M1 Informatique – Graphes II et Réseaux
Problème du flot maximum dans un réseau

Irena.Rusu@univ-nantes.fr
LS2N, bât. 34, bureau 303
tél. 02.51.12.58.16

- Introduction au cours
- Réseaux de transport
- Flot maximum
- Méthode de Ford-Fulkerson
- Méthode des préflots

- Introduction au cours
- Réseaux de transport
- Flot maximum
- Méthode de Ford-Fulkerson
- Méthode des préflots

- Objectifs
  - Glisser depuis la notion de graphe (vue « statique ») vers la notion de « réseau » (vue « dynamique »)
  - Etudier des problèmes de transport divers, formulés et résolus comme des problèmes de graphes
  - Connaître leurs applications
  - Faire le lien, de nouveau, entre des problèmes et méthodes de la R.O. et de la théorie des graphes

- Contenu
  - Problème du flot maximum
  - Problème du flot maximum de coût minimum
  - Problèmes de transbordement

  - En distanciel : problèmes de transbordement avec capacités (sem. 48, 49)

- CM : sans téléphones, tablettes etc.
- Distanciel : évalué à l'examen
- TP : binôme, avec un programme et un rapport à rendre

  $\rightarrow$ noteTP

- CC le jeudi 21/11/2019 à 9h30, amphi G (durée à fixer) $\rightarrow$ noteCC

Note finale = (NoteCC+NoteTP)/4 + NoteExam/2

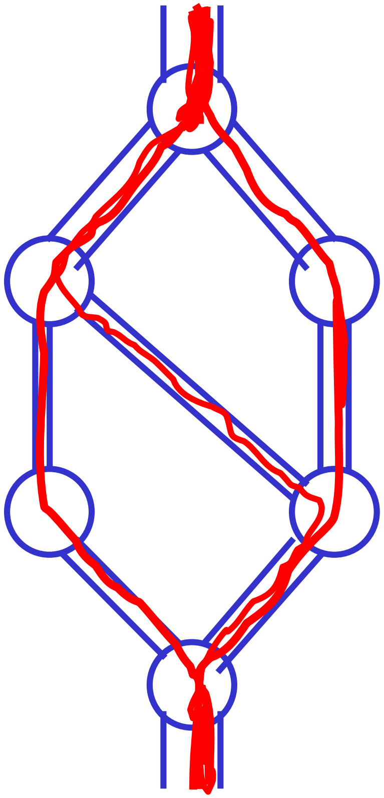
- Remarque :

  toute absence non-justifiée de plus d'1h20 en TP signifie non-participation, et donc une noteTP=0.

- Introduction au cours
- Réseaux de transport
- Flot maximum
- Méthode de Ford-Fulkerson
- Méthode des préflots

Graphe orienté valué $G = (S, A, c)$

$c(p, q)$: capacité de l'arc (p, q)
$f(p, q)$: débit ou flot de l'arc (p, q)



Exemples
Canalisations hydrauliques
Voies de circulation
Réseau de communication

Capacité $c : S \times S \rightarrow \mathrm{R}$ avec $c(p, q) \geq 0$ si $(p,q) \epsilon A$
et $c(p, q) = 0$ si $(p, q) \notin A$

Flot $f : S \times S \rightarrow \mathrm{R}$

source $s \in S$, puits $t \in S$

Accessiblité
tous les sommets sont sur un chemin de s à t

Contrainte de capacité
pour tous $p, q \in S \quad f(p, q) \leq c(p, q)$

Anti-symétrie

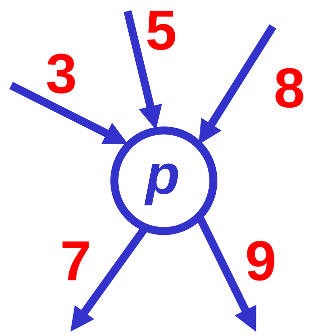
pour tous $p, q \in S \quad f(q, p) = - f(p, q)$

$p \xrightarrow{17} q$ flot direct

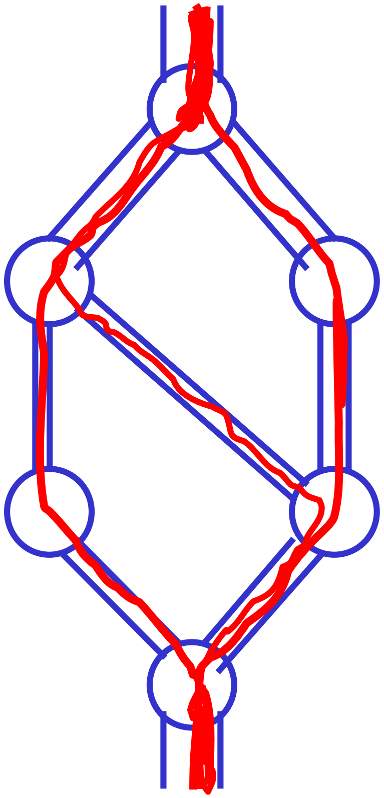
$p \xleftarrow{-17} q$ flot prélevable

Conservation du flot

pour tout $p \in S \setminus \{s,t\} \quad \Sigma\, (f(p, q) \mid q \in S) = 0$



- Introduction au cours
- Réseaux de transport
- Flot maximum
- Méthode de Ford-Fulkerson
- Méthode des préflots



Valeur du flot

$| f | = \Sigma\, (f\,(s, q) \mid q \in S\,)$

ce qui part de la source

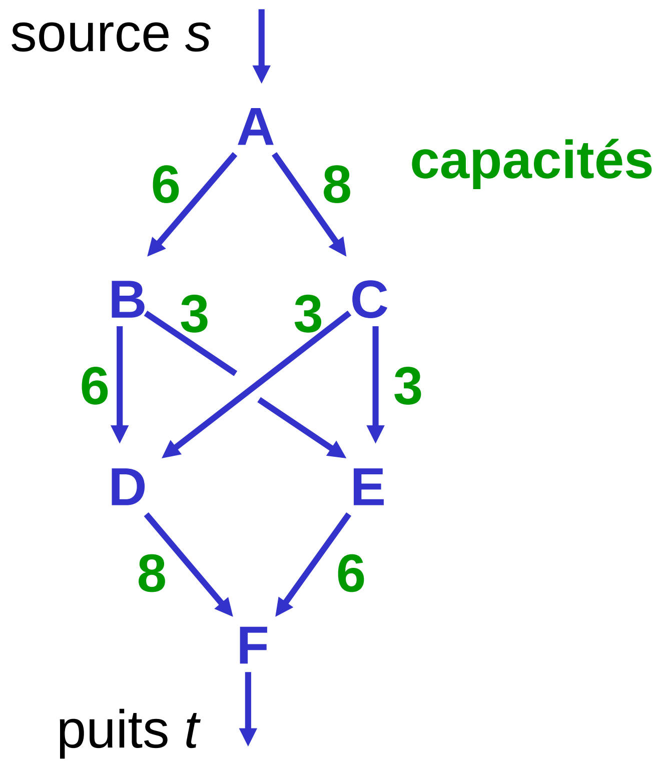
Propriétés

- pour tout $p \in S \quad f\,(p, p) = 0$
- pour tout $q \in S \setminus \{s,t\}\ \Sigma\, (f\,(p, q) \mid p \in S\,) = 0$
- $| f | = \Sigma\, (f\,(p, t) \mid p \in S\,)$

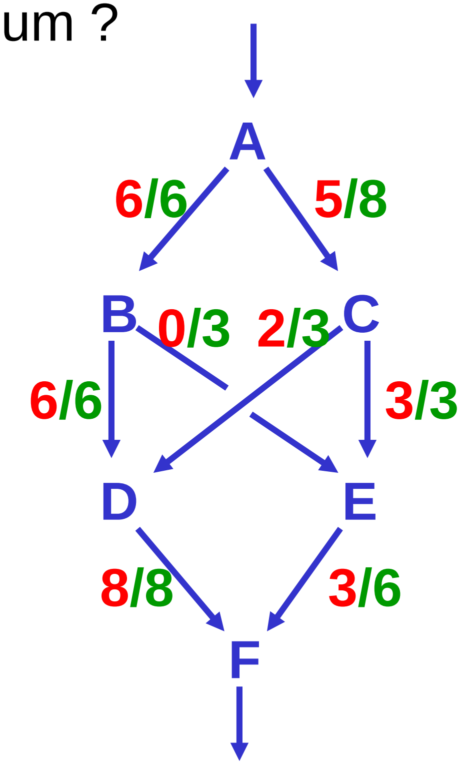
  ce qui arrive au puits

- Matrice d'adjacence +
  matrice des capacités +
  matrice des flots
- Listes des successeurs
  avec capacités et flots
- Graphique

Graphe orienté valué $G = (S, A, c)$
Calculer le flot maximum,
c'est-à-dire un flot f dont la valeur est maximum



un flot maximum ?
$| f | = 11$



- Coupe : partition S=X U Y avec s ϵ X et t ϵ Y

- Capacité d'une coupe S= X U Y:

$$c(X,Y)= \sum\{c(p,q) \mid p \in X \text{ et } q \in Y\}$$

- **Théorème** (MaxFlow-MinCut)

  Dans un réseau, le flot maximum = la capacité minimum d'une coupe.

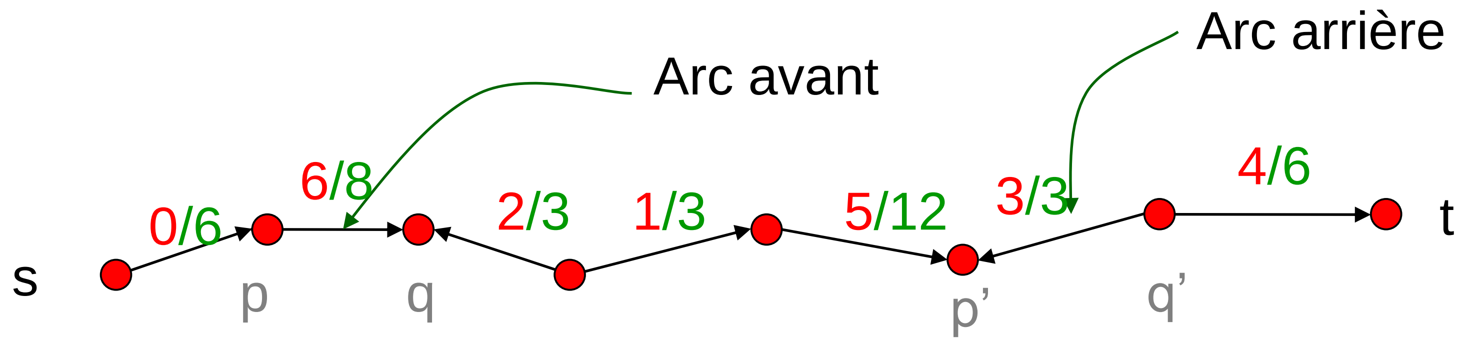
- Introduction au cours
- Réseaux de transport
- Flot maximum
- Méthode de Ford-Fulkerson
- Méthode des préflots

Idée générale

1 initialiser le flot *f* à 0

2 **tant qu**'il existe un chemin (pas forcément orienté) *P* de *s* à *t* sous-utilisé
faire augmenter le flot *f* sur ce chemin ;

3 retourner le flot *f*



Remarque. Un arc arrière n'est pris que si l'arc avant correspondant est absent ou rempli.

Capacités restant disponibles le long de *P* :

$$c_f(p,q) = c(p,q)-f(p,q),\ (p,q) \text{ arc de } P$$

Capacité résiduelle de *P*:

$$c_f(P)=\min\{c_f(p,q),\ (p,q) \text{ arc de } P\}$$

Théorème

1. Si *P* satisfait $c_f(P)>0$, alors la fonction f' définie par

   $f'(p,q) = f(p,q)+c_f(P);\ f'(q,p)=- f(p,q)$ si (p,q) arc de P

   $f'(p,q) = f(p,q)$ si $(p,q),(q,p)$ ne sont pas sur P

   est un flot de *G de valeur* $|f'|=|f|+c_f(P)>|f|$.

2. Soit $A_f =\{(p,q)\in SxS \ /\ c_f(p,q) > 0\}$ et soit $G_f=(S,A_f)$ le réseau résiduel de G. S'il n'existe aucun chemin de *s* à *t* dans G_f alors le flot est maximum.

Algorithme Ford-Fulkerson ($G=(S,A,c), s, t$)

Début

Pour chaque arc (p,q) de A faire $f(p,q) \leftarrow 0; f(q,p) \leftarrow 0;$

Tant qu'il existe dans G_f un chemin orienté P de s à t faire

$c_f(P)=min\{c_f(p,q), (p,q) \text{ arc de } P\}$;

pour chaque arc (p,q) de P faire

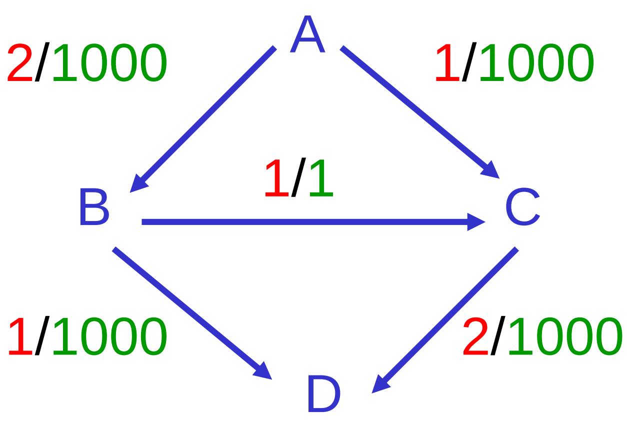
$f(p,q) \leftarrow f(p,q)+c_f(P)$

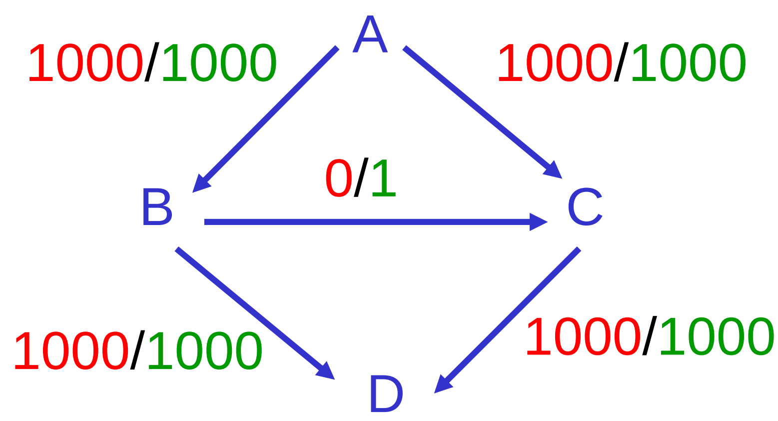
$f(q,p) \leftarrow - f(p,q)$

Fin

Remarque : une fois qu'il n'y a plus de chemin orienté, la partition X U Y donnée par la coupe minimum s'obtient en cherchant dans G_f tous les sommets accessibles depuis s (=X) et tous les autres (=Y)

Le nombre d'itérations dépend du choix des chemins





augmentations	chemins
1	ABCD
1	ACBD
1	ABCD
etc.	

augmentations	chemins
1000	ABD
1000	ACD
flot maximum	

Pour augmenter le flot :

choisir le plus court chemin

→ algorithme d’Edmonds-Karp

Théorème

Le calcul du flot maximum obtenu avec cette stratégie nécessite l'examen de moins de $|S|.|A|$ chemins.

Ford-Fulkerson

- peut ne pas converger
- en $O(|A|\ f^*)$ si les capacités sont entières, où f^* est le flot maximum.

Edmonds-Karp

- converge toujours
- en $O(|S|.|A|^2)$

Note. Edmonds et Karp étudient aussi la variante utilisant le chemin avec la plus grande capacité résiduelle : $O(|A|^2 \log f^*)$

- Introduction au cours
- Réseaux de transport
- Flot maximum
- Méthode de Ford-Fulkerson
- Méthode des préflots

- Implémentation simple: $O(|S|^2|A|)$
- Implémentation plus soignée : $O(|S|^3)$

- L'algorithme met à jour un *préflot:*

  une sorte de flot où la condition de conservation de flot est assouplie

Principes et vocabulaire

- Les sommets ont des *hauteurs.*
- Le flot est *avancé* le long d'un arc vers un sommet plus bas
- Certains sommets sont *débordants.*

Préflot : fonction f: S x S → R

- Anti-symétrie $f(u,v) = - f(v,u)$
- Contraintes de capacité $f(u,v) \leq c(u,v)$
- Conservation faible du flot dans v:

  $\Sigma\ (f(u,v)|u \in S) \geq 0$ (flot net arrivant)

Notations :

- $e(v) = \Sigma(f(u,v)|u \in S)$
- v est débordant si $v \neq s,t$ et $e(v) > 0$

Hauteur : fonction $h: S \rightarrow N$:

- $h(s) = |S|$
- $h(t) = 0$
- Pour tous $(u,v) \in A_f$ $h(u) \leq h(v)+1$

$\text{Initialiser-Préflot}(G, s) =$

$\quad$ **pour tout** $u \in S$ **faire**

$\quad\quad h[u] \leftarrow 0 \; ; \; e[u] \leftarrow 0$

$\quad h[s] \leftarrow |S|$

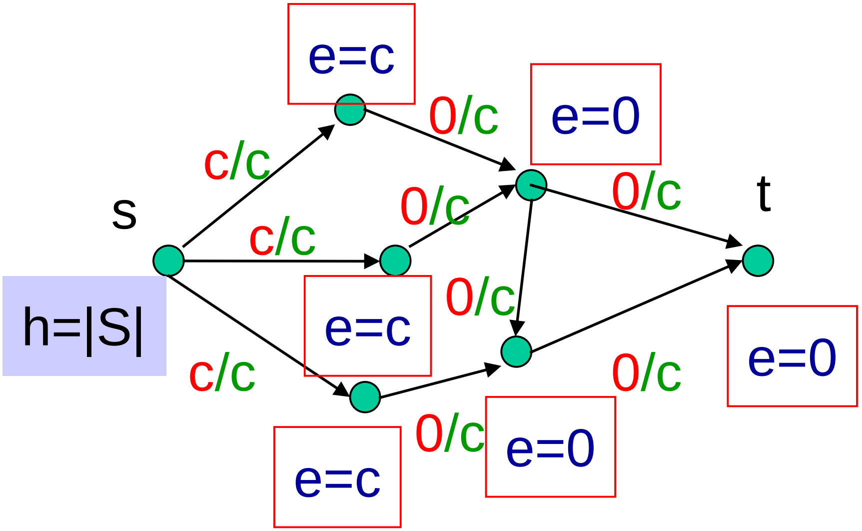
$\quad$ **pour tout** $(u, v) \in A$ **faire**

$\quad\quad f[u, v] \leftarrow 0 \; ; \; f[v, u] \leftarrow 0$

$\quad$ **pour tout** $u \in Adj(s)$ **faire**

$\quad\quad f[s, u] \leftarrow c(s, u) \; ; \; f[u, s] \leftarrow -c(s, u)$

$\quad\quad e[u] \leftarrow c(s, u) \; ; \; e[s] \leftarrow e[s] - c(s, u)$



Avancer(u,v) : *u ≠ s,t*

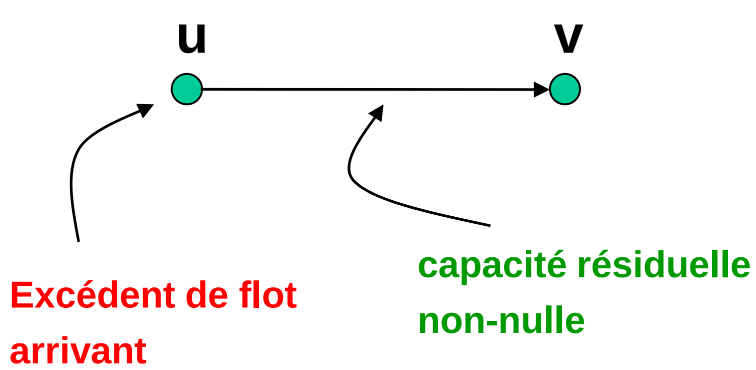
$$(\text{Précondition: } e[u] > 0,\ c_f(u,v) > 0, \text{ et } h[u] = h[v] + 1)$$

$$(\text{Effet: Ajoute } d_f(u,v) = \min(e[u], c_f(u,v)) \text{ au flot de } u \text{ à } v)$$

$$d_f(u,v) \leftarrow \min(e[u], c_f(u,v))$$

$$f[u,v] \leftarrow f[u,v] + d_f(u,v)\ ;\ f[v,u] = -f[u,v]$$

$$e[u] \leftarrow e[u] - d_f(u,v)\ ;\ e[v] \leftarrow e[v] + d_f(u,v)$$



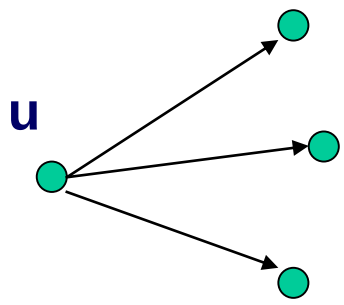
Une partie de l'excédent ne dépassant pas la capacité résiduelle peut être avancée.

Elever(u) : $u \neq s,t$

(Précondition: $e[u] > 0$ et $\forall (u,v) \in A_f.\ h[u] \leq h[v]$)

(Effet: Augmente la hauteur de u)

$$h[u] \leftarrow 1 + \min\{h[v] \ : \ (u,v) \in A_f\}$$



Donne à u la plus grande hauteur permise par les contraintes sur la fonction hauteur; prépare Avancer

Préflot-générique(*G,s,t*)

Initialiser-Préflot(G,s)

tant qu'il est possible d'appliquer une opération Avancer ou Elever

faire choisir une opération Avancer ou Elever applicable

l'exécuter

fin tant que

Propriétés.

1. un sommet débordant peut être soit avancé soit élevé
2. la fonction h calculée par l'algorithme est une fonction de hauteur.
3. à tout moment de l'algorithme, le réseau résiduel G_f par rapport au préflot f ne contient aucun chemin de s à t.

Théorème. A la fin de l'algorithme Préflot-Générique, le préflot calculé est un flot maximum de G.

Elévations:

- Au plus 2|S|-1 par sommet, puisque h(u)<=2|S|-1 pour tout u.
- Au plus (2|S|-1)(|S|-2) élévations c.à.d. $O(|S|^2)$ en tout

Avancées « saturantes » ($c_f(u,v)$ devient 0):

- Au plus 2|S|.|A|

Avancées non-saturantes:

- Au plus $4|S|^2(|S|+|A|)$

Temps global: $O(|S|^2|A|)$

On peut atteindre $O(|S|^3)$ avec une organisation plus précise des opérations Avancer et Elever.

- C'est une première version de problème de transport
- D'autres versions :
  - Flot maximum de coût minimum (cours suivant)
  - … dans un graphe avec capacités minimums
  - … dans un graphe avec plusieurs sources/puits
  - … en acheminant des produits divers ($\rightarrow$ multiflots)
  - … avec des offres/demandes dans les sommets (problème de transbordement, dernier cours)

T.H. Cormen, C. E. Leiserson, R.L. Rivest, C. Stein – Introduction à l’algorithmique

(Existe à la BU Sciences en différentes éditions et sur la page Madoc, en anglais)