ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Optimisation dans les réseaux

Recherche Opérationnelle
GC-SIE

Le problème du flot maximal

Introduction

Données :

- Graphe $(\mathcal{N},\mathcal{A})$
- Capacités : $x_{ij} \in [b_{ij},c_{ij}]$
- Un nœud source s
- Un nœud puits t

Problème :

- Faire passer un maximum de flot de s à t.
- Maximiser la divergence de s
- Minimiser la divergence de t.

Introduction

- Plus court chemin : coûts, mais pas de capacités
- Flot maximal : capacités, mais pas de coûts
- Transbordement : coûts et capacités

Flot maximal et coupe minimale

Idée de base :

- Un flot x peut être amélioré si l'on trouve un chemin de s à t qui soit non bloqué par rapport à x.
- Envoyer un flot le long de ce chemin augmente la divergence de s sans violer les contraintes de capacité.
- Question : si on ne trouve pas un tel chemin, sommes-nous à l'optimum ?

Coupes dans un graphe

Définition :

- Une coupe Q dans un graphe $(\mathcal{N},\mathcal{A})$ est une partition de l'ensemble des nœuds $\mathcal{N}$ en deux ensembles non vides $\mathcal{S}$ et $\mathcal{N} \setminus \mathcal{S}$.
- On notera

$$Q = [\mathcal{S},\mathcal{N} \setminus \mathcal{S}]$$

- Attention : $[\mathcal{S},\mathcal{N} \setminus \mathcal{S}] \neq [\mathcal{S} \setminus \mathcal{N},\mathcal{S}]$

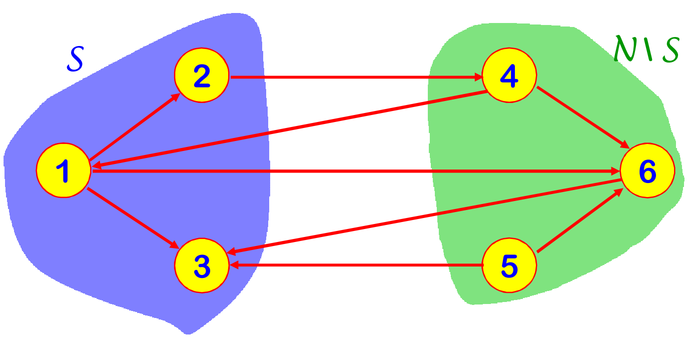
Coupes dans un graphe

- Notations :

$$Q^{+}=\{(i,j)\in \mathcal{A} \mid i\in \mathcal{S} \text{ et } j\notin \mathcal{S}\}$$
$$Q^{-}=\{(i,j)\in \mathcal{A} \mid i\notin \mathcal{S} \text{ et } j\in \mathcal{S}\}$$

- Q est non vide si $Q^{+}\cup Q^{-}\neq\emptyset$
- Q sépare s de t si $s\in\mathcal{S}$ et $t\notin\mathcal{S}$

Coupes dans un graphe



- S={1,2,3}
- Q^{+}={(2,4),(1,6)}
- Q^{-}={(4,1),(6,3),(5,3)}

Coupes dans un graphe

Définition

- Soit un vecteur de flots. Le flot F(Q) à travers une coupe non vide Q=[$\mathcal{S}$,$\mathcal{N}$ \ $\mathcal{S}$] est le flot total net sortant de $\mathcal{S}$.

$$F(Q) = \sum_{(i,j)\in Q^+} x_{ij} - \sum_{(i,j)\in Q^-} x_{ij}$$

- **On peut facilement montrer que**

$$F(Q) = \sum_{i\in\mathcal{S}} y_i$$

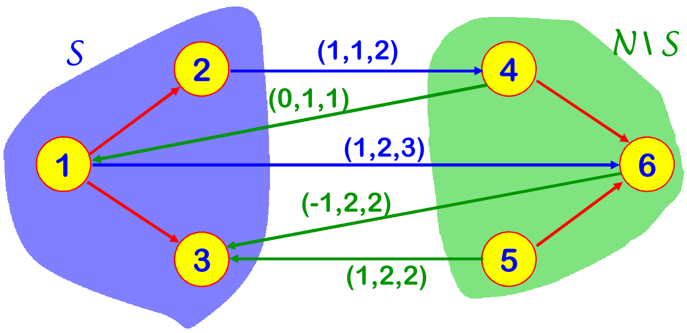
Coupes dans un graphe

Définition :

- Étant données les contraintes de capacité sur les flots, la capacité d'une coupe Q non vide est

$$C(Q) = \sum_{(i,j)\in Q^+} c_{ij} - \sum_{(i,j)\in Q^-} b_{ij}$$

Coupes dans un graphe



- Sur chaque arc : (b_{ij},x_{ij},c_{ij})
- F(Q)= 1+2-(1+2+2)=-2
- C(Q)=2+3-(0-1+1)=5

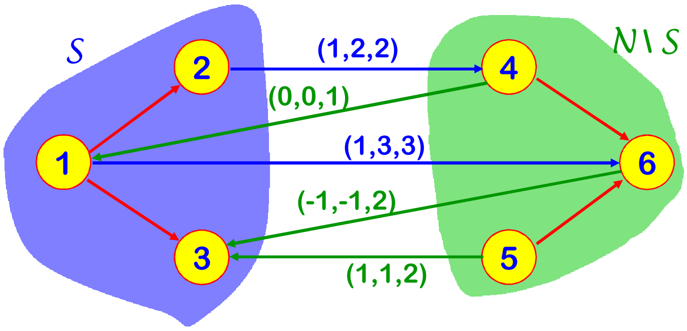
Coupes dans un graphe

- On a toujours

$$F(Q) \leq C(Q)$$

- Si F(Q)=C(Q), on dit que la coupe est saturée par rapport au vecteur de flots x.
- Par convention, une coupe vide est supposée saturée.

Coupes dans un graphe



- Sur chaque arc : (b_{ij},x_{ij},c_{ij})
- F(Q)= 2+3-(0-1+1)=5
- C(Q)=2+3-(0-1+1)=5

Coupes dans un graphe

Théorème

- Soit x un vecteur de flots vérifiant les contraintes de capacité.
- Soit s et t deux nœuds.
- Exactement une des deux affirmations suivantes est vérifiée :

1. Il existe un chemin simple de s à t non bloqué par rapport à x.
2. Il existe une coupe saturée séparant s de t.

Coupes dans un graphe

Algorithme du chemin non bloqué

Idée :

- On génère une suite d'ensembles de nœuds (T_k), avec $T_0 = \{s\}$
- T_k contient l'ensemble des nœuds qui peuvent être atteints à partir de s par un chemin non bloqué de k arcs.

Coupes dans un graphe

Algorithme du chemin non bloqué

Pour k=0,1,…

- Si $T_k = \emptyset$ ou $t \in T_k$, STOP
- $T_{k+1} = \emptyset$
- Pour tout nœud $i \in T_k$
  - Pour tout $(i,j) \in A$
    - Si $x_{ij} < c_{ij}$ et $j \notin T_0, \ldots T_k$ alors $T_{k+1} = T_{k+1} \cup \{j\}$
  - Pour tout $(j,i) \in A$
    - Si $x_{ij} > b_{ij}$ et $j \notin T_0, \ldots T_k$ alors $T_{k+1} = T_{k+1} \cup \{j\}$

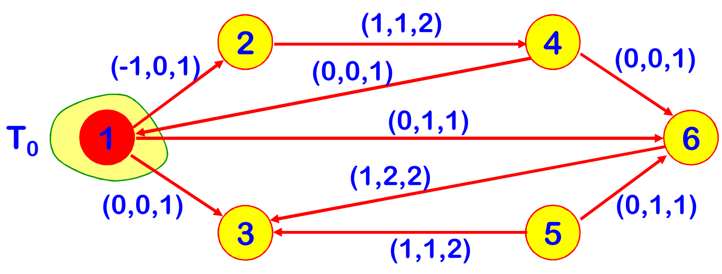
Coupes dans un graphe

Notes :

- Il y a deux manières pour cet algorithme de s'arrêter
  1. $t \in T_{k.}$ Dans ce cas, il existe un chemin non bloqué de s à t.
  2. $T_k = \emptyset$. Si on définit $S = \cup T_i$, alors la coupe $[\mathcal{S}, \mathcal{N} \setminus \mathcal{S}]$ est saturée.

Coupes dans un graphe

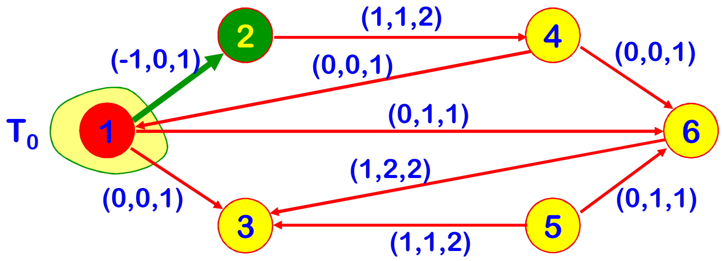
Exemple 1: s=1, t=5



- Sur chaque arc : (b_{ij}, x_{ij}, c_{ij})

Coupes dans un graphe

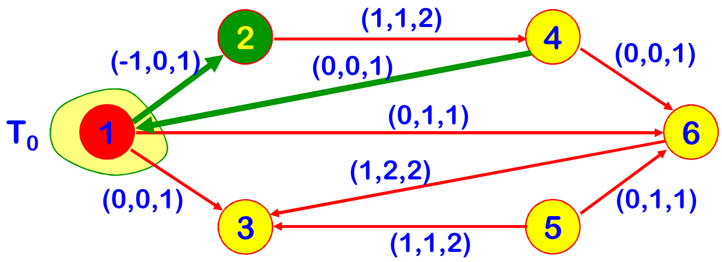
- **Exemple 1: s=1, t=5**



- Sur chaque arc : (b_{ij},x_{ij},c_{ij})

Coupes dans un graphe

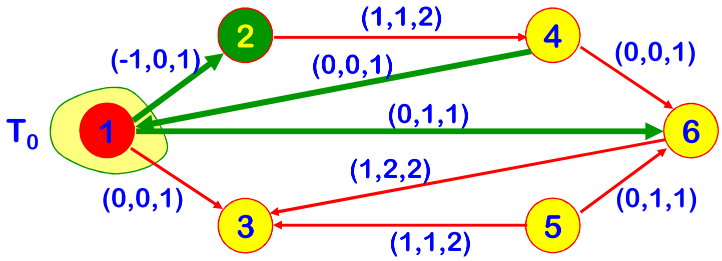
- **Exemple 1: s=1, t=5**



- Sur chaque arc : (b_{ij},x_{ij},c_{ij})

Coupes dans un graphe

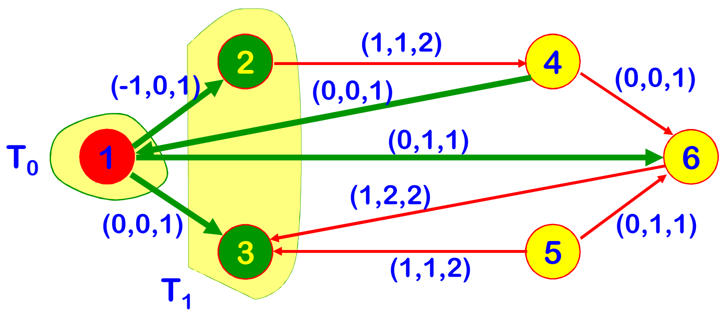
- **Exemple 1: s=1, t=5**



- Sur chaque arc : (b_{ij},x_{ij},c_{ij})

Coupes dans un graphe

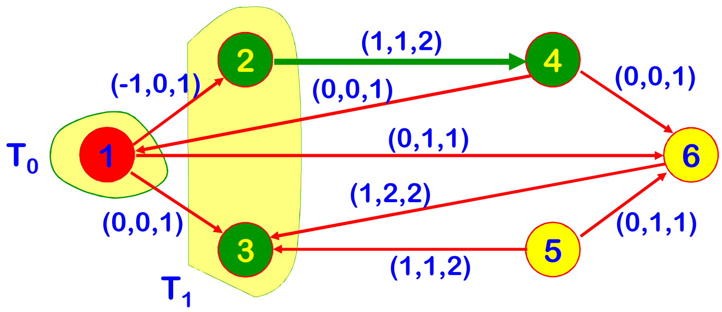
- **Exemple 1: s=1, t=5**



- Sur chaque arc : (b_{ij},x_{ij},c_{ij})

Coupes dans un graphe

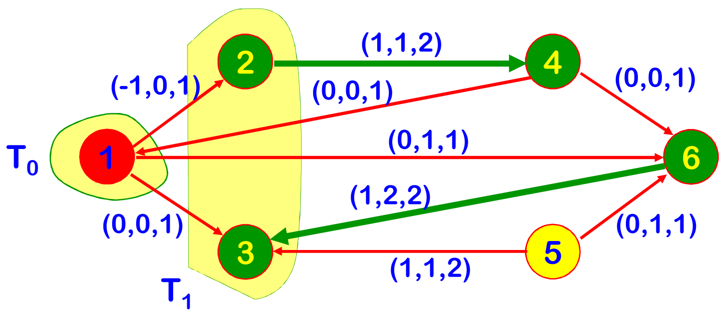
- **Exemple 1: s=1, t=5**



- Sur chaque arc : (b_{ij},x_{ij},c_{ij})

Coupes dans un graphe

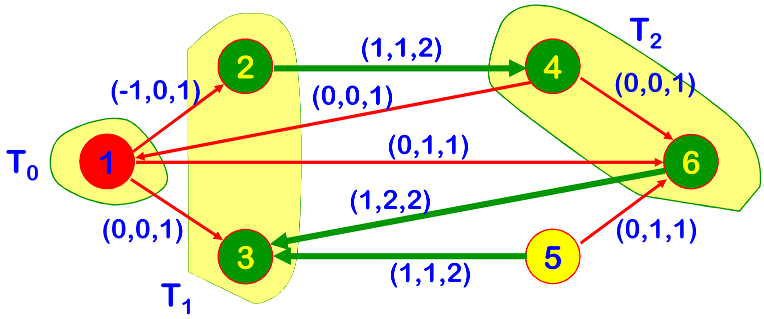
- **Exemple 1: s=1, t=5**



- Sur chaque arc : (b_{ij},x_{ij},c_{ij})

Coupes dans un graphe

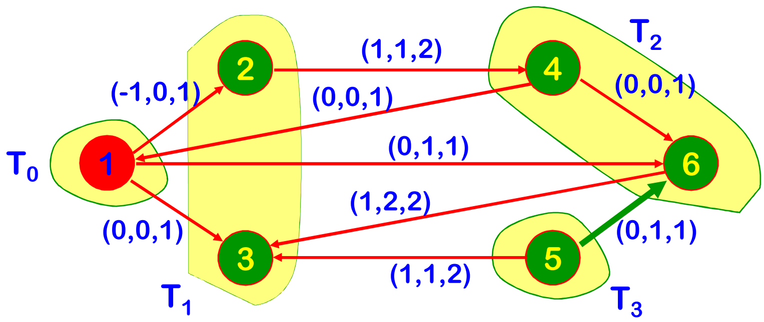
- **Exemple 1: s=1, t=5**



- Sur chaque arc : (b_{ij},x_{ij},c_{ij})

Coupes dans un graphe

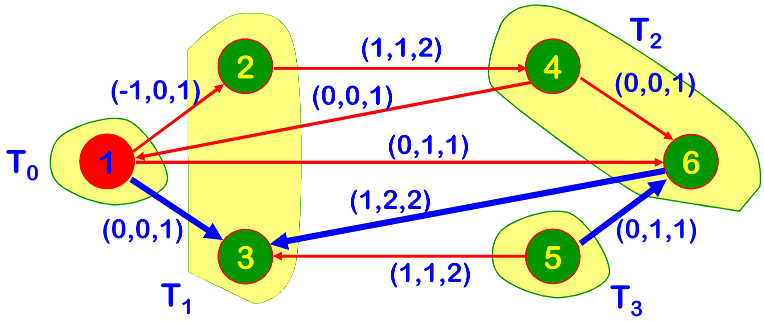
- **Exemple 1: s=1, t=5**



- Sur chaque arc : (b_{ij},x_{ij},c_{ij})

Coupes dans un graphe

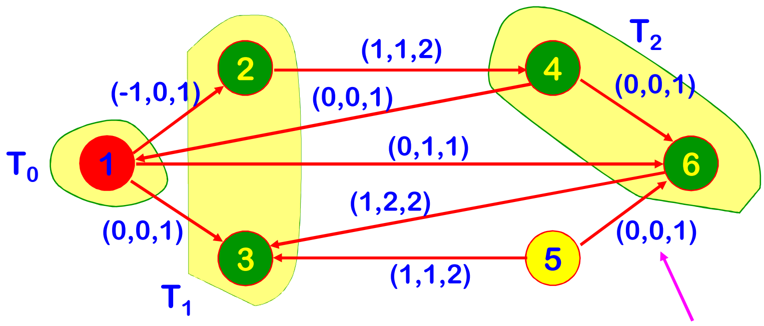
Exemple 1: s=1, t=5



- Sur chaque arc : (b_{ij},x_{ij},c_{ij})

Coupes dans un graphe

Exemple 2: s=1, t=5



- Coupe saturée {(5,3),(5,6)} séparant s et t

Coupe et flot maximal

- Dans le problème de flot maximal, on cherche parmi les vecteurs de flot admissibles, ayant une divergence nulle en tout nœud différente de s ou t, celui qui maximise la divergence de s.
- Soit un de ces vecteurs de flots, et une coupe Q=[$\mathcal{S}$,$\mathcal{N} \setminus \mathcal{S}$] séparant s de t.
- La divergence de s est le flot qui traverse Q. En effet, s est le seul nœud de $\mathcal{S}$ à divergence non nulle.

Coupe et flot maximal

- On a donc

  $\forall$Q, divergence s = F(Q) $\leq$ C(Q)

  flot maximal $\leq$ capacité de Q
- Si le problème de flot maximum possède une solution, alors il existe une coupe telle que l'inégalité soit une égalité.

Coupe et flot maximal

Théorème de flot maximal/coupe minimale

- Soit x* est une solution optimale du problème de flot maximum.
- Soit Q* la coupe de capacité minimum séparant s de t.
- Alors la divergence de s est égale à la capacité de Q*.

Algorithme de Ford-Fulkerson

Idée :

- Soit un vecteur de flots tel que
  - il vérifie les contraintes de capacité
  - divergence nœud i=0, si $i \neq s$, $i \neq t$
- Soit un chemin P non bloqué de s à t.
- On envoie le plus de flot possible le long de P : augmentation du flot.
- On dira que P est un chemin d'augmentation.

Algorithme de Ford-Fulkerson

Augmentation du flot :

$$m_{\text{avant}} = \min\{c_{ij} - x_{ij} | (i,j) \in P^+\}$$

$$m_{\text{arr.}} = \min\{x_{ij} - b_{ij} | (i,j) \in P^-\}$$

$$\delta = \min(m_{\text{avant}}, m_{\text{arr.}})$$

$$\bar{x}_{ij} = \begin{cases} x_{ij} + \delta & \text{si } (i,j) \in P^+ \\ x_{ij} - \delta & \text{si } (i,j) \in P^- \\ x_{ij} & \text{sinon} \end{cases}$$

Algorithme de Ford-Fulkerson

Initialisation

- Trouver un vecteur de flots admissible.
- Si toutes les capacités inférieures b_{ij} sont nulles, le vecteur nul est admissible.

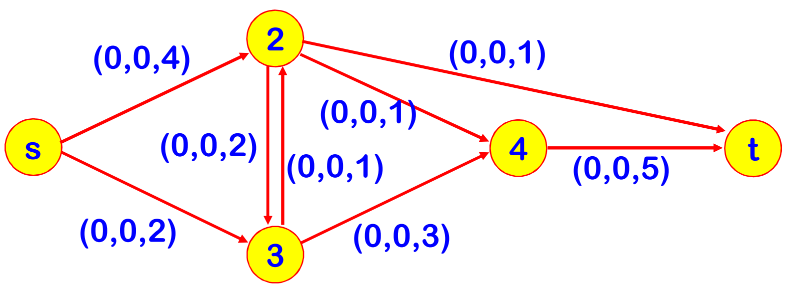
Itérations

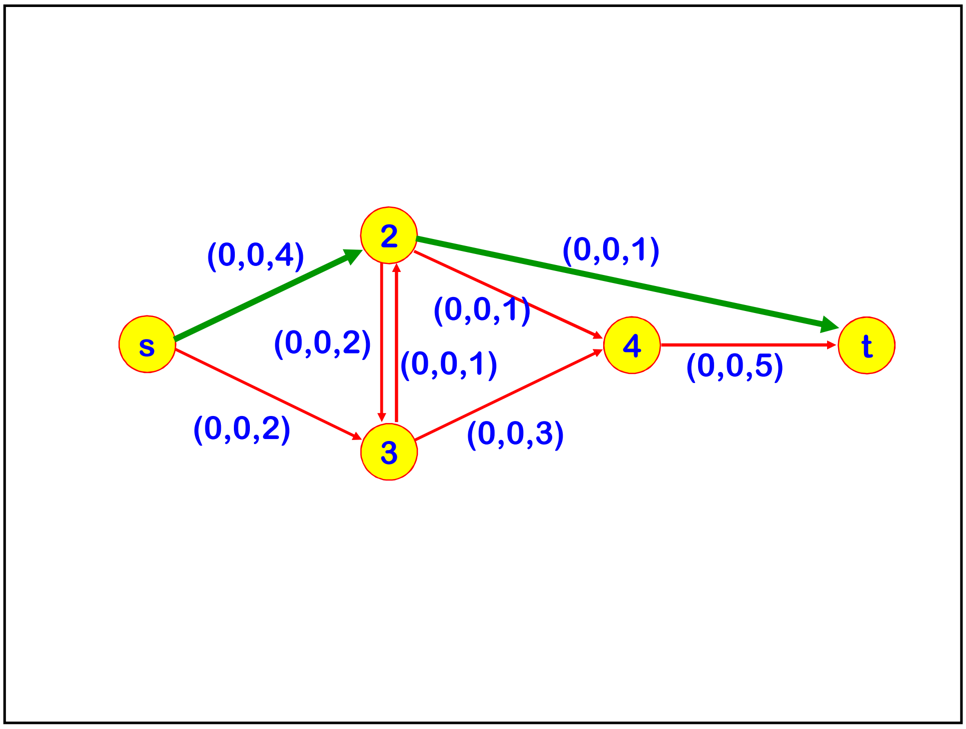
- Appliquer l'algorithme du chemin non bloqué.
- Si coupe saturée : STOP.
- Si chemin non bloqué : augmentation de flot.

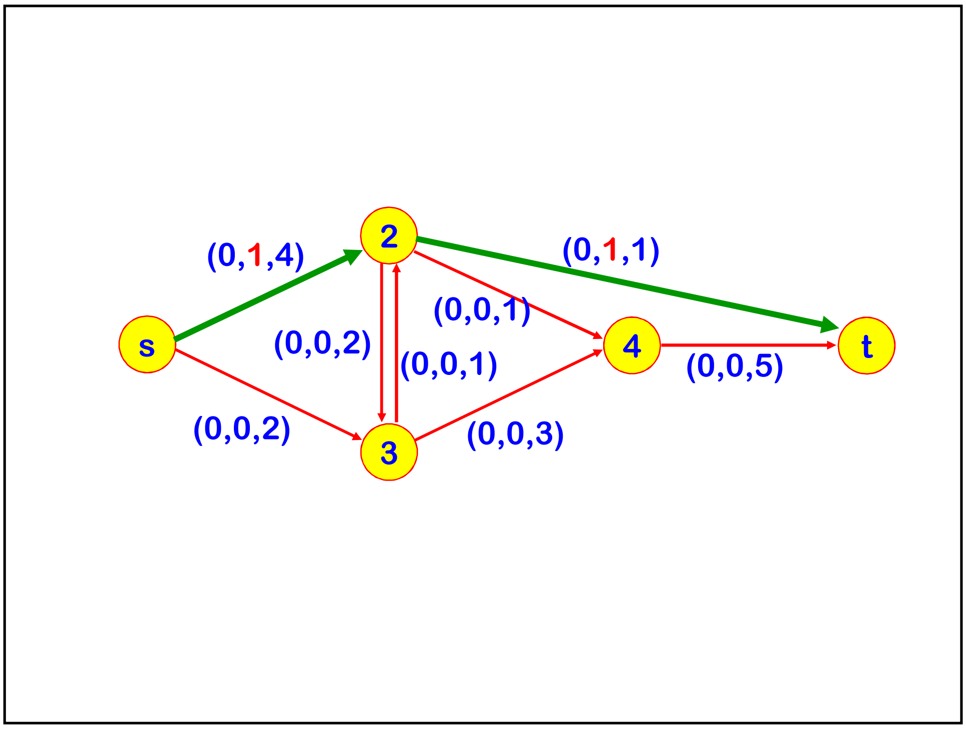
Algorithme de Ford-Fulkerson

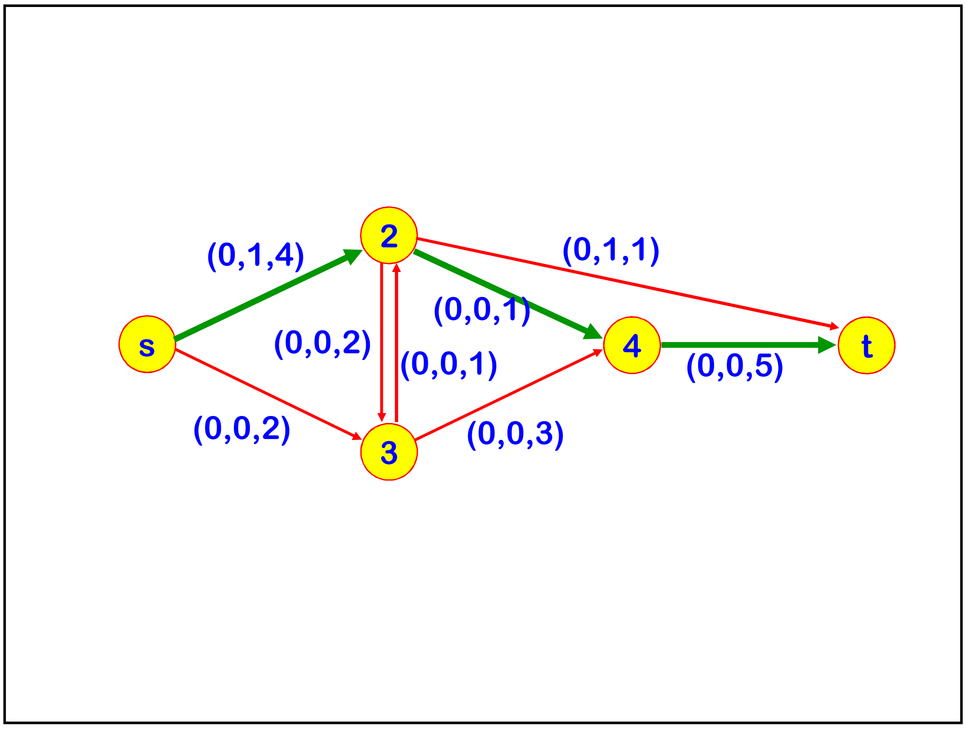
Notes :

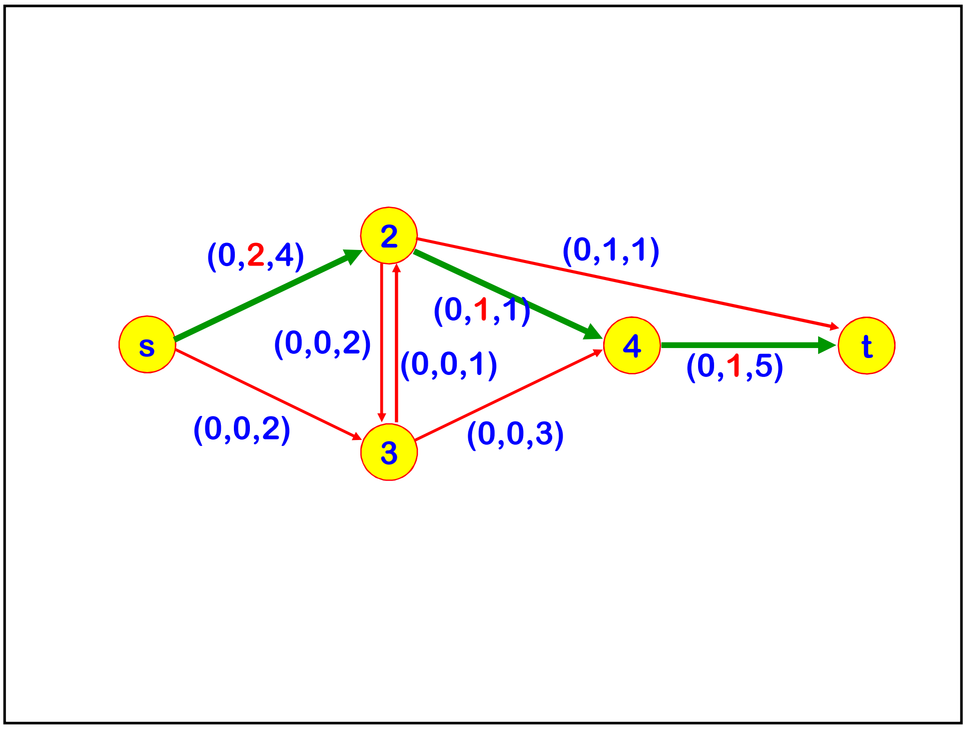
- Si le vecteur de flot initial ainsi que les bornes sont entiers ou rationnels, l'algorithme se terminera en un nombre fini d'itérations.
- Méthode du plus court chemin d'augmentation : parmi tous les chemins non bloqués, choisir comme chemin d'augmentation celui comportant le moins d'arcs.
- Si cette méthode est utilisée, l'algorithme convergera toujours.
- De plus, même avec des données entières, cette méthode est plus performante.

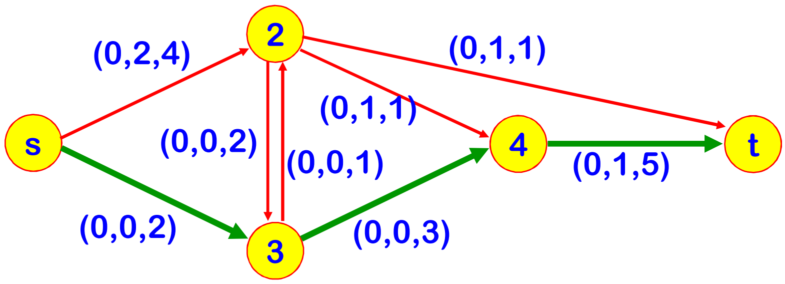


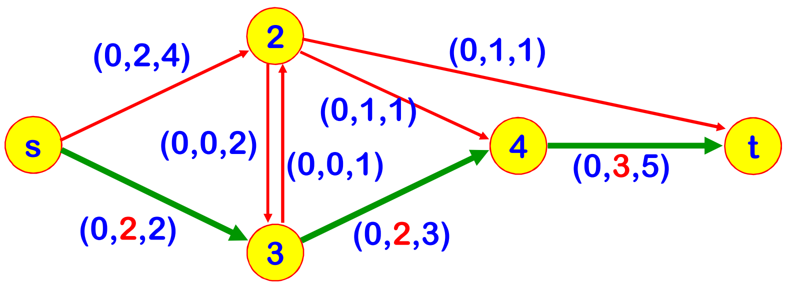

(0,0,4)
2
(0,0,1)
(0,0,1)
s
(0,0,2)
(0,0,1)
4
(0,0,5)
t
(0,0,2)
3
(0,0,3)

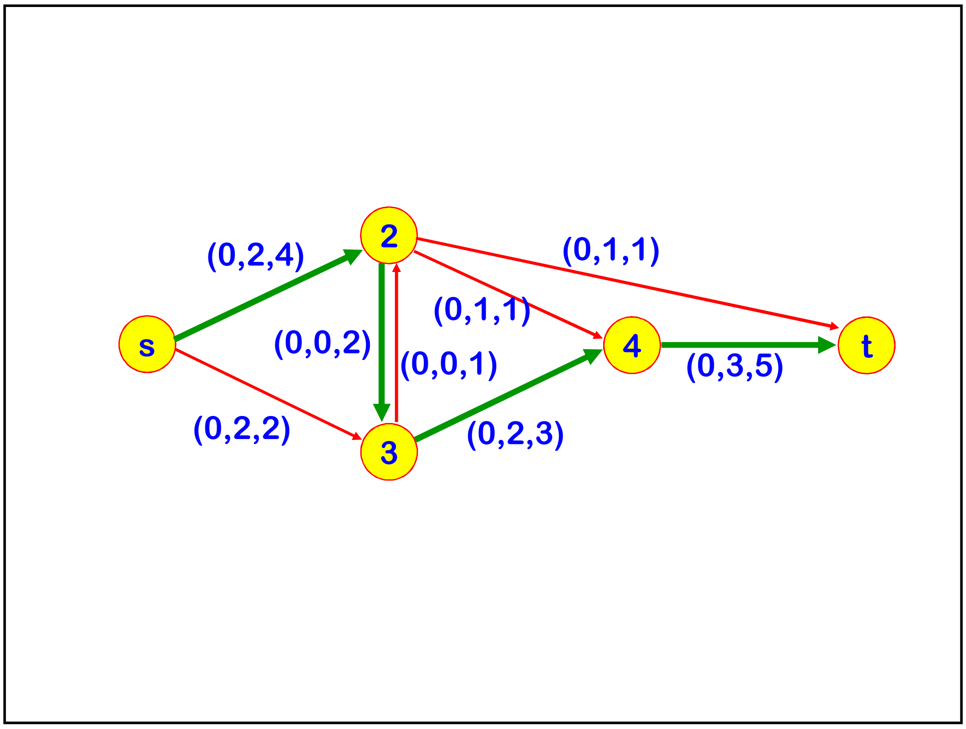

(0,1,4)
2
(0,1,1)
(0,0,1)
s
(0,0,2)
(0,0,1)
4
(0,0,5)
t
(0,0,2)
3
(0,0,3)

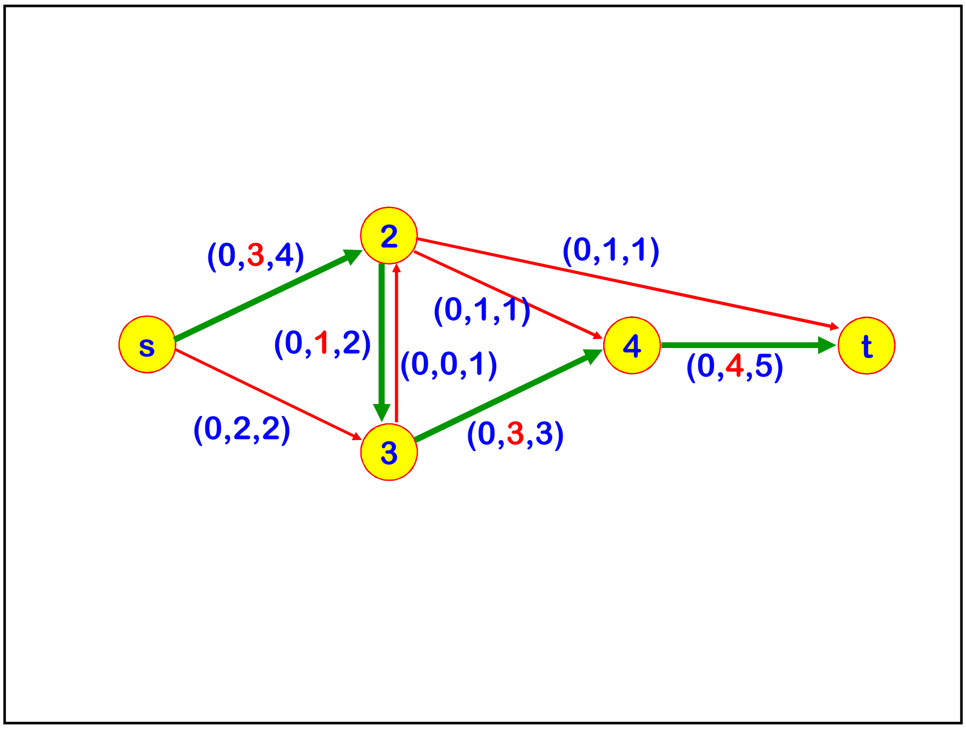

(0,1,4)
(0,1,1)
(0,0,1)
(0,0,2)
(0,0,1)
(0,0,5)
(0,0,2)
(0,0,3)
s
2
3
4
t

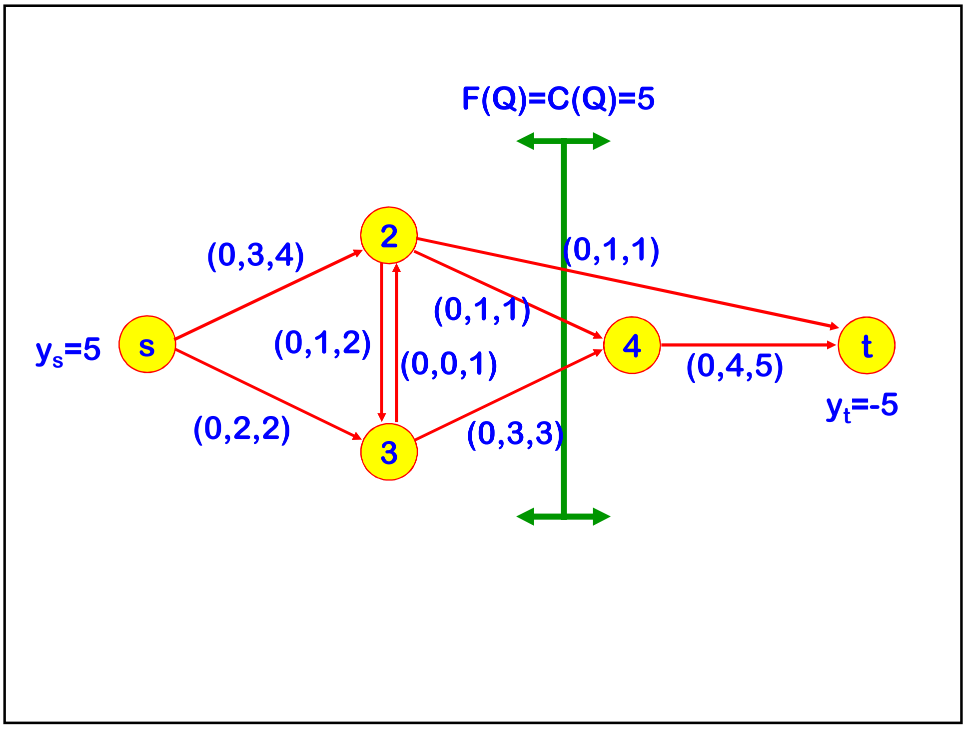

(0,2,4)
(0,1,1)
(0,1,1)
(0,0,2)
(0,0,1)
(0,1,5)
(0,0,2)
(0,0,3)
s
2
3
4
t


2
(0,2,4)
(0,1,1)
(0,1,1)
s
(0,0,2)
(0,0,1)
4
t
(0,1,5)
(0,0,2)
3
(0,0,3)


2
(0,2,4)
(0,1,1)
(0,1,1)
s
(0,0,2)
(0,0,1)
4
t
(0,3,5)
(0,2,2)
3
(0,2,3)


(0,2,4)
2
(0,1,1)
(0,1,1)
s
(0,0,2)
(0,0,1)
4
(0,3,5)
t
(0,2,2)
3
(0,2,3)


(0,3,4)
2
(0,1,1)
(0,1,1)
s
(0,1,2)
(0,0,1)
4
(0,4,5)
t
(0,2,2)
3
(0,3,3)


F(Q)=C(Q)=5
(0,3,4)
2
(0,1,1)
(0,1,1)
y_s=5
s
(0,1,2)
(0,0,1)
4
(0,4,5)
t
y_t=-5
(0,2,2)
3
(0,3,3)