Problème de flot maximum

- Trouver le flot maximum qui peut circuler dans un réseau de transport ou informatique.
- Représentation par un graphe où les arcs sont valués par les capacités (valeur maximale qui peut circuler sur cet arc).
- On rajoute (s'ils n'existent pas) une source et un puits.
- La source est reliée aux points de départ, avec comme valuations les disponibilités de ces points de départ.
- Les points d'arrivée sont reliés au puits avec comme valuations les demandes de ces points d'arrivée.
- On rajoute un arc de retour virtuel de p à s.

Définition du problème

Définition 1 : Un **flot** est une fonction s'appliquant sur les arcs d'un réseau de transport telle que :

- sur chaque arc, le flot est positif ou nul et inférieur à la capacité,
- en chaque noeud, la quantité totale de flot entrant est égale à la quantité totale de flot sortant (équivalent loi de Kirchhoff en électricité)

Le problème: Trouver un flot qui maximise $f(p, s)$ (= flot circulant dans le réseau).
(il peut exister plusieurs solutions de même valeur - on n'en cherche qu'une)

Définitions et Exemple

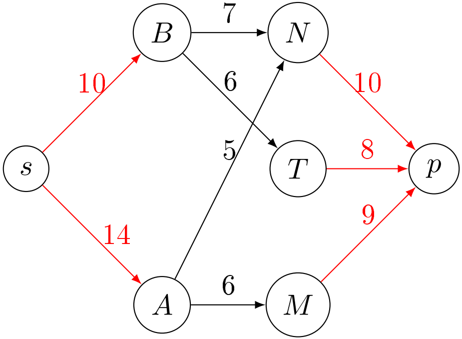
Définition 2: un arc (x, y) sur lequel le flot est égal à la capacité est dit **saturé**.

Définition 3: un flot tel que, sur chaque chemin allant de s à p, il existe au moins un arc saturé est dit **complet**.

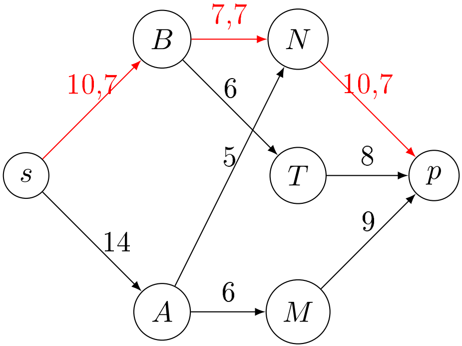
Exemple

On doit acheminer le maximum de marchandises de la Corse au Continent, sachant que l'on dispose de 10 tonnes à Bastia et 14 tonnes à Ajaccio. Les 3 ports d'accueil sont Nice, Toulon et Marseille qui disposent respectivement d'une demande de 10, 8 et 9 tonnes. Les liaisons maritimes disponibles et leurs capacités sont: Bastia $->$ Nice (7T), Bastia $->$ Toulon (6T), Ajaccio $->$ Nice (5T) et Ajaccio $->$ Marseille (6T).

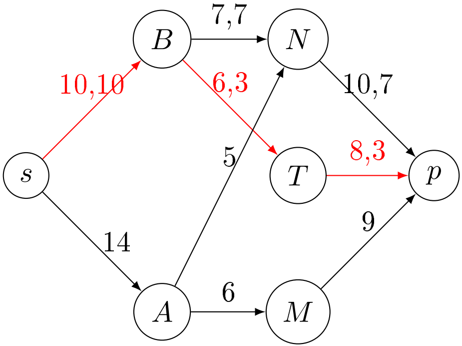
Un flot complet sur l'exemple



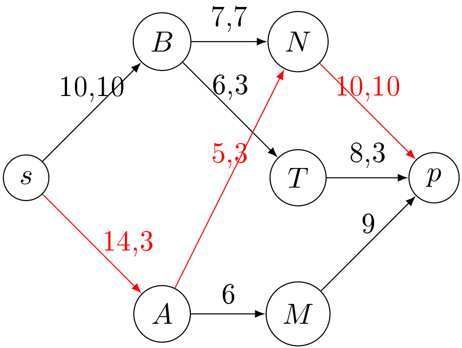
Un flot complet sur l'exemple(2)



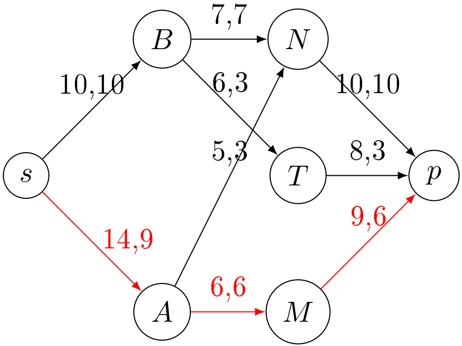
Un flot complet sur l'exemple(3)



Un flot complet sur l'exemple(4)



Un flot complet sur l'exemple(5)



Chaine améliorante

Remarque: Un flot complet n'est pas forcément optimal (mais un flot optimal est toujours complet)

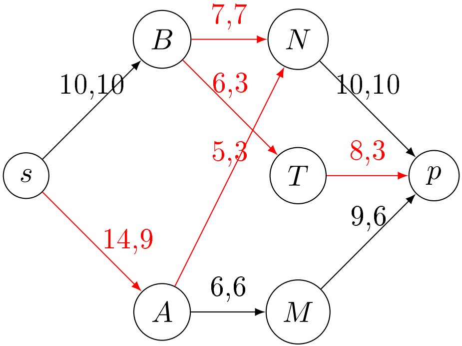
Procédure de marquage: on marque la source

- Marquage avant: Tout successeur d'un sommet marqué est marqué à son tour si l'arc entre les deux n'est pas saturé (augmentation possible du flot)
- Marquage arrière: Tout prédécesseur d'un sommet marqué est marqué à son tour si le flot sur l'arc entre les deux n'est pas nul (diminution possible du flot).

Si on arrive à marquer le puits, le flot actuel n'est pas optimal. La chaine qui a permis de marquer le puits est une **chaine améliorante** sur laquelle il convient d'augmenter autant que possible le flot.
Lorsqu'il n'est plus possible de marquer le puits, le flot est optimal. La frontière entre sommets marqués et sommets non marqués constitue le goulet d'étranglement du réseau. C'est une **coupe minimale**.

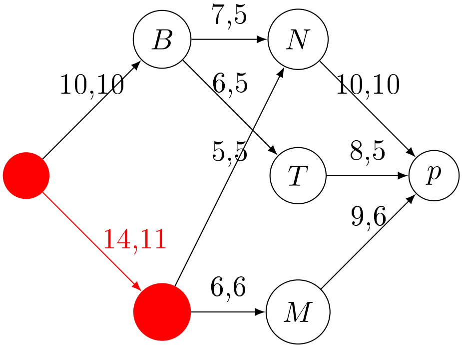
(coupe = somme des capacités des arcs séparant deux sous ensembles de sommets)

Une chaîne améliorante sur l'exemple



Sur la chaine $s- > A- > N < -B- > T- > p$, capacité d'amélioration de 2 (du fait de $A- > N$).

La solution optimale de valeur 21



La coupe minimale sépare les sommets marqués (s, A) du reste du graphe: arcs $(s, B), (A, N), (A, M)$ de capacité totale 21.

Graphe d'écart

Il est peut-être plus simple de présenter cela sur le graphe d'écart:

Graphe d'écart :

- Mêmes sommets
- Dès qu'un arc (a, b) de capacité c n'est pas saturé avec un flot $f < c$, créer l'arc (a, b) dans le graphe d'écart et le valuer par $c - f$ (capacité d'augmentation)
- Dés qu'un arc (a, b) a un flot $f > 0$, créer l'arc (b, a) dans le graphe d'écart et le valuer par f (capacité de diminution)

Tant qu'il existe un chemin de s à p dans le graphe d'écart, le flot n'est pas optimal. Ce chemin $=$ chaîne améliorante.

Algorithme sur le graphe d'écart

$VF = 0$. Graphe initial = Réseau de transport.
Tant qu'il existe un chemin de s à p faire:

- Soit w la valuation minimale sur ce chemin. $VF+ = w$
- Pour chaque arc (a, b) de ce chemin faire:
  - $capa(a, b) - = w$. Si $capa(a, b) = 0$, alors supprimer l'arc.
  - Si l'arc (b, a) n'existe pas, le créer. $capa(b, a) + = w$

Complexité

Edmonds et Karp ont proposé une implémentation particulière de l'algorithme de Ford-Fulkerson en parcours largeur (BFS) qui consiste à toujours choisir une chaine améliorante de s à p de plus petite longueur (avec le moins d'aretes possible).

Cet algorithme se termine toujours (même pour des capacités non entières, contrairement à Ford-Fulkerson) avec une complexité en $O(nm^2)$ (n sommets, m arêtes).

Couplage maximum

Un **couplage** est un ensemble d'arêtes qui n'ont pas de sommets en commun.
On peut aussi dire que c'est un ensemble de couples de sommets, tous différents, reliés par une arête (monogamie obligatoire).

Un **couplage maximum** est un couplage de cardinalité la plus grande possible, donc avec le plus grand nombre possible d'arêtes.

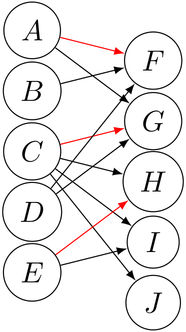
Dans le cas d'un graphe biparti

C'est le même algorithme que Ford-Fulkerson.
On rajoute une source qu'on relie avec tous les sommets d'une des 2 parties. Et un puits qu'on relie à tous les autres sommets. Toutes les capacités sont égales à 1.
Et de la même façon, on recherche des chaines améliorantes par marquage (avant et arrière) jusqu'à ce qu'on ne puisse plus marquer le puits.
Par contre complexité en $O(nm)$.

Un petit exemple

Chaîne améliorante: $B- > F < -A- > G < -C- > I$.

Couples: $(A, G), (B, F), (C, I), (E, H)$

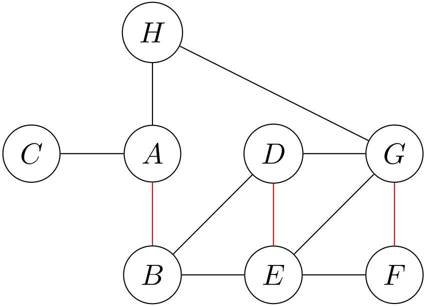


Dans le cas d'un graphe quelconque: algorithme d'Edmonds

Idée algorithme

- Plus possible de construire un réseau de transport et d'appliquer FF.
- Mais on continue à avoir des chaines améliorantes et à avoir la propriété: *l'optimum est atteint quand il n'y a plus de chaine améliorante*
- On prend d'abord (tant qu'on peut) des aretes indépendantes.
- Puis on recherche l'existence d'une chaine améliorante
  - qui commence et se termine par 2 sommets non couverts
  - qui alterne arete non prise - arete du couplage actuel

Un petit exemple



Un petit exemple (suite)

On a construit une solution initiale, en prenant les sommets dans l'ordre alphabétique. Donc, on prend (A, B), C ne peut plus être relié, on prend (D, E) puis (F, G).
Il faut trouver la chaîne améliorante: C - A - B - D - E - F - G - H, qui permet de trouver le couplage parfait avec 4 aretes.