Flots

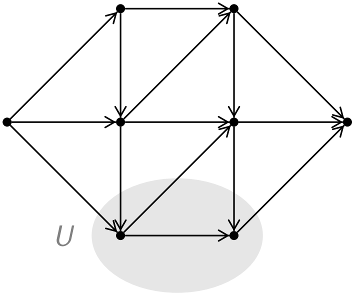
Plan

Plan

Arcs sortants et entrants

Définition

Soient $G = (V, A)$ un graphe orienté et $U \subseteq V$. Alors, on note $\delta^+(U)$ l'ensemble des arcs sortants de l'ensemble U (cocycle positif). De même, on note $\delta^-(U)$ l'ensemble des arcs entrants dans U (cocycle négatif).



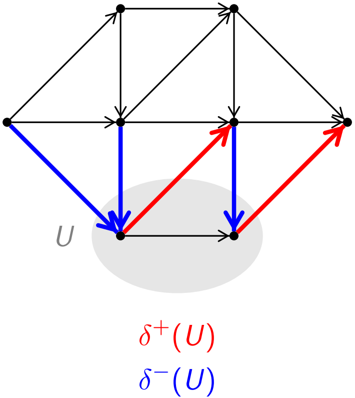
Rappel :

- arc sortant de l'ensemble U : arc dont le début est dans U et la fin dans $V \setminus U$.
- arc entrant de l'ensemble U : arc dont le début est dans $V \setminus U$ et la fin dans U.

Arcs sortants et entrants

Définition

Soient $G = (V, A)$ un graphe orienté et $U \subseteq V$. Alors, on note $\delta^+(U)$ l'ensemble des arcs sortants de l'ensemble U (cocycle positif). De même, on note $\delta^-(U)$ l'ensemble des arcs entrants dans U (cocycle négatif).



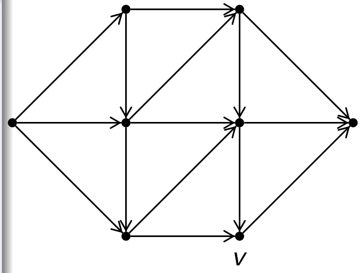
Rappel :

- arc sortant de l'ensemble U : arc dont le début est dans U et la fin dans $V \setminus U$.
- arc entrant de l'ensemble U : arc dont le début est dans $V \setminus U$ et la fin dans U.

Degré sortant et degré entrant

Définition

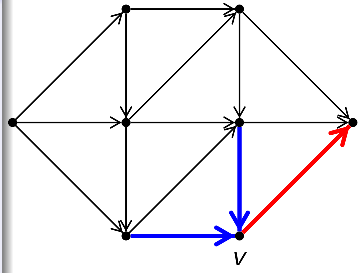
- le *degré entrant* $d^-(v)$, est le nombre d'arcs dont la fin est v.
- le *degré sortant* $d^+(v)$, est le nombre d'arcs dont le début est v.
- On a $d^-(v) = |\delta^-(\{v\})|$ et $d^+(v) = |\delta^+(\{v\})|$.
- v est une *source* si $d^-(v) = 0$.
- v est un *puits* si $d^+(v) = 0$.



Degré sortant et degré entrant

Définition

- le *degré entrant* $d^-(v)$, est le nombre d'arcs dont la fin est v.
- le *degré sortant* $d^+(v)$, est le nombre d'arcs dont le début est v.
- On a $d^-(v) = |\delta^-(\{v\})|$ et $d^+(v) = |\delta^+(\{v\})|$.
- v est une *source* si $d^-(v) = 0$.
- v est un *puits* si $d^+(v) = 0$.



$d^+(v) = 1$

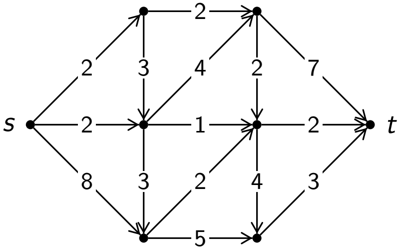
$d^-(v) = 2$

Réseaux

Définition

Un *réseau* est un graphe orienté $G = (V, A)$ avec :

- deux sommets spéciaux :
  - la source s tel que $d^{-}(s) = 0$
  - le puits t tel que $d^{+}(t) = 0$
- une fonction de *capacité* $c : A \to \mathbb{R}^{+}$.



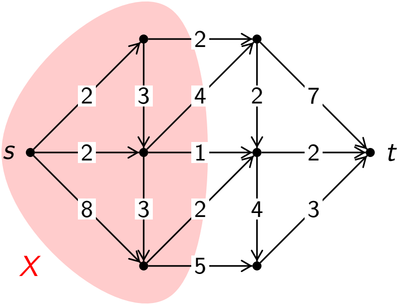
Coupes

Définition

$X \subset V$ est une $s - t$ *coupe* de G si $s \in X$ et $t \notin X$

Définition

La *capacité* d'une coupe X est $\text{cap}(X) = \sum_{a \in \delta^+(X)} c(a)$.

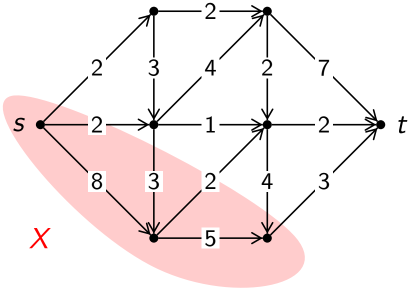


$\text{cap}(X) = 2 + 4 + 1 + 2 + 5 = 14$

Le problème de la coupe minimum

Problème

Trouver une $s-t$ coupe de capacité minimum.



$$\mathrm{cap}(X) = 2 + 2 + 2 + 3 = 9$$

Plan

Flot Maximum

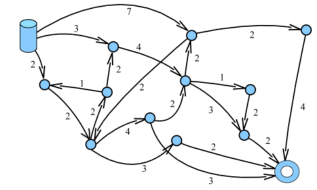
Un exemple

La compagnie pétrolière Inti T'schouff souhaite acheminer du pétrole par oléoduc vers un pays client. Le réseau d'oléoducs comporte plusieurs tronçons, chacun ayant une capacité maximale (en débit) à ne pas dépasser. Les tronçons sont directionnels.
Comment modéliser le problème par un graphe ?

Flot Maximum

Un exemple

Sur le graphe suivant, la compagnie pétrolière est représentée par le cylindre, le client par le jeton. La capacité maximale de chaque arc est indiquée.



Quel est le débit maximum que la compagnie pétrolière peut envoyer vers le client via le réseau ? En général, est-ce facile de trouver le débit maximum d'un tel réseau ?

Flot Maximum

Quelques exemples

- Les réseaux : réseaux de train, canalisations, tuyaux, routes, câbles électriques, réseaux d'information, réseaux routier, infrastructures de déplacement d'une station de ski
- La question : combien (d'information, d'eau, de touristes, de voitures, d'électrons...) peut-on faire transiter sur le réseau ?

Leur interprétation correspond à la circulation de flux physiques sur un réseau : distribution électrique, réseau d'adduction, acheminement de paquets sur Internet... Il s'agit d'acheminer la plus grande quantité possible de matière entre une source s et une destination t. Les liens permettant d'acheminer les flux ont une capacité limitée, et il n'y a ni perte, ni création de matière lors de l'acheminement : pour chaque noeud intermédiaire du réseau, le flux entrant (ce qui arrive) doit être égal au flux sortant (ce qui repart).

Flots

Définition

Soit $G = (V, A)$ un réseau

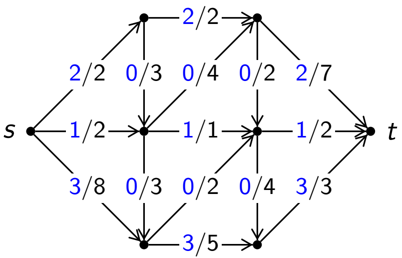
- de capacité $c : A \to \mathbb{R}^+$
- avec la source s et le puits t

Un $s - t$ *flot* est une fonction $f : A \to \mathbb{R}^+$ qui vérifie

- $\forall a \in A, \quad 0 \leq f(a) \leq c(a)$ (*contrainte de capacité*)
- $\forall v \in V \setminus \{s, t\}$, $\sum_{a \in \delta^-(\{v\})} f(a) = \sum_{a \in \delta^+(\{v\})} f(a)$ (*conservation de flot*).

Définition

La *valeur* d'un flot f est $\text{val}(f) = \sum_{a \in \delta^+(\{s\})} f(a)$.



Flots

Définition

Soit $G = (V, A)$ un réseau

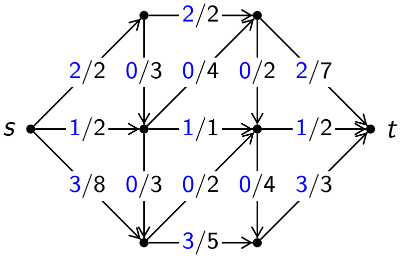
- de capacité $c : A \to \mathbb{R}^+$
- avec la source s et le puits t

Un $s - t$ *flot* est une fonction $f : A \to \mathbb{R}^+$ qui vérifie

- $\forall a \in A, \quad 0 \leq f(a) \leq c(a)$ (*contrainte de capacité*)
- $\forall v \in V \setminus \{s, t\}$, $\sum_{a \in \delta^-(\{v\})} f(a) = \sum_{a \in \delta^+(\{v\})} f(a)$ (*conservation de flot*).

Définition

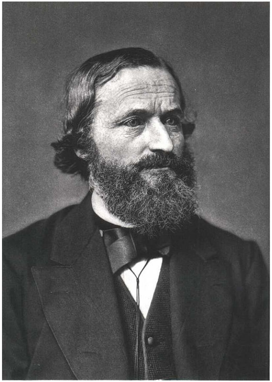
La *valeur* d'un flot f est $\text{val}(f) = \sum_{a \in \delta^+(\{s\})} f(a)$.

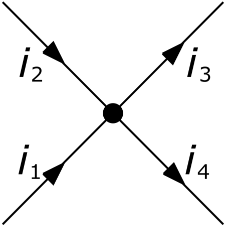


$$\text{val}(f) = 2 + 1 + 3 = 6$$

Flot Maximum

Gustav Kirchhoff, physicien allemand de la fin du XIXe siècle né à Königsberg (!) a énoncé deux lois (1845), la loi des mailles et la loi des nœuds : en chaque point du réseau électrique, la somme des intensités entrantes est égale à la somme des intensités sortantes.



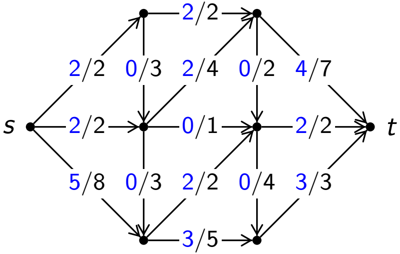


$$i_1 + i_2 = i_3 + i_4$$

Le problème du flot maximum

Problème

Trouver un $s - t$ flot de valeur maximum.



$$\text{val}(f) = 2 + 2 + 5 = 9$$

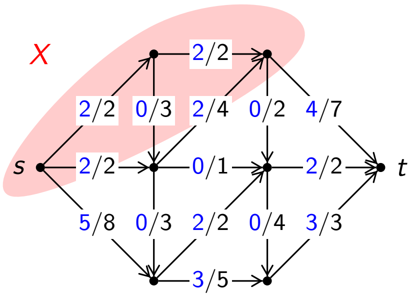
Plan

Flots et coupes

Lemme

Soit f un flot et X une $s-t$ coupe. Alors, le flot qui traverse la coupe est égal au flot qui sort du sommet s.

$$\sum_{a \in \delta^+(X)} f(a) - \sum_{a \in \delta^-(X)} f(a) = \text{val}(f).$$



$$\begin{aligned}\text{val}(f) &= 2+2+5\\ &= 5+2-2+4\end{aligned}$$

Flots et coupes

Démonstration

$$\begin{aligned}
\operatorname{val}(f) &= \sum_{a\in\delta^+(\{s\})} f(a) \\
&= \sum_{a\in\delta^+(\{s\})} f(a) - \sum_{a\in\delta^-(\{s\})} f(a) \\
&= \sum_{v\in X}\left(\sum_{a\in\delta^+(\{v\})} f(a) - \sum_{a\in\delta^-(\{v\})} f(a)\right) \quad (*) \\
&= \sum_{a\in\delta^+(X)} f(a) - \sum_{a\in\delta^-(X)} f(a).
\end{aligned}$$

(*) Par conservation de flot, tous les termes sont égaux à 0 sauf quand $v = s$

Dualité faible

Lemme

Soit f un flot et X une $s-t$ coupe quelconques. Alors, la valeur de f est inférieure ou égale à la capacité de la coupe X.

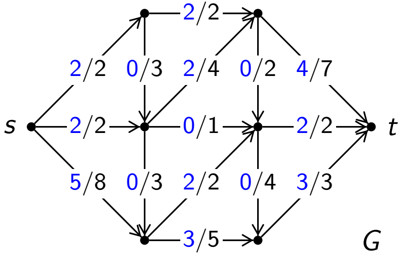
Démonstration

$$\begin{aligned} \mathrm{val}(f) &= \sum_{a \in \delta^+(X)} f(a) - \sum_{a \in \delta^-(X)} f(a) \\ &\leq \sum_{a \in \delta^+(X)} f(a) \\ &\leq \sum_{a \in \delta^+(X)} c(a) \\ &= \mathrm{cap}(X). \end{aligned}$$

Certificat d'optimalité

Corollaire

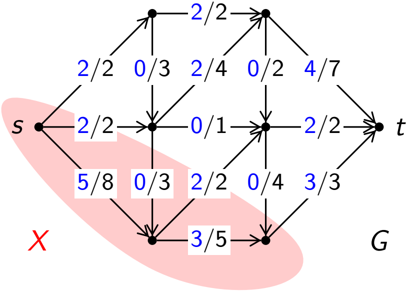
Soit f un flot et X une coupe. Si $\mathrm{val}(f) = \mathrm{cap}(X)$, alors f est un flot max et X est une coupe min.



Certificat d'optimalité

Corollaire

Soit f un flot et X une coupe. Si $\mathrm{val}(f) = \mathrm{cap}(X)$, alors f est un flot max et X est une coupe min.



$$\mathrm{val}(f) = 9$$

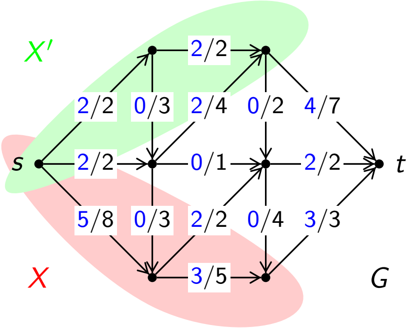
$$\mathrm{cap}(X) = 9$$

Le flot f est maximum.
La coupe X est minimum.

Certificat d'optimalité

Corollaire

Soit f un flot et X une coupe. Si $\mathrm{val}(f) = \mathrm{cap}(X)$, alors f est un flot max et X est une coupe min.



$$\mathrm{val}(f) = 9$$

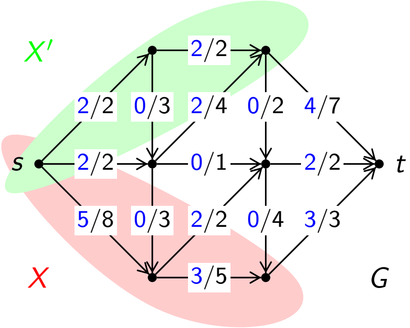
$$\mathrm{cap}(X) = 9$$

Le flot f est maximum.
La coupe X est minimum.
Attention : pour la capacité de la coupe, il faut bien prendre la valeur des capacités des arcs, et non la valeur du flot.

Certificat d'optimalité

Corollaire

Soit f un flot et X une coupe. Si $\mathrm{val}(f) = \mathrm{cap}(X)$, alors f est un flot max et X est une coupe min.



$$\mathrm{val}(f) = 9$$
$$\mathrm{cap}(X) = 9$$

Le flot f est maximum.
La coupe X est minimum.
Attention : pour la capacité de la coupe, il faut bien prendre la valeur des capacités des arcs, et non la valeur du flot.

$$\mathrm{cap}(X') = 22 > \mathrm{cap}(X)$$

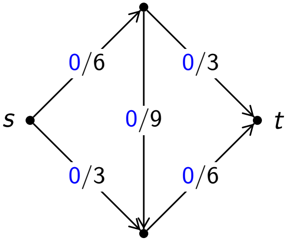
La coupe X' n'est pas minimum.

Plan

Vers un algorithme de flot max

Un algorithme glouton

- Commencer par le flot nul, càd, $f(a) = 0$ pour chaque arc $a \in A$.
- Trouver un $s - t$ chemin P où tout arc vérifie $f(a) < c(a)$.
- Augmenter le flot le long le chemin P.
- Répéter jusqu'à être coincé.

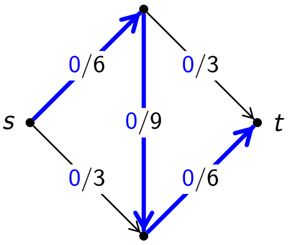


$\mathrm{val}(f) = 6$

Vers un algorithme de flot max

Un algorithme glouton

- Commencer par le flot nul, càd, $f(a) = 0$ pour chaque arc $a \in A$.
- Trouver un $s - t$ chemin P où tout arc vérifie $f(a) < c(a)$.
- Augmenter le flot le long le chemin P.
- Répéter jusqu'à être coincé.



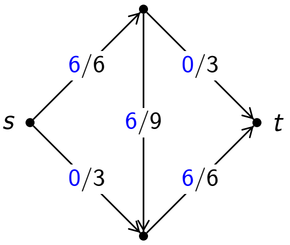
$\text{val}(f) = 6$

Vers un algorithme de flot max

Un algorithme glouton

- Commencer par le flot nul, càd, $f(a) = 0$ pour chaque arc $a \in A$.
- Trouver un $s - t$ chemin P où tout arc vérifie $f(a) < c(a)$.
- Augmenter le flot le long le chemin P.
- Répéter jusqu'à être coincé.

On est coincé, pourtant le flot n'est pas maximum !



$\text{val}(f) = 6$

Le graphe résiduel

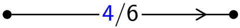
Arc originel :

- $a = (u, v) \in A$.
- Flot $f(a)$, capacité $c(a)$.

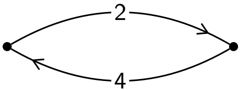
Arc résiduel :

- "annuler" le flot envoyé (ou une partie)
- $a = (u, v)$ et $a^R = (v, u)$.
- Capacité résiduelle :

$$c_f(a) = \begin{cases} c(a) - f(a) & \text{si } a \in A \\ f(a) & \text{si } a^R \in A. \end{cases}$$



arc de G



arcs de R_f

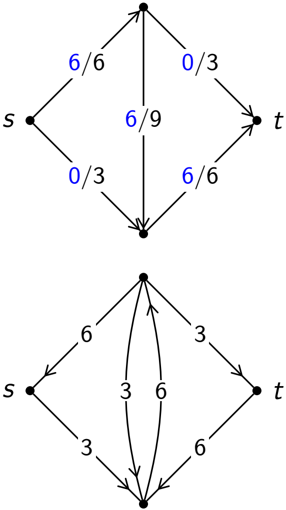
Le graphe résiduel

Graphe résiduel :

- Arcs résiduels avec capacité résiduelle positive.
- $A_f = \{a \mid f(a) < c(a)\} \cup \{a^R \mid f(a) > 0\}$.

Chemin augmentant :

- Un *chemin augmentant* est un s–t chemin simple (="sans répétition d'arc") dans le graphe résiduel R_f.
- La *capacité* $\text{cap}(R_f, P)$ d'un chemin augmentant P est le minimum des capacités résiduelles parmi tous les arcs de P.



Propriété clé

- Soit f un flot et P un chemin augmentant dans R_f.
- En envoyant un flot de valeur $\text{cap}(R_f, P)$ on obtient un nouveau flot f' de valeur $\text{val}(f') = \text{val}(f) + \text{cap}(R_f, P)$.

augmenter **: mettre à jour le flot selon un chemin augmentant**

Données :
$G = (V, A)$ $c : A \to \mathbb{R}^+$ flot f chemin augmentant P
Résultat : flot f mis à jour (augmenté)

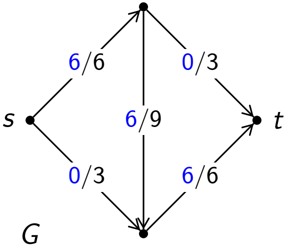
$\varepsilon \leftarrow \min\{c_f(a) : a \in A(P)\}$
pour *tout a arc de P* **faire**
  si $a \in A$ **alors**
    $f(a) \leftarrow f(a) + \varepsilon$
  sinon
    $f(a^R) \leftarrow f(a) - \varepsilon$
  retourner f

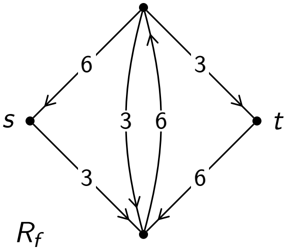
Propriété clé

***augmenter* : mettre à jour le flot (version abrégée)**

Données : voir page précédente
Résultat : flot f mis à jour

$\varepsilon \leftarrow \min\{c_f(a) : a \in A(P)\}$
pour *tout a arc de P* **faire**
  si $a \in A$ **alors**
    $f(a) \leftarrow f(a) + \varepsilon$
  sinon
    $f(a^R) \leftarrow f(a) - \varepsilon$
  retourner f



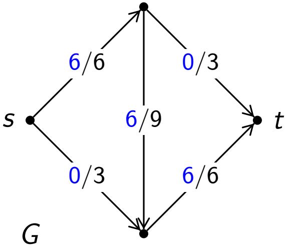


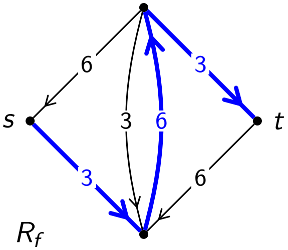
Propriété clé

***augmenter* : mettre à jour le flot (version abrégée)**

Données : voir page précédente
Résultat : flot f mis à jour

$\varepsilon \leftarrow \min\{c_f(a) : a \in A(P)\}$
pour *tout a arc de P* **faire**
  si $a \in A$ **alors**
    $f(a) \leftarrow f(a) + \varepsilon$
  sinon
    $f(a^R) \leftarrow f(a) - \varepsilon$
  retourner f



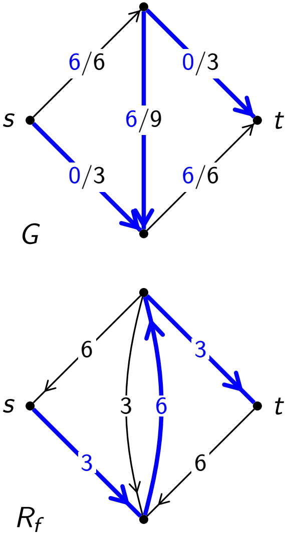


Propriété clé

augmenter **: mettre à jour le flot (version abrégée)**

Données : voir page précédente
Résultat : flot f mis à jour

$\varepsilon \leftarrow \min\{c_f(a) : a \in A(P)\}$
pour *tout a arc de P* **faire**
  si $a \in A$ **alors**
    $f(a) \leftarrow f(a) + \varepsilon$
  sinon
    $f(a^R) \leftarrow f(a) - \varepsilon$
  retourner f

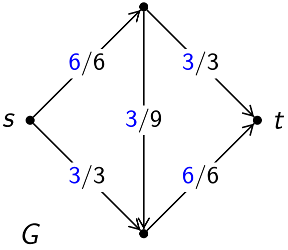


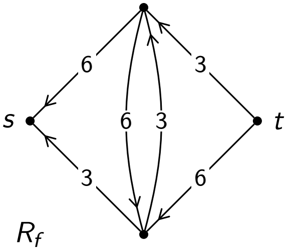
Propriété clé

augmenter : **mettre à jour le flot (version abrégée)**

Données : voir page précédente
Résultat : flot f mis à jour

$\varepsilon \leftarrow \min\{c_f(a) : a \in A(P)\}$
pour *tout a arc de P* **faire**
  si $a \in A$ **alors**
    $f(a) \leftarrow f(a) + \varepsilon$
  sinon
    $f(a^R) \leftarrow f(a) - \varepsilon$
  retourner f





Flot Maximum : algorithme de Ford-Fulkerson

Algorithme de Ford-Fulkerson

Données :

$G = (V, A)$
$c : A \to \mathbb{R}^+$
$s, t \in V$ (source et puits)

Résultat : Un flot f maximum

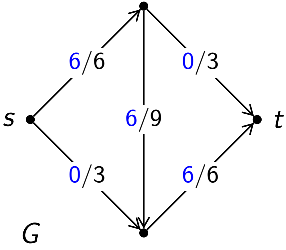
Initialisation

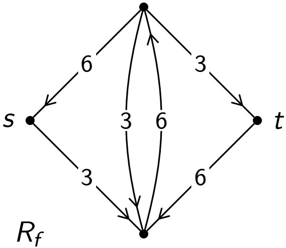
$f(a) = 0 \quad \forall a \in A$
$R_f \leftarrow$ graphe residuel

tant que *il existe un chemin augmentant* P **faire**

$f \leftarrow$ augmenter(G, f, c, P)
mettre a jour R_f

retourner f





Flot Maximum : algorithme de Ford-Fulkerson

Algorithme de Ford-Fulkerson

Données :

$G = (V, A)$

$c : A \to \mathbb{R}^+$

$s, t \in V$ (source et puits)

Résultat : Un flot f maximum

Initialisation

$f(a) = 0 \quad \forall a \in A$

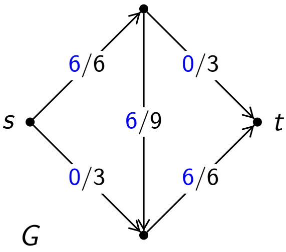
$R_f \leftarrow$ graphe residuel

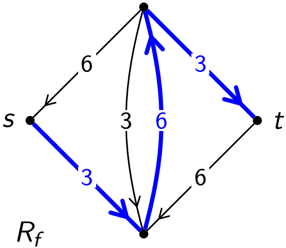
tant que *il existe un chemin augmentant* P **faire**

$f \leftarrow$ augmenter(G, f, c, P)

mettre a jour R_f

retourner f





Flot Maximum : algorithme de Ford-Fulkerson

Algorithme de Ford-Fulkerson

Données :

$G = (V, A)$

$c : A \to \mathbb{R}^+$

$s, t \in V$ (source et puits)

Résultat : Un flot f maximum

Initialisation

$f(a) = 0 \quad \forall a \in A$

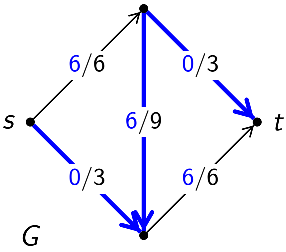
$R_f \leftarrow$ graphe residuel

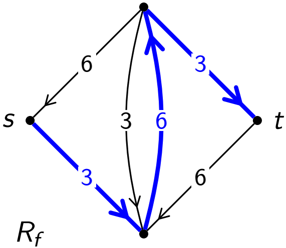
tant que *il existe un chemin augmentant* P **faire**

$f \leftarrow$ augmenter(G, f, c, P)

mettre a jour R_f

retourner f





Flot Maximum : algorithme de Ford-Fulkerson

Algorithme de Ford-Fulkerson

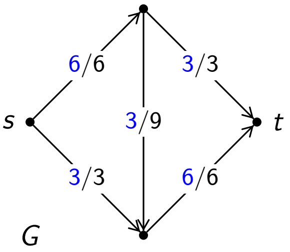
Données :
- $G = (V, A)$
- $c : A \rightarrow \mathbb{R}^+$
- $s, t \in V$ (source et puits)

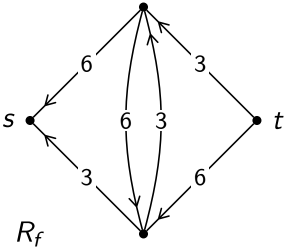
Résultat : Un flot f maximum

Initialisation
- $f(a) = 0 \quad \forall a \in A$
- $R_f \leftarrow$ graphe residuel

tant que *il existe un chemin augmentant* P **faire**
- $f \leftarrow$ augmenter(G, f, c, P)
- mettre a jour R_f

retourner f





Plan

Le théorème flot-max/coupe-min

Théorème des chemins augmentants

Un flot f est maximum ssi il n'y a pas de chemin augmentant.

Théorème (Elias–Feinstein–Shannon 1956 ; Ford-Fulkerson 1956)

La valeur maximum d'un flot est égale à la capacité minimum d'une coupe.

On va prouver les deux théorèmes en même temps en démontrant que les énoncés suivants sont équivalents :

(1) Il existe une coupe X telle que $\mathrm{val}(f) = \mathrm{cap}(X)$.

(2) Le flot f est maximum.

(3) Il n'existe pas de chemin augmentant par rapport à f.

Démonstration du théorème flot-max/coupe-min (1/2)

(1) Il existe une coupe X telle que $\mathrm{val}(f) = \mathrm{cap}(X)$.

(2) Le flot f est maximum.

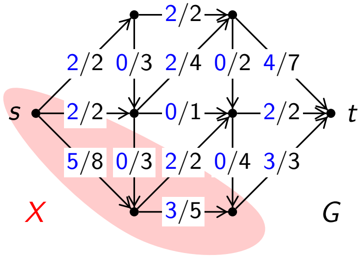
(3) Il n'existe pas de chemin augmentant par rapport à f.

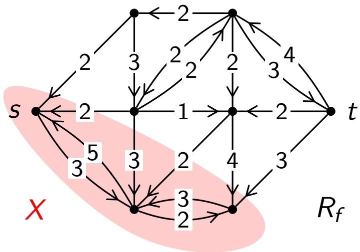
(1) $\Rightarrow$ (2) Corollaire à la dualité faible.
(2) $\Rightarrow$ (3) Soit f un flot. S'il existe un chemin augmentant P, on peut augmenter f en envoyant un flot le long P.
(3) $\Rightarrow$ (1)

- Soit f un flot sans chemin augmentant.
- Soit X un ensemble de sommets atteignables depuis s dans le graphe résiduel.
- Par la définition de X, $s \in X$.
- Par la définition de f, $t \notin X$.
- Donc X est une $s - t$ coupe

Démonstration du théorème flot-max/coupe-min (2/2)

$$\begin{aligned}\mathrm{val}(f) &= \sum_{a\in\delta^+(X)} f(a) - \sum_{a\in\delta^-(X)} f(a)\\ &= \sum_{a\in\delta^+(X)} c(a)\\ &= \mathrm{cap}(X)\end{aligned}$$





Comment trouver une coupe minimum ?

- Il suffit de prendre X comme l'ensemble des sommets atteignables à partir de s dans le graphe résiduel R_f.
- C'est-à-dire, $v \in X$ ssi il existe un chemin orienté dans R_f avec sommet de départ s et sommet d'arrivée v.
- Si vous trouvez que $t \in X$, alors il existe un chemin augmentant et f n'est pas maximum — dans ce cas, il faut encore faire tourner Ford-Fulkerson !

Conclusion

Quand on vous demande de trouver un flot maximum dans un graphe, il faut :

- appliquer l'algorithme de Ford-Fulkerson,
- vérifier que vous ne vous êtes pas arrêtés trop tôt en cherchant une $s-t$ coupe de capacité égale à la valeur du flot trouvé.