Primal-Dual Approximation Algorithms for Integral Flow and Multicut in Trees

N. Garg, V. V. Vazirani, M. Yannakakis

Algorithmica (1997) 18:3-20

Plan de la présentation

Définitions et présentation des problèmes

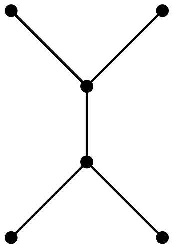
Problèmes étudiés

- Données : un arbre A dont chaque arête uv est munie d'une capacité entière $capa(u, v)$, un ensemble de p paires sources-puits (s_i, s'_i).

Problèmes étudiés

- Données : un arbre A dont chaque arête uv est munie d'une capacité entière $capa(u, v)$, un ensemble de p paires sources-puits (s_i, s_i').

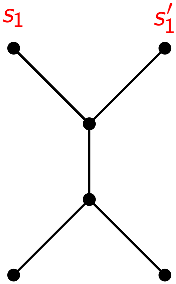
Capacités $= 1$



Problèmes étudiés

- <u>Données</u> : un arbre A dont chaque arête uv est munie d'une capacité entière $capa(u,v)$, un ensemble de p paires sources-puits (s_i, s_i').

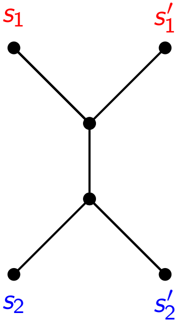
Capacités $= 1$



Problèmes étudiés

- Données : un arbre A dont chaque arête uv est munie d'une capacité entière $capa(u, v)$, un ensemble de p paires sources-puits (s_i, s'_i).

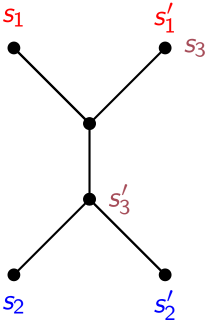
Capacités $= 1$



Problèmes étudiés

- Données : un arbre A dont chaque arête uv est munie d'une capacité entière $capa(u,v)$, un ensemble de p paires sources-puits (s_i, s'_i).

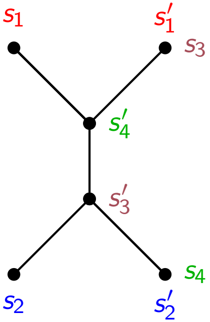
Capacités $= 1$



Problèmes étudiés

- Données : un arbre A dont chaque arête uv est munie d'une capacité entière $capa(u, v)$, un ensemble de p paires sources-puits (s_i, s'_i).

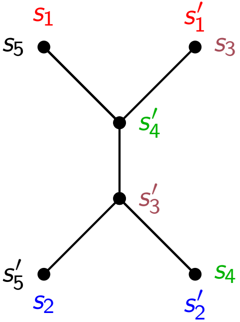
Capacités $= 1$



Problèmes étudiés

- Données : un arbre A dont chaque arête uv est munie d'une capacité entière $capa(u, v)$, un ensemble de p paires sources-puits (s_i, s_i').

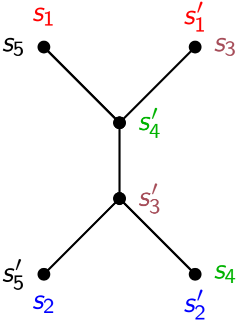
Capacités $= 1$
p $= 5$

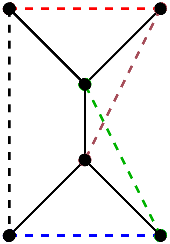


Problèmes étudiés

- <u>Données</u> : un arbre A dont chaque arête uv est munie d'une capacité entière $capa(u, v)$, un ensemble de p paires sources-puits (s_i, s_i').

Capacités = 1
p = 5

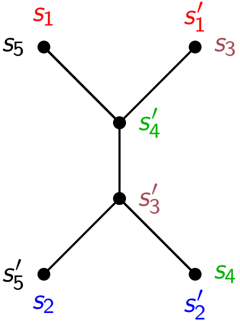


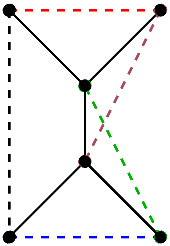


Problèmes étudiés

- Données : un arbre A dont chaque arête uv est munie d'une capacité entière $capa(u, v)$, un ensemble de p paires sources-puits (s_i, s_i').
- Problème 1 (**Multiflot entier maximum**) : maximiser le nombre d'unités de flot circulant dans A de façon à ce que chaque unité soit acheminée de s_i à s_i' pour un i, sans violer les capacités sur les arêtes.

Capacités $= 1$
p $= 5$

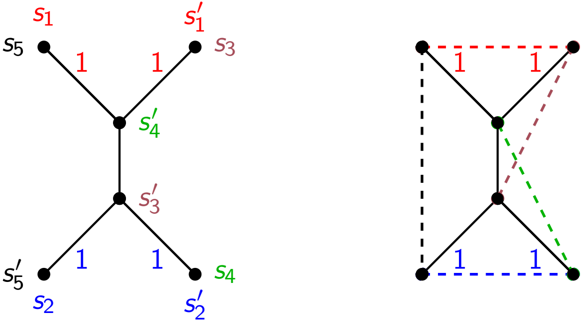




Problèmes étudiés

- Données : un arbre A dont chaque arête uv est munie d'une capacité entière $capa(u, v)$, un ensemble de p paires sources-puits (s_i, s_i').
- Problème 1 (**Multiflot entier maximum**) : maximiser le nombre d'unités de flot circulant dans A de façon à ce que chaque unité soit acheminée de s_i à s_i' pour un i, sans violer les capacités sur les arêtes.

Capacités $= 1$
p $= 5$

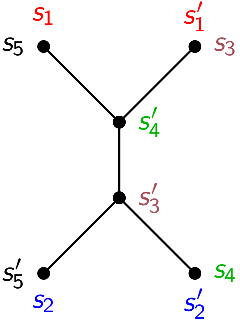


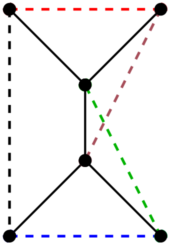
Multiflot entier de valeur 1+1=2

Problèmes étudiés

- Données : un arbre A dont chaque arête uv est munie d'une capacité entière $capa(u, v)$, un ensemble de p paires sources-puits (s_i, s'_i).
- Problème 2 (**Multicoupe minimum**) : sélectionner un ensemble d'arêtes de capacité totale minimum qui "couvre" les p paires, c.-à-d. dont la suppression ne laisse aucune chaîne entre s_i et s'_i, pour tout i.

Capacités = 1
p = 5



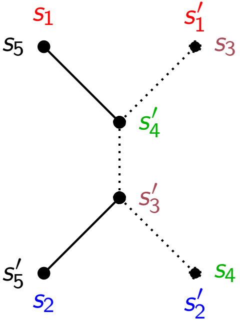


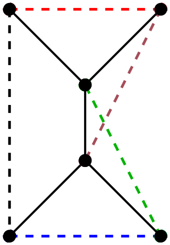
Problèmes étudiés

- <u>Données</u> : un arbre A dont chaque arête uv est munie d'une capacité entière $capa(u, v)$, un ensemble de p paires sources-puits (s_i, s_i').
- <u>Problème 2</u> (**Multicoupe minimum**) : sélectionner un ensemble d'arêtes de capacité totale minimum qui "couvre" les p paires, c.-à-d. dont la suppression ne laisse aucune chaîne entre s_i et s_i', pour tout i.

Capacités = 1

p = 5

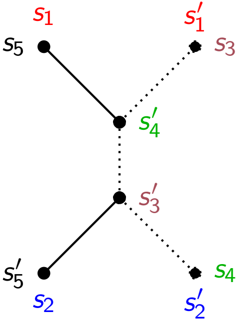


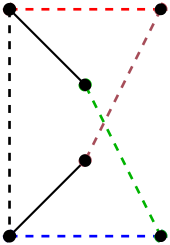


Problèmes étudiés

- Données : un arbre A dont chaque arête uv est munie d'une capacité entière $capa(u, v)$, un ensemble de p paires sources-puits (s_i, s'_i).
- Problème 2 (**Multicoupe minimum**) : sélectionner un ensemble d'arêtes de capacité totale minimum qui "couvre" les p paires, c.-à-d. dont la suppression ne laisse aucune chaîne entre s_i et s'_i, pour tout i.

Capacités = 1
p = 5

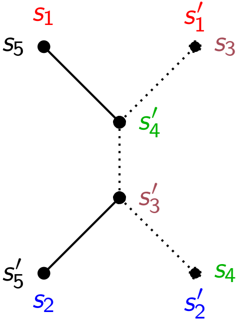


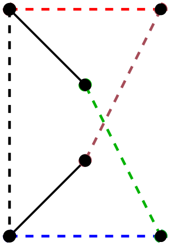


Problèmes étudiés

- Données : un arbre A dont chaque arête uv est munie d'une capacité entière $capa(u, v)$, un ensemble de p paires sources-puits (s_i, s_i').
- Problème 2 (**Multicoupe minimum**) : sélectionner un ensemble d'arêtes de capacité totale minimum qui "couvre" les p paires, c.-à-d. dont la suppression ne laisse aucune chaîne entre s_i et s_i', pour tout i.

Capacités = 1
p = 5





Multicoupe de valeur 1+1+1=3

Un bref état de l'art (graphes non orientés)

	Multiflot entier maximum	**Multicoupe minimum**
$p = 1$	Polynomial (flot maximum)	Polynomial (coupe minimum)
$p = 2$	NP-difficile [Even, Itai, Shamir (1976)]	Polynomial [Yannakakis et al. (1983)]
$p \geq 3$	NP-difficile [Even, Itai, Shamir (1976)]	NP-difficile [Dahlhaus et al. (1994)]

Modèles PLNE associés (multiflot entier maximum)

- Variables (une variable $f_i \geq 0$ par paire (s_i, s_i')) : $f_i =$ valeur du flot (entière ou non) sur l'unique chaîne μ_i entre s_i et s_i' dans A.

Modèles PLNE associés (multiflot entier maximum)

- Variables (une variable $f_i \geq 0$ par paire (s_i, s'_i)) : f_i = valeur du flot (entière ou non) sur l'unique chaîne μ_i entre s_i et s'_i dans A.
- Fonction objectif : $\max \sum_{i=1}^{p} f_i$

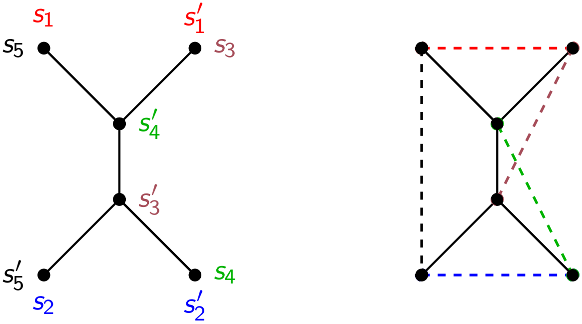
Modèles PLNE associés (multiflot entier maximum)

- Variables (une variable $f_i \geq 0$ par paire $(s_i, s_i'))$: $f_i =$ valeur du flot (entière ou non) sur l'unique chaîne μ_i entre s_i et s_i' dans A.
- Fonction objectif : $\max \sum_{i=1}^{p} f_i$
- Contraintes : $\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u, v)$, pour toute arête uv

Modèles PLNE associés (multiflot entier maximum)

- Variables (une variable $f_i \geq 0$ par paire (s_i, s'_i)) : f_i = valeur du flot (entière ou non) sur l'unique chaîne μ_i entre s_i et s'_i dans A.
- Fonction objectif : $\max \sum_{i=1}^{p} f_i$
- Contraintes : $\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u, v)$, pour toute arête uv

Capacités = 1
p = 5



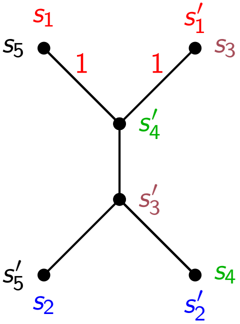
Modèles PLNE associés (multiflot entier maximum)

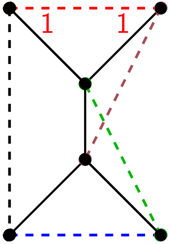
- Variables (une variable $f_i \geq 0$ par paire (s_i, s_i')) : f_i = valeur du flot (entière ou non) sur l'unique chaîne μ_i entre s_i et s_i' dans A.
- Fonction objectif : $\max \sum_{i=1}^{p} f_i$
- Contraintes : $\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u,v)$, pour toute arête uv

Capacités = 1

p = 5

$f_1 = 1$





Modèles PLNE associés (multiflot entier maximum)

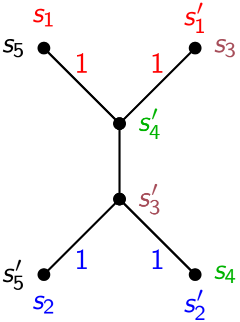
- Variables (une variable $f_i \geq 0$ par paire (s_i, s'_i)) : f_i = valeur du flot (entière ou non) sur l'unique chaîne μ_i entre s_i et s'_i dans A.
- Fonction objectif : $\max \sum_{i=1}^{p} f_i$
- Contraintes : $\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u, v)$, pour toute arête uv

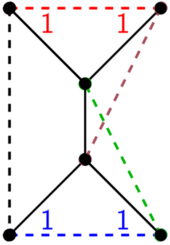
Capacités = 1

p = 5

$f_1 = 1$

$f_2 = 1$





Modèles PLNE associés (multiflot entier maximum)

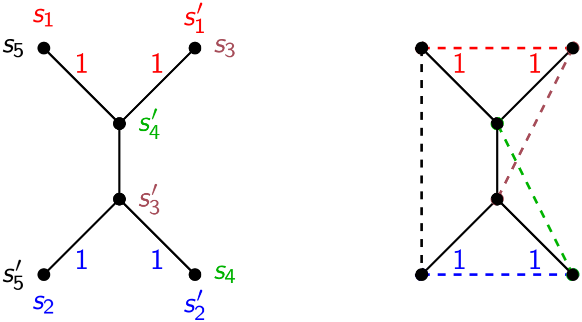
- Variables (une variable $f_i \geq 0$ par paire (s_i, s'_i)) : f_i = valeur du flot (entière ou non) sur l'unique chaîne μ_i entre s_i et s'_i dans A.
- Fonction objectif : $\max \sum_{i=1}^{p} f_i$
- Contraintes : $\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u, v)$, pour toute arête uv

Capacités = 1

p = 5

$f_1 = 1$

$f_2 = 1$



Multiflot entier de valeur $f_1 + f_2 + f_3 + f_4 + f_5 = 1 + 1 = 2$

Modèles PLNE associés (multicoupe minimum)

- Variables (une variable $y_{uv} \in \{0,1\}$ par arête uv) : $y_{uv} = 1$ ssi l'arête uv est sélectionnée (pour être supprimée).

Modèles PLNE associés (multicoupe minimum)

- Variables (une variable $y_{uv} \in \{0, 1\}$ par arête uv) : $y_{uv} = 1$ ssi l'arête uv est sélectionnée (pour être supprimée).
- Fonction objectif : $\min \sum_{\text{arêtes } uv \in A} capa(u, v) y_{uv}$

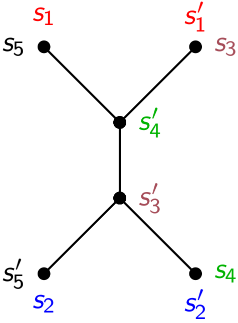
Modèles PLNE associés (multicoupe minimum)

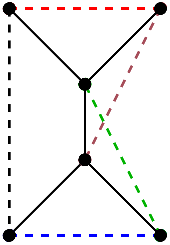
- Variables (une variable $y_{uv} \in \{0, 1\}$ par arête uv) : $y_{uv} = 1$ ssi l'arête uv est sélectionnée (pour être supprimée).
- Fonction objectif : $\min \sum_{\text{arêtes } uv \in A} capa(u, v) y_{uv}$
- Contraintes : $\sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1$, pour toute paire (s_i, s_i')

Modèles PLNE associés (multicoupe minimum)

- Variables (une variable $y_{uv} \in \{0, 1\}$ par arête uv) : $y_{uv} = 1$ ssi l'arête uv est sélectionnée (pour être supprimée).
- Fonction objectif : $\min \sum_{\text{arêtes } uv \in A} capa(u, v) y_{uv}$
- Contraintes : $\sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1$, pour toute paire (s_i, s'_i)

Capacités = 1
p = 5





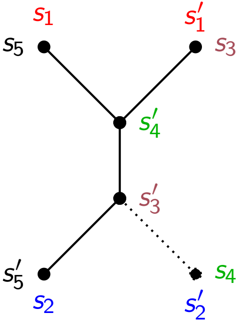
Modèles PLNE associés (multicoupe minimum)

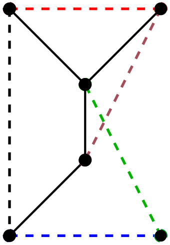
- Variables (une variable $y_{uv} \in \{0, 1\}$ par arête uv) : $y_{uv} = 1$ ssi l'arête uv est sélectionnée (pour être supprimée).
- Fonction objectif : $\min \sum_{\text{arêtes } uv \in A} capa(u, v) y_{uv}$
- Contraintes : $\sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1$, pour toute paire (s_i, s_i')

Capacités $= 1$

p $= 5$

$y_{s_2' s_3'} = 1$





Modèles PLNE associés (multicoupe minimum)

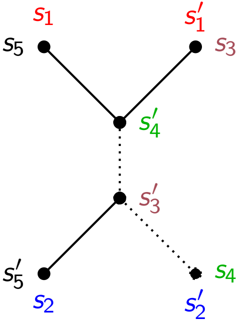
- Variables (une variable $y_{uv} \in \{0,1\}$ par arête uv) : $y_{uv} = 1$ ssi l'arête uv est sélectionnée (pour être supprimée).
- Fonction objectif : $\min \sum_{\text{arêtes } uv \in A} capa(u,v) y_{uv}$
- Contraintes : $\sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1$, pour toute paire (s_i, s'_i)

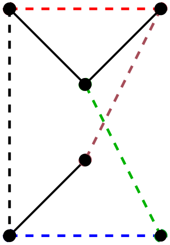
Capacités $= 1$

p $= 5$

$y_{s'_2 s'_3} = 1$

$y_{s'_3 s'_4} = 1$





Modèles PLNE associés (multicoupe minimum)

- Variables (une variable $y_{uv} \in \{0,1\}$ par arête uv) : $y_{uv} = 1$ ssi l'arête uv est sélectionnée (pour être supprimée).
- Fonction objectif : $\min \sum_{\text{arêtes } uv \in A} capa(u,v) y_{uv}$
- Contraintes : $\sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1$, pour toute paire (s_i, s_i')

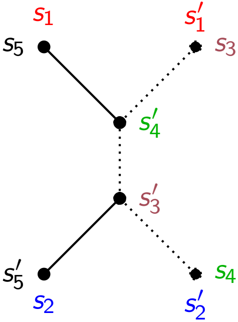
Capacités $= 1$

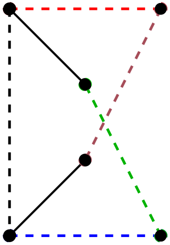
p $= 5$

$y_{s_2' s_3'} = 1$

$y_{s_3' s_4'} = 1$

$y_{s_4' s_1'} = 1$





Modèles PLNE associés (multicoupe minimum)

- Variables (une variable $y_{uv} \in \{0,1\}$ par arête uv) : $y_{uv} = 1$ ssi l'arête uv est sélectionnée (pour être supprimée).
- Fonction objectif : $\min \sum_{\text{arêtes } uv \in A} capa(u,v) y_{uv}$
- Contraintes : $\sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1$, pour toute paire (s_i, s'_i)

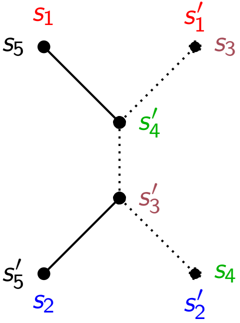
Capacités $= 1$

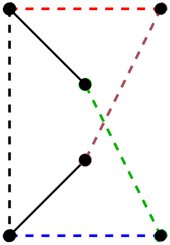
p $= 5$

$y_{s'_2 s'_3} = 1$

$y_{s'_3 s'_4} = 1$

$y_{s'_4 s'_1} = 1$





Multicoupe de valeur $y_{s'_2 s'_3} + y_{s'_3 s'_4} + y_{s'_4 s'_1} = 1 + 1 + 1 = 3$

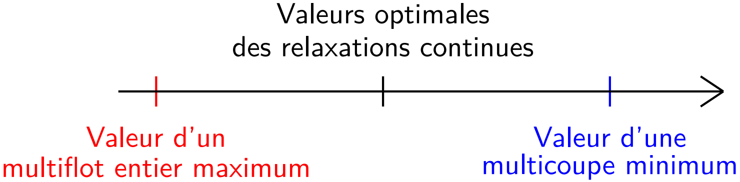
Dualité

Théorème
Les relaxations continues de ces deux PLNE sont duales l'une de l'autre.

Dualité

Théorème

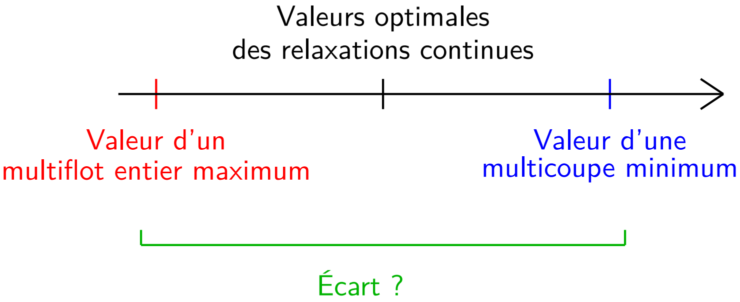
Les relaxations continues de ces deux PLNE sont duales l'une de l'autre.



Dualité

Théorème

Les relaxations continues de ces deux PLNE sont duales l'une de l'autre.



Solutions optimales des relaxations continues sur l'exemple

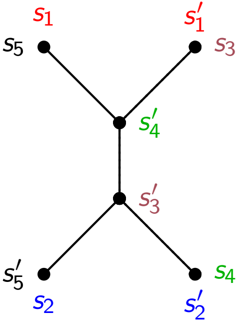
$$\max \sum_{i=1}^{p} f_i$$

$$\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u,v), \ \forall \text{ arête } uv$$

$$f_i \geq 0$$

Capacités = 1

p = 5



Solutions optimales des relaxations continues sur l'exemple

$$\max \sum_{i=1}^{p} f_i$$

$$\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u,v), \ \forall \text{ arête } uv$$

$$f_i \geq 0$$

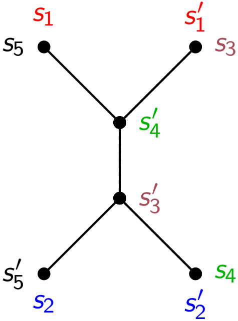
$$\min \sum_{\text{arêtes } uv \in A} capa(u,v) y_{uv}$$

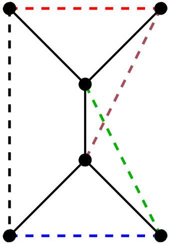
$$\sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1, \ \forall (s_i, s_i')$$

$$y_{uv} \geq 0$$

Capacités $= 1$

p $= 5$





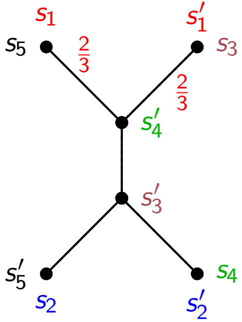
Solutions optimales des relaxations continues sur l'exemple

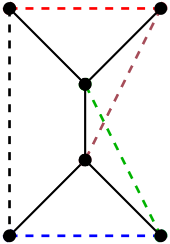
$$\max \sum_{i=1}^{p} f_i$$
$$\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u,v), \ \forall \text{ arête } uv$$
$$f_i \geq 0$$

$$\min \sum_{\text{arêtes } uv \in A} capa(u,v) y_{uv}$$
$$\sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1, \ \forall (s_i, s_i')$$
$$y_{uv} \geq 0$$

Capacités $= 1$

p $= 5$





Multiflot continu de valeur : $\frac{2}{3}$

Solutions optimales des relaxations continues sur l'exemple

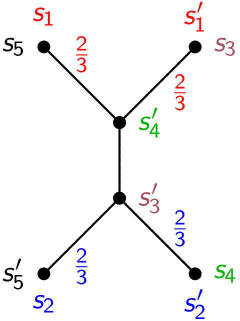
$$\max \sum_{i=1}^{p} f_i \qquad\qquad \min \sum_{\text{arêtes } uv \in A} capa(u,v)\, y_{uv}$$

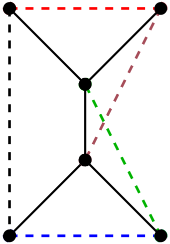
$$\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u,v),\ \forall \text{ arête } uv \qquad\qquad \sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1,\ \forall (s_i, s_i')$$

$$f_i \geq 0 \qquad\qquad y_{uv} \geq 0$$

Capacités = 1

p = 5





Multiflot continu de valeur :

$\frac{2}{3} + \frac{2}{3}$

Solutions optimales des relaxations continues sur l'exemple

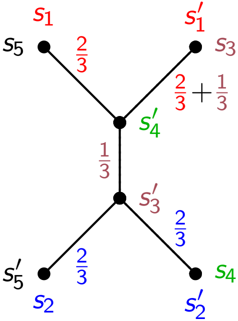
$$\max \sum_{i=1}^{p} f_i \qquad\qquad \min \sum_{\text{arêtes } uv \in A} capa(u,v) y_{uv}$$

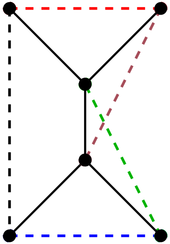
$$\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u,v), \ \forall \text{ arête } uv \qquad\qquad \sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1, \ \forall (s_i, s_i')$$

$$f_i \geq 0 \qquad\qquad y_{uv} \geq 0$$

Capacités = 1

p = 5





Multiflot continu de valeur :

$\frac{2}{3}+\frac{2}{3}+\frac{1}{3}$

Solutions optimales des relaxations continues sur l'exemple

$$\max \sum_{i=1}^{p} f_i$$

$$\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u,v), \ \forall \text{ arête } uv$$

$$f_i \geq 0$$

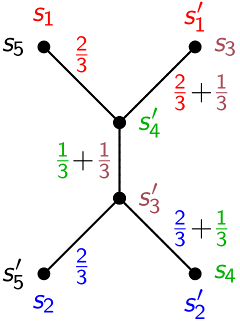
$$\min \sum_{\text{arêtes } uv \in A} capa(u,v) y_{uv}$$

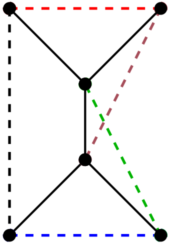
$$\sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1, \ \forall (s_i, s_i')$$

$$y_{uv} \geq 0$$

Capacités = 1

p = 5





Multiflot continu de valeur :

$\frac{2}{3}+\frac{2}{3}+\frac{1}{3}+\frac{1}{3}$

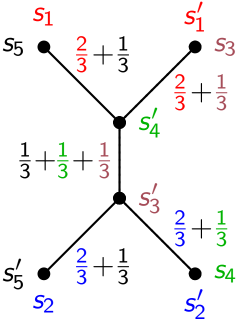
Solutions optimales des relaxations continues sur l'exemple

$$\max \sum_{i=1}^{p} f_i$$
$$\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u,v), \ \forall \text{ arête } uv$$
$$f_i \geq 0$$

$$\min \sum_{\text{arêtes } uv \in A} capa(u,v) y_{uv}$$
$$\sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1, \ \forall (s_i, s_i')$$
$$y_{uv} \geq 0$$

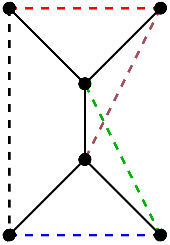
Capacités = 1

p = 5



Multiflot continu de valeur :

$\frac{2}{3}+\frac{2}{3}+\frac{1}{3}+\frac{1}{3}+\frac{1}{3}$



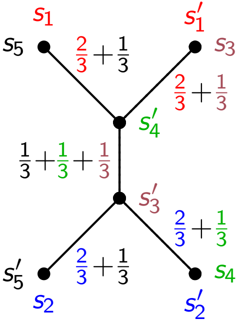
Solutions optimales des relaxations continues sur l'exemple

$$\max \sum_{i=1}^{p} f_i$$
$$\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u,v),\ \forall \text{ arête } uv$$
$$f_i \geq 0$$

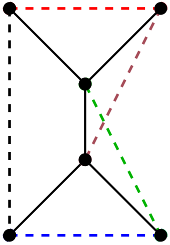
$$\min \sum_{\text{arêtes } uv \in A} capa(u,v) y_{uv}$$
$$\sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1,\ \forall (s_i, s_i')$$
$$y_{uv} \geq 0$$

Capacités = 1

p = 5



Multiflot continu de valeur :
$\frac{2}{3}+\frac{2}{3}+\frac{1}{3}+\frac{1}{3}+\frac{1}{3} = \frac{7}{3}$



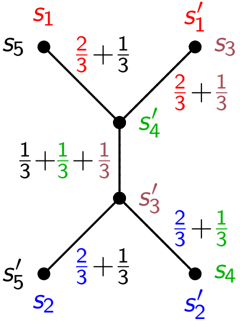
Solutions optimales des relaxations continues sur l'exemple

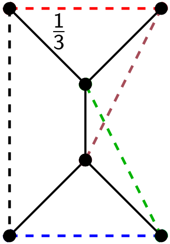
$$\max \sum_{i=1}^{p} f_i$$
$$\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u,v), \ \forall \text{ arête } uv$$
$$f_i \geq 0$$

$$\min \sum_{\text{arêtes } uv \in A} capa(u,v) y_{uv}$$
$$\sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1, \ \forall (s_i, s_i')$$
$$y_{uv} \geq 0$$

Capacités = 1

p = 5





Multiflot continu de valeur :
$$\frac{2}{3}+\frac{2}{3}+\frac{1}{3}+\frac{1}{3}+\frac{1}{3} = \frac{7}{3}$$

Solution duale de valeur :
$$\frac{1}{3}$$

Solutions optimales des relaxations continues sur l'exemple

$$\max \sum_{i=1}^{p} f_i$$

$$\sum_{i : \mu_i \text{ contient } uv} f_i \leq capa(u,v), \ \forall \text{ arête } uv$$

$$f_i \geq 0$$

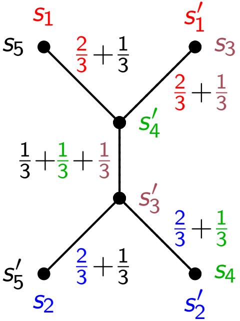
$$\min \sum_{\text{arêtes } uv \in A} capa(u,v) y_{uv}$$

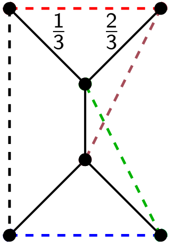
$$\sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1, \ \forall (s_i, s_i')$$

$$y_{uv} \geq 0$$

Capacités = 1

p = 5





Multiflot continu de valeur :

$$\frac{2}{3}+\frac{2}{3}+\frac{1}{3}+\frac{1}{3}+\frac{1}{3} = \frac{7}{3}$$

Solution duale de valeur :

$$\frac{1}{3}+\frac{2}{3}$$

Solutions optimales des relaxations continues sur l'exemple

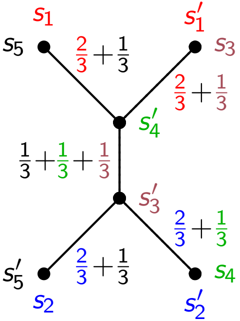
$$\max \sum_{i=1}^{p} f_i \qquad\qquad \min \sum_{\text{arêtes } uv \in A} capa(u,v) y_{uv}$$

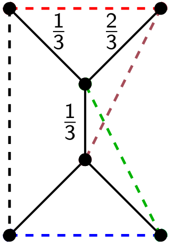
$$\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u,v), \; \forall \text{ arête } uv \qquad\qquad \sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1, \; \forall (s_i, s_i')$$

$$f_i \geq 0 \qquad\qquad y_{uv} \geq 0$$

Capacités = 1

p = 5





Multiflot continu de valeur :
$\frac{2}{3}+\frac{2}{3}+\frac{1}{3}+\frac{1}{3}+\frac{1}{3} = \frac{7}{3}$

Solution duale de valeur :
$\frac{1}{3}+\frac{2}{3}+\frac{1}{3}$

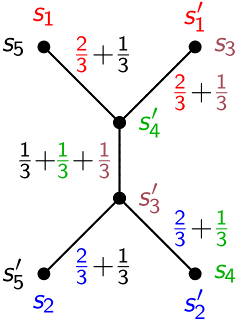
Solutions optimales des relaxations continues sur l'exemple

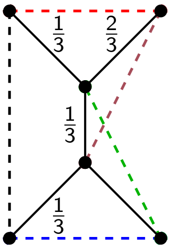
$$\max \sum_{i=1}^{p} f_i$$
$$\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u,v), \ \forall \text{ arête } uv$$
$$f_i \geq 0$$

$$\min \sum_{\text{arêtes } uv \in A} capa(u,v) y_{uv}$$
$$\sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1, \ \forall (s_i, s_i')$$
$$y_{uv} \geq 0$$

Capacités $= 1$

p $= 5$





Multiflot continu de valeur :
$\frac{2}{3}+\frac{2}{3}+\frac{1}{3}+\frac{1}{3}+\frac{1}{3}=\frac{7}{3}$

Solution duale de valeur :
$\frac{1}{3}+\frac{2}{3}+\frac{1}{3}+\frac{1}{3}$

Solutions optimales des relaxations continues sur l'exemple

$$\max \sum_{i=1}^{p} f_i$$

$$\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u,v), \ \forall \text{ arête } uv$$

$$f_i \geq 0$$

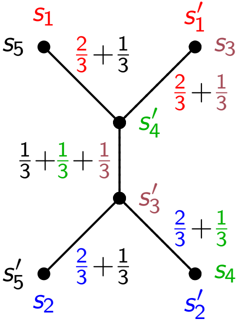
$$\min \sum_{\text{arêtes } uv \in A} capa(u,v) y_{uv}$$

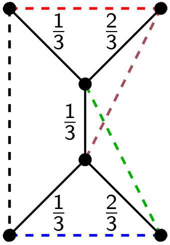
$$\sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1, \ \forall (s_i, s_i')$$

$$y_{uv} \geq 0$$

Capacités = 1

p = 5





Multiflot continu de valeur : $\frac{2}{3}+\frac{2}{3}+\frac{1}{3}+\frac{1}{3}+\frac{1}{3}=\frac{7}{3}$

Solution duale de valeur : $\frac{1}{3}+\frac{2}{3}+\frac{1}{3}+\frac{1}{3}+\frac{2}{3}$

Solutions optimales des relaxations continues sur l'exemple

$$\max \sum_{i=1}^{p} f_i$$

$$\sum_{i:\mu_i \text{ contient } uv} f_i \leq capa(u,v),\ \forall \text{ arête } uv$$

$$f_i \geq 0$$

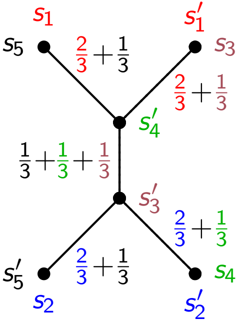
$$\min \sum_{\text{arêtes } uv \in A} capa(u,v) y_{uv}$$

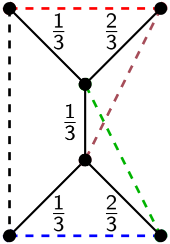
$$\sum_{\text{arêtes } uv \text{ appartenant à } \mu_i} y_{uv} \geq 1,\ \forall (s_i, s_i')$$

$$y_{uv} \geq 0$$

Capacités = 1

p = 5

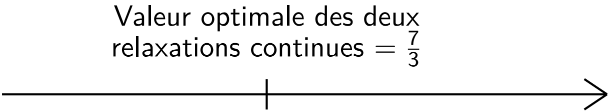




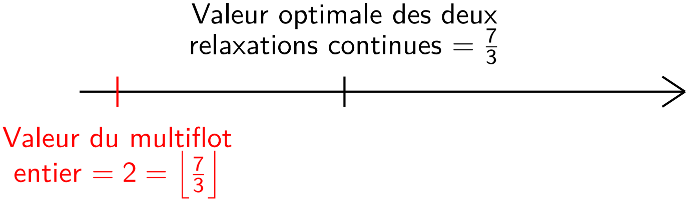
Multiflot continu de valeur :
$\frac{2}{3}+\frac{2}{3}+\frac{1}{3}+\frac{1}{3}+\frac{1}{3} = \frac{7}{3}$

Solution duale de valeur :
$\frac{1}{3}+\frac{2}{3}+\frac{1}{3}+\frac{1}{3}+\frac{2}{3} = \frac{7}{3}$

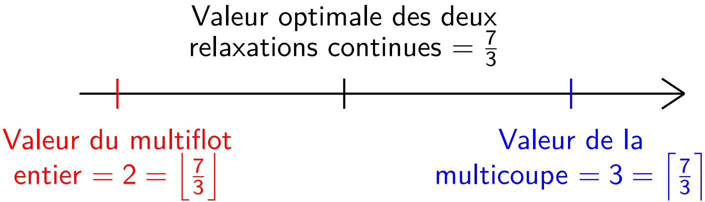
Solutions optimales entières dans cet exemple



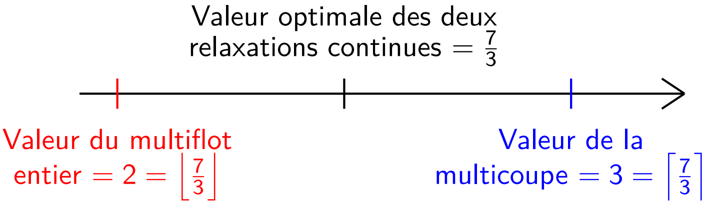
Solutions optimales entières dans cet exemple



Solutions optimales entières dans cet exemple



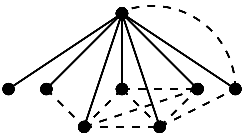
Solutions optimales entières dans cet exemple



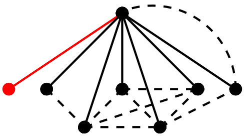
$\Rightarrow$ Les deux solutions entières précédentes sont donc optimales !

Étoiles avec capacités 1

Correspondance entre les feuilles et les sources/puits

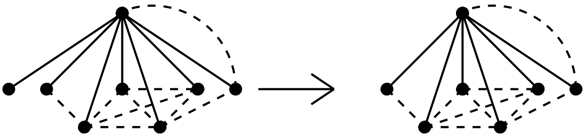


Correspondance entre les feuilles et les sources/puits



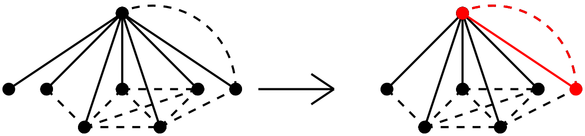
On supprime les feuilles
sans sources/puits

Correspondance entre les feuilles et les sources/puits



On supprime les feuilles
sans sources/puits

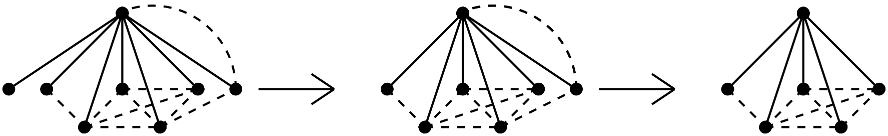
Correspondance entre les feuilles et les sources/puits



On supprime les feuilles sans sources/puits

On supprime toute paire source-puits entre le centre et une feuille :
Multicoupe $\Rightarrow$ arête supprimée
Multiflot entier $\Rightarrow$ arête saturée

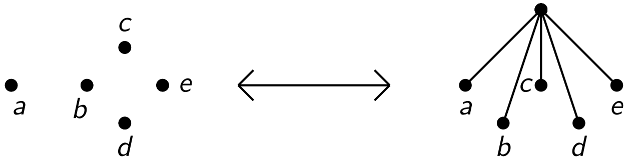
Correspondance entre les feuilles et les sources/puits



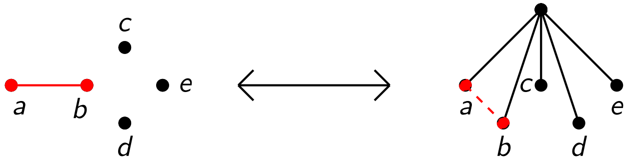
On supprime les feuilles sans sources/puits

On supprime toute paire source-puits entre le centre et une feuille :
Multicoupe ⇒ arête supprimée
Multiflot entier ⇒ arête saturée

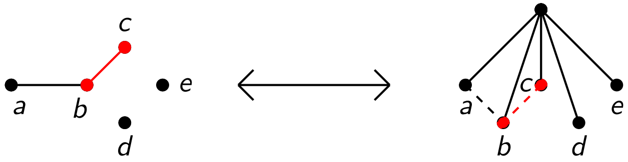
Graphe associé aux paires source-puits



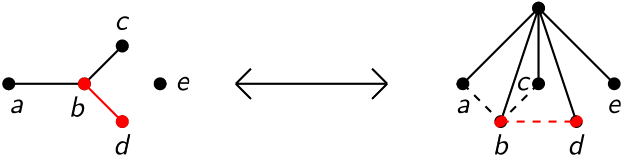
Graphe associé aux paires source-puits



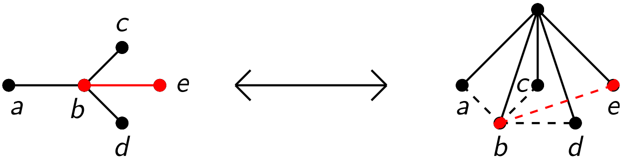
Graphe associé aux paires source-puits



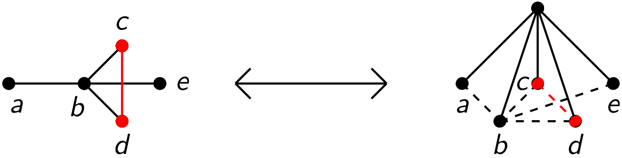
Graphe associé aux paires source-puits



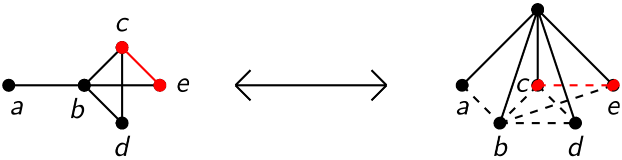
Graphe associé aux paires source-puits



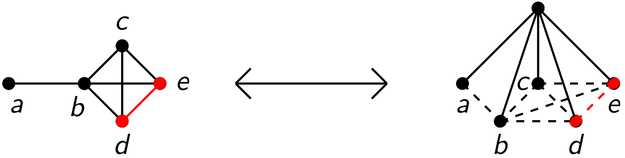
Graphe associé aux paires source-puits



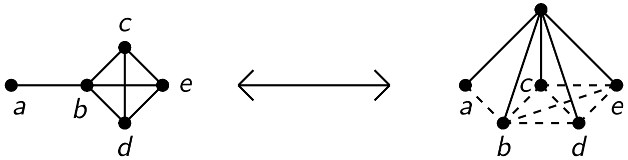
Graphe associé aux paires source-puits



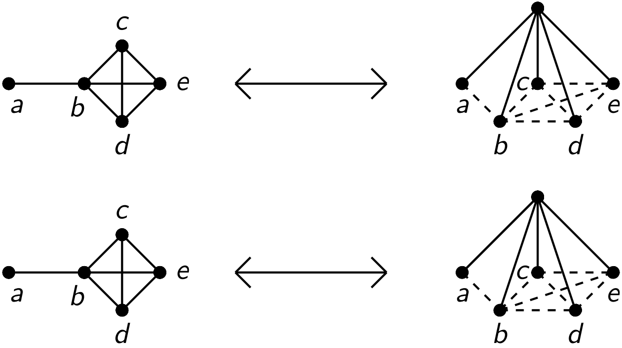
Graphe associé aux paires source-puits



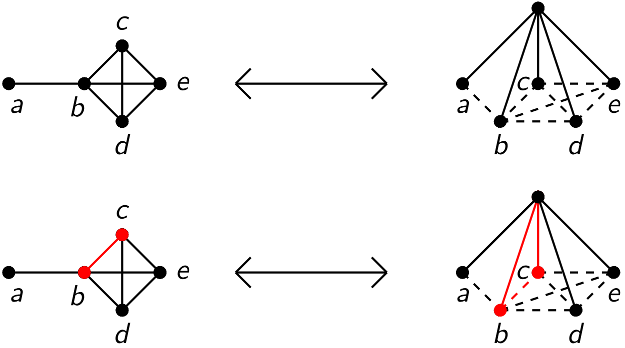
Graphe associé aux paires source-puits



Graphe associé aux paires source-puits

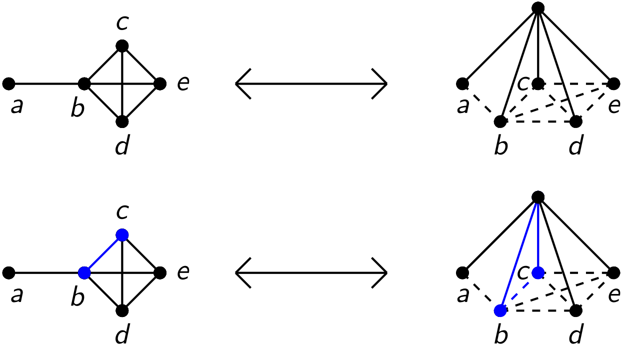


Graphe associé aux paires source-puits



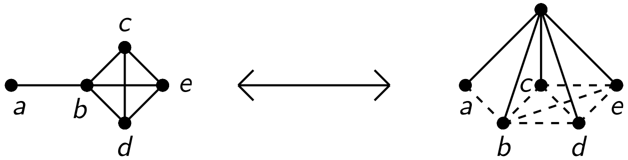
Couplage (matching)

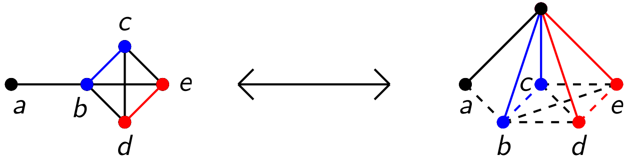
Graphe associé aux paires source-puits



Couplage (matching)

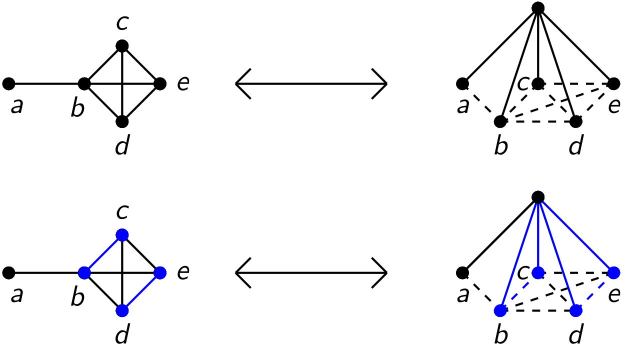
Graphe associé aux paires source-puits





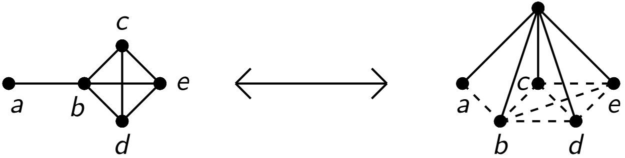
Couplage (matching)

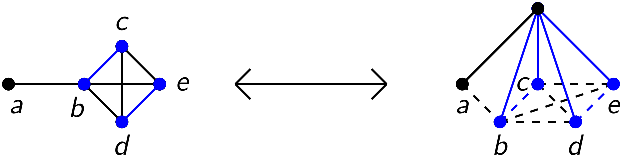
Graphe associé aux paires source-puits



Couplage (matching)

Graphe associé aux paires source-puits

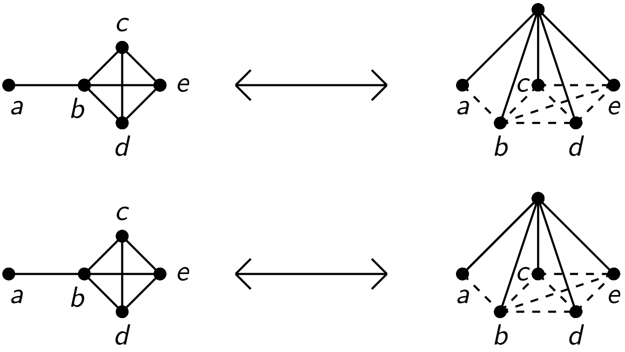




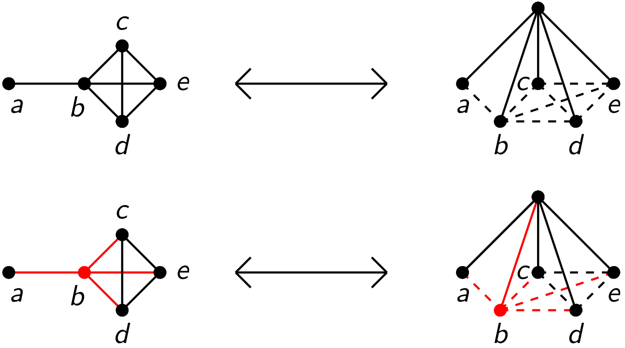
Couplage (matching)

Multiflot entier
(de même valeur)
$=$ chemins disjoints

Graphe associé aux paires source-puits

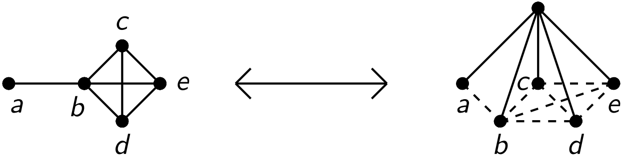


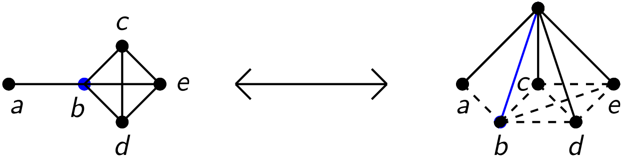
Graphe associé aux paires source-puits



Transversal (vertex cover)

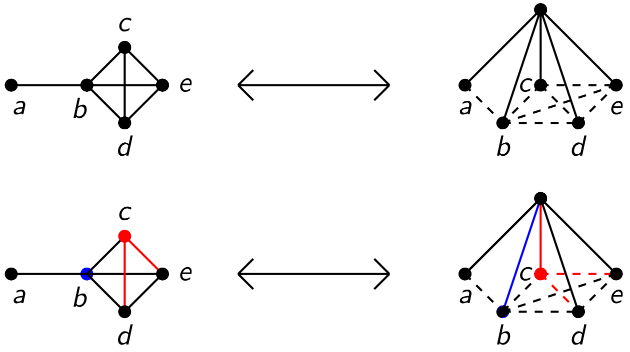
Graphe associé aux paires source-puits





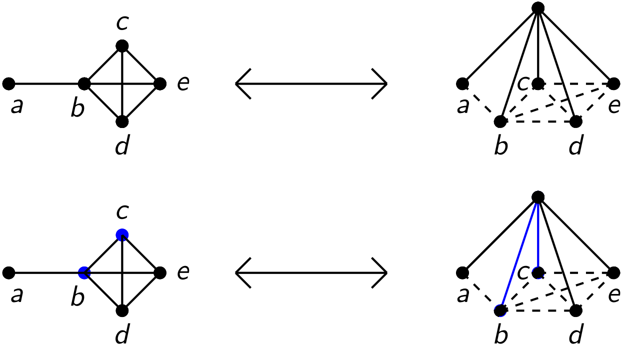
Transversal (vertex cover)

Graphe associé aux paires source-puits



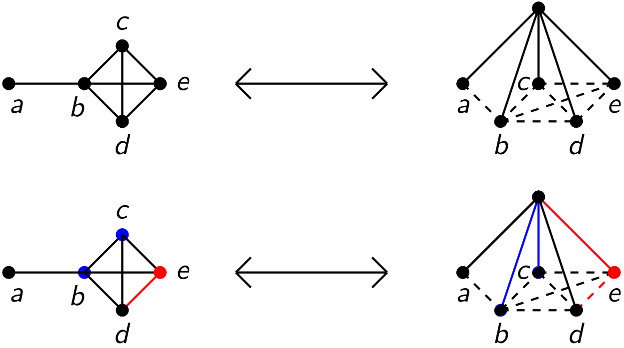
Transversal (vertex cover)

Graphe associé aux paires source-puits



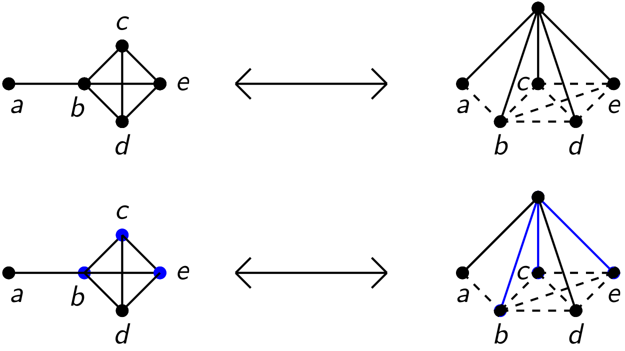
Transversal (vertex cover)

Graphe associé aux paires source-puits



Transversal (vertex cover)

Graphe associé aux paires source-puits



Transversal (vertex cover)

Multicoupe
(de même valeur)

Borne inférieure sur l'écart entre les deux problèmes

Que peut-il se passer dans le pire cas ?

Borne inférieure sur l'écart entre les deux problèmes

Que peut-il se passer dans le pire cas ?

Théorème

*Il est possible que la valeur d'une multicoupe minimum soit **deux fois** celle d'un multiflot entier maximum, même dans des étoiles avec capacités 1.*

Borne inférieure sur l'écart entre les deux problèmes

Que peut-il se passer dans le pire cas ?

Théorème

*Il est possible que la valeur d'une multicoupe minimum soit **deux fois** celle d'un multiflot entier maximum, même dans des étoiles avec capacités 1.*

Par exemple, dans un triangle :

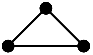
Borne inférieure sur l'écart entre les deux problèmes

Que peut-il se passer dans le pire cas ?

Théorème

*Il est possible que la valeur d'une multicoupe minimum soit **deux fois** celle d'un multiflot entier maximum, même dans des étoiles avec capacités 1.*

Par exemple, dans un triangle :



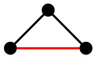
Borne inférieure sur l'écart entre les deux problèmes

Que peut-il se passer dans le pire cas ?

Théorème

Il est possible que la valeur d'une multicoupe minimum soit ***deux fois*** *celle d'un multiflot entier maximum, même dans des étoiles avec capacités 1.*

Par exemple, dans un triangle :



Couplage maximum de taille 1

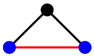
Borne inférieure sur l'écart entre les deux problèmes

Que peut-il se passer dans le pire cas ?

Théorème

*Il est possible que la valeur d'une multicoupe minimum soit **deux fois** celle d'un multiflot entier maximum, même dans des étoiles avec capacités 1.*

Par exemple, dans un triangle :



Couplage maximum de taille 1

Transversal minimum de taille 2

Complexité des problèmes dans les étoiles avec capacités 1

Bilan : des problèmes plus généraux qu'il n'y paraît.

Complexité des problèmes dans les étoiles avec capacités 1

Bilan : des problèmes plus généraux qu'il n'y paraît.

Théorème

Multiflot entier maximum $\Leftrightarrow$ *couplage maximum (graphe quelconque).*

$\Rightarrow$ ***Problème polynomial***

Complexité des problèmes dans les étoiles avec capacités 1

Bilan : des problèmes plus généraux qu'il n'y paraît.

Théorème

Multiflot entier maximum $\Leftrightarrow$ *couplage maximum (graphe quelconque).*
$\Rightarrow$ ***Problème polynomial***

Théorème

Multicoupe minimum $\Leftrightarrow$ *transversal minimum (graphe quelconque).*
$\Rightarrow$ ***Problème NP-difficile***

Multiflot entier maximum dans les arbres quelconques

Algorithme polynomial dans les arbres avec capacités 1

Multiflot entier = ensemble de chemins disjoints.
Étoile $\Rightarrow$ recherche d'un couplage maximum.

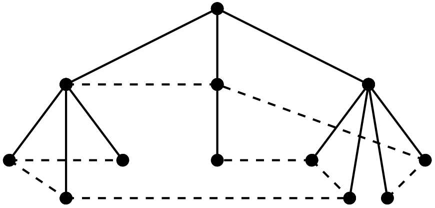
Algorithme polynomial dans les arbres avec capacités 1

Multiflot entier = ensemble de chemins disjoints.
Étoile $\Rightarrow$ recherche d'un couplage maximum.
Arbre quelconque ? Traiter une étoile de l'arbre, puis le réduire.

Algorithme polynomial dans les arbres avec capacités 1

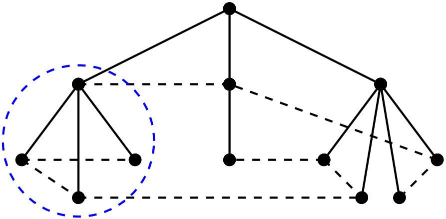
Multiflot entier = ensemble de chemins disjoints.
Étoile ⇒ recherche d'un couplage maximum.
Arbre quelconque ? Traiter une étoile de l'arbre, puis le réduire.

Racine quelconque



Algorithme polynomial dans les arbres avec capacités 1

Multiflot entier = ensemble de chemins disjoints.
Étoile ⇒ recherche d'un couplage maximum.
Arbre quelconque ? Traiter une étoile de l'arbre, puis le réduire.

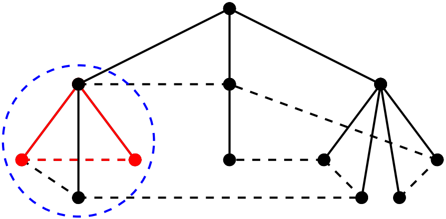


Algorithme polynomial dans les arbres avec capacités 1

Multiflot entier = ensemble de chemins disjoints.

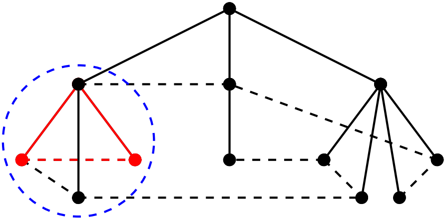
Étoile ⇒ recherche d'un couplage maximum.

Arbre quelconque ? Traiter une étoile de l'arbre, puis le réduire.



Algorithme polynomial dans les arbres avec capacités 1

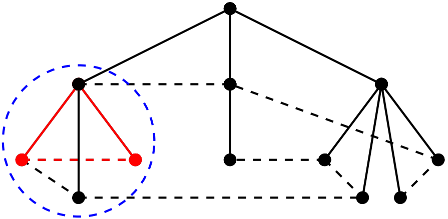
Multiflot entier = ensemble de chemins disjoints.
Étoile ⇒ recherche d'un couplage maximum.
Arbre quelconque ? Traiter une étoile de l'arbre, puis le réduire.



Multiflot entier maximum de valeur 1

Algorithme polynomial dans les arbres avec capacités 1

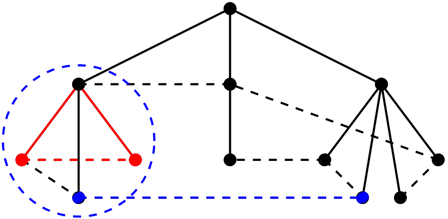
Multiflot entier = ensemble de chemins disjoints.
Étoile ⇒ recherche d'un couplage maximum.
Arbre quelconque ? Traiter une étoile de l'arbre, puis le réduire.



⇒ On envoie cette unité de flot

Algorithme polynomial dans les arbres avec capacités 1

Multiflot entier = ensemble de chemins disjoints.
Étoile ⇒ recherche d'un couplage maximum.
Arbre quelconque ? Traiter une étoile de l'arbre, puis le réduire.



⇒ On envoie cette unité de flot

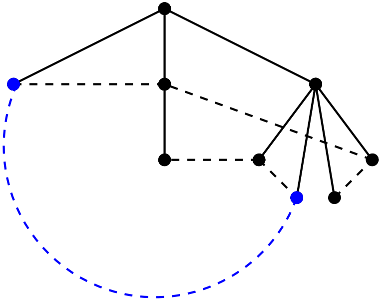
⇒ On contracte l'étoile, et on "remonte" une paire sources-puits

Algorithme polynomial dans les arbres avec capacités 1

Multiflot entier = ensemble de chemins disjoints.

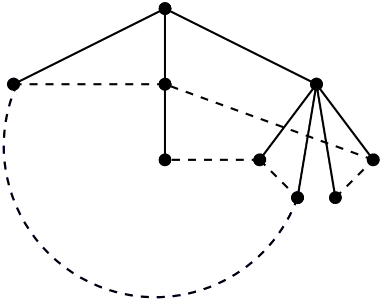
Étoile ⇒ recherche d'un couplage maximum.

Arbre quelconque ? Traiter une étoile de l'arbre, puis le réduire.



Algorithme polynomial dans les arbres avec capacités 1

Multiflot entier = ensemble de chemins disjoints.
Étoile ⇒ recherche d'un couplage maximum.
Arbre quelconque ? Traiter une étoile de l'arbre, puis le réduire.



⇒ Puis, on recommence...

NP-difficulté dans les arbres avec capacités 1 et 2

Réduction à partir du **problème NP-complet de la partition en triplets disjoints** : dans un ensemble T de N triplets $(x_i \in X, y_j \in Y, z_k \in Z)$ avec $|X| = |Y| = |Z| = n$, déterminer s'il existe n triplets disjoints.

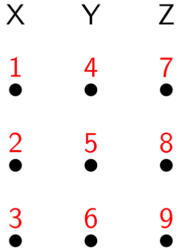
Exemple : $n = 3$, $N = 7$, $X = \{1, 2, 3\}$, $Y = \{4, 5, 6\}$ et $Z = \{7, 8, 9\}$.

(3,4,8)
(1,4,7)
(1,5,7)
(3,6,9)
(2,6,9)
(1,5,9)
(2,6,8)

NP-difficulté dans les arbres avec capacités 1 et 2

Réduction à partir du **problème NP-complet de la partition en triplets disjoints** : dans un ensemble T de N triplets $(x_i \in X, y_j \in Y, z_k \in Z)$ avec $|X| = |Y| = |Z| = n$, déterminer s'il existe n triplets disjoints.
Exemple : $n = 3$, $N = 7$, $X = \{1, 2, 3\}$, $Y = \{4, 5, 6\}$ et $Z = \{7, 8, 9\}$.

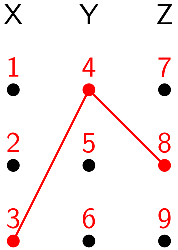
(3,4,8)
(1,4,7)
(1,5,7)
(3,6,9)
(2,6,9)
(1,5,9)
(2,6,8)



NP-difficulté dans les arbres avec capacités 1 et 2

Réduction à partir du **problème NP-complet de la partition en triplets disjoints** : dans un ensemble T de N triplets $(x_i \in X, y_j \in Y, z_k \in Z)$ avec $|X| = |Y| = |Z| = n$, déterminer s'il existe n triplets disjoints.
Exemple : $n = 3$, $N = 7$, $X = \{1, 2, 3\}$, $Y = \{4, 5, 6\}$ et $Z = \{7, 8, 9\}$.

(3,4,8)
(1,4,7)
(1,5,7)
(3,6,9)
(2,6,9)
(1,5,9)
(2,6,8)

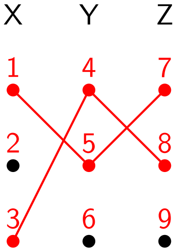


NP-difficulté dans les arbres avec capacités 1 et 2

Réduction à partir du **problème NP-complet de la partition en triplets disjoints** : dans un ensemble T de N triplets $(x_i \in X, y_j \in Y, z_k \in Z)$ avec $|X| = |Y| = |Z| = n$, déterminer s'il existe n triplets disjoints.

Exemple : $n = 3$, $N = 7$, $X = \{1, 2, 3\}$, $Y = \{4, 5, 6\}$ et $Z = \{7, 8, 9\}$.

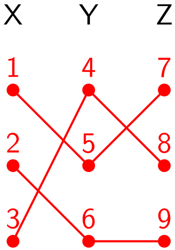
(3,4,8)
(1,4,7)
(1,5,7)
(3,6,9)
(2,6,9)
(1,5,9)
(2,6,8)



NP-difficulté dans les arbres avec capacités 1 et 2

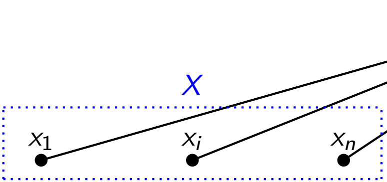
Réduction à partir du **problème NP-complet de la partition en triplets disjoints** : dans un ensemble T de N triplets $(x_i \in X, y_j \in Y, z_k \in Z)$ avec $|X| = |Y| = |Z| = n$, déterminer s'il existe n triplets disjoints.
Exemple : $n = 3$, $N = 7$, $X = \{1, 2, 3\}$, $Y = \{4, 5, 6\}$ et $Z = \{7, 8, 9\}$.

(3,4,8)
(1,4,7)
(1,5,7)
(3,6,9)
(2,6,9)
(1,5,9)
(2,6,8)



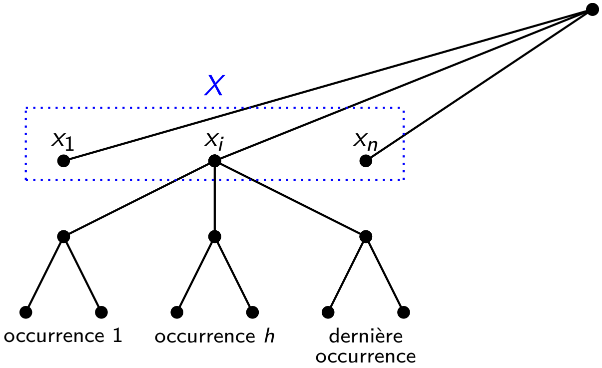
NP-difficulté dans les arbres avec capacités 1 et 2

Réduction à partir du **problème NP-complet de la partition en triplets disjoints** : dans un ensemble T de N triplets $(x_i \in X, y_j \in Y, z_k \in Z)$ avec $|X| = |Y| = |Z| = n$, déterminer s'il existe n triplets disjoints.



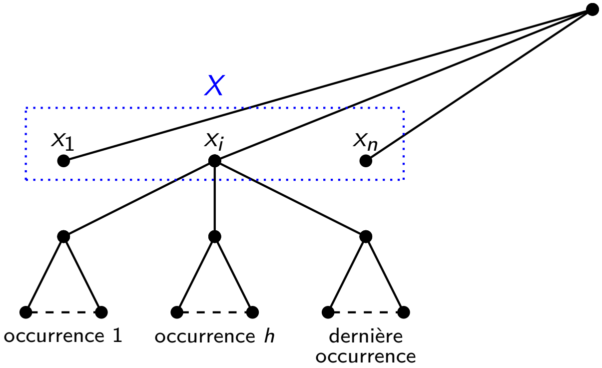
NP-difficulté dans les arbres avec capacités 1 et 2

Réduction à partir du **problème NP-complet de la partition en triplets disjoints** : dans un ensemble T de N triplets $(x_i \in X, y_j \in Y, z_k \in Z)$ avec $|X| = |Y| = |Z| = n$, déterminer s'il existe n triplets disjoints.



NP-difficulté dans les arbres avec capacités 1 et 2

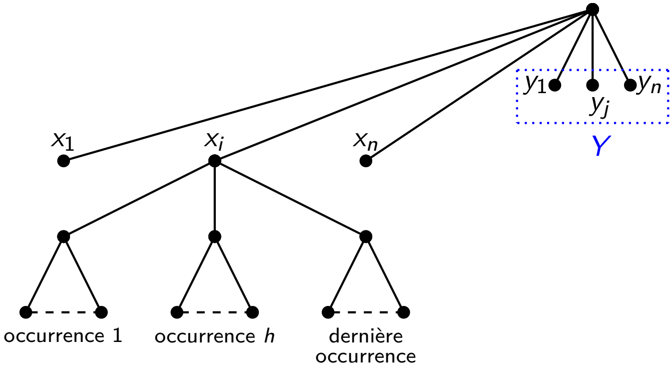
Réduction à partir du **problème NP-complet de la partition en triplets disjoints** : dans un ensemble T de N triplets $(x_i \in X, y_j \in Y, z_k \in Z)$ avec $|X| = |Y| = |Z| = n$, déterminer s'il existe n triplets disjoints.



N = nombre de telles paires sources-puits

NP-difficulté dans les arbres avec capacités 1 et 2

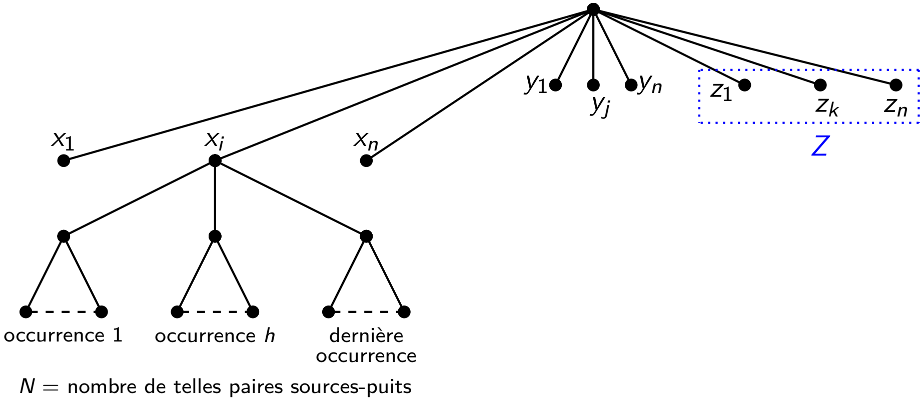
Réduction à partir du **problème NP-complet de la partition en triplets disjoints** : dans un ensemble T de N triplets $(x_i \in X, y_j \in Y, z_k \in Z)$ avec $|X| = |Y| = |Z| = n$, déterminer s'il existe n triplets disjoints.



N = nombre de telles paires sources-puits

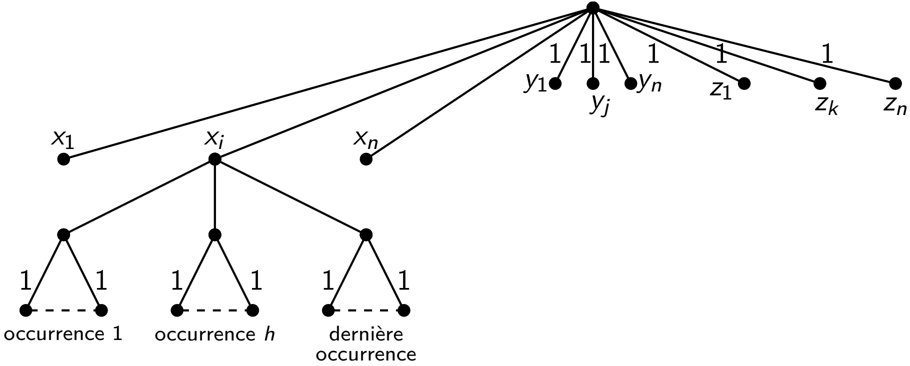
NP-difficulté dans les arbres avec capacités 1 et 2

Réduction à partir du **problème NP-complet de la partition en triplets disjoints** : dans un ensemble T de N triplets $(x_i \in X, y_j \in Y, z_k \in Z)$ avec $|X| = |Y| = |Z| = n$, déterminer s'il existe n triplets disjoints.



NP-difficulté dans les arbres avec capacités 1 et 2

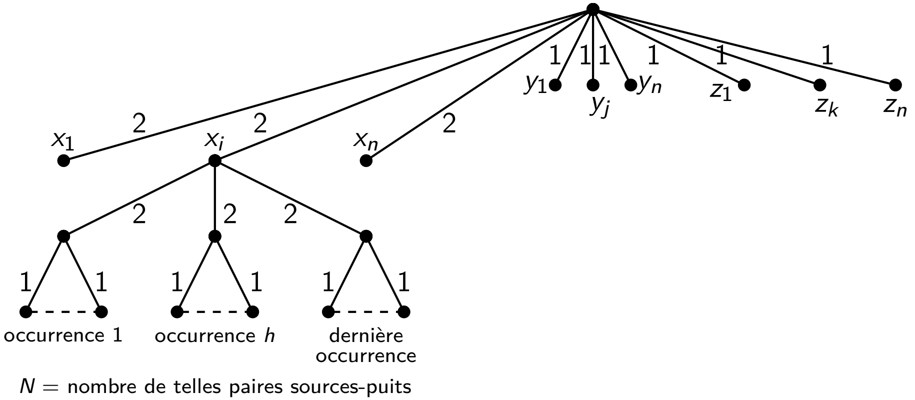
Réduction à partir du **problème NP-complet de la partition en triplets disjoints** : dans un ensemble T de N triplets $(x_i \in X, y_j \in Y, z_k \in Z)$ avec $|X| = |Y| = |Z| = n$, déterminer s'il existe n triplets disjoints.



N = nombre de telles paires sources-puits

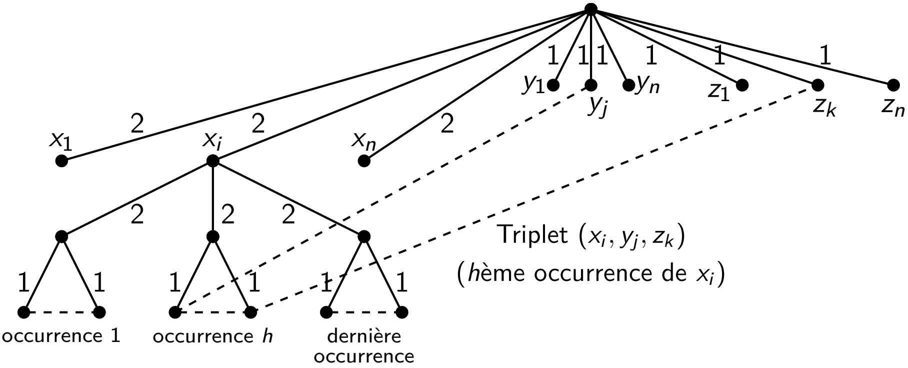
NP-difficulté dans les arbres avec capacités 1 et 2

Réduction à partir du **problème NP-complet de la partition en triplets disjoints** : dans un ensemble T de N triplets $(x_i \in X, y_j \in Y, z_k \in Z)$ avec $|X| = |Y| = |Z| = n$, déterminer s'il existe n triplets disjoints.



NP-difficulté dans les arbres avec capacités 1 et 2

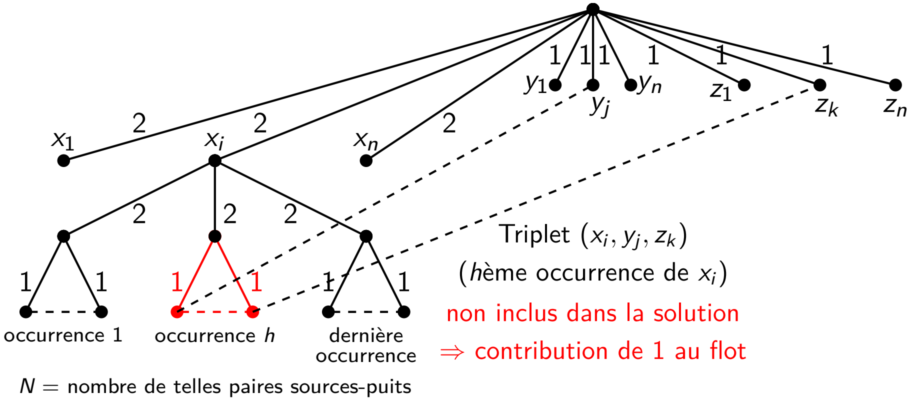
Réduction à partir du **problème NP-complet de la partition en triplets disjoints** : dans un ensemble T de N triplets $(x_i \in X, y_j \in Y, z_k \in Z)$ avec $|X| = |Y| = |Z| = n$, déterminer s'il existe n triplets disjoints.



N = nombre de telles paires sources-puits

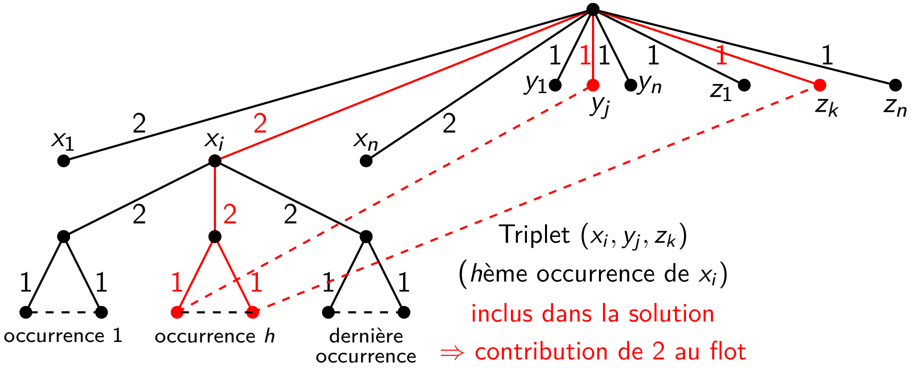
NP-difficulté dans les arbres avec capacités 1 et 2

Réduction à partir du **problème NP-complet de la partition en triplets disjoints** : dans un ensemble T de N triplets $(x_i \in X, y_j \in Y, z_k \in Z)$ avec $|X| = |Y| = |Z| = n$, déterminer s'il existe n triplets disjoints.



NP-difficulté dans les arbres avec capacités 1 et 2

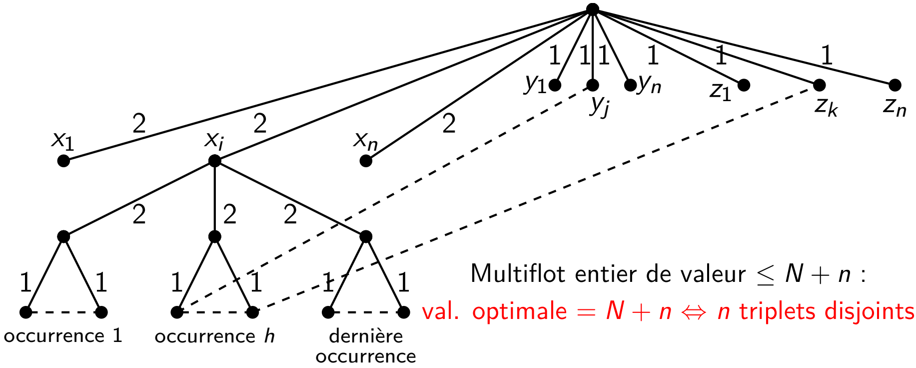
Réduction à partir du **problème NP-complet de la partition en triplets disjoints** : dans un ensemble T de N triplets $(x_i \in X, y_j \in Y, z_k \in Z)$ avec $|X| = |Y| = |Z| = n$, déterminer s'il existe n triplets disjoints.



N = nombre de telles paires sources-puits

NP-difficulté dans les arbres avec capacités 1 et 2

Réduction à partir du **problème NP-complet de la partition en triplets disjoints** : dans un ensemble T de N triplets $(x_i \in X, y_j \in Y, z_k \in Z)$ avec $|X| = |Y| = |Z| = n$, déterminer s'il existe n triplets disjoints.



N = nombre de telles paires sources-puits

Multiflot entier de valeur $\leq N + n$:

val. optimale $= N + n \Leftrightarrow n$ triplets disjoints

Bilan sur la complexité du problème du multiflot entier maximum dans les arbres quelconques

Théorème

Dans les arbres, le problème du multiflot entier maximum est polynomial si les capacités sont 1.

Bilan sur la complexité du problème du multiflot entier maximum dans les arbres quelconques

Théorème

Dans les arbres, le problème du multiflot entier maximum est polynomial si les capacités sont 1.

Théorème

Dans les arbres, le problème du multiflot entier maximum est NP-difficile, même si les capacités possibles sont 1 et 2.

Multiflots et multicoupes approchés dans les arbres quelconques

Algorithme approché primal-dual (glouton)

- On enracine l'arbre en un sommet quelconque.

Algorithme approché primal-dual (glouton)

- On enracine l'arbre en un sommet quelconque.
- **Multiflot entier** :

Algorithme approché primal-dual (glouton)

- On enracine l'arbre en un sommet quelconque.
- **Multiflot entier** :
  - **On parcourt l'arbre de bas en haut** : pour tout i tel que μ_i devient entièrement contenue dans le sous-arbre enraciné au sommet courant, on choisit f_i aussi grand que possible.

Algorithme approché primal-dual (glouton)

- On enracine l'arbre en un sommet quelconque.
- **Multiflot entier** :
  - **On parcourt l'arbre de bas en haut** : pour tout i tel que μ_i devient entièrement contenue dans le sous-arbre enraciné au sommet courant, on choisit f_i aussi grand que possible.
  - On stocke les arêtes ainsi saturées.

Algorithme approché primal-dual (glouton)

- On enracine l'arbre en un sommet quelconque.
- **Multiflot entier** :
  - **On parcourt l'arbre de bas en haut** : pour tout i tel que μ_i devient entièrement contenue dans le sous-arbre enraciné au sommet courant, on choisit f_i aussi grand que possible.
  - On stocke les arêtes ainsi saturées.
- **Multicoupe** :

Algorithme approché primal-dual (glouton)

- On enracine l'arbre en un sommet quelconque.
- **Multiflot entier** :
  - **On parcourt l'arbre de bas en haut** : pour tout i tel que μ_i devient entièrement contenue dans le sous-arbre enraciné au sommet courant, on choisit f_i aussi grand que possible.
  - On stocke les arêtes ainsi saturées.
- **Multicoupe** :
  - **On parcourt l'arbre de haut en bas** : on inclut dans la multicoupe les arêtes "les plus hautes" parmi celles saturées par le multiflot.

Propriétés des solutions approchées obtenues

Lemme

Les 2 solutions obtenues par l'algorithme primal-dual sont admissibles.

Propriétés des solutions approchées obtenues

Lemme

Les 2 solutions obtenues par l'algorithme primal-dual sont admissibles.

Lemme

La multicoupe obtenue ne contient que des arêtes saturées, et chaque unité du multiflot entier traverse au plus deux arêtes de la multicoupe.

Propriétés des solutions approchées obtenues

Lemme

Les 2 solutions obtenues par l'algorithme primal-dual sont admissibles.

Lemme

La multicoupe obtenue ne contient que des arêtes saturées, et chaque unité du multiflot entier traverse au plus deux arêtes de la multicoupe.

Ces deux lemmes impliquent :

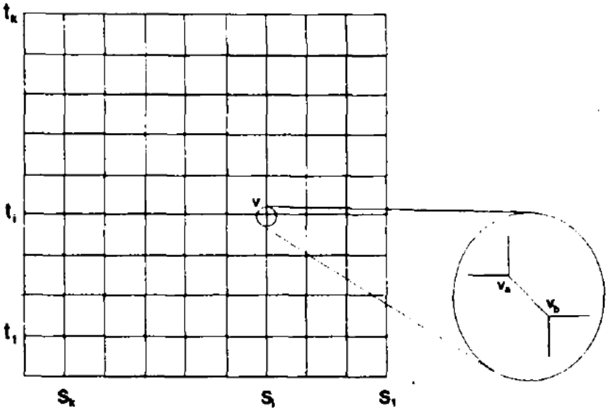
Théorème

La valeur de la multicoupe obtenue par l'algorithme est au plus deux fois celle du multiflot entier obtenu*.*

Écart entre les valeurs optimales

Théorème

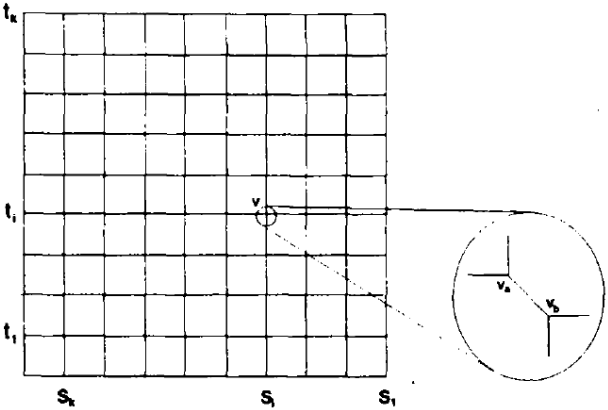
Dans le pire cas, le ratio entre la valeur d'une multicoupe minimum et celle d'un multiflot entier maximum est au moins $\max(p, \sqrt{n})$ ***dans les graphes planaires à*** n ***sommets.***



Écart entre les valeurs optimales

Théorème

Dans le pire cas, le ratio entre la valeur d'une multicoupe minimum et celle d'un multiflot entier maximum est au moins $\max(p, \sqrt{n})$ ***dans les graphes planaires à*** n ***sommets***.



[Kleinberg (2005)] $\Rightarrow O(1)$ si pas de sommet interne de degré impair.

Bilan et conclusion

Pour conclure...

- Problèmes très généraux, même dans les arbres (liens avec couplages, transversaux, partitions en triplets, etc.).

Pour conclure...

- Problèmes très généraux, même dans les arbres (liens avec couplages, transversaux, partitions en triplets, etc.).
- Quelques cas particuliers "faciles", résolus via des algorithmes élégants et non triviaux.

Pour conclure...

- Problèmes très généraux, même dans les arbres (liens avec couplages, transversaux, partitions en triplets, etc.).
- Quelques cas particuliers "faciles", résolus via des algorithmes élégants et non triviaux.
- Un algorithme approché, dont la garantie est la meilleure possible, et qui est en outre une belle illustration des méthodes primales-duales.

Pour conclure...

- Problèmes très généraux, même dans les arbres (liens avec couplages, transversaux, partitions en triplets, etc.).
- Quelques cas particuliers "faciles", résolus via des algorithmes élégants et non triviaux.
- Un algorithme approché, dont la garantie est la meilleure possible, et qui est en outre une belle illustration des méthodes primales-duales.
- Encore aujourd'hui, principale classe de graphes admettant un algorithme approché avec une garantie aussi bonne pour les deux problèmes, et un algorithme polynomial pour calculer un multiflot entier maximum avec des capacités 1.

Pour conclure...

- Problèmes très généraux, même dans les arbres (liens avec couplages, transversaux, partitions en triplets, etc.).
- Quelques cas particuliers "faciles", résolus via des algorithmes élégants et non triviaux.
- Un algorithme approché, dont la garantie est la meilleure possible, et qui est en outre une belle illustration des méthodes primales-duales.
- Encore aujourd'hui, principale classe de graphes admettant un algorithme approché avec une garantie aussi bonne pour les deux problèmes, et un algorithme polynomial pour calculer un multiflot entier maximum avec des capacités 1.

Quelques points non évoqués :

Pour conclure...

- Problèmes très généraux, même dans les arbres (liens avec couplages, transversaux, partitions en triplets, etc.).
- Quelques cas particuliers "faciles", résolus via des algorithmes élégants et non triviaux.
- Un algorithme approché, dont la garantie est la meilleure possible, et qui est en outre une belle illustration des méthodes primales-duales.
- Encore aujourd'hui, principale classe de graphes admettant un algorithme approché avec une garantie aussi bonne pour les deux problèmes, et un algorithme polynomial pour calculer un multiflot entier maximum avec des capacités 1.

Quelques points non évoqués :

- Lien avec les b-couplages.

Pour conclure...

- Problèmes très généraux, même dans les arbres (liens avec couplages, transversaux, partitions en triplets, etc.).
- Quelques cas particuliers "faciles", résolus via des algorithmes élégants et non triviaux.
- Un algorithme approché, dont la garantie est la meilleure possible, et qui est en outre une belle illustration des méthodes primales-duales.
- Encore aujourd'hui, principale classe de graphes admettant un algorithme approché avec une garantie aussi bonne pour les deux problèmes, et un algorithme polynomial pour calculer un multiflot entier maximum avec des capacités 1.

Quelques points non évoqués :

- Lien avec les *b*-couplages.
- Lien avec les couvertures d'ensembles représentables par des arbres.

Pour conclure...

- Problèmes très généraux, même dans les arbres (liens avec couplages, transversaux, partitions en triplets, etc.).
- Quelques cas particuliers "faciles", résolus via des algorithmes élégants et non triviaux.
- Un algorithme approché, dont la garantie est la meilleure possible, et qui est en outre une belle illustration des méthodes primales-duales.
- Encore aujourd'hui, principale classe de graphes admettant un algorithme approché avec une garantie aussi bonne pour les deux problèmes, et un algorithme polynomial pour calculer un multiflot entier maximum avec des capacités 1.

Quelques points non évoqués :

- Lien avec les b-couplages.
- Lien avec les couvertures d'ensembles représentables par des arbres.
- Preuve d'inapproximabilité pour le multiflot entier maximum.

Aller plus loin ?

Pour la multicoupe :

Aller plus loin ?

Pour la multicoupe :

- Saut d'intégrité de $O(1)$ dans les graphes planaires [Tardos, Vazirani (1993)].

Aller plus loin ?

Pour la multicoupe :

- Saut d'intégrité de $O(1)$ dans les graphes planaires [Tardos, Vazirani (1993)].
- Saut d'intégrité de $O(\log p)$ dans les graphes généraux [Garg et al. (1996)].

Merci pour votre attention !

Des questions ?