Université de Nantes

M1 Informatique – Graphes II et Réseaux
Flots maximum de coût minimum

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

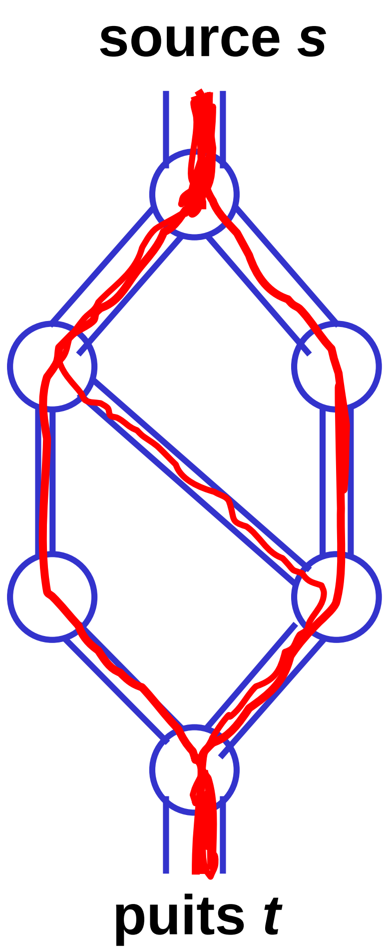
- Réseaux de transport avec des coûts
- Le problème
- Approches
  - Plus courts chemins de coût minimum
  - Circuits de coût négatif

- Réseaux de transport avec des coûts
- Le problème
- Approches
  - Plus courts chemins de coût minimum
  - Circuits de coût négatif

- Un nouveau problème de flot maximum entre une source et un puits
- Il s'agit de trouver le flot maximum qui minimise le coût
- Beaucoup de similarités mais aussi des différences avec le problème du flot maximum

Graphe orienté valué $G = (S, A, c, w)$

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



Capacité $c : S \times S \to \mathbf{Z}$ avec $c(p, q) \geq 0$ si $(p,q) \in A$
et $c(p, q) = 0$ si $(p, q) \notin A$

Flot $f : S \times S \to \mathbf{Z}$

Coût $w : S \times S \to \mathbf{Z}$ avec $c(p, q)$ *arbitraire* si $(p,q) \in A$
et $c(p, q) = 0$ si $(p, q) \notin A$

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 $p, q \in S$ $f(p, q) \leq c(p, q)$

Anti-symétrie

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

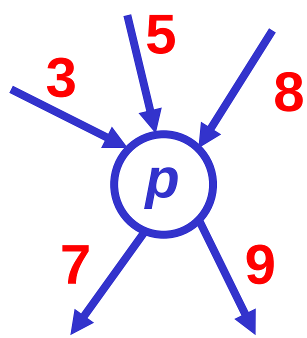
$w(q,p) = -w(p,q)$

$$p \xrightarrow{\mathbf{17,\ 5}} q \qquad\qquad p \xleftarrow{\mathbf{-17,\ -5}} q$$

flot direct — flot prélevable

Conservation du flot

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



- Matrice d'adjacence +
  matrice des capacités +
  matrice des coûts+
  matrice des flots

- Listes des successeurs
  avec capacités, coûts et flots

- Graphique

- Réseaux de transport avec des coûts
- Le problème
- Approches
  - Plus courts chemins de coût minimum
  - Circuits de coût négatif

Graphe orienté valué $G = (S, A, c, w)$ avec source s et puits t

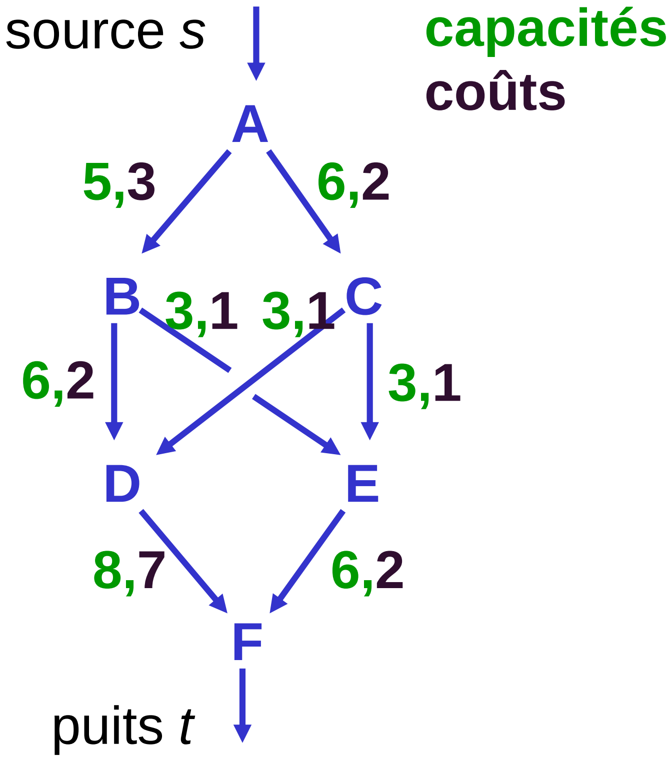
Coût d'un flot f :

$$w(f)=\sum(f(p,q)*w(p,q) \mid f(p,q)>0)$$

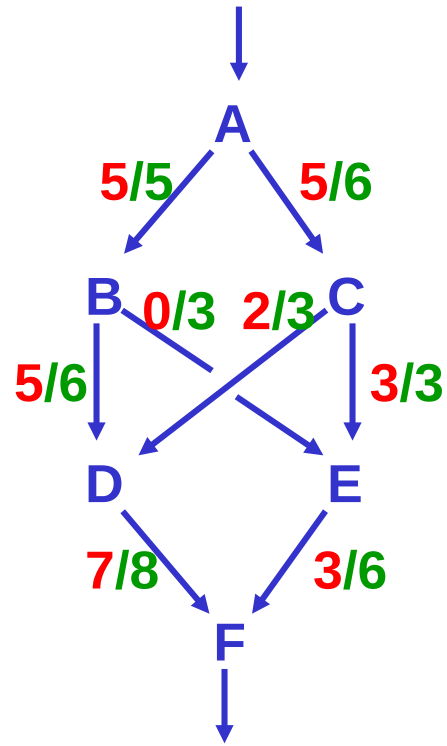
Problème

Calculer un flot maximum de coût minimum.

(c'est-à-dire un flot *f* dont la valeur est maximum et qui a, en plus, le coût minimum parmi les flots de valeur maximum)



un flot maximum de coût minimum ?
| f | = 10
Coût= 95



Plusieurs flots max. possibles
Celui de coût minimum ?

- Réseaux de transport avec des coûts
- Le problème
- Approches
  - Plus courts chemins de coût minimum
  - Circuits de coût négatif

- Par les plus courts chemins du point de vue du coût (= les plus économiques)
  - Comme Ford-Fulkerson
  - En augmentant le flot par le chemin le plus économique, à chaque fois

- En corrigeant le flot maximum
  - Trouver un flot maximum (Ford-Fulkerson ou méthode de préflots ou autre)
  - Le corriger, en lui faisant emprunter des bouts de chemins plus économiques

- Réseaux de transport avec des coûts
- Le problème
- Approches
  - Plus courts chemins de coût minimum
  - Circuits de coût négatif

- Utiliser l'algorithme de Ford-Fulkerson
- A chaque étape, dans le réseau résiduel calculer le chemin de coût minimum de s à t

  → Algorithme de Bellman-Ford car les coûts peuvent être négatifs
- Utiliser ce chemin pour augmenter le flot
- Quand il n'y a plus de chemin, on a fini

- Mise en garde. Cette approche ne fonctionne que si le réseau résiduel ne contient pas de circuits de coût négatif.

- Soient : B la capacité maximum (entière !) sur un arc sortant de s
  D le nombre d'arcs sortant de s ($D \leq n$)

- Alors
  - Nombre d'étapes : au plus BD, càd en $O(Bn)$, car – au pire - sur chaque arc le flot sera augmenté d'1 à chaque étape
  - Donc autant d'appels à Bellman-Ford, qui est en $O(nm)$
  - Complexité totale : $O(Bn^2m)$

Essayer une approche comme l'algorithme de Johnson

- Utiliser l'algorithme de Bellman-Ford une fois pour calculer le chemin de coût minimum de s à tout sommet *u*. Soit *p(u)* ce coût.
- Repondérer les coûts :

$$w'(u, v) \leftarrow w(u, v) + p(u) - p(v)$$

(Ils seront positifs car $p(v) < p(u) + w(u, v)$ par l'inégalité triangulaire)

- Utiliser Ford-Fulkerson
  - A chaque étape, dans le réseau résiduel calculer le chemin de coût minimum de s à t, à l'aide des coûts *w'*

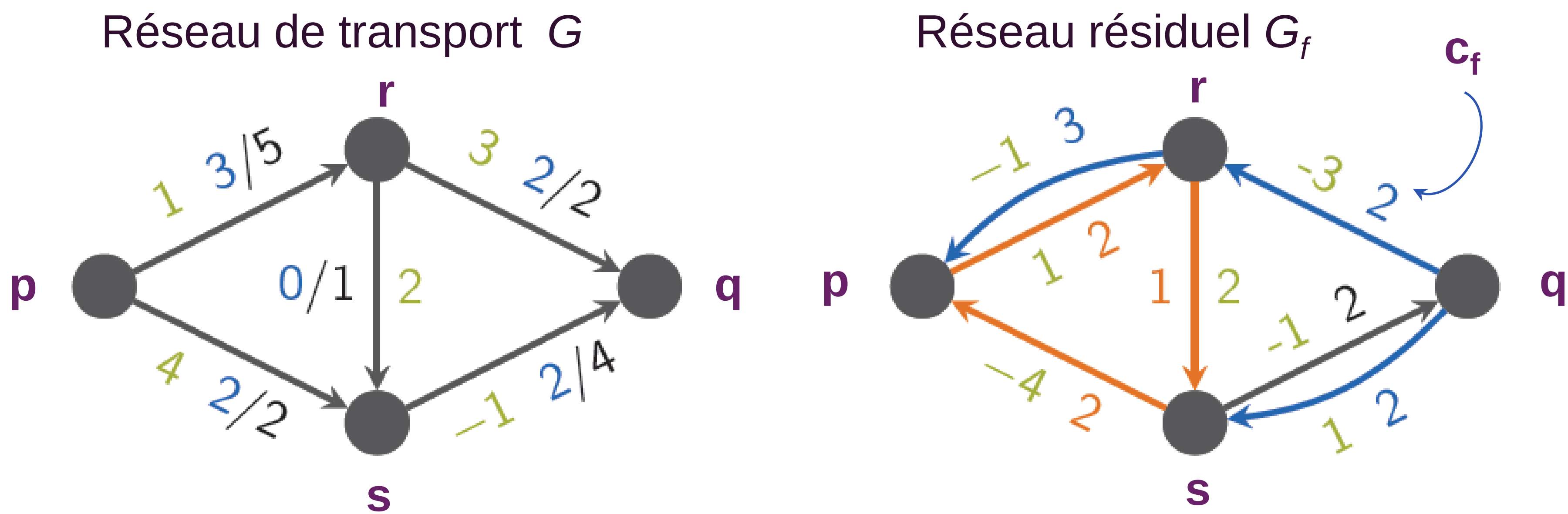
    → Algorithme de Dijkstra car les poids sont positifs
  - Utiliser ce chemin pour augmenter le flot

- Mise à jour de *p*(*u*) et des *w'(u,v)* à chaque étape :
  - Lorsqu'on applique Dijkstra, on a les chemins de coût minimum de s vers tous les u
  - On appelle ces coûts *p(u)* (pas evident, mais ça marche !)
  - On met à jour tous les arcs en O(m) (mais Dijkstra est déjà en $O(n^2)$, donc pas de surcoût)

- Complexité :
  - Même nombre d'étapes maximum O(Bn)
  - Mais complexité par étape $O(n^2)$
  - Total : $O(Bn^3)$ (dépend quand-même fortement de B)

- Réseaux de transport avec des coûts
- Le problème
- Approches
  - Plus courts chemins de coût minimum
  - Circuits de coût négatif

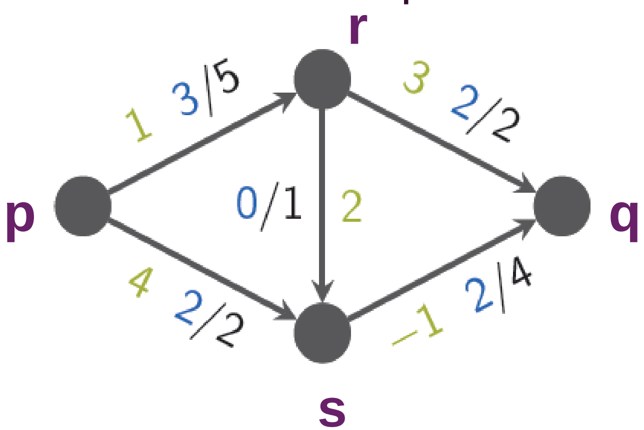
- Soit *A_f ={(p,q)∈SxS / c_f(p,q) > 0}* et soit *G_f=(S,A_f)* le réseau résiduel de G (incluant les coûts)



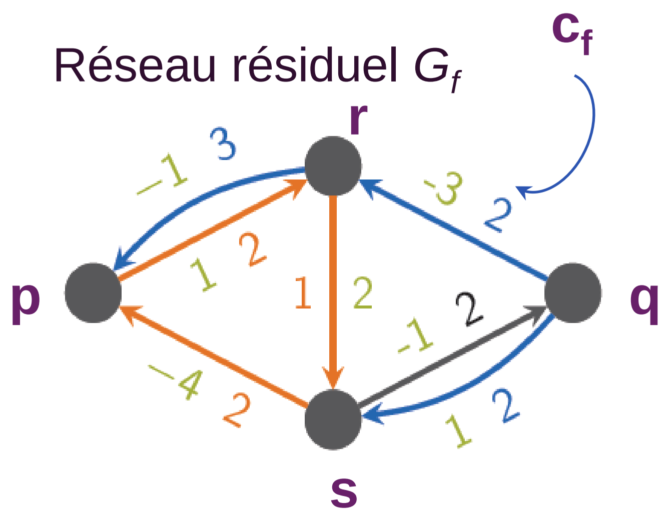
coût
flot/capacité

Note. Il s'agit ici d'une partie seulement d'un réseau de transport.

Réseau de transport G



Réseau résiduel G_f



coût
flot/capacité

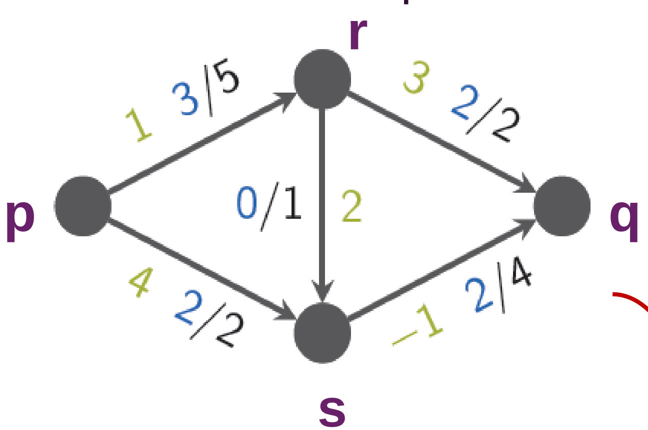
Circuit $C : p \rightarrow r \rightarrow s$ de coût total négatif

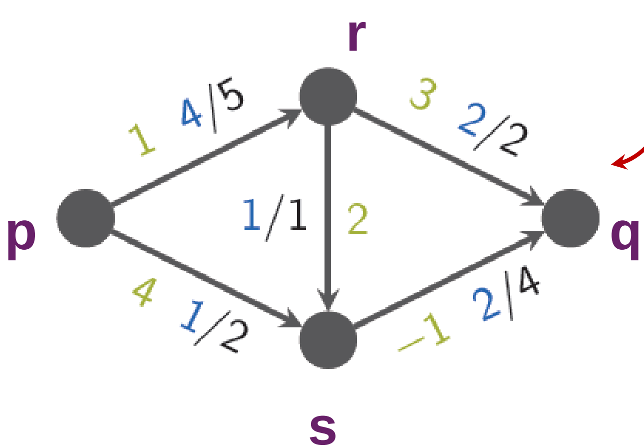
Faire circuler du flot dans ce circuit permet de baisser le coût total du flot.

Combien de flot ?

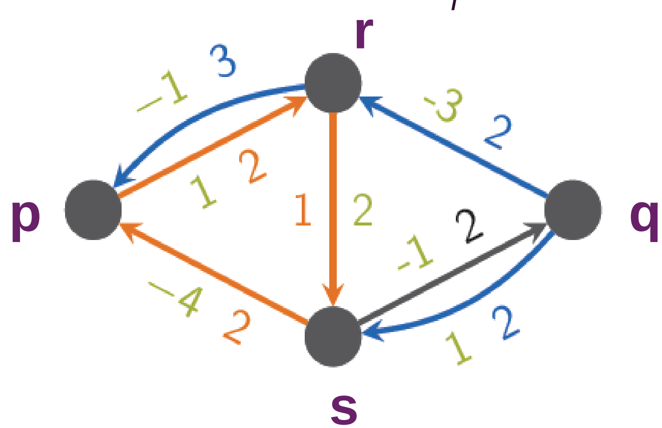
$$C_f(C) = Min\{c_f(a) |\ a \text{ arc du circuit}\}$$

Réseau de transport *G*





Réseau résiduel G_f



$C_f(C) = Min\{c_f(a) \mid a$ arc du circuit$\} = 1$

Coût initial du flot :

3x1+0x2+2x4 + 2x3+2x(-1) = 15

Nouveau coût du flot :

4x1+1x2+1x4+2x3+2x(-1) = 14

Idée générale

1 Calculer un flot maximum f de s à t

2 **tant qu**'il existe un circuit dans G_f de coût total négatif

faire augmenter le flot f sur ce circuit ;

3 retourner le flot f

Théorème

1. Si C est un circuit de coût négatif qui satisfait $c_f(C)>0$, alors la fonction f' définie par

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

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

   est un flot de G *de valeur* $|f'|=|f|$ et de *coût* $w(f')=w(f)+c_f(C)*\sum(w(a)|a \text{ arc de } C)$.

2. Soit $A_f =\{(p,q)\in SxS \mid c_f(p,q) > 0\}$ et soit $G_f=(S,A_f)$ le réseau résiduel de G. S'il n'existe aucun circuit de coût négatif dans G_f alors le flot est de coût minimum.

Algorithme Klein ($G=(S,A,c,w)$, s, t)

Début

Calculer un flot maximum f de s à t

Tant qu'il existe dans G_f un circuit C de coût total négatif faire

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

pour chaque arc (p,q) de C faire

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

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

Fin

- Trouver un circuit de coût négatif :

  - Algorithme de Bellman-Ford indique si (oui/non) il existe un circuit négatif

  - Pour récupérer le circuit : modifier l'algorithme de Bellman-Ford afin de garder le prédécesseur de chaque sommet u sur le plus court chemin de s à u. Complexité : O(mn)

Complexité de l'algorithme de Klein :

- Le coût du flot peut ne baisser que d'1 à chaque passage dans la boucle

- Coût initial au pire du flot : O(mWB), où W=coût maximum d'un arc, en valeur absolue, et B la capacité maximum d'un arc

- Donc O(mWB) étapes au pire, chacune à O(mn)

- Total : $O(m^2nWB)$ en plus du calcul du flot max.

Circuit de coût moyen minimum : circuit C qui minimise la fonction

$$\frac{\sum_{pq \text{ arc de C}} w(p,q)}{|C|}$$

Algorithme de Goldberg-Tarjan (*G=(S,A,c,w), s, t)*

Début

Calculer un flot maximum *f* de *s* à *t*

Tant qu'il existe dans G_f un circuit de coût total négatif faire

soit C un circuit de coût moyen minimum

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

pour chaque arc *(p,q)* de *C* faire

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

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

Fin

Complexité : $O(nm^2\log^2 n)$

(difficile à établir)

Ne dépend plus des capacités, ni des coûts !

- Jeff Erikson - Lecture 25: extensions of maximum flow, http://www.cs.uiuc.edu/~jeffe/teaching/algorithms
- J. Matuschke – Network Flows, 2016,
  http://www-m9.ma.tum.de/SS2016/NetworkFlows
- J. Chiu, R. Kumar – Problem Solving in Computer Science – Min Cost Flow, University of British Columbia. https://www.ugrad.cs.ubc.ca/~cs490/2016W2/lectures/cpsc-490-lecture-26.pdf