Problèmes de flot maximum

Auteur: Julien David

Remerciements

Ce cours est basé sur celui d'Anthony Labarre, enseignant-chercheur à l'Université Gustave-Eiffel, qui a eu la gentillesse de me laisser utiliser ses supports de cours pour préparer celui-ci.

Problème de flot maximum

Problème On souhaite télécharger un fichier volumineux depuis un serveur distant

- Il existe de multiples chemins entre nous et le serveur.
- Entre chaque serveur, la bande passante n'est pas forcément la même.

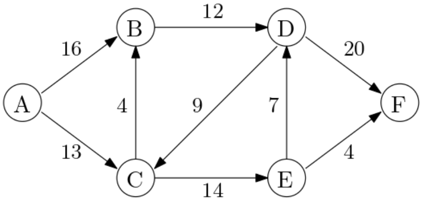
La question est donc: quelle est la vitesse maximale à laquelle on peut télécharger le fichier?

Contexte: les graphes orientés et pondérés

Definition Graphe orienté pondéré

Un graphe orienté pondéré est un triplet $G = (V, A, c)$ tel que

- V est l'ensemble des sommets,
- $A \subseteq V \times V$ est l'ensemble des arcs du graphe, i.e. un ensemble de couples de sommets.
- l'application $c : A \mapsto \mathbb{R}$ associe à chaque arc du graphe sa capacité.

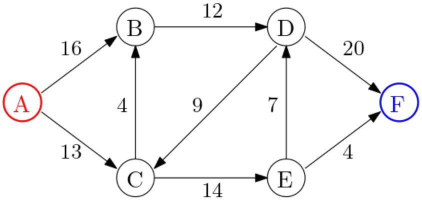


Source et Puit

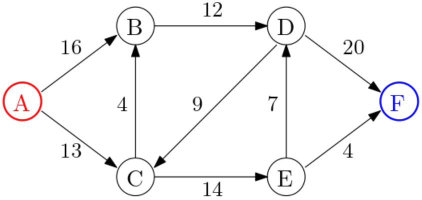
Definition Source et Puit

Dans un graphe orienté:

- une **source** est un sommet sans arcs entrant.
- un **puit** est un sommet sans arcs sortant.



Application

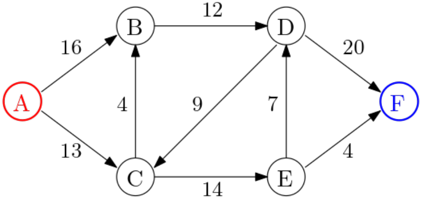


Remarque ViaMichelin, GoogleMaps, Waze

Lorsque vous utilisez un de ces logiciels:

- on parle de **poids** et non de **capacité** des arcs.
- il calcule le chemin de poids minimal d'une source à une destination.
- le graphe est **dynamique**, les poids sur les arcs varient dans le temps.

Application



Remarque ViaMichelin, GoogleMaps, Waze

Lorsque vous utilisez un de ces logiciels:

- on parle de **poids** et non de **capacité** des arcs.
- il calcule le chemin de poids minimal d'une source à une destination.
- le graphe est **dynamique**, les poids sur les arcs varient dans le temps.

Mais ce n'est pas le sujet d'aujourd'hui.

Réseau de flots

Definition Réseau de flots

Un réseau de flot est un graphe orienté pondéré $G = (V, A, c)$ tel que

- tout arc $(u, v) \in A$ possède une capacité $c(u, v) \geq 0$,
- il existe une unique source $s \in V$
- il existe un unique puits $p \in V$

Réseau de flots

Definition — Réseau de flots

Un réseau de flot est un graphe orienté pondéré $G = (V, A, c)$ tel que

- tout arc $(u, v) \in A$ possède une capacité $c(u, v) \geq 0$,
- il existe une unique source $s \in V$
- il existe un unique puits $p \in V$

Remarque — On pourra supposer que

- tous les autres sommets se trouvent sur un chemin entre s et t,
- si $(u, v) \in A$ alors $(v, u) \notin A$,

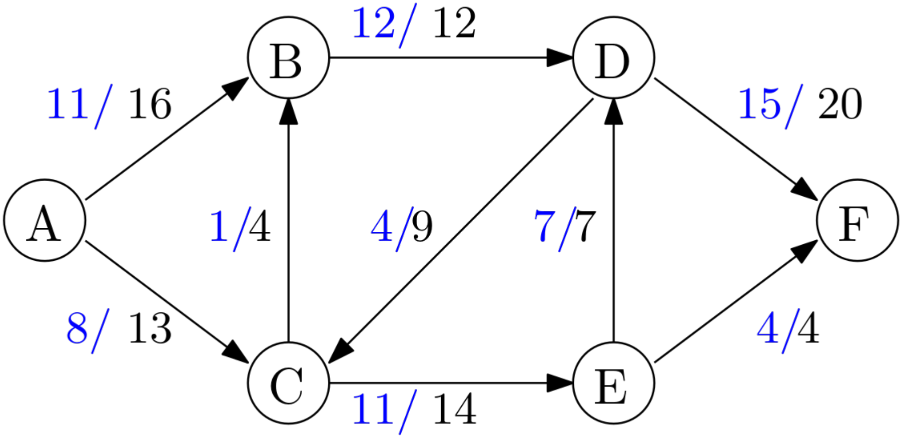
Flot: définition

Definition Flot

Un flot dans un réseau de flot $G = (V, A, c)$ est une fonction $f : V \times V \mapsto \mathbb{R}$ satisfaisant les deux propriétés suivantes:

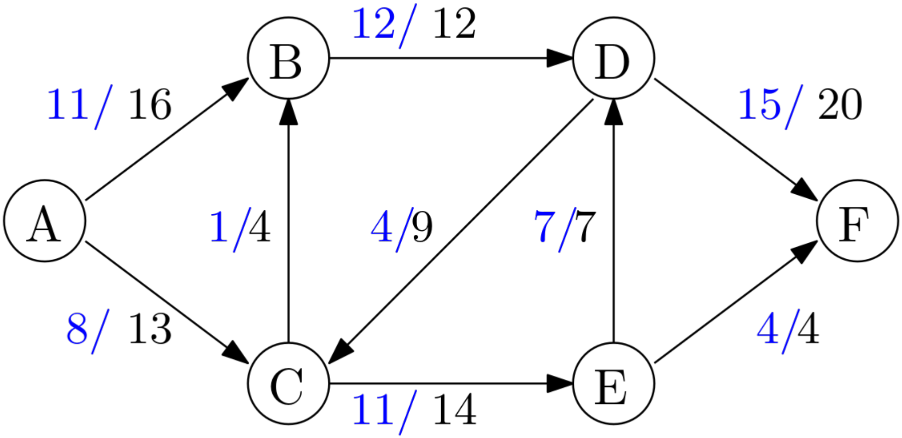
1. **Contrainte de capacité**: $\forall u, v \in V$, $0 \leq f(u, v) \leq c(u, v)$
2. **Conservation des flots**: $\forall u \in V \setminus \{s, p\}$, $\sum_{v \in V} f(u, v) = \sum_{v \in V} f(v, u)$

Flot: exemple



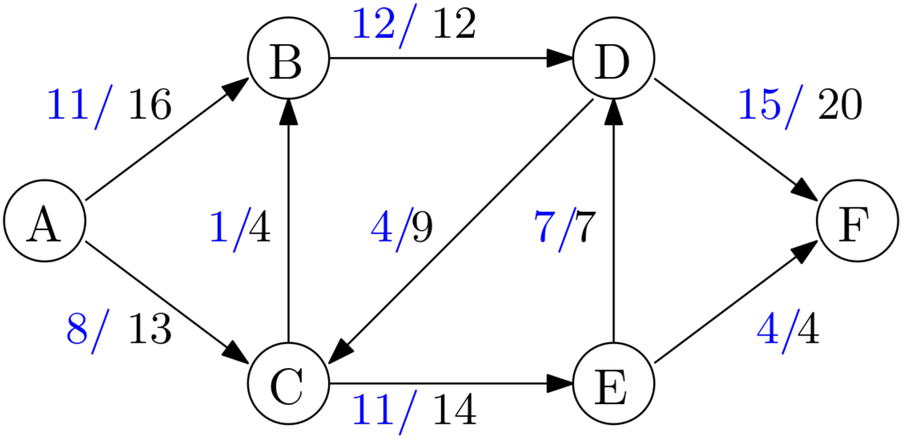
Pour chaque arc, le flot est noté en bleu et la capacité en noir.

Flot: contrainte de capacité



Contrainte de capacité: $\forall u, v \in V$, $0 \leq f(u, v) \leq c(u, v)$
Le flot d'un arc est toujours inférieur ou égal à la capacité de l'arc.

Flot: conservation des flots



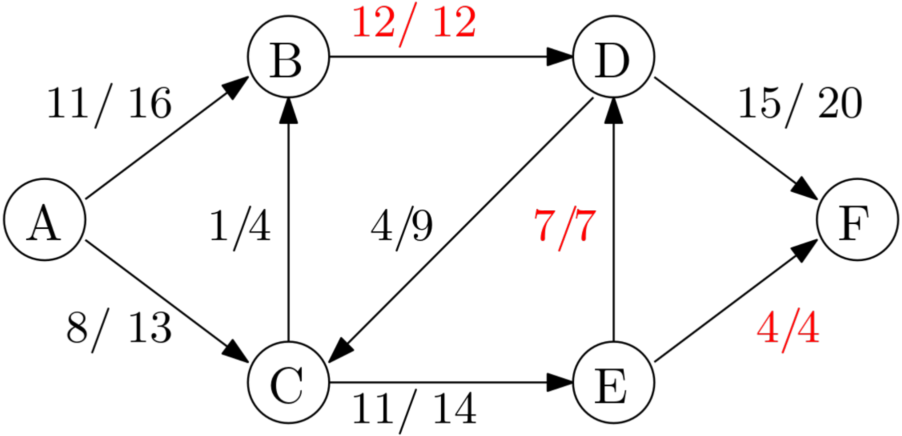
Conservation des flots: $\forall u \in V \setminus \{s, p\}, \sum_{v \in V} f(u, v) = \sum_{v \in V} f(v, u)$
Le flot total entrant dans chaque sommet est égal au flot total en sortie.

Arc saturé

Definition Arc saturé par un flot

Soit un réseau de flot $G = (V, A, c)$ muni d'un flot f. On dit qu'un arc $(u, v) \in A$ est **saturé** par le flot f si $f(u, v) = c(u, v)$.

Arcs saturés: exemple



Flot: valeur

Definition Valeur d'un flot

Soit un réseau de flot $G = (V, A, c)$, une source $s \in V$ et un puits $p \in V$.
La valeur d'un flot f est

$$|f| = \sum_{v \in V} f(s, v) - f(v, s)$$

Flot: valeur

Definition Valeur d'un flot

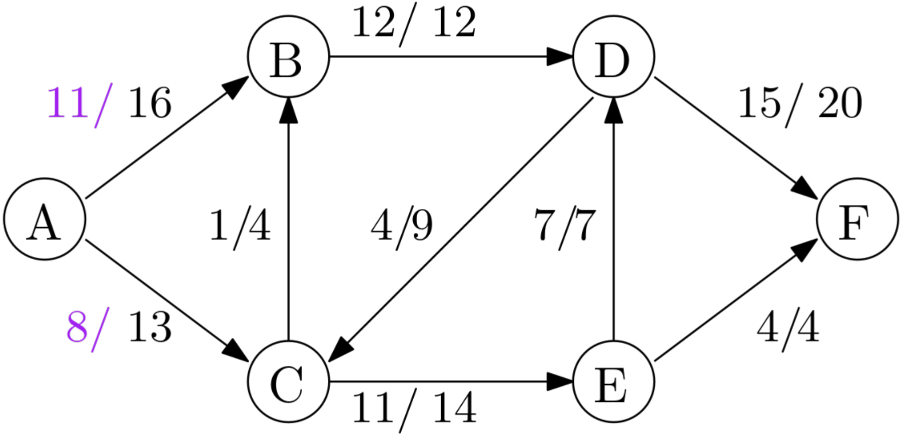
Soit un réseau de flot $G = (V, A, c)$, une source $s \in V$ et un puits $p \in V$.
La valeur d'un flot f est

$$|f| = \sum_{v \in V} f(s, v) - f(v, s)$$

Remarque Compliqué pour rien?

La partie $-f(v, s)$ peut sembler inutile pour le moment puisque le sommet s n'a pas d'arc entrant, mais prendra du sens par la suite.

Flot: contrainte de capacité



$|f| = \sum_{v \in V} f(s,v) - f(v,s) = 11 + 8 - 0 = 19$

Problème de flot: version formelle

Problème Flot maximum

Entrée: un réseau de flot $G = (V, A, c)$, une source s et un puits $p \in V$
Sortie: un flot f de G tel que pour tout flot f' de G, on a $|f| \geq |f'|$.

Méthode de Ford-Fulkerson

1956

Méthode de Ford-Fulkerson

Explication — L'idée générale

- On construit le flot maximal en partant d'un flot nul et en l'augmentant progressivement.

Méthode de Ford-Fulkerson

Explication L'idée générale

- On construit le flot maximal en partant d'un flot nul et en l'augmentant progressivement.
- On augmente le flot le long d'un chemin, un chemin à la fois.

Méthode de Ford-Fulkerson

Explication L'idée générale

- On construit le flot maximal en partant d'un flot nul et en l'augmentant progressivement.
- On augmente le flot le long d'un chemin, un chemin à la fois.
- Un tel chemin est appelé **chemin augmentant**.

Méthode de Ford-Fulkerson

Explication L'idée générale

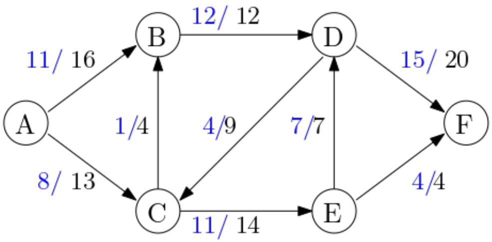
- On construit le flot maximal en partant d'un flot nul et en l'augmentant progressivement.
- On augmente le flot le long d'un chemin, un chemin à la fois.
- Un tel chemin est appelé **chemin augmentant**.
- si le flot change, la capacité des arcs du réseau peut-être modifiée. On parle de **capacité résiduelle**.

Méthode de Ford-Fulkerson

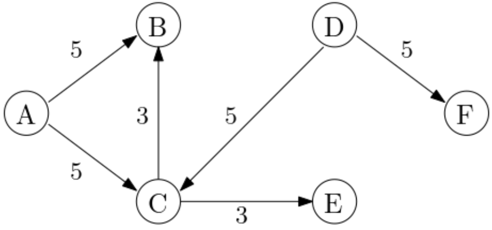
Explication | L'idée générale

- On construit le flot maximal en partant d'un flot nul et en l'augmentant progressivement.
- On augmente le flot le long d'un chemin, un chemin à la fois.
- Un tel chemin est appelé **chemin augmentant**.
- si le flot change, la capacité des arcs du réseau peut-être modifiée. On parle de **capacité résiduelle**.
- si la capacité des arcs change, le réseau change également. On parle de **réseau résiduel**.

Idée (incomplète) de réseau résiduel

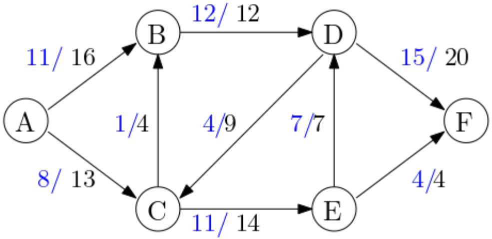


Réseau muni d'un flot

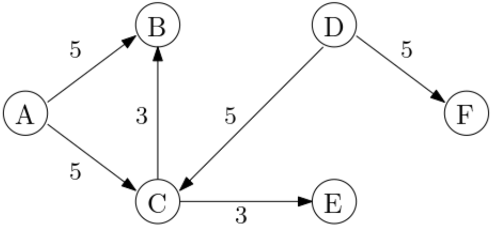


Réseau résiduel (incomplet)

Idée (incomplète) de réseau résiduel



Réseau muni d'un flot



Réseau résiduel (incomplet)

Problème: le flot actuel n'est pas maximum, or il n'existe plus de chemin de A à F. On a perdu la trace des flots précédement utilisés, qui ne peuvent plus être redirigés.

Flot: capacité résiduelle

Definition Capacité résiduelle

Soit un réseau de flot $G = (V, A, c)$ muni d'un flot f.
La fonction $c_f : A \times A \mapsto \mathbb{R}$ appelée capacité résiduelle est définie comme suit. Pour tout couple de sommet $u, v \in A$:

$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v), \text{ si } (u,v) \in A \\ f(u,v), \text{ si } (v,u) \in A \\ 0, \text{ sinon} \end{cases}$$

Flot: capacité résiduelle

Definition Capacité résiduelle

Soit un réseau de flot $G = (V, A, c)$ muni d'un flot f.
La fonction $c_f : A \times A \mapsto \mathbb{R}$ appelée capacité résiduelle est définie comme suit. Pour tout couple de sommet $u, v \in A$:

$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v), \text{ si } (u,v) \in A \\ f(u,v), \text{ si } (v,u) \in A \\ 0, \text{ sinon} \end{cases}$$

Remarque Apparition

Un couple (u, v) peut donc avoir une capacité résiduelle même si $(u, v) \notin A$.

Réseau résiduel

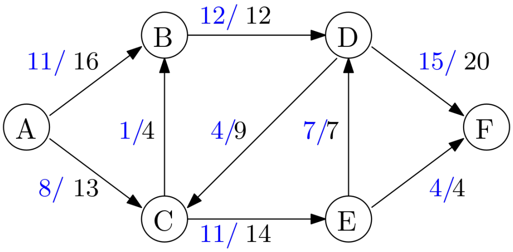
Definition Réseau résiduel

Soit un réseau de flot $G = (V, A, c)$ muni d'un flot f.
On définit $G_f = (V, A_f, c_f)$, le réseau résiduel de G muni de f, tel que:

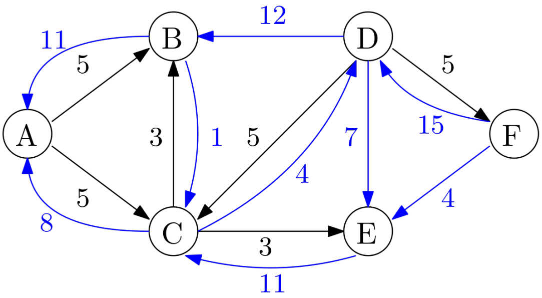
- V est l'ensemble des sommets de G.
- c_f est la capacité résiduelle de G muni de f.
- A_F est l'ensemble d'arêtes défini comme suit:

$$A_f = \{(u, v) \in V \times V \mid c_f(u, v) > 0\}$$

Réseau résiduel: exemple



G muni d'un flot f.



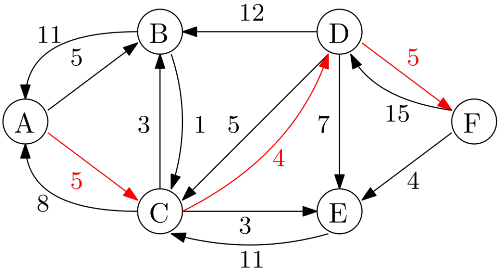
Réseau résiduel de G muni de f.

Chemin augmentant

Definition Chemin augmentant

Soit un réseau résiduel $G_f = (V, A_f, c_f)$, une source $s \in V$ et un puits $p \in V$.
Un chemin augmentant est un chemin simple[a] P de s à t dans G_f.

[a] chaque sommet est visité au plus une fois



Chemin $P = ((A, C), (C, D), (D, F))$

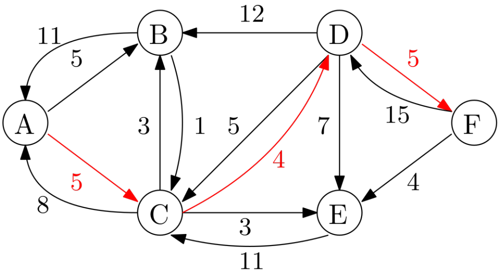
Capacité d'un chemin augmentant

Definition — Capacité d'un chemin augmentant

Soit un réseau résiduel $G_f = (V, A_f, c_f)$, et un chemin augmentant P de s à t. La capacité du chemin P, notée $c_f(P)$, est égale à

$$c_f(P) = min_{(u,v)\in P} c_f(u,v)$$

soit la capacité résiduelle minimale des arcs de P.



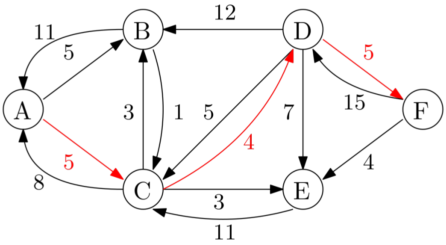
Chemin $P = ((A,C),(C,D),(D,F))$
Capacité $c_f(P) = 4$

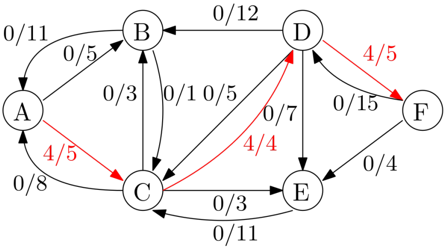
Flot d'un chemin augmentant

Definition — Flot d'un chemin augmentant

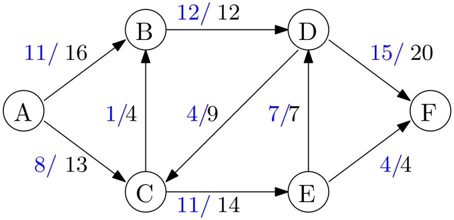
Soit un réseau résiduel $G_f = (V, A_f, c_f)$, et un chemin augmentant P de s à t. Le flot associé au chemin P, noté f_P, est donné par:

$$f_P(u,v) = \begin{cases} c_f(P), & \text{si } (u,v) \in P \\ 0, & \text{sinon} \end{cases}$$

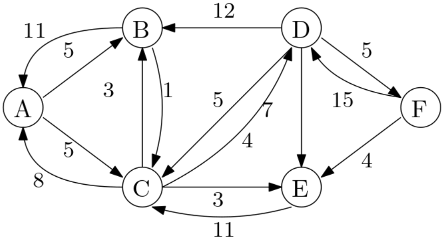




Augmentation d’un flot

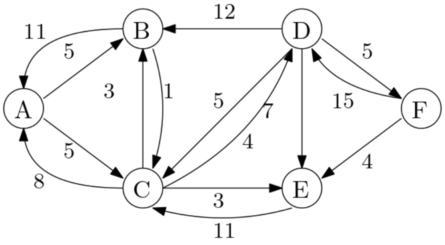


Flot initial
sur le réseau de flot

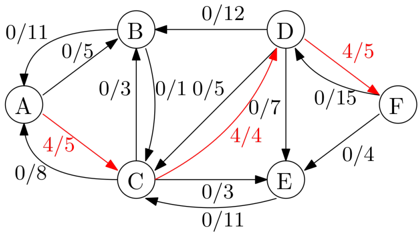


Réseau résiduel associé

Augmentation d’un flot

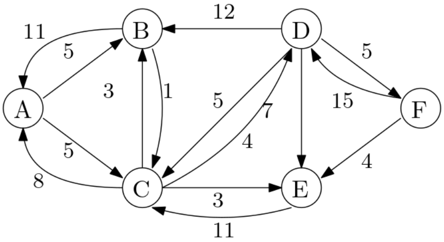


Réseau résiduel

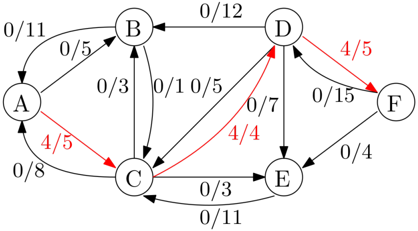


Flot du chemin augmentant
sur le réseau résiduel.

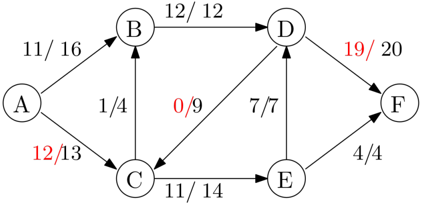
Augmentation d'un flot



Réseau résiduel

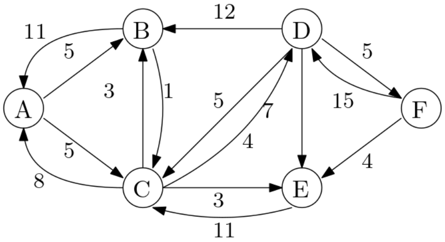


Flot du chemin augmentant sur le réseau résiduel.

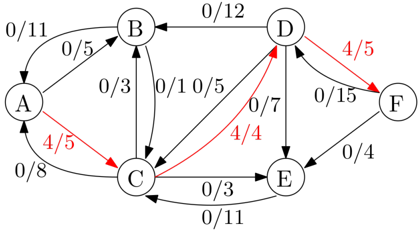


Flot augmenté

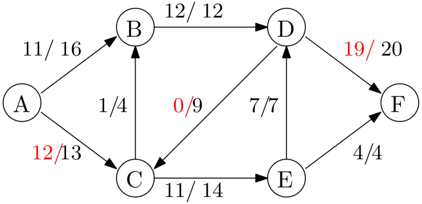
Augmentation d'un flot



Réseau résiduel



Flot du chemin augmentant
sur le réseau résiduel.



Flot augmenté

Augmenter le flot sur (C, D) signifie en réalité le diminuer sur (D, C)

Méthode de Ford-Fulkerson: Algorithme

Algorithme : $FordFulkerson(G)$

Entrées : un réseau de flot $G = (V, A, c)$, deux sommets $source$ et $puits$
Sortie : un flot maximum de G
$flot \leftarrow$ Initialiser un tableau associatif;
pour tous les $\underline{(u, v) \in A}$ **faire**
    $flot[(u, v)] = 0$;
fin
$G_f \leftarrow G$;
$chemin \leftarrow cheminAugmentant(G_f, source, puits)$;
tant que $\underline{chemin \neq None}$ **faire**
    $flot \leftarrow augmenterFlot(flot, chemin)$;
    $G_f \leftarrow mettreAJourResiduel(G, G_f, chemin, flot)$;
    $chemin \leftarrow cheminAugmentant(G_f, source, puits)$;
fin
return $\underline{flot}$;

Augmentation d'un flot

Algorithme : $augmenterFlot(f, P)$

Entrées : un flot f, un chemin P
Sortie : un flot augmenté

```
capacite_min ← min{c(u, v) | (u, v) ∈ P};
pour tous les (u, v) ∈ P faire
    si (u, v) ∈ f alors
        f(u, v) ← f(u, v) + capacite_min;
    fin
    sinon
        f(v, u) ← f(v, u) − capacite_min;
    fin
fin
return f;
```

Augmentation d'un flot

Algorithme : $augmenterFlot(f, P)$

Entrées : un flot f, un chemin P
Sortie : un flot augmenté

$capacite\_min \leftarrow \min\{c(u,v) \mid (u,v) \in P\}$;
pour tous les $\underline{(u,v) \in P}$ **faire**
  si $\underline{(u,v) \in f}$ **alors**
    $f(u,v) \leftarrow f(u,v) + capacite\_min$;
  fin
  sinon
    $f(v,u) \leftarrow f(v,u) - capacite\_min$;
  fin
fin
return $\underline{f}$;

Lemme Complexité

- Temps: $\Theta(|P|)$
- Espace: $\Theta(1)$

Mise à jour du réseau résiduel

Algorithme : $mettreAJourResiduel(G, G_f, P, flot)$

Entrées : Un réseau résiduel $G_f = (V, A_f, c_f)$, un chemin P, un flot f_P
Sortie : Un réseau résiduel G_f modifié

pour tous les $(u, v) \in P$ **faire**
  $c_f(u, v) \leftarrow c_f(u, v) - f_P(u, v)$;
  si $c_f(u, v) = 0$ **alors**
    $A_f = A_f \setminus \{(u, v)\}$;
  fin
  si $c_f(v, u) = 0$ **alors**
    $A_f = A_f \cup \{(v, u)\}$;
  fin
  $c_f(v, u) \leftarrow f_P(u, v)$;
fin
return $G_f = (V, A_f, c_f)$;

Mise à jour du réseau résiduel

Algorithme : $mettreAJourResiduel(G, G_f, P, flot)$

Entrées : Un réseau résiduel $G_f = (V, A_f, c_f)$, un chemin P, un flot f_P
Sortie : Un réseau résiduel G_f modifié

pour tous les $(u, v) \in P$ **faire**
  $c_f(u, v) \leftarrow c_f(u, v) - f_P(u, v)$;
  si $c_f(u, v) = 0$ **alors**
    $A_f = A_f \setminus \{(u, v)\}$;
  fin
  si $c_f(v, u) = 0$ **alors**
    $A_f = A_f \cup \{(v, u)\}$;
  fin
  $c_f(v, u) \leftarrow f_P(u, v)$;
fin
return $G_f = (V, A_f, c_f)$;

Lemme Complexité

- Temps: $\Theta(|P|)$
- Espace: $\Theta(1)$

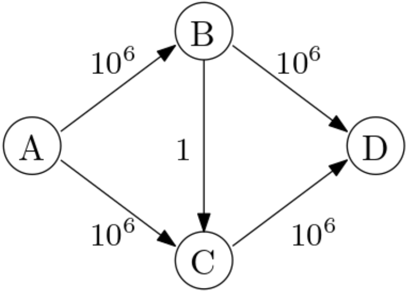
Méthode de Ford-Fulkerson

Problème Détail manquant

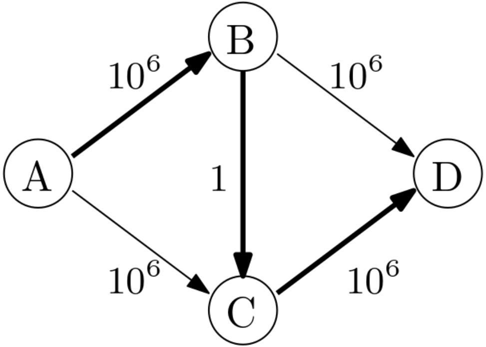
La méthode de Ford-Fulkerson ne précise pas comment trouver les chemins augmentant.

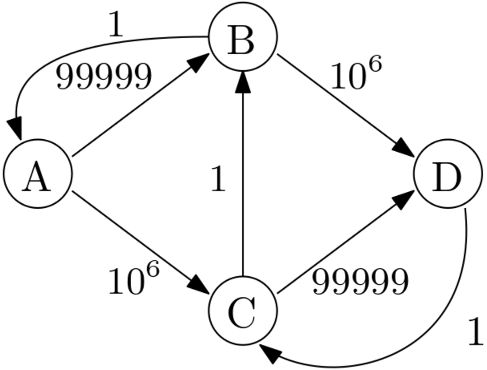
- Il est donc possible de construire plusieurs algorithmes à l'aide de cette méthode.
- Sa complexité va dépendre de l'algorithme qui permet de trouver un chemin augmentant et de **combien de fois il faudra trouver un chemin augmentant**
- Mais il faut également montrer que cette méthode calcule bien un flot maximum

Cas Pathologique

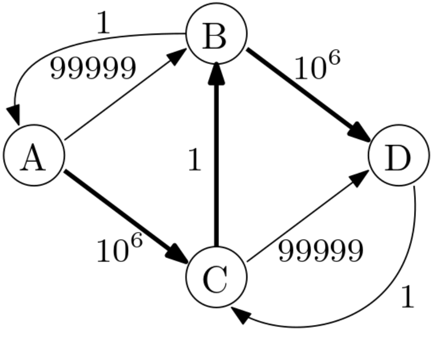


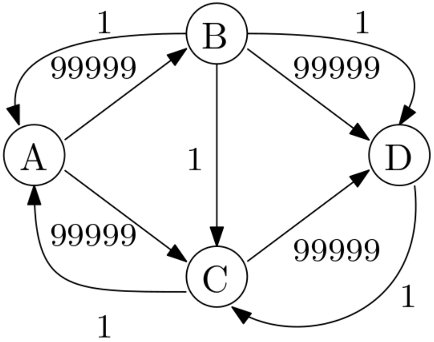
Cas Pathologique





Cas Pathologique





On va faire 2×10^6 chemins augmentants pour un graphe de 5 arcs ...

L'approche d'Edmonds-Karp

1972

L'approche d'Edmonds-Karp

Explication L'approche d'Edmonds-Karp

Edmonds et Karp proposent un méthode pour trouver un chemin augmentant.

- Il s'agit d'un simple parcours en largeur permettant de trouver un plus court chemin de s à t
- Nous avons vu cet algorithme en L3 et connaissons déjà sa complexité: $\Theta(|A|)$

Construction d'un chemin augmentant: algorithmes

Algorithme : $cheminAugmentant(G_f = (V, A_f, c_f))$

Entrées : Un réseau résiduel $G_f = (V, A_f, c_f)$, deux sommets s et p
Sortie : un chemin augmenté P

```
f ← File(s);
parent ← Tableau initialisé à None;
parent[s] = s;
tant que f n'est pas vide et top(f) ≠ p faire
    (f, u) ← defiler(f);
    pour tous les v voisins de u faire
        si parent[v] == None alors
            parent[v] ← u;
            f ← enfiler(f, v);
        fin
    fin
fin
/* On reconstruit le chemin */
chemin ← [];
v ← p;
tant que v ≠ s faire
    Ajouter (parent[v], v) au début de chemin;
    v ← parent[v];
fin
return chemin;
```

Construction d'un chemin augmentant: algorithmes

Algorithme : $cheminAugmentant(G_f = (V, A_f, c_f))$

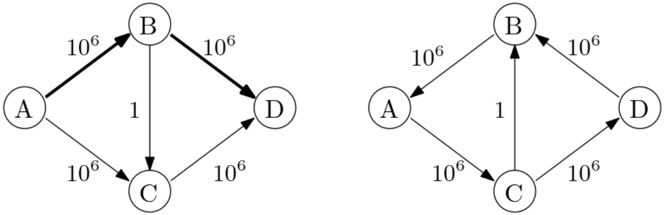
Entrées : Un réseau résiduel $G_f = (V, A_f, c_f)$, deux sommets s et p
Sortie : un chemin augmenté P

$f \leftarrow File(s)$;
$parent \leftarrow$ Tableau initialisé à $None$;
$parent[s] = s$;
tant que f n'est pas vide et $top(f) \neq p$ **faire**
  $(f, u) \leftarrow defiler(f)$;
  pour tous les v voisins de u **faire**
    si $parent[v] == None$ **alors**
      $parent[v] \leftarrow u$;
      $f \leftarrow enfiler(f, v)$;
    fin
  fin
fin
/* On reconstruit le chemin */
chemin $\leftarrow$ [];
$v \leftarrow p$;
tant que $v \neq s$ **faire**
  Ajouter $(parent[v], v)$ au début de $chemin$;
  $v \leftarrow parent[v]$;
fin
return $chemin$;

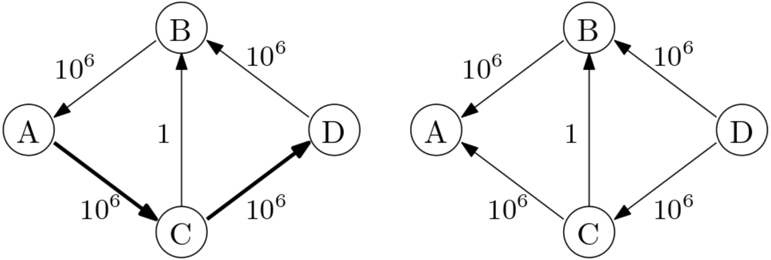
Lemme Complexité

- Temps: $\Theta(|A|)$
- Espace: $\Theta(|V|)$

Cas Pathologique



Cas Pathologique



Plus de chemin possible de A vers D.
On a trouvé le flot max en ajoutant seulement 2 chemins.

Pourquoi ça marche?

Le théorème max-flow min-cut

L'idée

Explication

- On veut montrer que, lorsqu'un graphe résiduel ne contient plus de chemin augmentant de la source vers le puits, alors on a atteint un flot maximum.
- Pour cela, on va passer par une notion intermédiaire: les coupes.

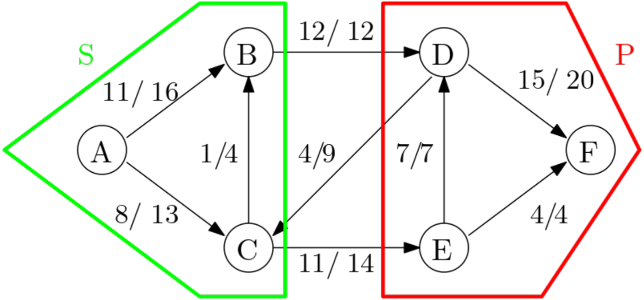
Coupe

Definition — Coupe

Soit un réseau de flot $G = (V, A, c)$ muni d'une source s et un puits p.
Une coupe (S, P) est une partition de V en deux sous-ensembles tels que

- $s \in S$
- $p \in P$

Coupe: exemple



Coupe et flot

On veut relier la notion de coupe à la notion de flot.
On va donc définir:

- le flot associé à une coupe
- la capacité d'une coupe

et montrer les liens entre flots, coupes et graphes résiduels.

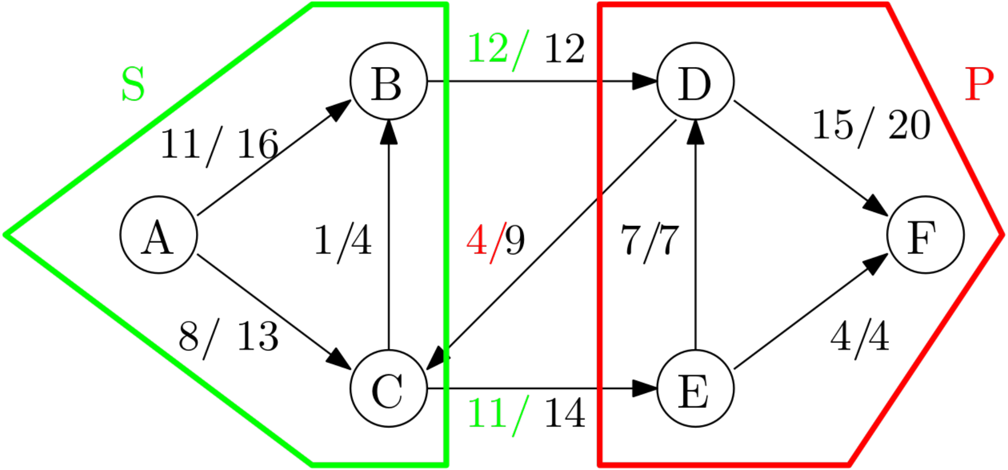
Flot associé à une coupe

Definition — Flot associé à une coupe

Soit un réseau de flot $G = (V, A, c)$ muni d'une source s, d'un puits p et d'une coupe (S, P). Le flot associé à la coupe (S, P) est le flot circulant entre les arcs entre en S et P

$$f(S, P) = \sum_{u \in S} \sum_{v \in P} f(u, v) - \sum_{u \in S} \sum_{v \in P} f(v, u)$$

Coupe: exemple



$$f(S,P) = \sum_{u \in S} \sum_{v \in P} f(u,v) - \sum_{u \in S} \sum_{v \in P} f(v,u) = 12 + 11 - 4 = 19$$

Flot associé à une coupe

Remarque Flot associé à une coupe

Soit un réseau de flot $G = (V, A, c)$ muni d'une source s, d'un puits p d'un flot f.

- pour tout $S \subseteq V$, on a $f(S, S) = 0$,
- pour tout $S, P \subseteq V$, on a $f(S, P) = -f(P, S)$

Flot associé à une coupe

Remarque Flot associé à une coupe

Soit un réseau de flot $G = (V, A, c)$ muni d'une source s, d'un puits p d'un flot f.

- pour tout $S \subseteq V$, on a $f(S, S) = 0$,
- pour tout $S, P \subseteq V$, on a $f(S, P) = -f(P, S)$

Le principe de **conservation des flots**:

$$\forall u \in V \setminus \{s, p\}, \sum_{v \in V} f(u, v) = \sum_{v \in V} f(v, u)$$

Lien coupes et flots

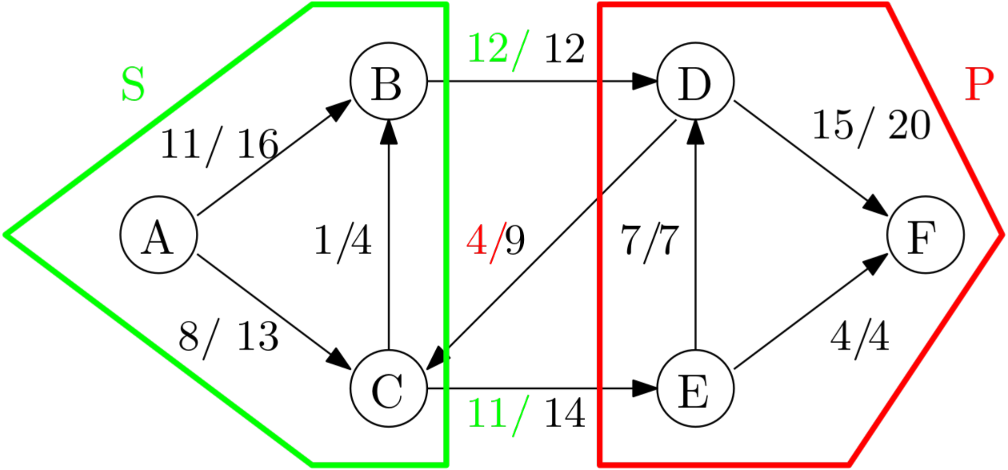
Lemme

Soit un réseau de flot $G = (V, A, c)$ muni d'un flot f et d'une coupe (S, P). On a

$$|f| = f(S, P)$$

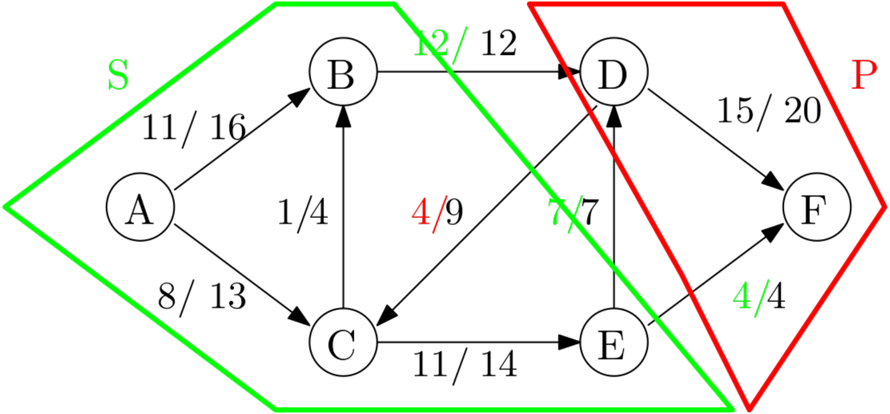
autrement dit, la valeur du flot f est égale au flot associé toute coupe.

Coupe: exemple



$$f(S,P) = \sum_{u \in S} \sum_{v \in P} f(u,v) - \sum_{u \in S} \sum_{v \in P} f(v,u) = 12 + 11 - 4 = 19$$

Coupe: exemple



$$f(S,P) = \sum_{u\in S}\sum_{v\in P} f(u,v) - \sum_{u\in S}\sum_{v\in P} f(v,u) = 12+7+4-4 = 19$$

Lien coupes et flots

Lemme

Soit un réseau de flot $G = (V, A, c)$ muni d'un flot f et d'une coupe (S, P). On a

$$|f| = f(S, P)$$

autrement dit, la valeur du flot f est égale au flot associé toute coupe.

Lien coupes et flots

Lemme

Soit un réseau de flot $G = (V, A, c)$ muni d'un flot f et d'une coupe (S, P). On a

$$|f| = f(S, P)$$

autrement dit, la valeur du flot f est égale au flot associé toute coupe.

Explication L'idée de la preuve

Formule de la valeur d'un flot

$$|f| = \sum_{v \in V} f(s, v) - f(v, s)$$

Valeur des flots allant de la source à un sommet v moins la valeur des flots allant de sommets v à la source.

Lien coupes et flots

Lemme

Soit un réseau de flot $G = (V, A, c)$ muni d'un flot f et d'une coupe (S, T). On a

$$|f| = f(S, T)$$

autrement dit, la valeur du flot f est égale au flot associé toute coupe.

Explication — L'idée de la preuve

Flot associé à une coupe

$$f(S,P) = \color{lime}{\sum_{u\in S}\sum_{v\in P} f(u,v)} - \color{red}{\sum_{u\in S}\sum_{v\in P} f(v,u)}$$

Valeur du flot partant du sommet S, vers le puits, moins la valeur du flot allant du puits à la source.

Capacité d'une coupe

Definition Capacité d'une coupe

Soit un réseau de flot $G = (V, A, c)$ muni d'une source s, d'un puits p et d'une coupe (S, P). La capacité de la coupe (S, P) est la somme des capacités des arcs de S vers T

$$c(S, P) = \sum_{u \in S} \sum_{v \in P} c(u, v)$$

Lien coupes et flots

Lemme

Soit un réseau de flot $G = (V, A, c)$.
Pour tout flot f et toute coupe $V = (S, P)$ on a

$$|f| \leq c(S, P)$$

Lien coupes et flots

Lemme

Soit un réseau de flot $G = (V, A, c)$.
Pour tout flot f et toute coupe $V = (S, P)$ on a

$$|f| \leq c(S, P)$$

$$\begin{aligned}
|f| &= f(S, T) && \text{(Lemme précédent)} \\
&= \sum_{u \in S} \sum_{v \in P} f(u, v) - \sum_{u \in S} \sum_{v \in P} f(v, u) && \text{(Définition du flot associé à une coupe)} \\
&\leq \sum_{u \in S} \sum_{v \in P} f(u, v) \leq \sum_{u \in S} \sum_{v \in P} c(u, v) \\
&= c(S, P) && \text{(Définition de la capacité d'une coupe)}
\end{aligned}$$

Coupe-Min

Definition Coupe minimum

Soit un réseau de flot $G = (V, A, c)$ muni d'une source s, d'un puits p et d'une coupe (S, P). La coupe (S, P) est **minimum** si pour toute coupe (S', P') de G, on a

$$c(S, P) \leq c(S', P')$$

Coupe-Min

Definition Coupe minimum

Soit un réseau de flot $G = (V, A, c)$ muni d'une source s, d'un puits p et d'une coupe (S, P). La coupe (S, P) est **minimum** si pour toute coupe (S', P') de G, on a

$$c(S, P) \leq c(S', P')$$

Une coupe min indique donc la capacité minimum à retirer du graphe afin qu'il n'y ait plus de chemin de s à p.

Coupe-Min

Lemme

- Soit un réseau de flot $G = (V, A, c)$ muni d'un flot f et d'un graphe résiduel G_f.
- Soit une coupe $V = (S, V \setminus S)$ où S est l'ensemble des sommets accessible depuis s dans G_f.
- **Si G_f ne contient pas de chemin augmentant, alors $(S, V \setminus S)$ est une coupe minimum.**

- Si G_f ne contient pas de chemin augmentant, les seuls arcs entre S et $V \setminus S$ vont de $V \setminus S$ à S.

Coupe-Min

Lemme

- Soit un réseau de flot $G = (V, A, c)$ muni d'un flot f et d'un graphe résiduel G_f.
- Soit une coupe $V = (S, V \setminus S)$ où S est l'ensemble des sommets accessible depuis s dans G_f.
- **Si G_f ne contient pas de chemin augmentant, alors $(S, V \setminus S)$ est une coupe minimum.**

Coupe-Min

Lemme

- Soit un réseau de flot $G = (V, A, c)$ muni d'un flot f et d'un graphe résiduel G_f.
- Soit une coupe $V = (S, V \setminus S)$ où S est l'ensemble des sommets accessible depuis s dans G_f.
- **Si G_f ne contient pas de chemin augmentant, alors $(S, V \setminus S)$ est une coupe minimum.**

- **Cas (a)** Pour tout arc (v, u) de $V \setminus S$ à S dans G, on a $f(v, u) = 0$
  (sinon (u,v) est dans G_f, mais on vient de dire que c'est impossible)

Coupe-Min

Lemme

- Soit un réseau de flot $G = (V, A, c)$ muni d'un flot f et d'un graphe résiduel G_f.
- Soit une coupe $V = (S, V \setminus S)$ où S est l'ensemble des sommets accessible depuis s dans G_f.
- **Si G_f ne contient pas de chemin augmentant, alors $(S, V \setminus S)$ est une coupe minimum.**

- **Cas (b)** Pour tout arc (u, v) de S à $V \setminus S$ dans G, on a $f(v, u) = c(u, v)$
  (sinon (u,v) est dans G_f, mais on vient de dire que c'est impossible)

Coupe-min

- **Cas (a)** Pour tout arc (v, u) de $V \setminus S$ à S dans G_f, on a $f(v, u) = 0$
  (sinon (u,v) est dans G_f, mais on vient de dire que c'est impossible)
- **Cas (b)** Pour tout arc (u, v) de S à $V \setminus S$, on a $f(v, u) = c(u, v)$
  (sinon (u,v) est dans G_f, mais on vient de dire que c'est impossible)

Coupe-min

- **Cas (a)** Pour tout arc (v, u) de $V \setminus S$ à S dans G_f, on a $f(v, u) = 0$
  (sinon (u,v) est dans G_f, mais on vient de dire que c'est impossible)
- **Cas (b)** Pour tout arc (u, v) de S à $V \setminus S$, on a $f(v, u) = c(u, v)$
  (sinon (u,v) est dans G_f, mais on vient de dire que c'est impossible)

$$
\begin{aligned}
|f| &= f(S, T) && \text{(Lemme précédent)} \\
&= \sum_{u \in S} \sum_{v \in P} f(u, v) - \sum_{u \in S} \sum_{v \in P} f(v, u) && \text{(Définition du flot associé à une coupe)} \\
&= \sum_{u \in S} \sum_{v \in P} f(u, v) && Cas(a) \\
&= \sum_{u \in S} \sum_{v \in P} c(u, v) && Cas(b) \\
&= c(S, P) && \text{(Définition de la capacité d'une coupe)}
\end{aligned}
$$

On a donc $|f| = c(S, P)$, ce qui, d'après le Lemme précédent, est la valeur minimum car $|f| \leq c(S, P)$.

Le théorème max-flow min-cut

Théorème max-flow min-cut (Ford-Fulkerson 1954)

Soit un réseau de flot $G = (V, A, c)$ muni d'une source s, d'un puits p et d'un flot f. Les propositions suivantes sont équivalentes:

1. f est un flot maximum,
2. le réseau résiduel G_f ne contient pas de chemin augmentant,
3. il existe une coupe (S, P) pour G telle que $|f| = c(S, P)$.

Le théorème max-flow min-cut

Théorème max-flow min-cut (Ford-Fulkerson 1954)

Soit un réseau de flot $G = (V, A, c)$ muni d'une source s, d'un puits p et d'un flot f. Les propositions suivantes sont équivalentes:

1. f est un flot maximum,
2. le réseau résiduel G_f ne contient pas de chemin augmentant,
3. il existe une coupe (S, P) pour G telle que $|f| = c(S, P)$.

- $(1 \implies 2)$: si G_f contenait un chemin augmentant, alors il serait possible d'augmenter le flot, qui ne serait pas maximum.

Le théorème max-flow min-cut

Théorème max-flow min-cut (Ford-Fulkerson 1954)

Soit un réseau de flot $G = (V, A, c)$ muni d'une source s, d'un puits p et d'un flot f. Les propositions suivantes sont équivalentes:

1. f est un flot maximum,
2. le réseau résiduel G_f ne contient pas de chemin augmentant,
3. il existe une coupe (S, T) pour G telle que $|f| = c(S, T)$.

- $(2 \Longrightarrow 3)$: si G_F ne contient pas de chemin augmentant, alors on peut obtenir une coupe min en prenant S comme étant l'ensemble des sommets que l'on peut atteindre depuis s.

Le théorème max-flow min-cut

Théorème max-flow min-cut (Ford-Fulkerson 1954)

Soit un réseau de flot $G = (V, A, c)$ muni d'une source s, d'un puits p et d'un flot f. Les propositions suivantes sont équivalentes:

1. f est un flot maximum,
2. le réseau résiduel G_f ne contient pas de chemin augmentant,
3. il existe une coupe (S, T) pour G telle que $|f| = c(S, T)$.

- $(3 \Longrightarrow 1)$: on sait grâce au dernier Lemme, que la valeur de tout flot f est inférieure à $c(S, T)$. S'il existe une coupe telle que $|f| = c(S, T)$ alors le flot est maximum.

Complexité de Ford-Fulkerson avec la méthode d'Edmonds-Karp

Lemme

Dans un réseau de flot $G = (V, A, c)$, le nombre maximum de chemins augmentant découvert par la méthode de Edmonds-Karp est $\mathcal{O}(|V| \times |A|)$.

Explication

L'idée est la suivante:

- A chaque fois que l'on calcule un chemin augmentant, une arête du graphe est saturée. La distance entre la source et cet arc saturé augmente.[a]
- Sur un chemin augmentant, la distance entre la source et un arc saturé est au plus $|V|$
- chaque arc de A peut donc être saturé au plus $|V|$ fois.

[a] preuve par l'absurde

Complexité de Ford-Fulkerson avec la méthode d'Edmonds-Karp

Théorème 1972

Dans un réseau de flot $G = (V, A, c)$, la complexité de l'algorithme de Edmonds-Karp dans le pire des cas est $\mathcal{O}(|V| \times |A|^2)$

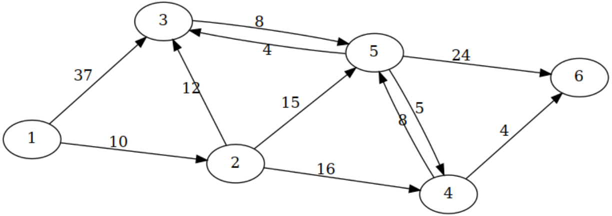
Algorithme de Dinitz

Théorème 1970

Dinitz est en réalité le premier à décrire l'algorithme de Edmonds-Karp, mais en Russie, avant la chute du rideau de fer. Il décrit ensuite une amélioration en $\mathcal{O}(|V|^2 \times |A|)$

Mise en pratique

Appliquons l'algorithme sur l'exemple suivant:



On indiquera un flot maximum et une coupe minimum.