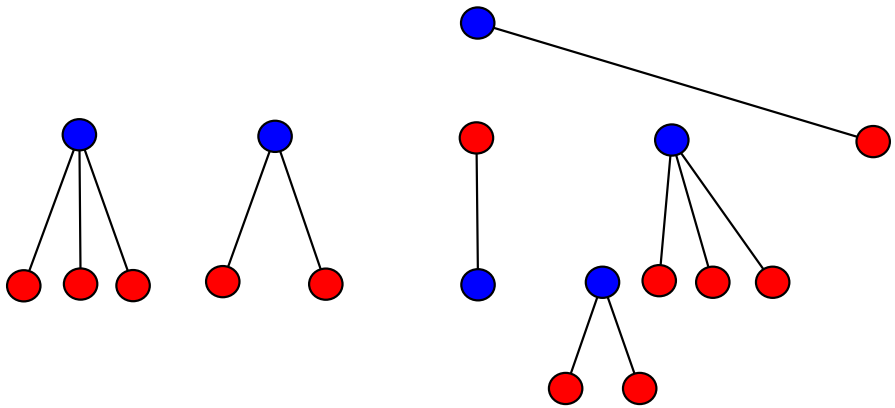


Le cas des arbres

Algorithme de partition

Programmation dynamique

- Enraciner l'arbre
- Calculer la partition optimale de chaque sous-arbre
- Calculer un transversal minimum
  1. La racine n'est pas dans le transversal
  2. La racine est dans le transversal



Complexité de l'algorithme

Quelques remarques

- À chaque étape
  - Calcul dynamique du transversal $O(d)$
  - Affectation de la racine en $O(d)$ ou $O(d \log(d))$
- Les sommets sont parcourus une seule et unique fois

Complexité en $O(n \log n)$

Évaluation empirique du critère

Partition en sous-graphes denses

- NP-Difficile en général
- Polynomial sur les arbres
- Évaluation du critère sur les instances de la littérature ?

Résolution pratique

- Moteur LocalSolver
- Échanges locaux entre les classes
- *First improve* + Recuit simulé

Différentes instances

- Arbres
- Littérature

LocalSolver

Solver de programmation mathématique basé sur la recherche locale

- Trouve de bonnes solutions en quelques minutes
- Pour des problèmes de grandes tailles
- Fortement combinatoires et non linéaires

Model & run

- Approche déclarative
- Dans un langage de modélisation innovant
- Ou via des API C++, Java, .NET

Modèle LocalSolver

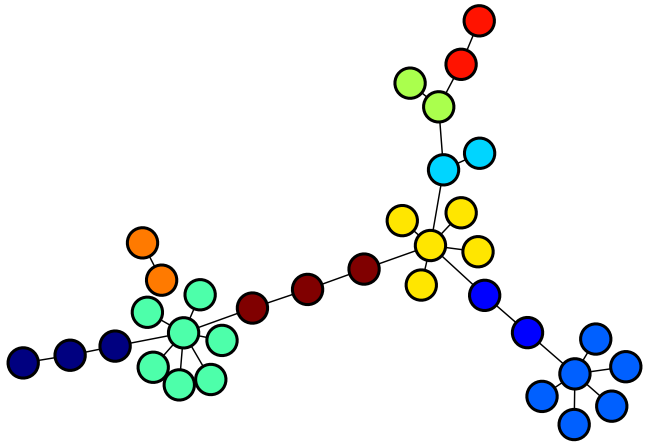
$$\max_{\mathcal{P} \in \Pi} \sum_{X \in \mathcal{P}} \frac{|E(G[X])|}{|X|}$$

```
x[1..n][1..C] <- bool();
for[i in 1..n]{
  constraint sum[j in 1..C] (x[i][j]) == 1;
}

for[c in 1..C]{
  card[c] <- max(1.0,sum[i in 1..n] (x[i][c]));
  edges[c] <- sum[i in 1..m](x[origin[i]][c] * x[dest[i]][c]);
}

obj <- sum[c in 1..C] ((edges[c]) / (card[c]));
maximize obj;
```

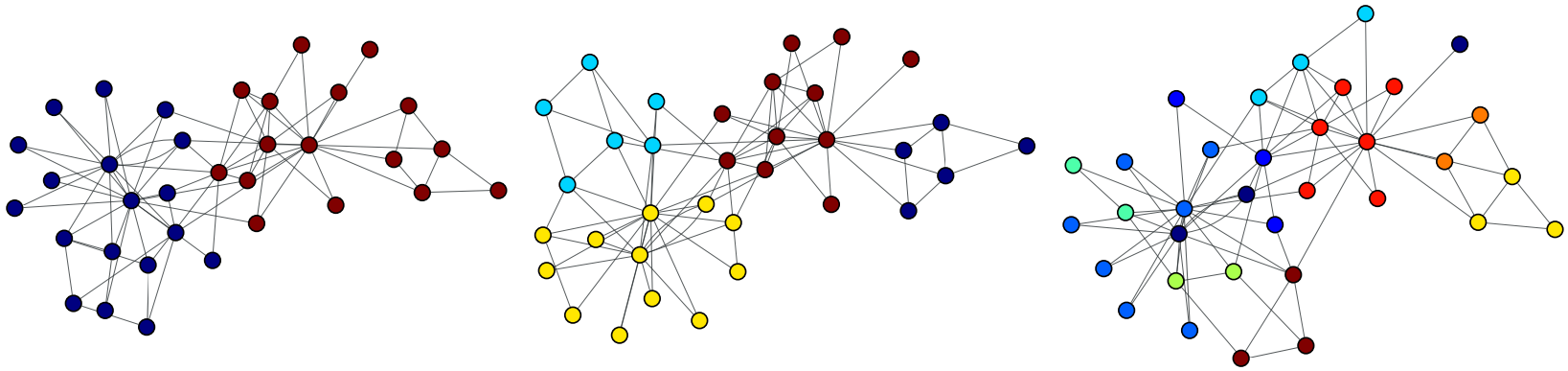
Expérimentations sur les arbres



Instance	n	t	Opt	LocalSolver 3.1
zachary_tree	34	10s	6.6213	6.6213
dolphins_tree	62	10s	14.3633	14.3633
polbooks_tree	105	60s	23.8033	23.75334
football_tree	115	60s	27.3217	27.1217
netscience_tree	1589	5min	335.0197	328.0876

Table: Moyenne des densités trouvées sur 10 instances

Expérimentations sur les données de la littérature



Instance	n	m	$\lvert C \rvert$	$\lvert C^*_Q \rvert$	Δ
zachary	34	78	10	4	0%
dolphins	62	159	20	5	25%
polbooks	105	254	32	5	10%
football	115	613	22	10	1%
netscience	1589	914	517	407*	16%
as-22july06	22963	48436	45*	-	-

Table: Solution après 5 minutes de calcul vs Modularité

Conclusion

Recherche locale avec LocalSolver :

- Performante sur les arbres
- Modèle quadratique : limite à 1000 sommets

Critère de partition :

- NP-difficile en général
- Tendance à créer beaucoup de classes
- Propose des solutions différentes de la modularité