Recherche Opérationnelle

Chapitre 3 : Flot maximal dans un graphe

J.-F. Scheid
Institute Elie Cartan de Lorraine/Télécom Nancy
Université de Lorraine

2026-27 (version du 3/11/2026)

Table des matières

1. Graphes
2. Flot dans un graphe
   - Introduction
   - Flot
3. Problème de flot maximal dans un graphe
   - Introduction
   - Coupe minimale dans un graphe
4. Détermination du flot maximal
   - Chaîne améliorante
   - Algorithme de Ford-Fulkerson
   - Parcours de graphe
   - Un exemple
   - Cas des capacités inférieures non-nulles

I. Graphes

De nombreux phénomènes/situations peuvent être modélisés par des graphes. Quelques exemples :

- réseau routier : les sommets sont les intersections des routes, les arêtes représentent les routes. Problème du voyageur de commerce (TSP).
- cheminement dans un réseau informatique.
- web modélisé par un graphe. Les sommets sont les pages Web et les arêtes sont les liens hypertexte entre ces différentes pages.

Définition 1.

Un graphe *orienté* $G = (V, E)$ est constitué d'un ensemble fini V de **sommets** (*vertex*) et d'un ensemble fini E de couples ordonnés (i, j) avec $i, j \in V$. Les éléments de E sont appelés les **arêtes** (*edge*) du graphe.

Notation (les flèches indiquent l'orientation des arêtes) :

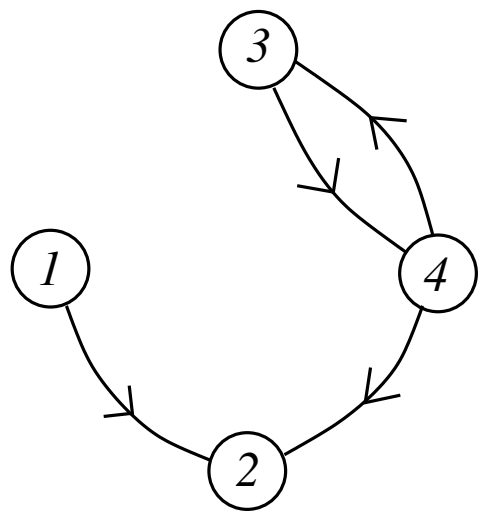


Remarques :

- Pour un graphe **orienté**, les arêtes (i, j) et (j, i) sont distinctes.
- Pour un graphe **non-orienté**, ces arêtes sont confondues et sont représentées par un seul arc sans *flèche* ; l'arête est notée $\{i, j\}$.
- Dans un graphe orienté, un sommet i peut avoir une arête sur lui-même, i.e. une arête (i, i) ce qu'on appelle une *boucle*.
  Un graphe orienté sans boucle est dit *simple*.

Exemple de graphe orienté :

$V = \{1, 2, 3, 4\}$; $E = \{(1, 2), (3, 4), (4, 2), (4, 3)\}$



Définition 2.

L'ensemble $G = (V, E, c)$ est un **graphe valué** si (V, E) est un graphe auquel on associe une fonction positive $c : E \to \mathbb{R}^+$ appelée **valuation** ou **capacité**. La capacité de l'arête (i, j) est notée c_{ij}.

Exemple : La capacité c_{ij} représente par exemple :

- la distance ou longueur du tronçon de route (i, j) entre deux villes i et j (cf. TSP)
- le nombre maximal de voitures par unité de temps entre deux villes i et j
- le débit maximal entre deux points i et j d'approvisionnement en eau/gaz.
- ...

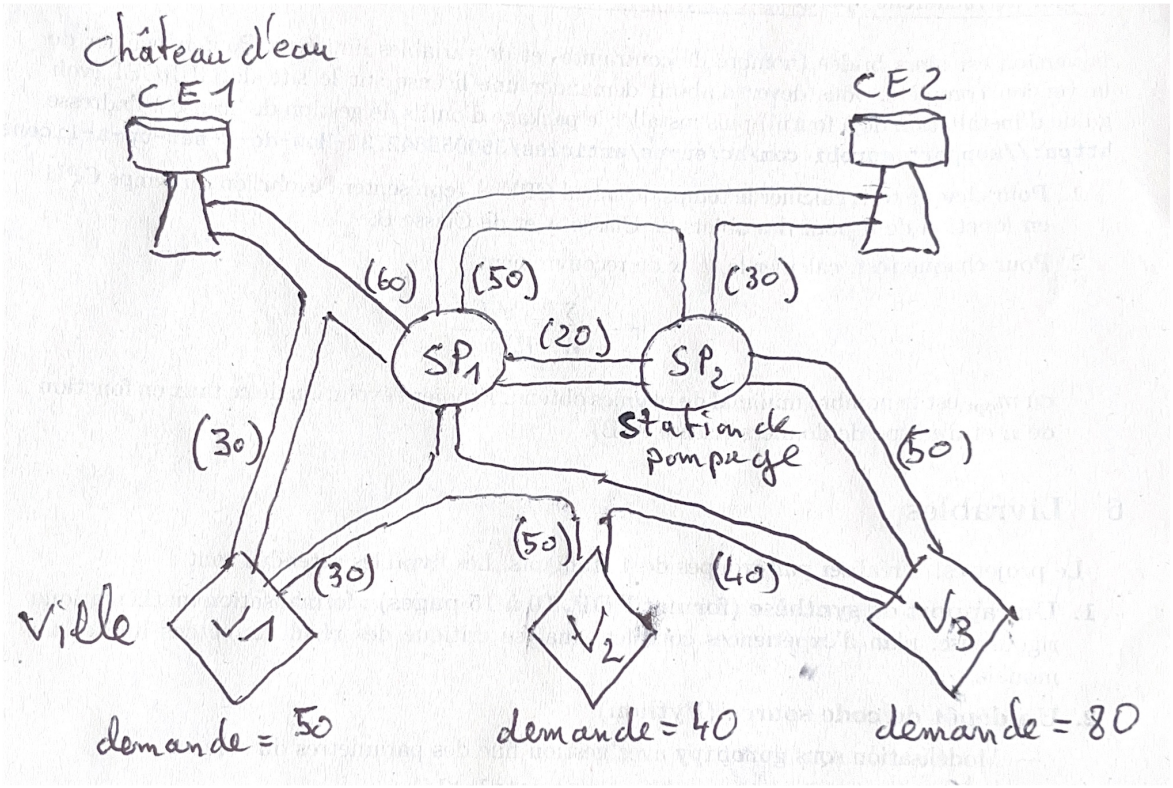
II. Flot dans un graphe

1. Exemple d'introduction

Deux châteaux d'eau alimentent 3 villes à travers un réseau de canalisations comportant également 2 stations de pompage.

- Les châteaux d'eau ont une capacité limitée qui s'élève pour chacun d'eux à $100\,000\,m^3$.
- Les villes ont exprimé les demandes suivantes :
  - au minimum $50\,000\,m^3$ pour la ville 1
  - au minimum $40\,000\,m^3$ pour la ville 2
  - au minimum $80\,000\,m^3$ pour la ville 3
- Les canalisations entre les châteaux d'eau et les villes ont des débits limités. Par exemple :
  - le débit maximal reliant le chateau 1 à la ville 1 est de $30\,000\,m^3$.
  - le débit maximal reliant la station de pompage 1 à la ville 2 est de $50\,000\,m^3$.

Le réseau est représenté schématiquement par le graphique suivant.



Les débits maximaux sont indiqués entre parenthèses le long des canalisations (en milliers de m^3).

☛ Déterminer s'il est possible de satisfaire la demande des 3 villes à travers ce réseau et comment ? Déterminer le flot maximal.

On va modéliser le réseau par un graphe dont les sommets sont les châteaux d'eau, les stations de pompages et les villes. Les arêtes représentent les canalisations. Le graphe est valué par les débits maximaux des canalisations. Pour résoudre le problème, on introduit la notion de flot dans un graphe.

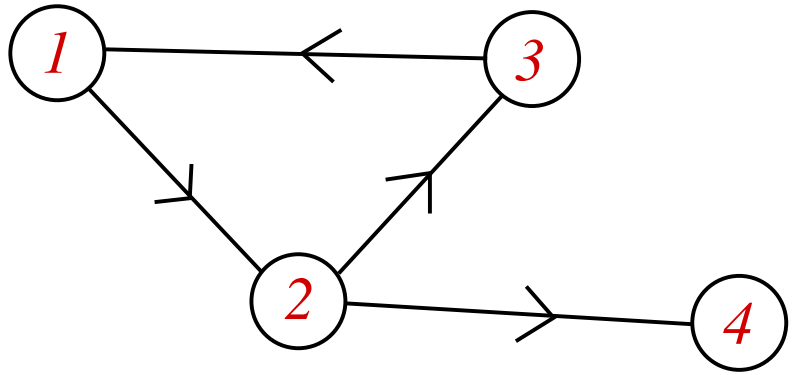
2. Flot dans un graphe.

On considère un graphe orienté et valué $G = (V, E, c)$ où V est l'ensemble de sommets, E l'ensemble des arêtes et c les capacités supérieures associées aux arêtes.

On note $u = (i, j) \in E$ l'arête qui a le sommet i de départ et le sommet j d'arrivée (on tient compte de l'orientation). On désigne par :

- $S(i)$ l'ensemble des sommets successeurs de i (les sommets j tel que l'arête (i, j) existe)
- $P(i)$ l'ensemble des sommets prédécesseurs de i (les sommets j tel que l'arête (j, i) existe).

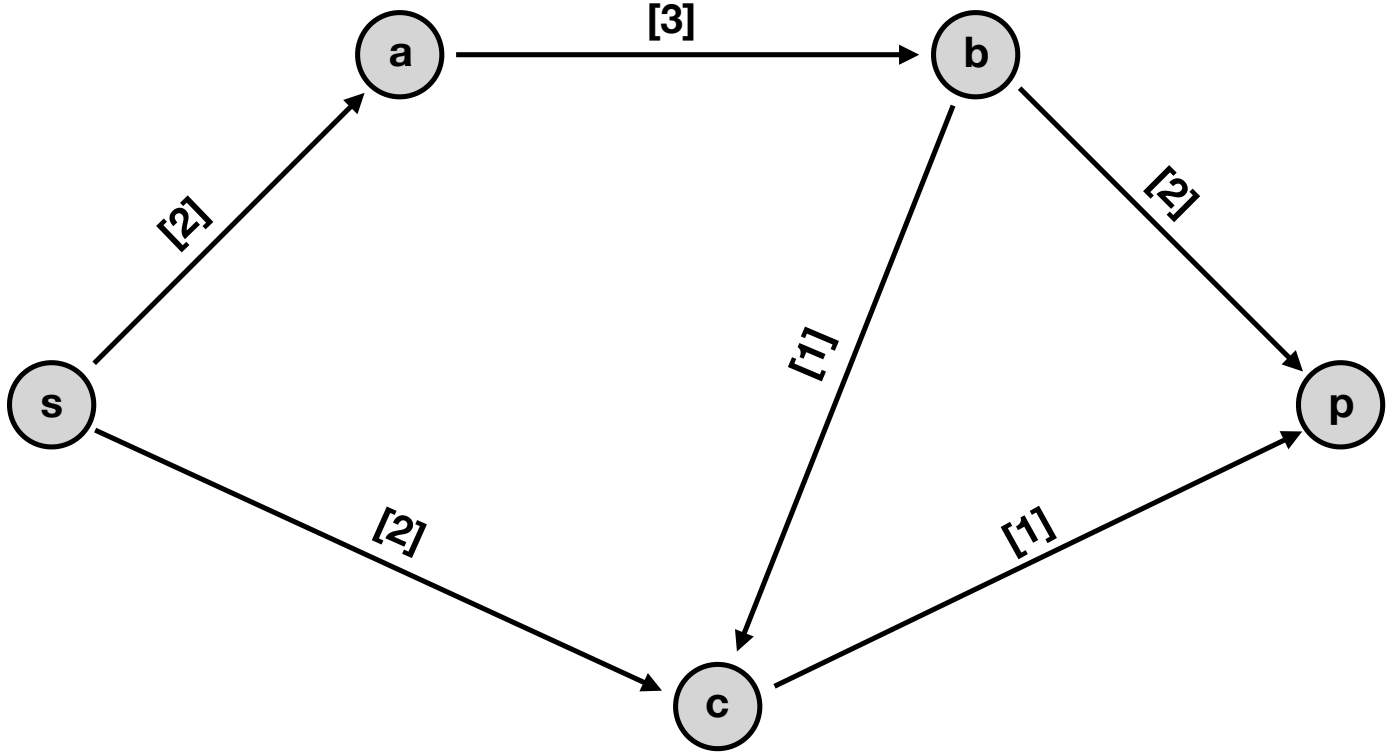
Exemple : $V = \{1, 2, 3, 4\}$; $E = \{(1, 2), (2, 3), (3, 1), (2, 4)\}$



sommet	successeur S	prédécesseur P
1	2	3
2	3, 4	1
3	1	2
4	–	2

On suppose aussi que le graphe G comporte deux sommets particuliers :

- un sommet *source* s : le seul sommet du graphe qui n'a pas de prédécesseur.
- un sommet *puits* p : le seul sommet du graphe qui n'a pas de successeur.



Les valeurs indiquées sur le graphe entre "[.]" représentent les capacités associées aux arêtes. Ainsi, $c_{sa} = 2$, $c_{sc} = 2$,$c_{ab} = 3$,...

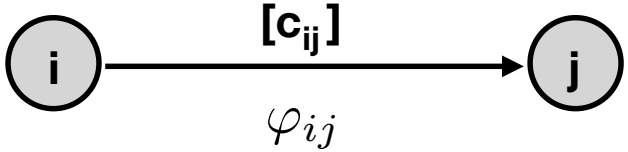
Définition d'un flot dans un graphe

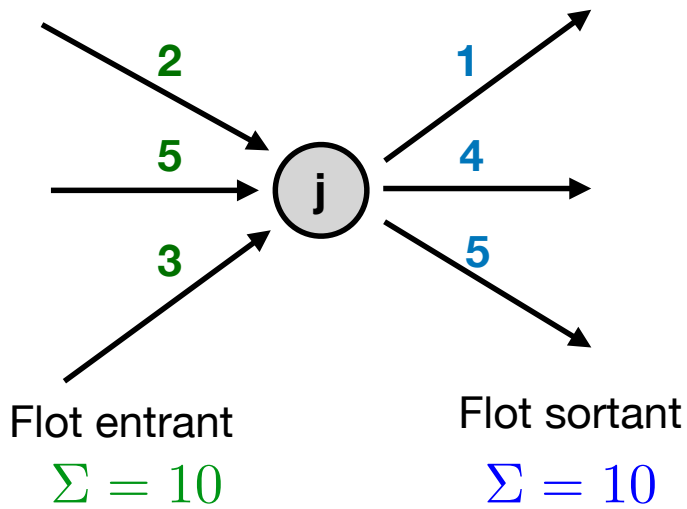
- Un **flot** de s à p est une fonction $\varphi : E \to \mathbb{R}$ définie sur les arêtes tq

$$\sum_{i \in P(j)} \varphi_{ij} = \sum_{k \in S(j)} \varphi_{jk} \quad \text{où} \quad \varphi_{ij} \overset{def}{=} \varphi(i,j)$$

pour tout sommet $j \neq s, p$. On dit qu'il y a conservation du flux au sommet j ("ce qui rentre égale ce qui sort").

La valeur φ_{ij} est le **flot sur l'arête** (i,j).





Définition (suite)

- Le flot est dit **réalisable** si pour toute arête $(i, j) \in E$, on a

$$0 \leq \varphi_{ij} \leq c_{ij} \tag{1}$$

- La quantité $v = \sum_{i \in P(p)} \varphi_{ip}$ est la **valeur du flot** de s à p.

Remarque. Les contraintes de capacités (1) se généralisent à :

$$d_{ij} \leq \varphi_{ij} \leq c_{ij} \tag{2}$$

i.e. avec des capacités inférieures non nécessairement nulles.

III. Problème de flot maximal dans un graphe

1. Introduction

Des châteaux d'eau alimentent des villes (les sommets) à travers un réseau de canalisation (les arêtes). Les canalisations ont des débits limités (les capacités). On cherche le débit maximal à faire passer dans le réseau.

Problème de flot maximal

Etant donné un graphe valué (avec une seule source et un seul puits), trouver un flot réalisable maximal (i.e. dont la valeur est la plus grande possible).

2. Coupe minimale dans un graphe

Définition d'une coupe

Une **coupe** d'un graphe valué $G = (V, E, c)$ avec une seule source s et un seul puits p, est une partition des sommets notée $(X, \overline{X})$ telle que :

- $V = X \cup \overline{X}, \quad X \cap \overline{X} = \emptyset$
- $s \in X$ et $p \in \overline{X}$

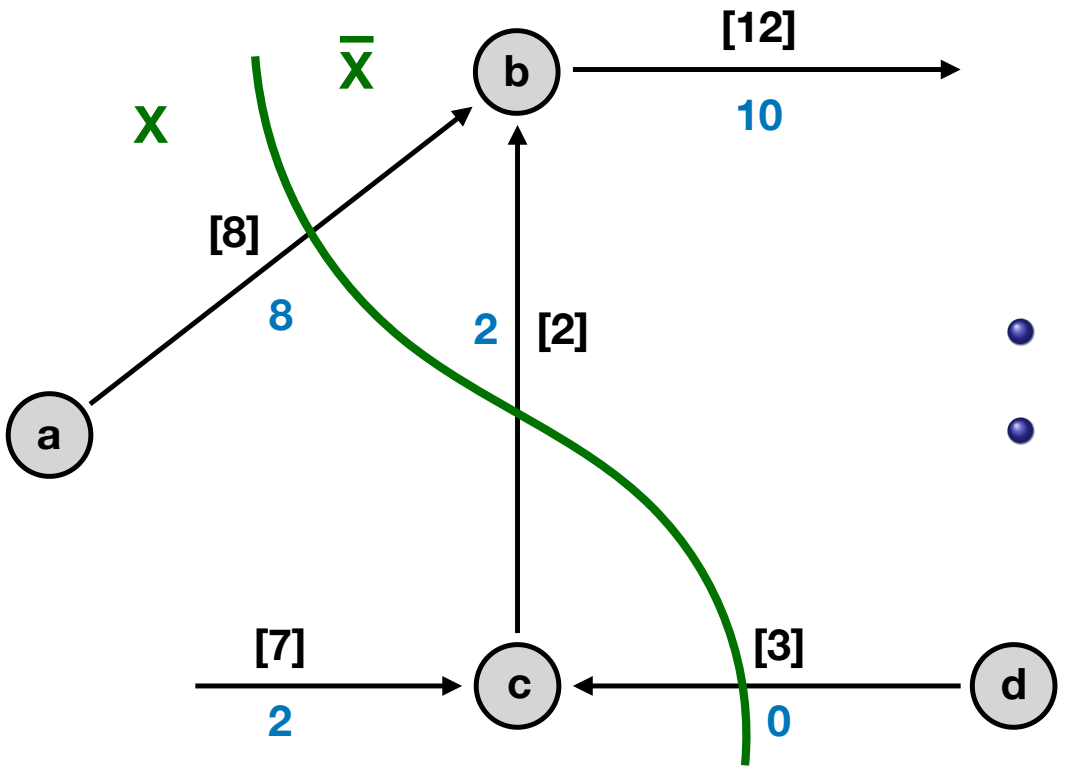
La **capacité de la coupe** est définie par $c(X, \overline{X}) = \sum_{\substack{i \in X \\ j \in \overline{X}}} c_{ij}$

La capacité d'une coupe est donc la somme des capacités des arêtes dont un sommet est dans X et l'autre dans $\overline{X}$.

Définition d'une coupe minimale

Une coupe $(X, \overline{X})$ est dite **minimale** pour le flot φ si :

- toute arête de X vers $\overline{X}$ est *saturée* c-à-d $\varphi_{ij} = c_{ij}$ pour $i \in X$ et $j \in \overline{X}$.
- toute arête de $\overline{X}$ vers X est *insaturée* c-à-d $\varphi_{ij} = 0$ pour $i \in \overline{X}$ et $j \in X$.



- arêtes (a, b) et (c, b) **saturées**
- arête (d, c) **insaturée**

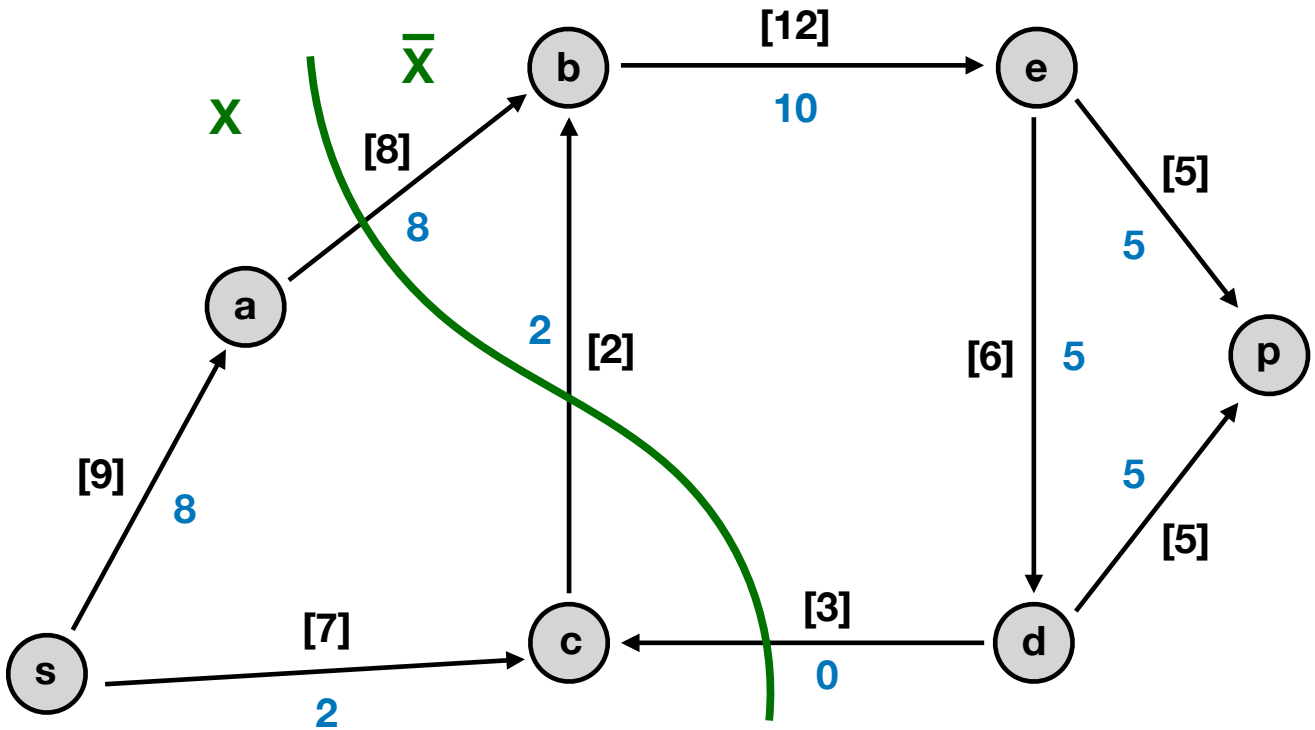
Quel lien entre flot max et coupe minimale ?

Proposition

S'il existe une coupe minimale pour un flot φ réalisable, alors ce flot est maximal.

On peut montrer qu'on a tjs $v(\varphi) \leq c(X, \overline{X})$ pour toute coupe $(X, \overline{X})$.

Coupe minimale / flot maximal



$\Rightarrow$ flot maximal $v(\varphi) = c(X, \overline{X}) = 10$

IV. Détermination du flot maximal (Ford-Fulkerson)

On cherche un flot maximal.

1. Chaîne améliorante

Comme on va le voir, c'est la notion de *chaîne* et non pas de *chemin* qui est essentielle.

Définition d'une chaîne.

Une chaîne d'un graphe est une suite de sommets

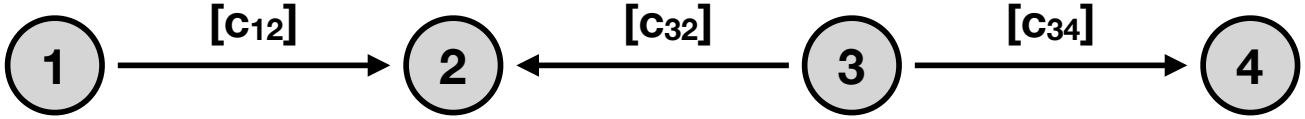
$$\mathcal{C} = (i_1, i_2, \cdots, i_k, i_{k+1}, \cdots, i_m)$$

reliés entre eux par des arêtes c'est-à-dire tels que :

$$(i_k, i_{k+1}) \in E \quad \text{ou} \quad (i_{k+1}, i_k) \in E$$

(arête **directe**) (arête **inverse**)

Une chaîne ne tient pas compte de l'orientation des arêtes reliant les sommets (chaîne $\neq$ chemin).

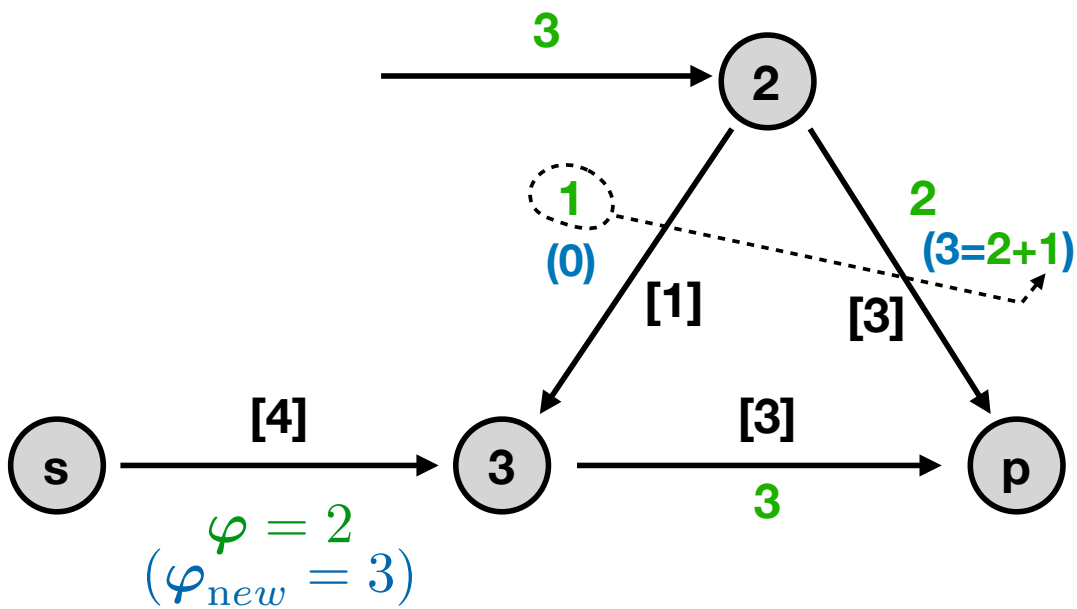


Définition d'une chaîne améliorante.

Une chaîne $\mathcal{C} = (s, i_1, i_2, \cdots, i_k, i_{k+1}, \cdots, i_m, p)$ est dite **améliorante** pour un flot réalisable φ donné si :

1. $\varphi(i_k, i_{k+1}) < c(i_k, i_{k+1})$ si $(i_k, i_{k+1}) \in E$: **arête directe non saturée**
2. $\varphi(i_{k+1}, i_k) > 0$ si $(i_{k+1}, i_k) \in E$: **arête inverse non insaturée**

Remarque : ce qui compte ici, ce sont les inégalites strictes. Pour une arête directe, on cherche à ajouter du flot. Pour une arête inverse, on cherche à *enlever* du flot pour le mettre ailleurs (par conservation) et ce faisant permettre d'augmenter globalement le flot.



L'algorithme de Ford-Fulkerson permet de trouver un flot maximal par recherche de chaînes améliorantes. Il est basé sur le résultat suivant :

Proposition

Un flot réalisable est maximal si et seulement s'il n'existe pas de chaîne améliorante.

On recherche de façon itérative une chaîne améliorante en parcourant le graphe de s à p selon un parcours défini (DFS, BFS,...) et en utilisant une procédure de marquage des sommets.

Une fois qu'une chaîne améliorante est trouvée, on met à jour les flots.

On s'arrête lorsqu'il n'y a plus de chaîne améliorante de s à p. On a alors trouvé le flot maximal.

2. Algorithme de Ford-Fulkerson

- Initialisation par un flot initial réalisable ($\varphi = 0$)
- Tant que le flot n'est pas maximal
  - Marquage de la source s
  - Tant qu'on marque des sommets

    Pour tout sommet marqué i

    Marquer les sommets j non marqués tq
    $\varphi(i,j) < c(i,j)$ pour une arête directe
    ou
    $\varphi(j,i) > 0$ pour une arête inverse

    Fin pour

    Fin Tant que
  - Si le puits p n'est pas marqué alors le flot est maximal
    Sinon amélioration du flot

  Fin Tant que

Amélioration du flot

- trouver une chaîne qui a permis de marquer p et calculer l'amélioration $\varepsilon = \min(\varepsilon_1, \varepsilon_2) > 0$ avec

$$\varepsilon_1 = \min\{c(i_k, i_{k+1}) - \varphi(i_k, i_{k+1}) \text{ avec } (i_k, i_{k+1}) \in E \text{ (arête directe)}\}$$

$$\varepsilon_2 = \min\{\varphi(i_{k+1}, i_k) \text{ avec } (i_{k+1}, i_k) \in E \text{ (arête inverse)}\}$$

- déterminer le nouveau flot φ' :
  - Si $(i_k, i_{k+1}) \in E$ (arête directe) alors $\varphi'(i_k, i_{k+1}) = \varphi(i_k, i_{k+1}) + \varepsilon$
  - Si $(i_{k+1}, i_k) \in E$ (arête inverse) alors $\varphi'(i_k, i_{k+1}) = \varphi(i_k, i_{k+1}) - \varepsilon$

3. Parcours de graphe

- **Parcours profondeur** (**D**eep **F**irst **S**earch)

  Exploration en profondeur des chemins : pour chaque sommet, on prend et on marque le premier sommet successeur jusqu'à ce qu'un sommet n'ait plus de successeur ou bien que tous ses successeurs soient déjà marqués.

  On utilise généralement **une pile**[1] pour l'exploration des sommets

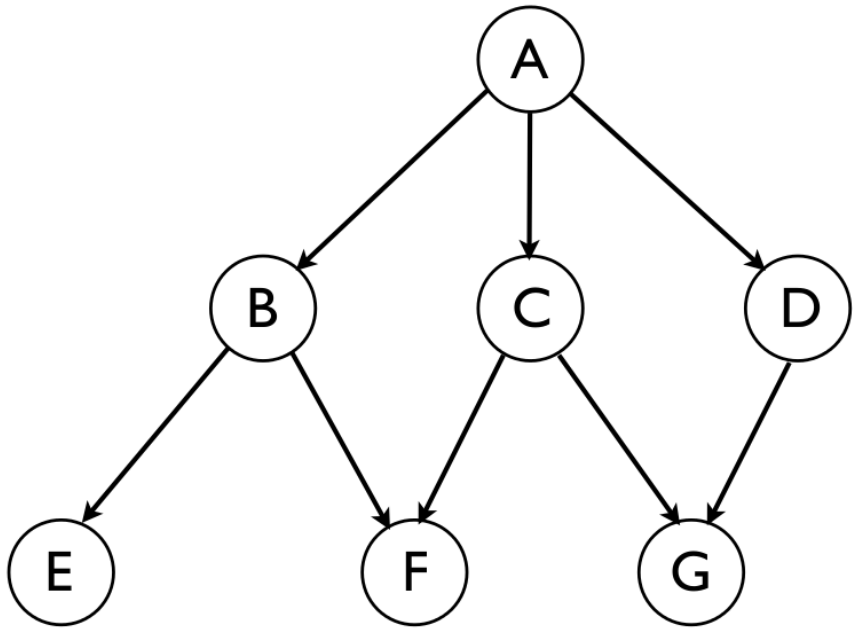
- **Parcours largeur** (**B**reath **F**irst **S**earch)

  A partir d'un sommet, on liste et on marque tous les sommets successeurs non encore marqués, jusqu'à ce qu'un sommet n'ait plus de successeur ou bien que tous ses successeurs soient déjà marqués.

  On utilise généralement **une file**[2] pour l'exploration des sommets.

1. Une pile est une structure de données basée sur le principe « dernier arrivé, premier sorti » (en anglais LIFO « last in, first out ») : le dernier élément ajouté à la pile est le premier à en sortir.
2. Une file est une structure de données basé sur le principe « premier entré, premier sorti » (en anglais FIFO « first in, first out ») : les premiers éléments ajoutés à la file seront les premiers à en être retirés.

Exemple de parcours DFS / BFS



Parcours profondeur DFS : A,B,E,F,C,G,D

Parcours largeur BFS : A,B,C,D,E,F,G

4. Un exemple

Considérations pratiques :

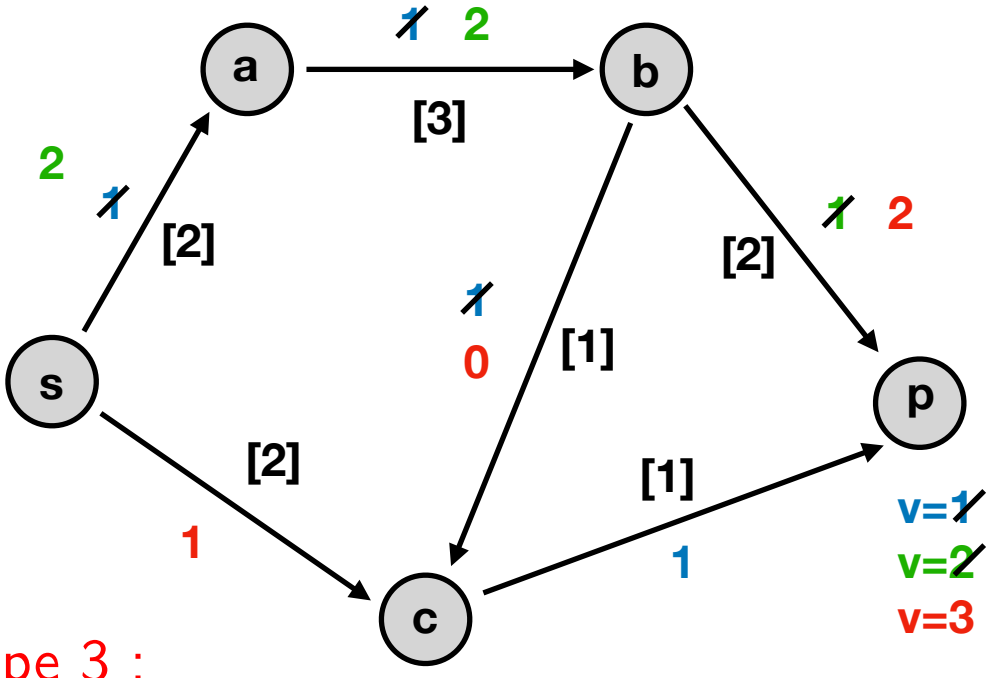
Dans la pratique, on utilise plusieurs tableaux

- $\mathbb{S}=$ ensemble des sommets marqués et non complètement examinés.
- tableau `origine` qui indique l'origine du marquage :

  arête directe $(i_k, i_{k+1}) \in E \rightarrow \texttt{origine}\ (i_{k+1}) = i_k$

  arête inverse $(i_{k+1}, i_k) \in E \rightarrow \texttt{origine}\ (i_k) = -i_{k+1}$
- tableau ε : amélioration globale possible du flot à chaque marquage.

Marquage profondeur (DFS avec pile)



Etape 1 :

$\mathbb{S}$	s	a	b	c	p
orig	$-$	s	a	b	c
ε	∞	2	2	1	$\varepsilon = 1$

Etape 2 :

$\mathbb{S}$	s	a	b	p
orig	$-$	s	a	b
ε	∞	1	1	$\varepsilon = 1$

Etape 3 :

$\mathbb{S}$	s	c	b	a	p
orig	$-$	s	$-c$	$-b$	b
ε	∞	2	1	1	$\varepsilon = 1$

on dépile ↑

Etape 4 :

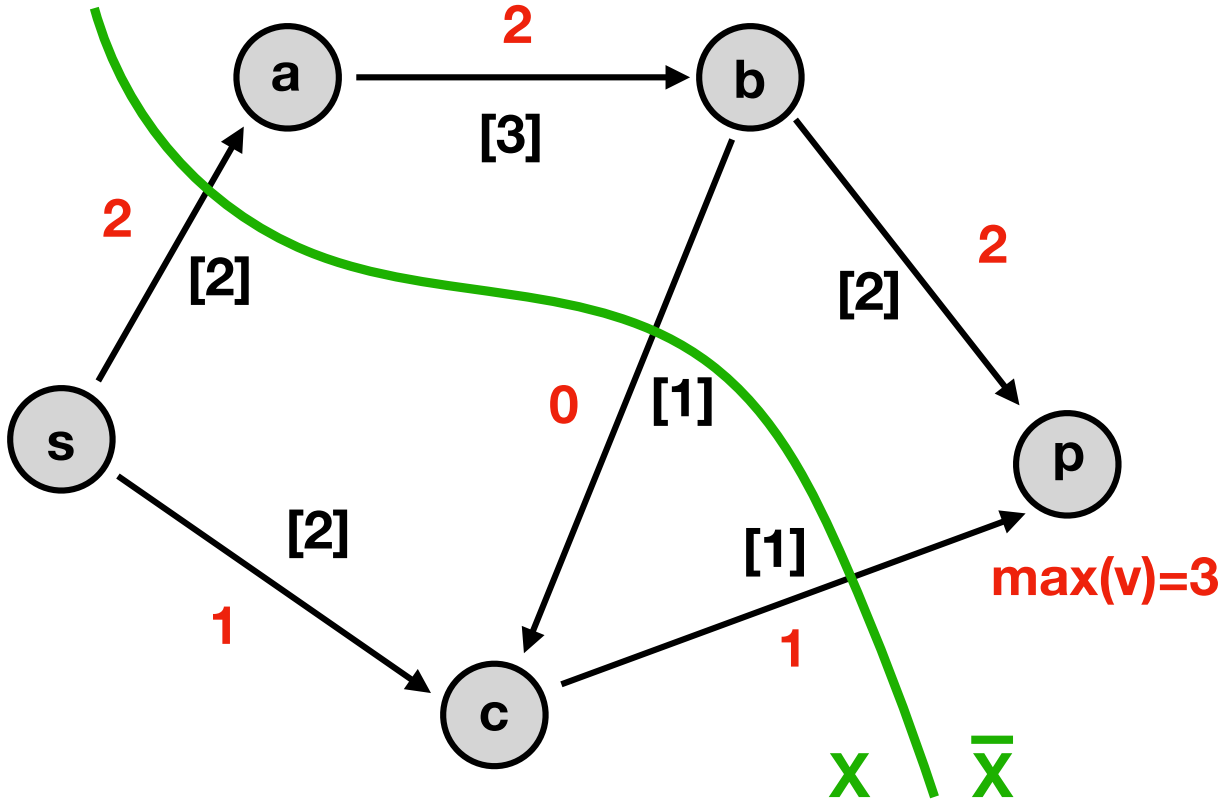
$\mathbb{S}$	s	c
orig	$-$	s
ε	∞	1

on dépile ↑ ↑

$\Rightarrow$ pile vide ($\mathbb{S} = \emptyset$).

Le puits p n'est pas marqué $\Rightarrow$ pas de chaîne améliorante $\Rightarrow$ flot maximal

Coupe minimale : $(X, \overline{X})$ avec $X = \{s, c\}$ et $\overline{X} = \{a, b, p\}$.
X est formé des sommets marqués à la dernière étape (pile vide).



5. Cas des capacités inférieures non nulles.

De façon plus générale, on peut considérer des capacités inférieures non nulles avec les contraintes de bornes sur les flots :

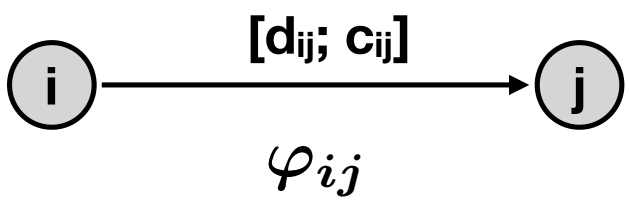
$$d_{ij} \leq \varphi_{ij} \leq c_{ij}$$

☛ Quelles modifications apporter à l'algo. de Ford-Fulkerson ?

Presque rien ! On est toujours à la recherche d'une chaîne améliorante. Ce qui change c'est le critère portant sur une *arête inverse*, le critère sur une arête directe restant inchangé.

Condition chaîne améliorante/amélioration :

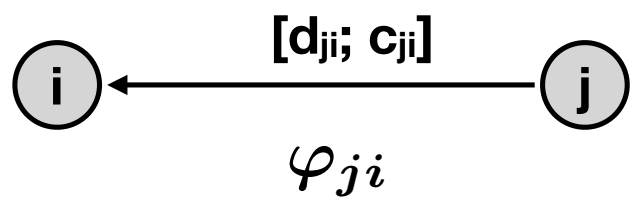
arête directe



$\epsilon_{ij} := c_{ij} - \varphi_{ij} > 0$

(inchangé)

arête inverse



$\epsilon_{ji} := \varphi_{ji} - d_{ji} > 0$

Remarques.

- Avec un parcours largeur BFS, l'algo. porte le nom d'Edmonds-Karp (1972). Sa complexité est en $\mathcal{O}(|V||E|^2)$. Généralement meilleur qu'un parcours DFS.
- Un autre algo. dû à Dinic, découvert en parallèle, possède une complexité en $\mathcal{O}(|V|^2|E|)$. Généralement plus performant qu'Edmonds-Karp (du fait aussi qu'on peut utiliser une structure de données particulière et efficace, les *arbres dynamiques*).
- Il existe d'autres algo. encore plus efficaces, par ex. Goldberg-Tarjan (1988) en $\mathcal{O}(|V|^2\sqrt{|E|})$.