Calcul de flots maximum
Algorithme de Ford-Fulkerson

Frederic.Guinand@univ-lehavre.fr

Master I - Le Havre

Réseau

- Soit $G = (S, A)$ un graphe orienté.
- Chaque arc est valué par une valeur correspondant à la capacité du lien associé à cet arc.
- Soit la fonction $c : A \rightarrow R^+$ qui associe à chaque arc une valeur réelle positive de capacité.
- Soient deux sommets particuliers : un sommet source s et un sommet puits p, tels que $\forall v \in S$ il existe un chemin qui va de s à p et qui passe par v.
- Un réseau est défini par la donnée du quadruplet $R = (G, c, s, p)$.

Un flot est une fonction $f : A \rightarrow R^+$ telle que :

$$f((a,b)) \leq c((a,b)), \forall (a,b) \in A \quad (1)$$

$$\sum_{j \in \text{voisins(s)}} f((s,j)) - \sum_{j \in \text{voisins(s)}} f((j,s)) = v \quad (2)$$

$$\sum_{j \in \text{voisins(p)}} f((p,j)) - \sum_{j \in \text{voisins(p)}} f((j,p)) = -v \quad (3)$$

$$\sum_{j \in \text{voisins(i)}} f((i,j)) - \sum_{j \in \text{voisins(i)}} f((j,i)) = 0, \forall i \in S \backslash \{s,p\} \quad (4)$$

- v représente la valeur du flot.
- Equation (4) : loi de Kirchhoff.

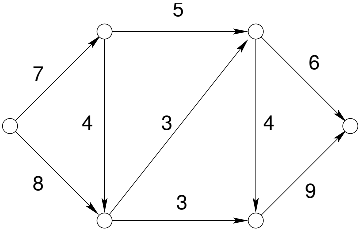
Coupe

Soit $G = (S, A)$, et $R = (G, c, s, p)$, une coupe est une partition de S en deux ensembles V et $\bar{V}$, tels que :

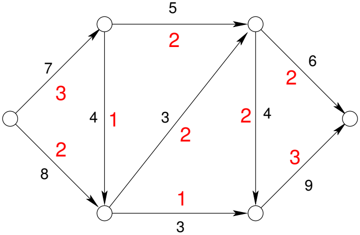
- $V \cup \bar{V} = S$
- $s \in V$ et $p \in \bar{V}$

la capacité de la coupe est égale à la somme des capacités des arcs qui font la coupe.

Un réseau



un flot réalisable pour ce réseau



Problème du calcul du flot maximum

Connectivité

- Pour un graphe connexe, la connectivité est définie comme le nombre minimum de sommets dont la suppression entraîne la perte de connexité.
- Théorème de Menger (1927) :
  Pour deux sommets non adjacents quelconques d'un graphe connexe G, le nombre minimum d'arêtes dont la suppression entraîne leur déconnexion est égal au nombre maximum de chaînes deux-à-deux arêtes disjointes les connectant dans G.
- Ce résultat a été re-formulé en termes de réseaux en 1956 par Ford et Fulkerson et est connu sous le nom de théorème *maximum flow/minimum cut* ou max flow-min cut.

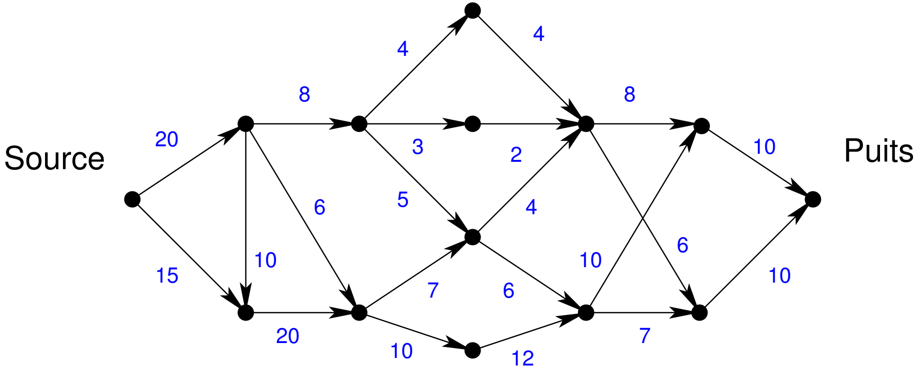
Enoncé du problème

1. **Intitulé du problème :** *Flot maximum*.
2. **Description des paramètres :** un graphe orienté $G = (S, A)$ dont chaque arête est valuée par une capacité, un sommet source et un sommet puits,
3. **Question :** quel est le flot maximum qu'il est possible de faire passer dans ce réseau depuis la source vers le puits ?

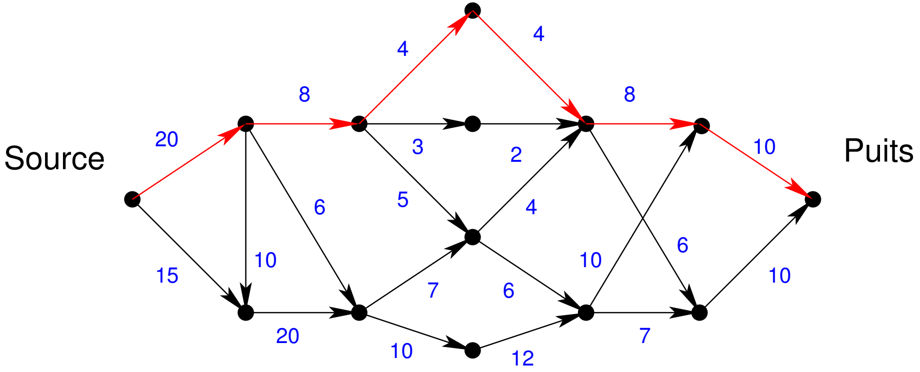
Enoncé du problème

- Intuitivement cela fait référence aux problèmes de plomberie ou de trafic routier.
- On dispose d'une source et d'un puits et le flot doit s'écouler de la source vers le puits et il doit être maximum en fonction de la capacité des arcs.

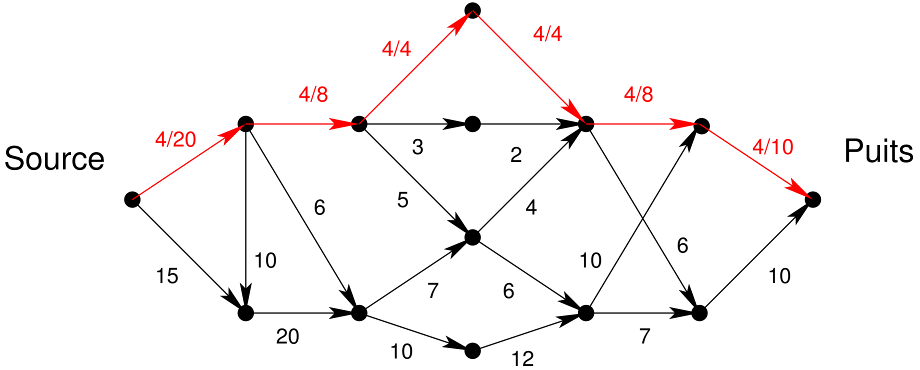
Exemple



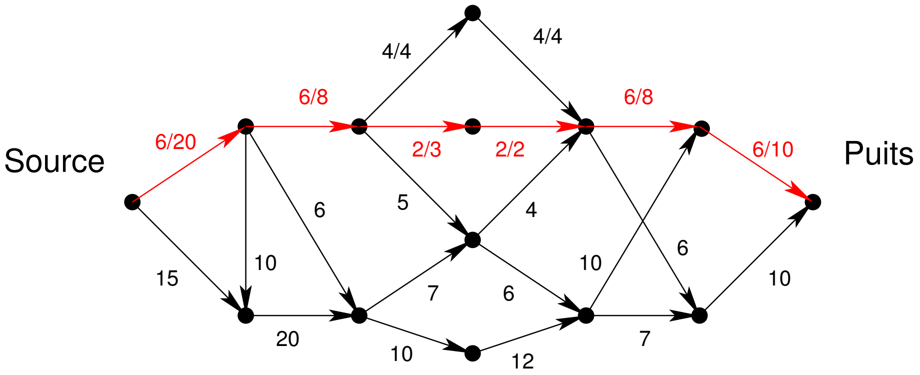
Exemple



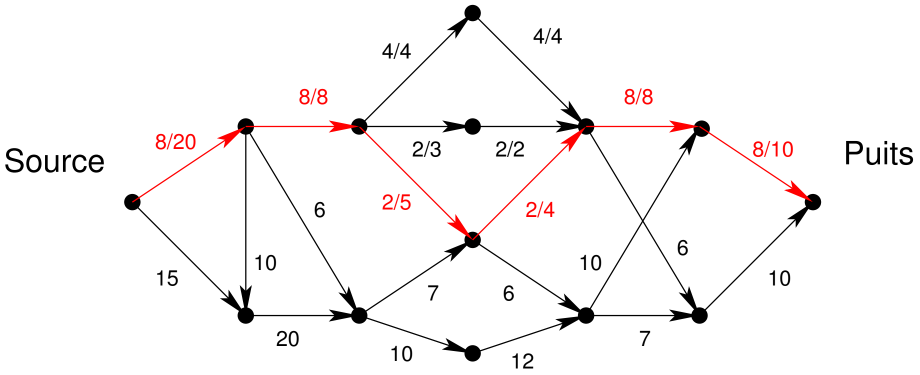
Exemple



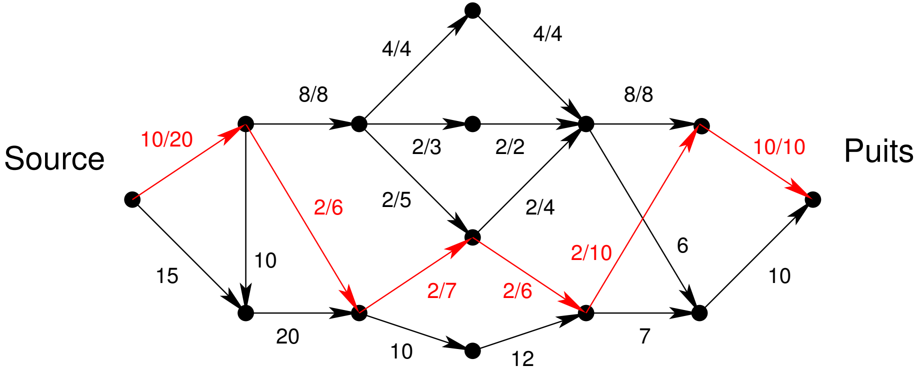
Exemple


Source
Puits
6/20
6/8
4/4
4/4
6/8
2/3
2/2
6/10
15
10
6
5
4
6
10
7
6
10
20
10
12
7

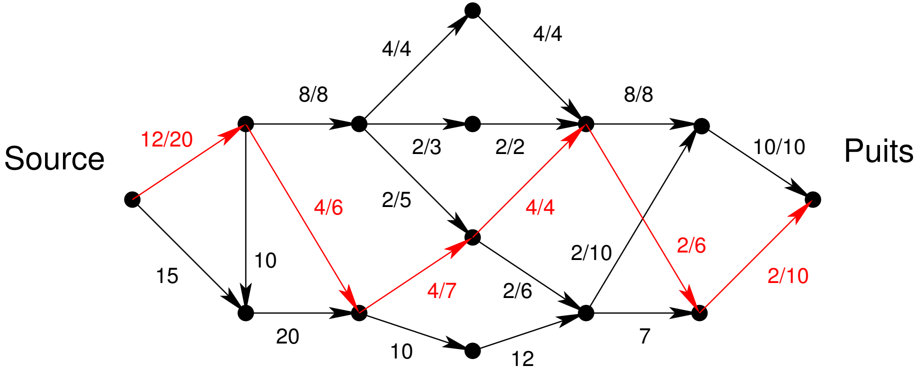
Exemple


Source
Puits
8/20
15
8/8
10
6
4/4
2/3
2/5
20
7
10
4/4
2/2
2/4
6
12
8/8
10
6
7
8/10
10

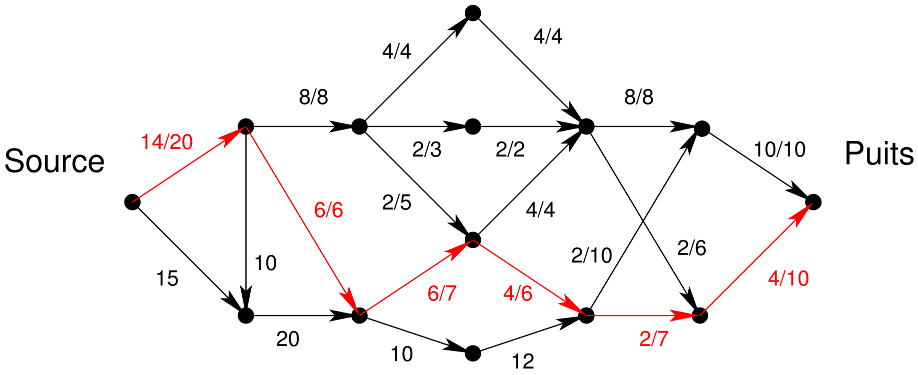
Exemple


Source
Puits
10/20
15
8/8
10
2/6
20
4/4
2/3
2/5
10
2/7
4/4
2/2
2/4
12
2/6
8/8
2/10
6
7
10/10
10

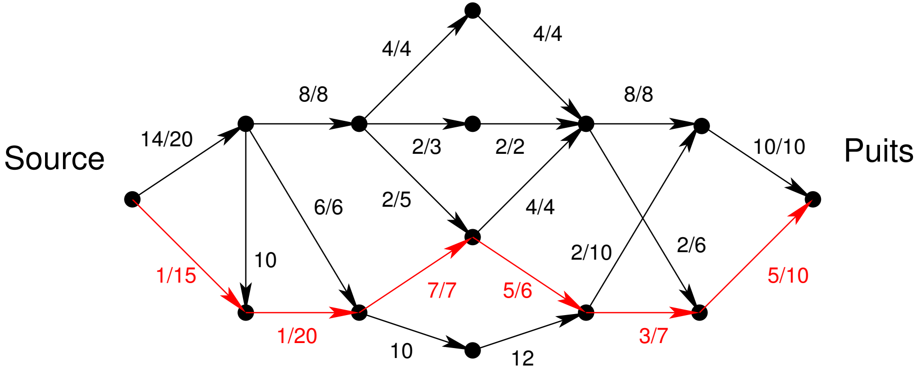
Exemple


4/4
4/4
8/8
8/8
12/20
2/3
2/2
10/10
Source
Puits
4/6
2/5
4/4
2/6
2/10
10
15
4/7
2/6
2/10
20
10
12
7

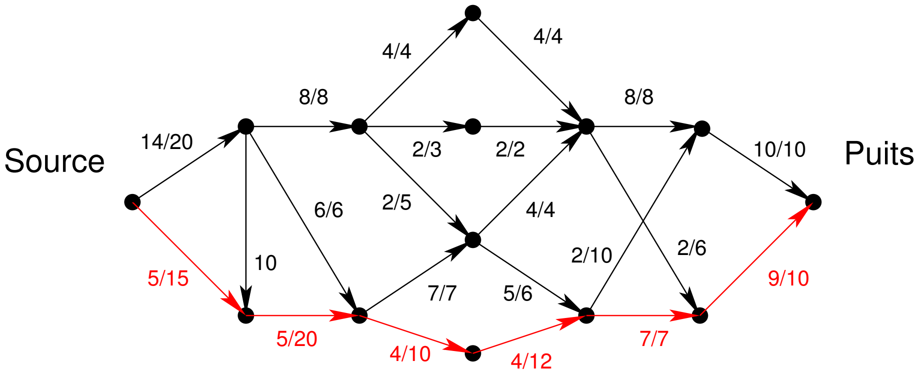
Exemple


Source
Puits
14/20
15
10
6/6
8/8
20
4/4
2/3
2/5
6/7
10
4/4
2/2
4/4
4/6
12
8/8
2/10
2/6
2/7
10/10
4/10

Exemple

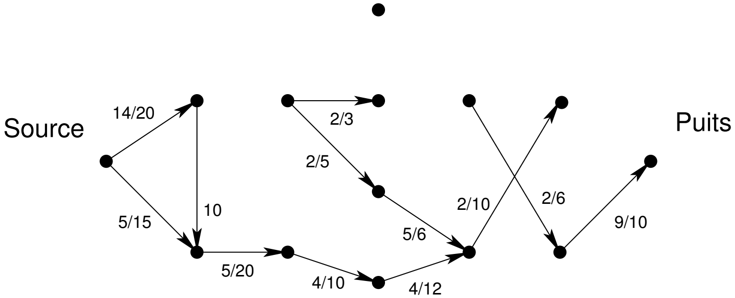

Source
Puits
14/20
1/15
10
8/8
6/6
1/20
4/4
2/3
2/5
7/7
10
4/4
2/2
4/4
5/6
12
2/10
8/8
2/6
3/7
10/10
5/10

Exemple


Source
Puits
14/20
5/15
10
8/8
6/6
5/20
4/4
2/3
2/5
7/7
4/10
4/4
2/2
4/4
5/6
4/12
8/8
2/10
2/6
7/7
10/10
9/10

Rapport avec la connexité

Il suffit de supprimer les arcs pour lesquels la valeur du flot est égale à la capacité :



Capacité résiduelle

$$c_{(s_i,s_j)}^{\text{résiduelle}} = c_{(s_i,s_j)} - f((s_i, s_j))$$

la capacité résiduelle, pour un arc (s_i, s_j) donné, représente la quantité de flot pouvant encore passer par cet arc.

Réseau résiduel

$G_{\text{résiduel}} = (S, A_{\text{résiduel}})$ tel que

$A_{\text{résiduel}} = \{(s_i, s_j) \in A, c_{(s_i,s_j)}^{\text{résiduelle}} > 0\}$

le réseau résiduel, est le graphe partiel obtenu à partir du graphe d'origine mais dont ont été retiré tous les arcs dont la capacité résiduelle est nulle.

Chemin améliorant ou augmentant

Un chemin améliorant ou chemin augmentant, est un chemin de $G_{\text{résiduel}}$, allant de s à p et sans circuit

Capacité résiduelle d’un chemin améliorant

La capacité résiduelle d’un chemin améliorant est le minimum des capacités résiduelles des arcs appartenant au chemin

Arc saturé

Un arc est dit saturé si sa capacité résiduelle est nulle.

Algorithme de Ford-Fulkerson

Principe de résolution

Principe

Tant qu'il existe un chemin augmentant dans le graphe, on ajoute un flot le long de ce chemin.

Algorithme de Ford-Fulkerson

Principe

Créer un graphe résiduel $G_r = G$
Initialement tous les arcs sont valués par leur capacité
Déterminer un chemin augmentant
TantQue il existe un chemin augmentant de s vers p **Faire**
    mettre à jour le graphe résiduel
    chercher un nouveau chemin augmentant
finTantQue

Algorithme de Ford-Fulkerson

Principaux points

- déterminer un chemin augmentant
- mettre à jour le graphe résiduel

Algorithme de Ford-Fulkerson

Mise à jour du graphe résiduel

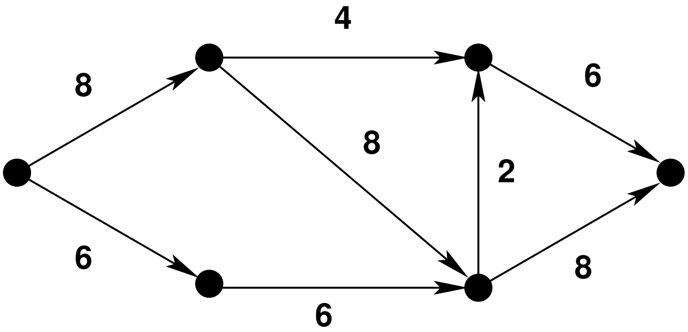
Soit c un chemin augmentant
Soit c_r la capacité résiduelle de ce chemin
Pour tous les arcs (s_i, s_j) appartenant au chemin **Faire**
 mettre à jour la capacité résiduelle de l'arc
finPour

Algorithme de Ford-Fulkerson

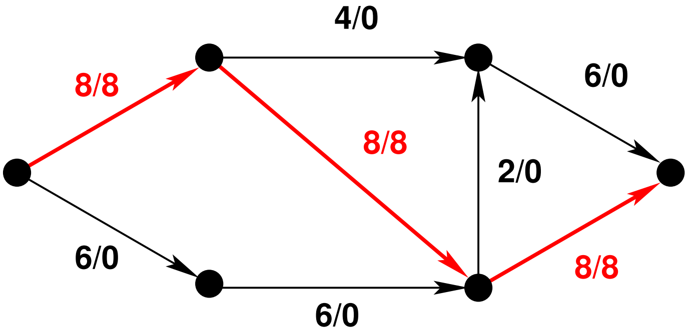
Détermination d'un chemin augmentant

- Plus délicat.
- Examen des chemins du graphe résiduel $\rightarrow$ parcours du graphe résiduel.
- Il peut y avoir remise en question de flots déjà validés.

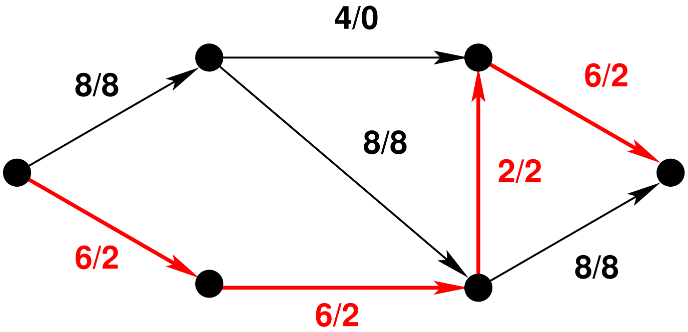
Exemple



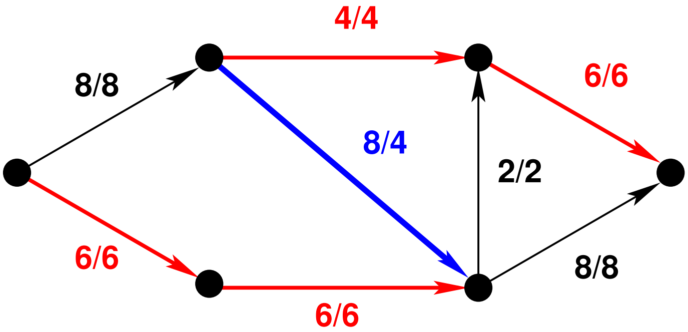
Exemple



Exemple



Exemple



Algorithme de Ford-Fulkerson

Départ de la source
Voisins éligibles $\leftarrow$ ensemble des voisins...
...dont les arcs entrants depuis la source ne sont pas saturés
TantQue (il reste des voisins non visités) ET (chemin pas trouvé) Faire
    choisir un voisin pour poursuivre le chemin augmentant
    Si ce voisin est le puits Alors
        le chemin construit est le chemin augmentant
        valider le flot
        mettre à jour le graphe résiduel
        retour au départ pour la recherche d'un nouveau chemin
    finSi
finTantQue

Algorithme de Ford-Fulkerson

Voisins éligibles

- Les voisins éligibles d'un sommet u sont :
  - les successeurs v de ce sommet tels que l'arc (u, v) n'est pas saturée
  - les prédecesseurs v' de ce sommet tels que l'arc (v', u) présente un flot strictement positif

Algorithme de Ford-Fulkerson

Condition de terminaison

- Le calcul du flot maximum se termine lorsqu'il n'existe plus de chemin augmentant de s vers p.
- Détecter une telle situation suppose de marquer les sommets à partir desquels aucun nouveau chemin augmentant n'est possible. Cette procédure de marquage peut s'appuyer par exemple sur un algorithme de parcours classique :
  - profondeur d'abord : lorsque la pile est vide et qu'aucun chemin augmentant de s à p n'a pu être déterminé
  - largeur d'abord : lorsque la file est vide et qu'aucun chemin augmentant de s à p n'a pu être déterminé

Analyse de la complexité

Complexité

- Exercice.

Références

Références

1. *Maximal Flow Through a Network*. L. R. Ford, Jr. and D. R. Fulkerson. Canadian Journal of Mathematics, vol. 8, pages 399-404, 1956.
2. *Ford-Fulkerson Max-Flow Labeling Algorithm*. Harvey J. Greenberg. Rapport de recherche. *University of Colorado, Denver*. Decembre 1998.