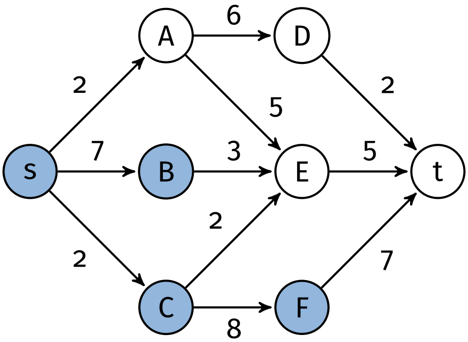
RO03 : Recherche opérationnelle

Problèmes de flots

Dritan Nace, David Savourey



Réseau de transport

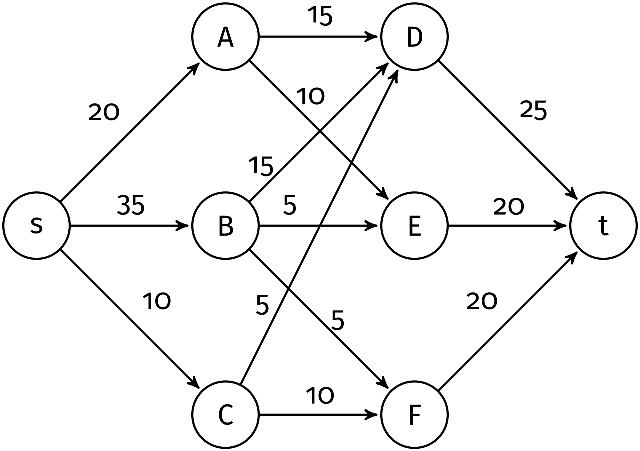
Un premier exemple de réseau de transport

Exemple : On dispose respectivement de 20, 35 et 10 tonnes de marchandise dans les dépôts A, B et C. On doit acheminer respectivement 25, 20 et 20 tonnes aux destinations D, E et F. On dispose de camions pouvant acheminer les quantités suivantes entre les différents points :

	D	**E**	**F**
A	15	10	0
B	15	5	5
C	5	0	10

Problème : Déterminer un plan de transport permettant de satisfaire la demande en respectant les contraintes d'acheminement.

Modélisation sous forme de graphe

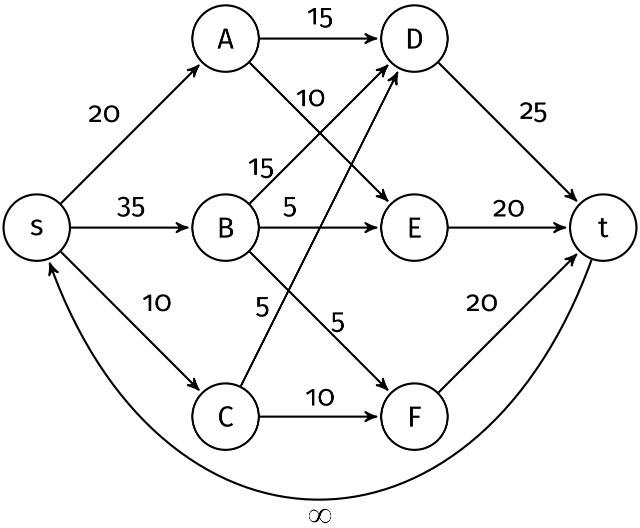


Réseau de transport

Définition

On appelle **réseau de transport** un graphe $G = (X, U, c)$ valué positivement sans boucle, ayant une racine s, un puits t et contenant l'arc (t, s) de valuation infinie. Les valuations des arcs sont appelées **capacités**.

Réseau de transport : exemple



Flot

Définition

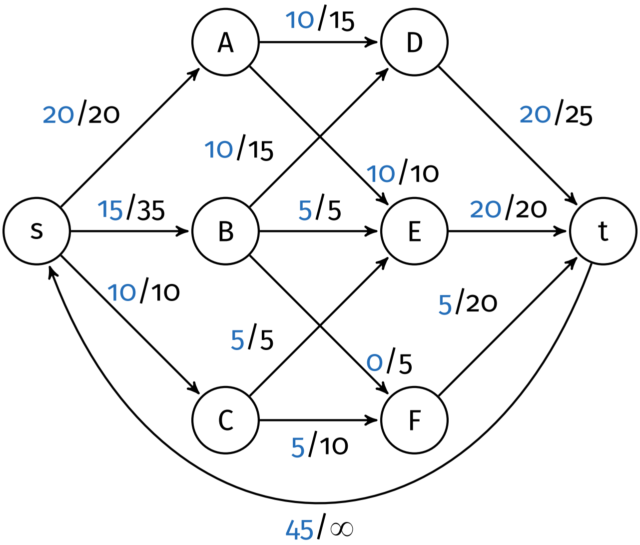
On appelle **flot** sur un réseau de transport $G = (X, U, c)$ toute application $F : U \rightarrow \mathbb{R}$ telle que :

- $\forall (i, j) \in U, 0 \leqslant f_{ij} \leqslant c_{ij}$ (contraintes de capacités)
- $\forall i \in X, \sum_{j \in U^+(i)} f_{ij} = \sum_{k \in U^-(i)} f_{ki}$ (contraintes de conservation)

Remarque

Les contraintes de conservation sont aussi appelées contraintes de Kirchoff.

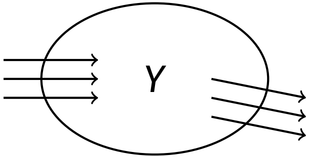
Réseau de transport : exemple



Lemme de Kirchoff

Lemme

Soit Y est un sous-ensemble de X. Le flot sortant de Y est égal au flot entrant en Y.



Flot complet

Flot complet

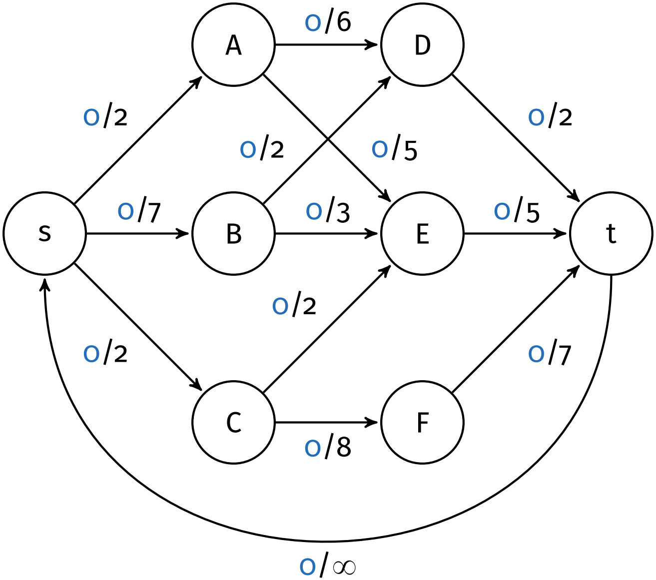
Définition

On dit qu'un flot est un **flot complet** si tout chemin du réseau de transport allant de s à t contient au moins un arc saturé, c'est-à-dire un arc (i, j) tel que $f_{ij} = c_{ij}$.

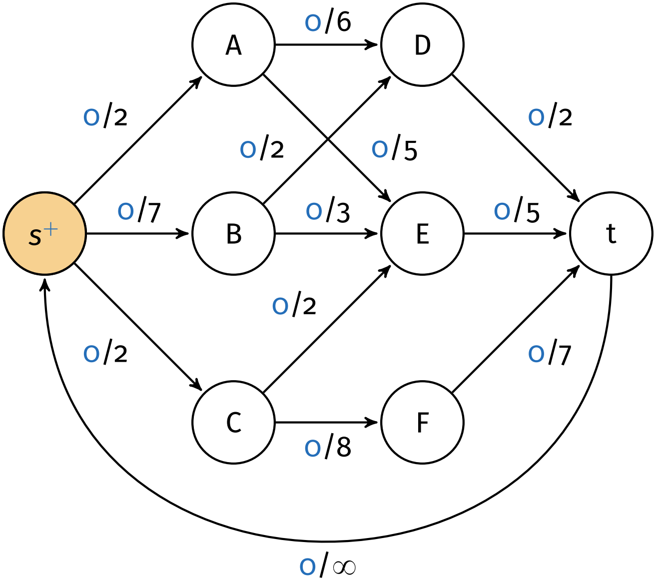
Algorithme : Algorithme de flot complet

Marquer s
tant que il existe un sommet i **marqué non examiné faire**
    si j **successeur de** i **non marqué et** $f_{ij} < c_{ij}$ **alors**
        Marquer j avec $+i$
    si p **est marqué alors**
        Améliorer le flot en utilisant les marques
        Effacer les marques
        Recommencer l'algorithme

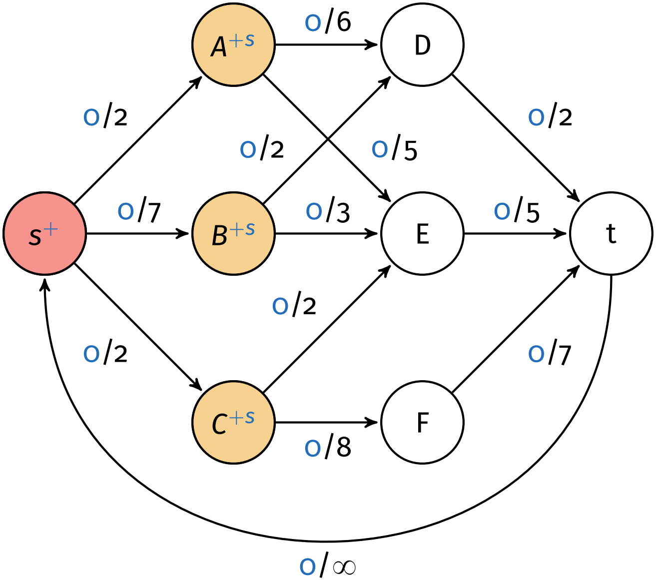
Exemple



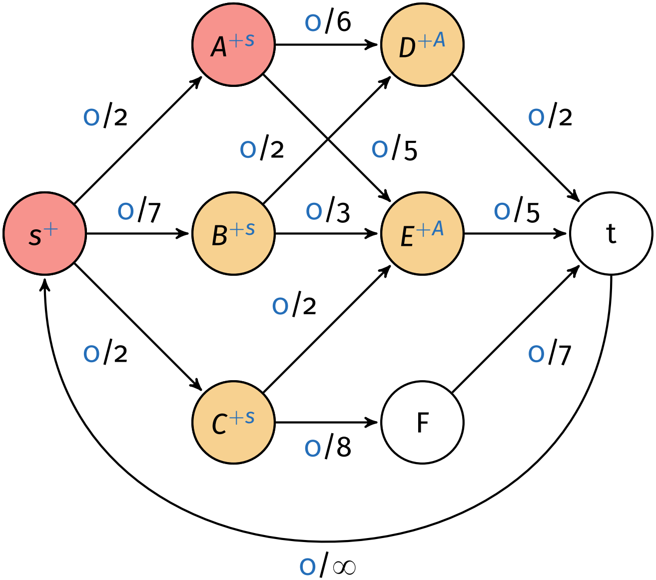
Exemple



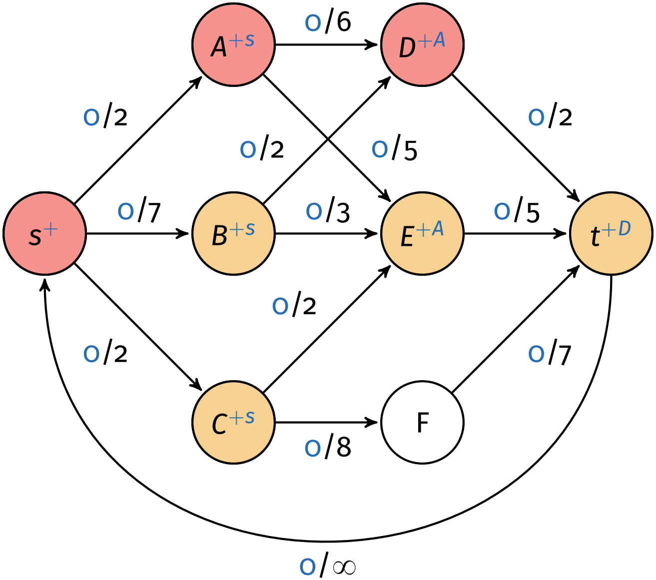
Exemple



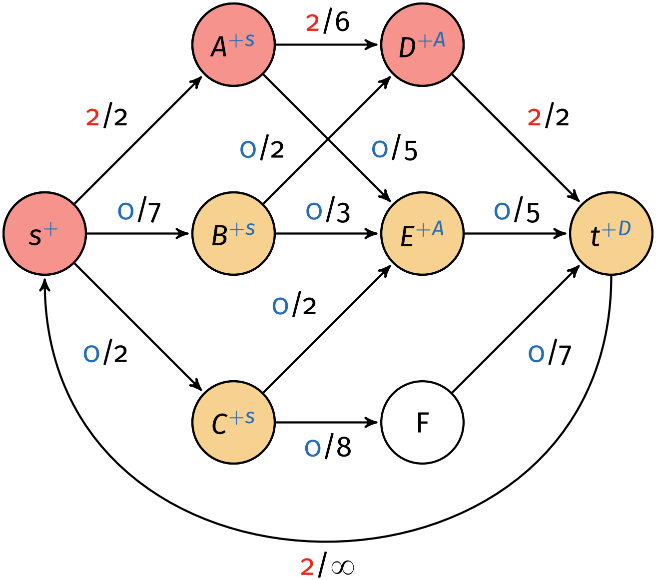
Exemple



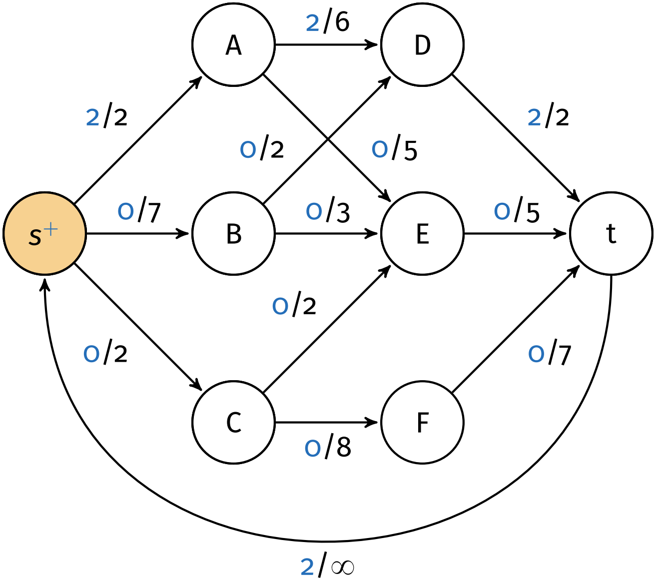
Exemple



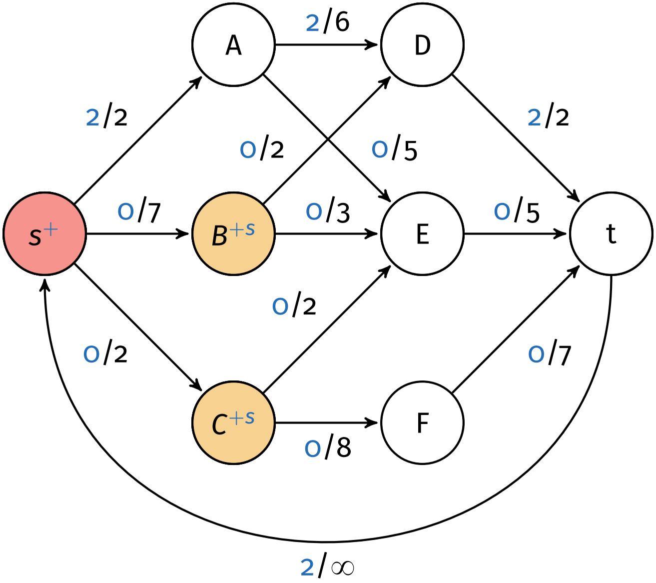
Exemple



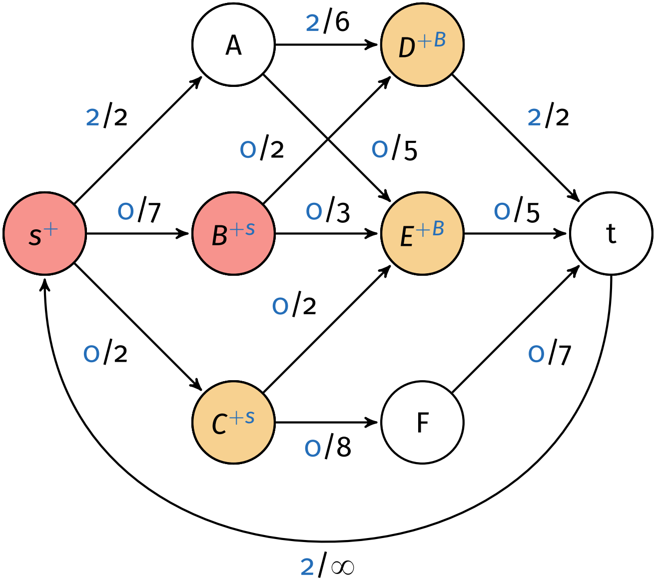
Exemple



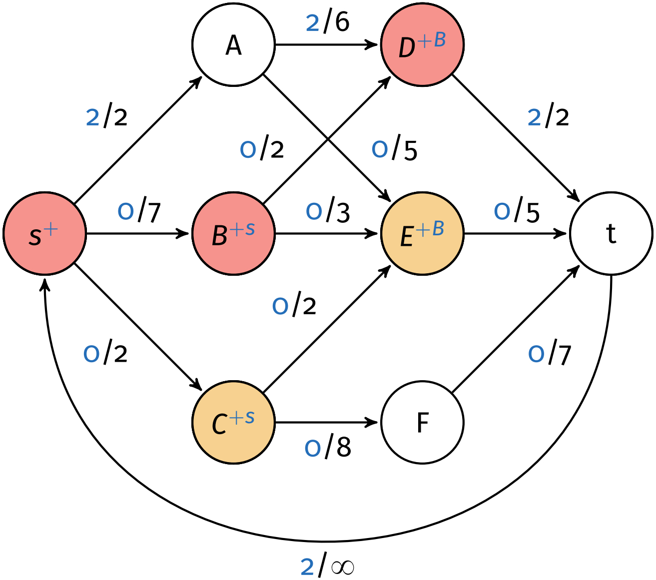
Exemple



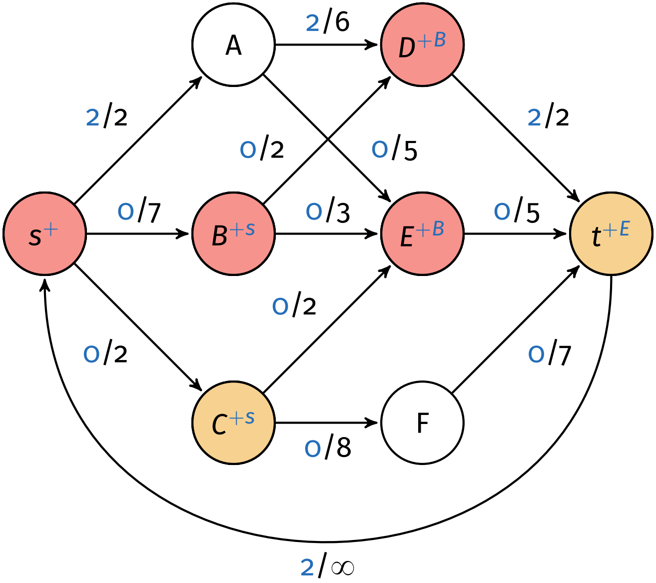
Exemple



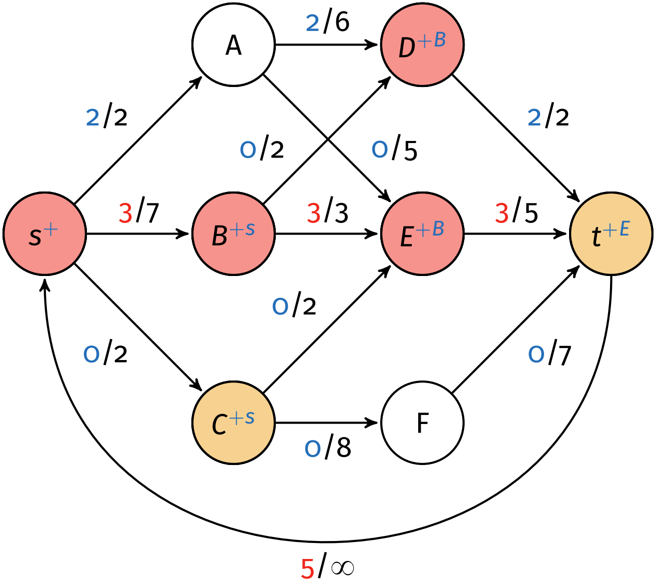
Exemple



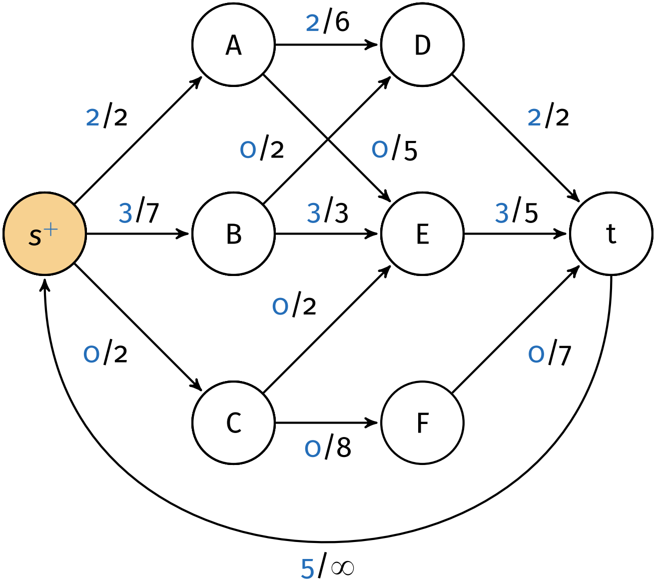
Exemple



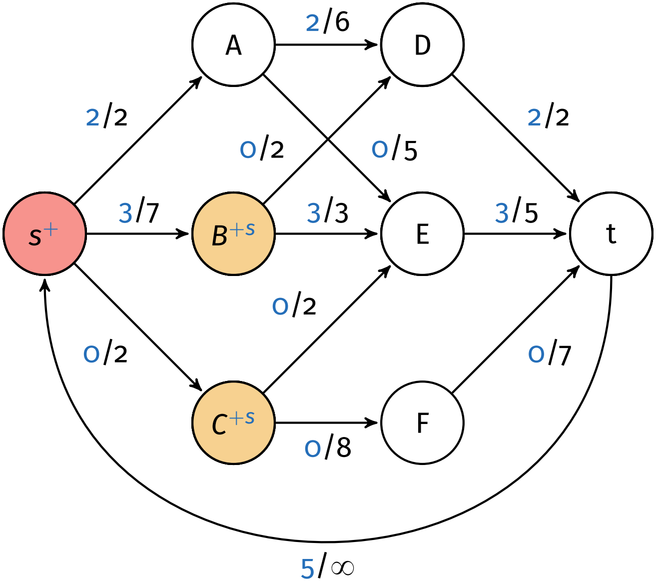
Exemple



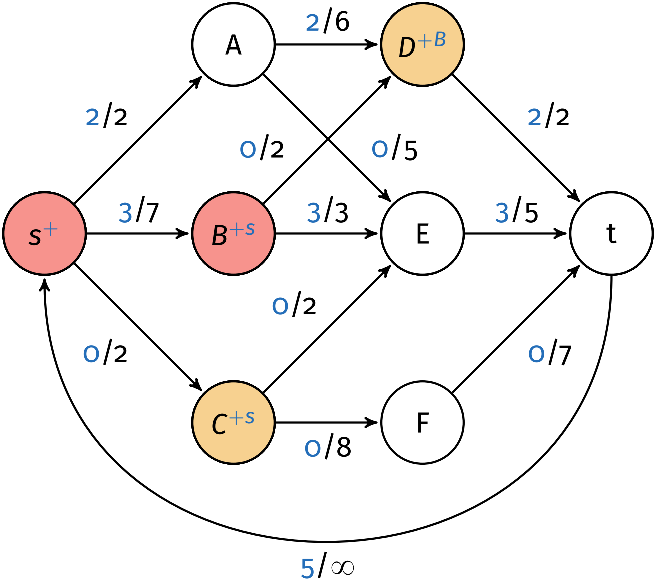
Exemple



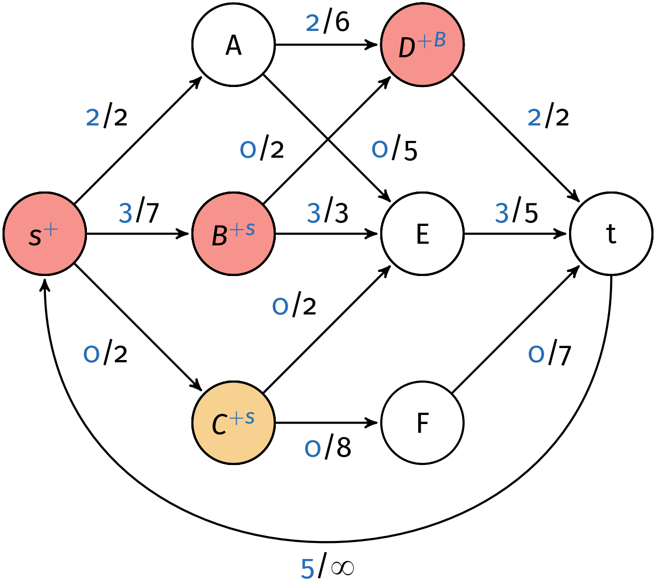
Exemple



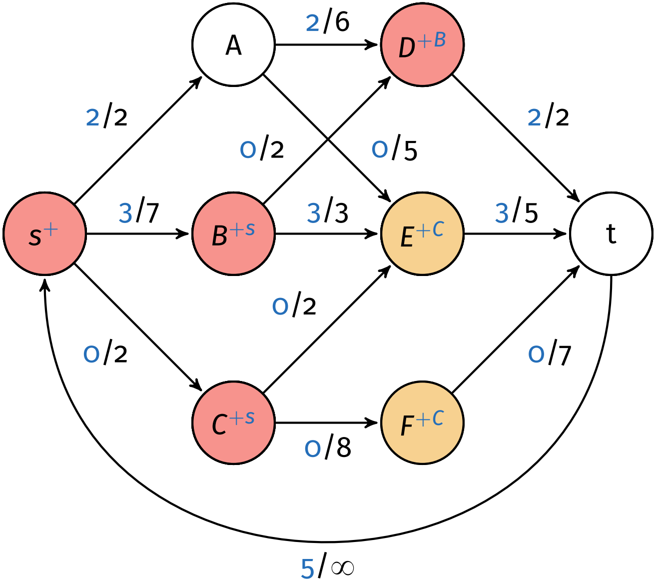
Exemple



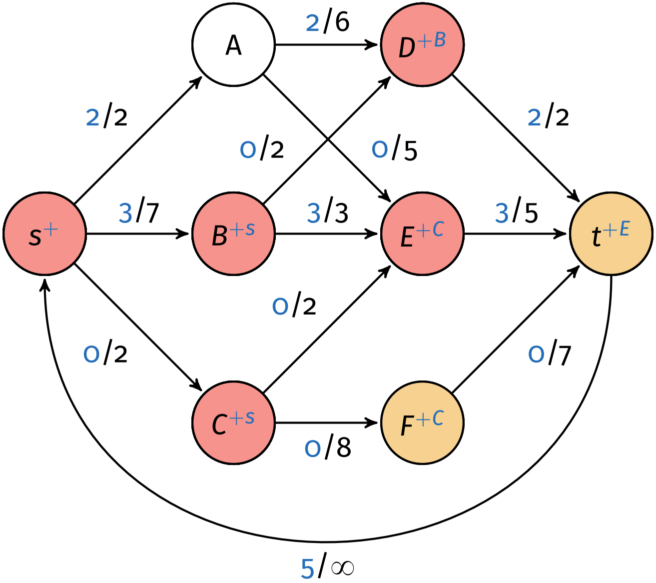
Exemple



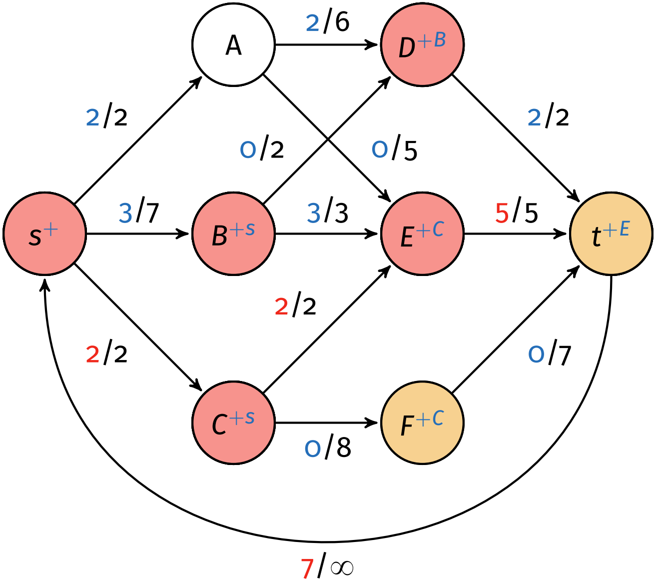
Exemple



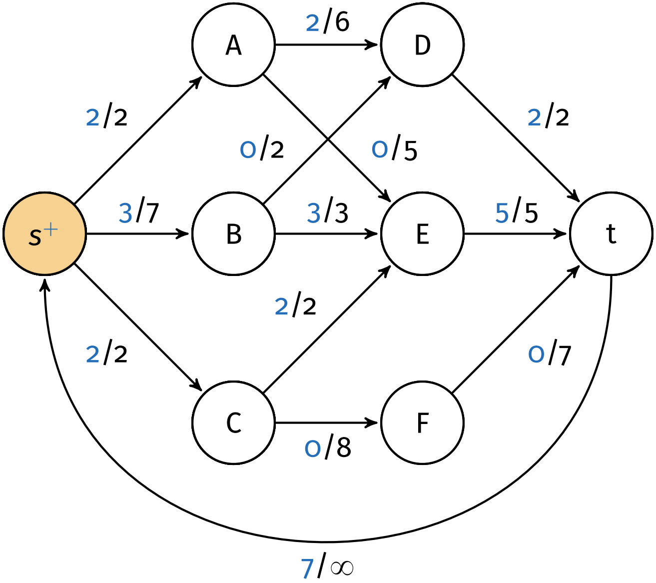
Exemple



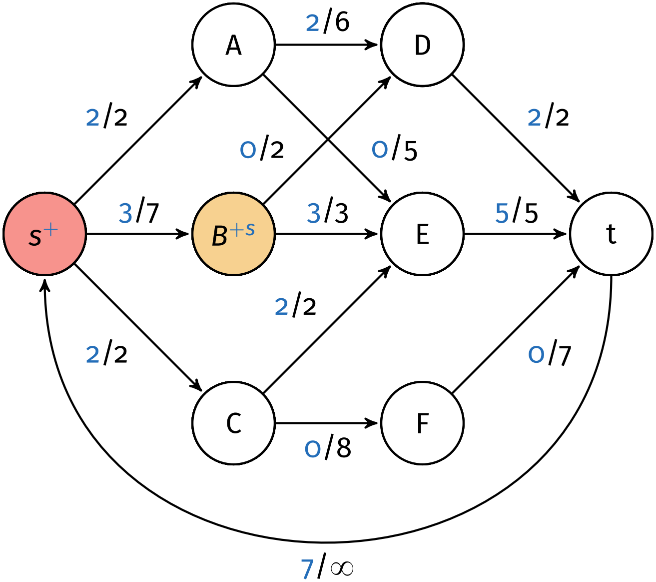
Exemple



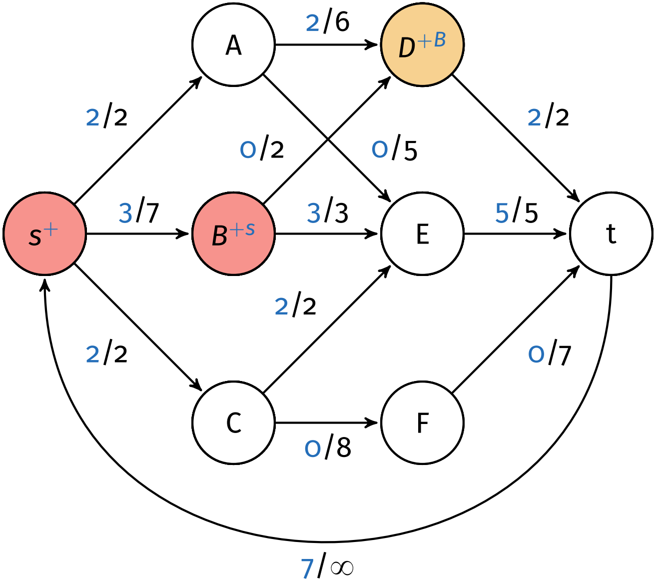
Exemple



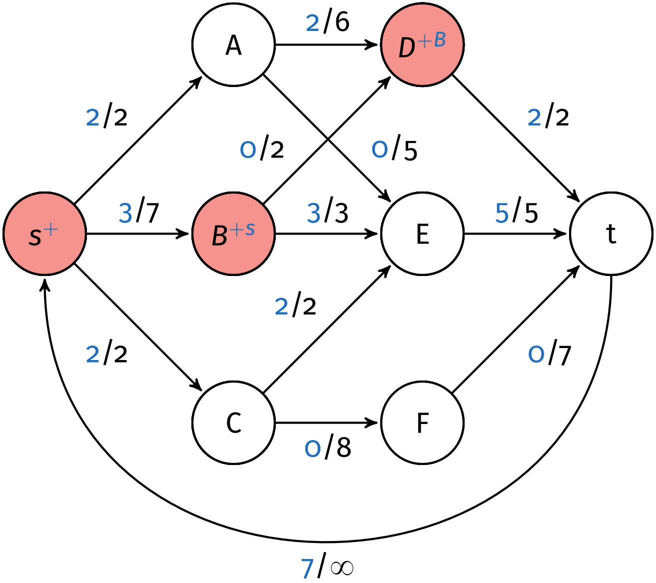
Exemple



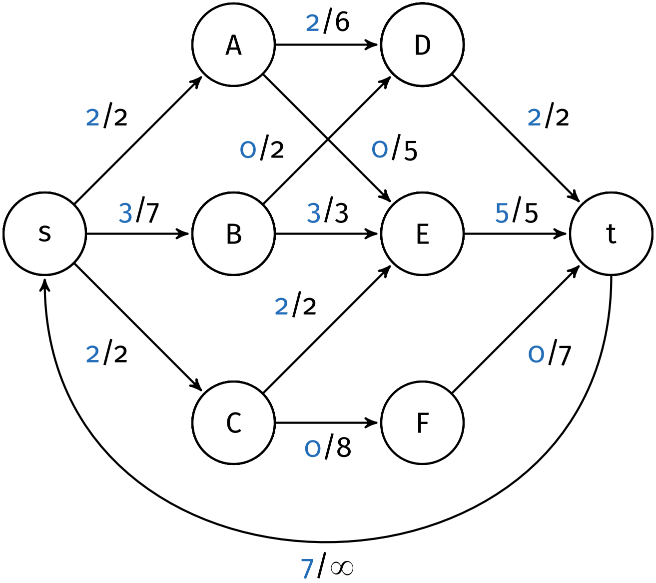
Exemple



Exemple



Exemple

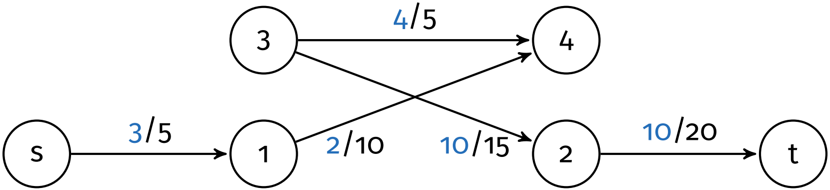


Chaîne améliorante

Définition

Soit $G = (X, U, c)$ un réseau de transport et f un flot sur G. Une chaîne de s à t est dite **améliorante** pour f si :

- tous les arcs qu'elle emprunte « dans le bon sens » sont insaturés ($f_{ij} < c_{ij}$) ;
- tous les arcs qu'elle emprunte « dans le mauvais sens » ont un flot non nul ($f_{ij} > 0$).

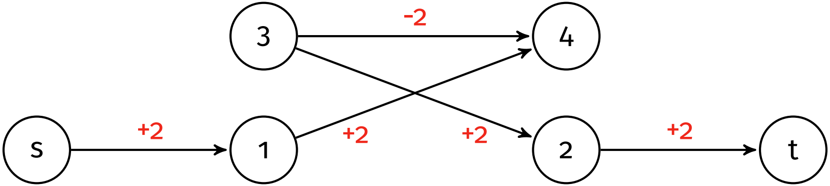


Chaîne améliorante

Définition

Soit $G = (X, U, c)$ un réseau de transport et f un flot sur G. Une chaîne de s à t est dite **améliorante** pour f si :

- tous les arcs qu'elle emprunte « dans le bon sens » sont insaturés ($f_{ij} < c_{ij}$) ;
- tous les arcs qu'elle emprunte « dans le mauvais sens » ont un flot non nul ($f_{ij} > 0$).

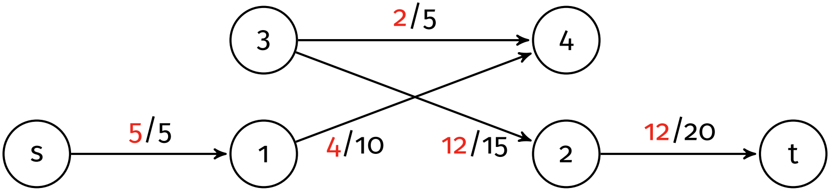


Chaîne améliorante

Définition

Soit $G = (X, U, c)$ un réseau de transport et f un flot sur G. Une chaîne de s à t est dite **améliorante** pour f si :

- tous les arcs qu'elle emprunte « dans le bon sens » sont insaturés $(f_{ij} < c_{ij})$;
- tous les arcs qu'elle emprunte « dans le mauvais sens » ont un flot non nul $(f_{ij} > 0)$.



Flot maximal

Définition

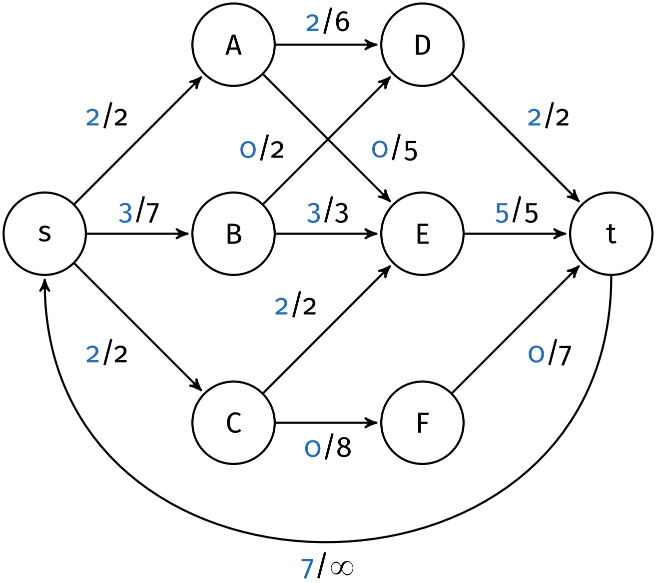
On dit qu'un flot est un **flot maximal** dans un réseau de transport s'il n'existe pas de flot de plus grande valeur.

Algorithme de Ford-Fulkerson

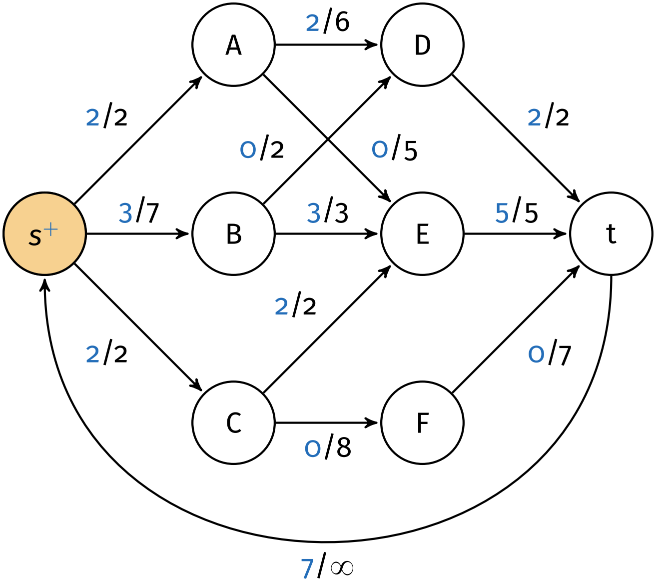
Algorithme : Algorithme de Ford-Fulkerson

Marquer s
tant que il existe un sommet i **marqué non examiné faire**
  si j **successeur de** i **non marqué et** $f_{ij} < c_{ij}$ **alors**
    Marquer j avec $+i$
  si k **prédecesseur de** i **non marqué et** $f_{ki} > 0$ **alors**
    Marquer k avec $-i$
  si p **est marqué alors**
    Améliorer le flot en utilisant les marques
    Effacer les marques
    Recommencer l'algorithme

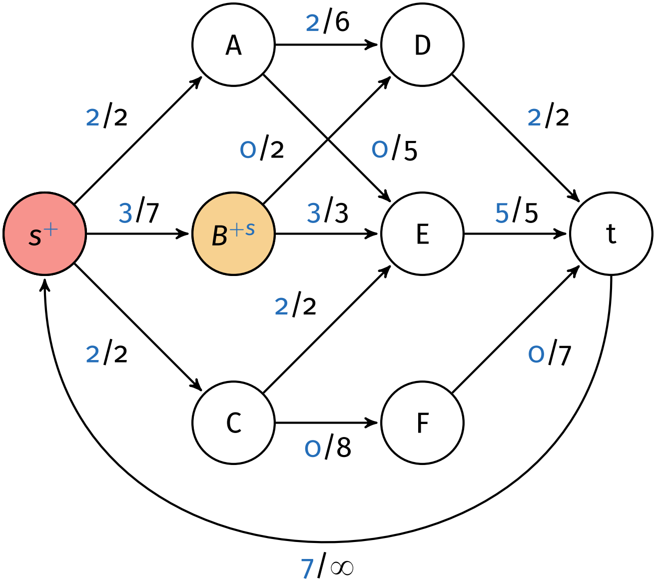
Exemple



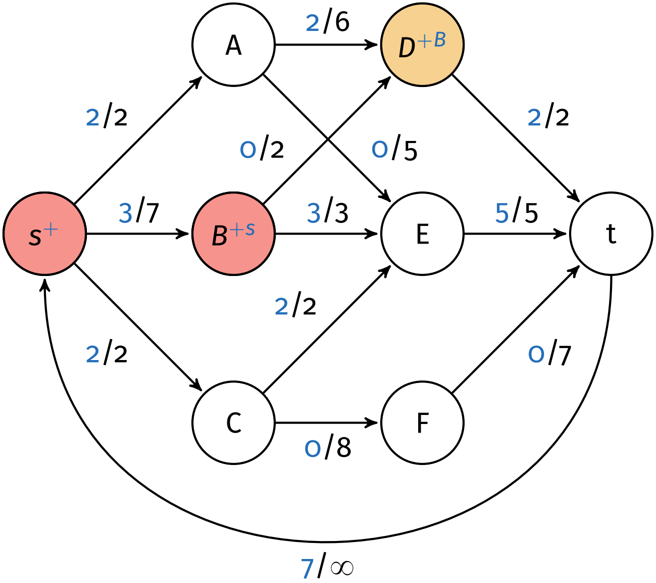
Exemple



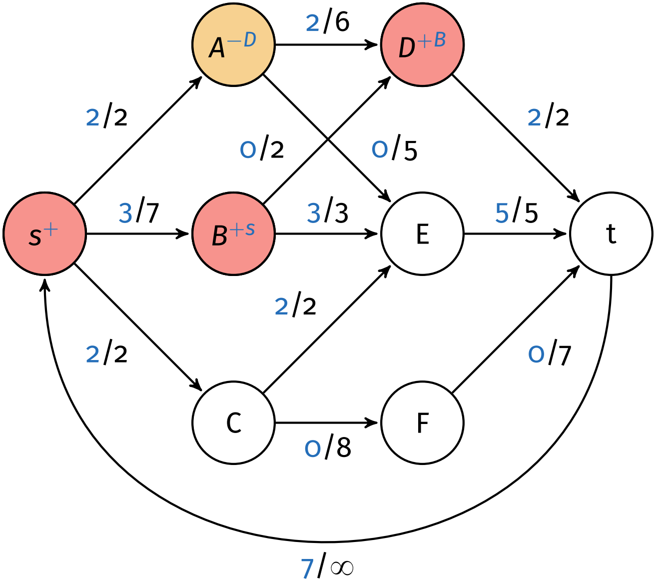
Exemple



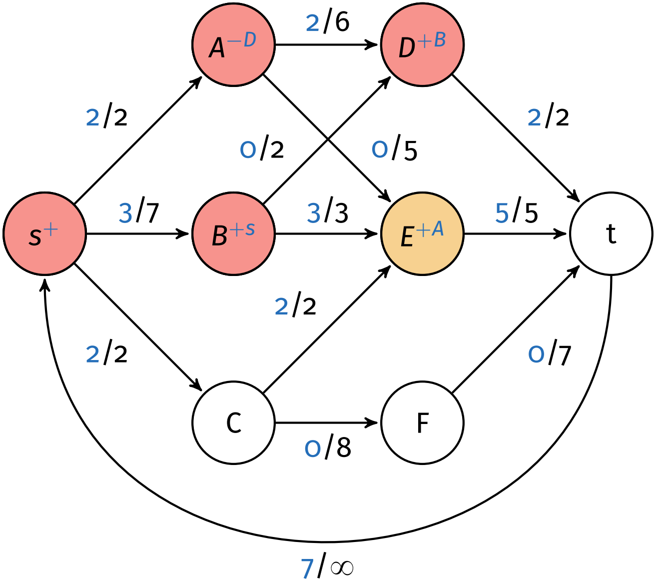
Exemple



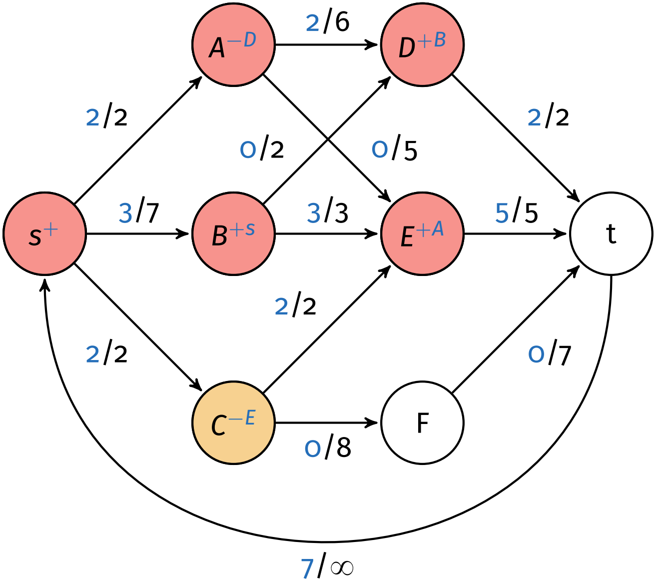
Exemple



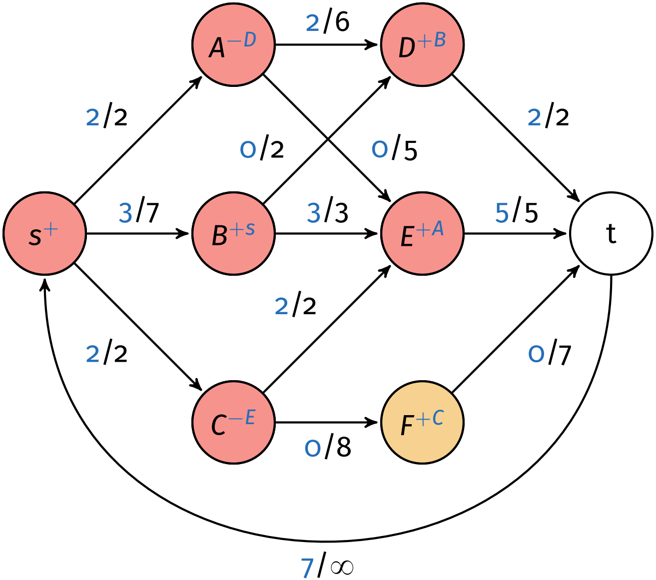
Exemple



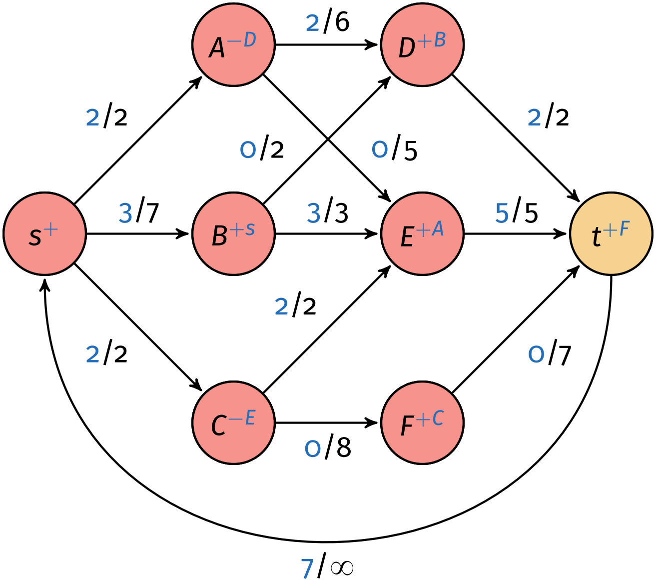
Exemple



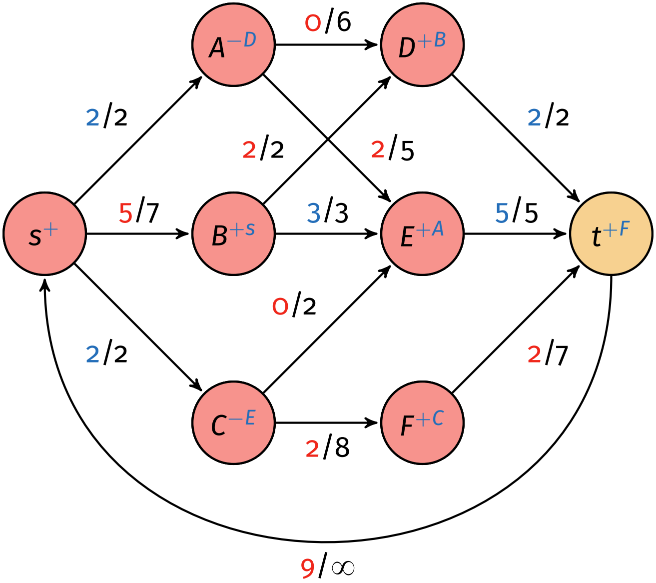
Exemple



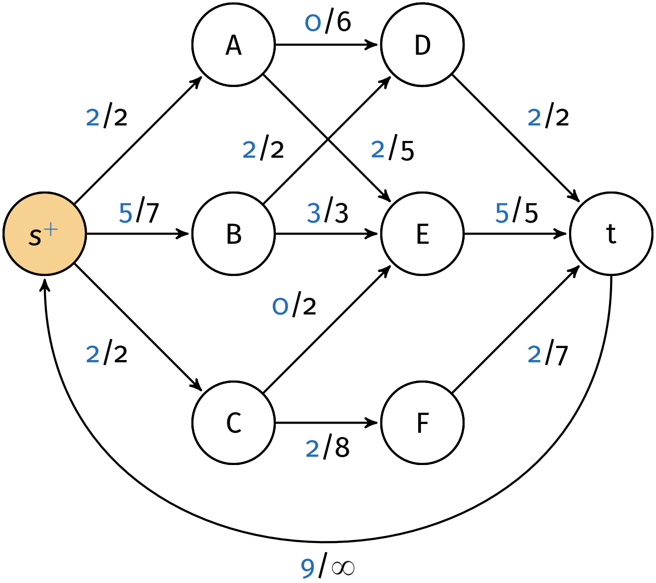
Exemple



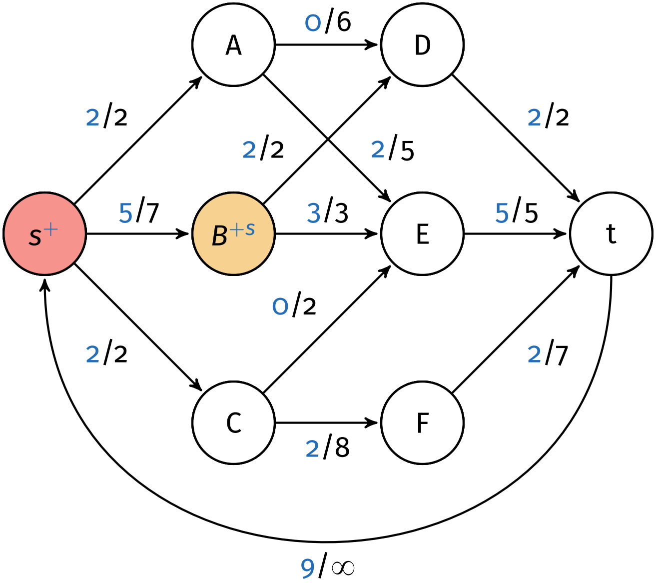
Exemple



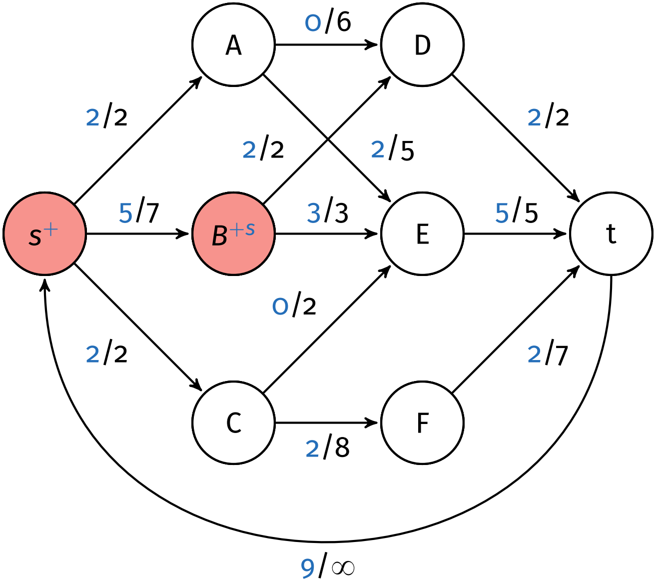
Exemple



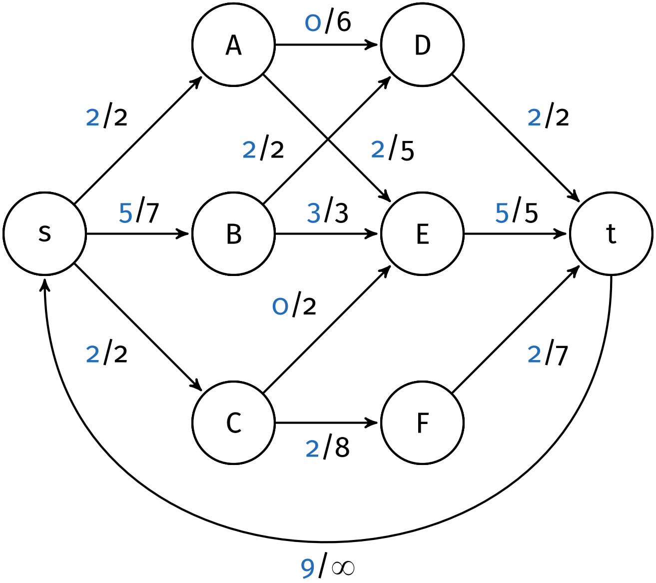
Exemple



Exemple

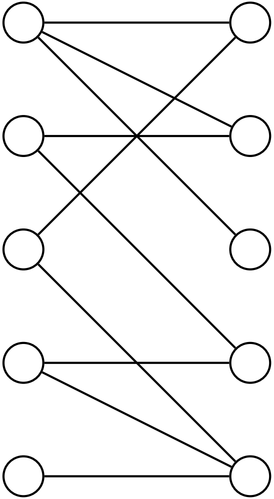


Exemple



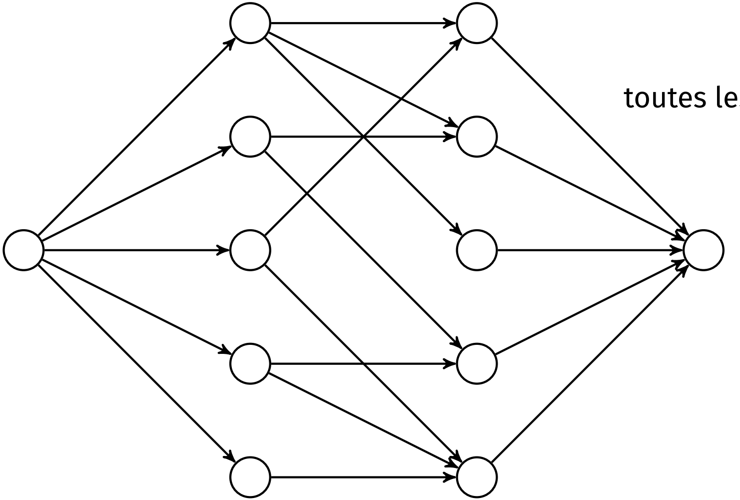
Couplage maximal

Trouver un **couplage maximal** dans un graphe biparti :



Couplage maximal

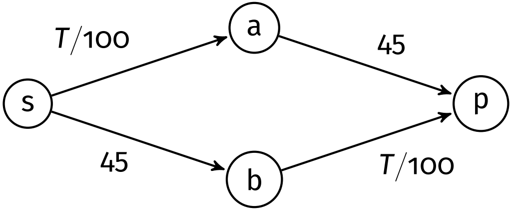
Trouver un **couplage maximal** dans un graphe biparti :



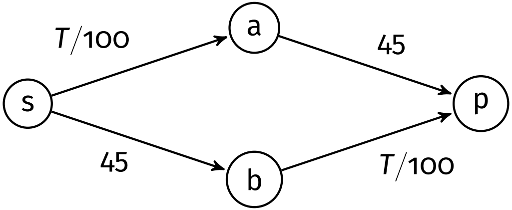
toutes les capacités $= 1$

Paradoxe de Braess

(aparté) : Considérons un réseau de transport et un ensemble de 4000 personnes souhaitant l'emprunter.

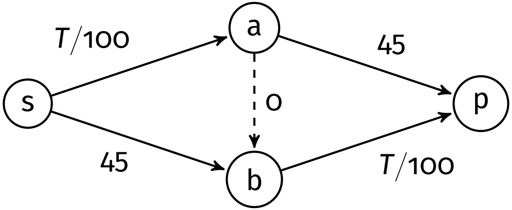


(aparté) : Considérons un réseau de transport et un ensemble de 4000 personnes souhaitant l'emprunter.



Les routes ont la même espérance de temps, donc les conducteurs vont se répartir équitablement entre le haut et le bas (donc $T = 2000$ sur (s, a) et $T = 2000$ sur (b, p), donc le temps de s à p est de $20 + 45 = 65$.

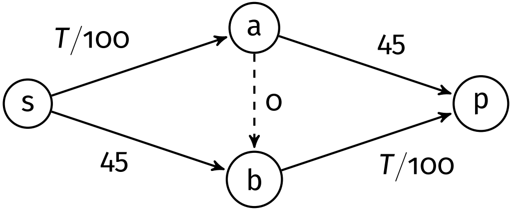
(aparté) : Considérons un réseau de transport et un ensemble de 4000 personnes souhaitant l'emprunter.



On ajoute l'arc (a, b) de temps nul

Paradoxe de Braess

(aparté) : Considérons un réseau de transport et un ensemble de 4000 personnes souhaitant l'emprunter.

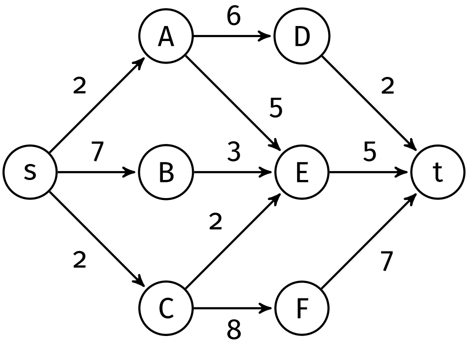


Tous les usagers souhaitent désormais utiliser (s, a) plutôt que (s, b) ($40 < 45$), de même pour (b, p) versus (a, p). Donc le temps est maintenant de 80 (contre 65 avant ajout de l'arc).

Coupe

Définition

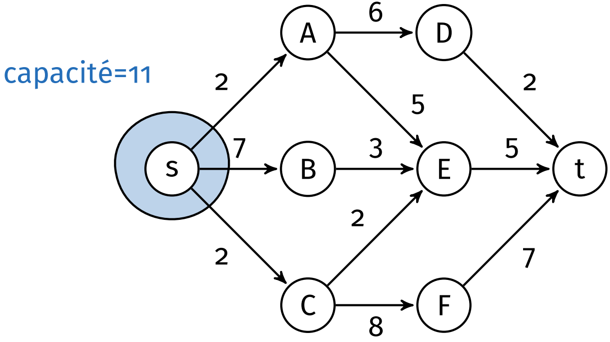
Une **coupe** est un ensemble de sommets contenant s et pas p. La **capacité d'une coupe** est égale à la somme des arcs sortant de cette coupe.



Coupe

Définition

Une **coupe** est un ensemble de sommets contenant s et pas p. La **capacité d'une coupe** est égale à la somme des arcs sortant de cette coupe.

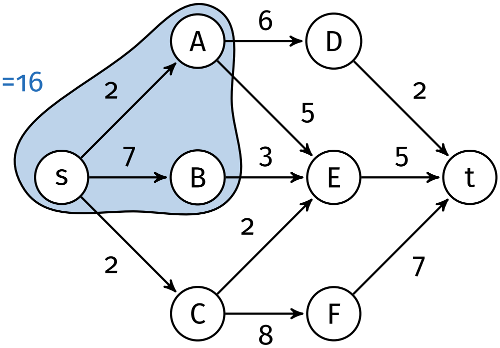


Coupe

Définition

Une **coupe** est un ensemble de sommets contenant s et pas p. La **capacité d'une coupe** est égale à la somme des arcs sortant de cette coupe.

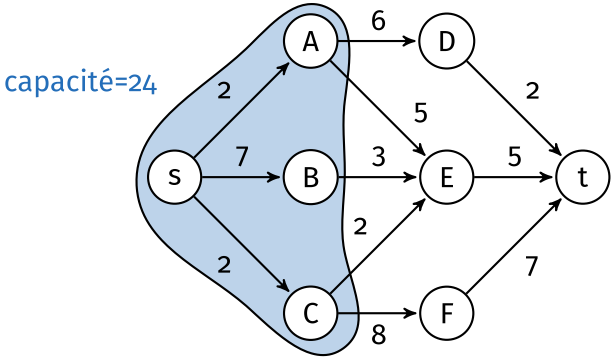
capacité=16



Coupe

Définition

Une **coupe** est un ensemble de sommets contenant s et pas p. La **capacité d'une coupe** est égale à la somme des arcs sortant de cette coupe.



Coupe minimale

Définition

Une coupe est dite **minimale** si elle a la plus petite capacité parmi toutes les coupes possibles.

Flot max, Coupe min

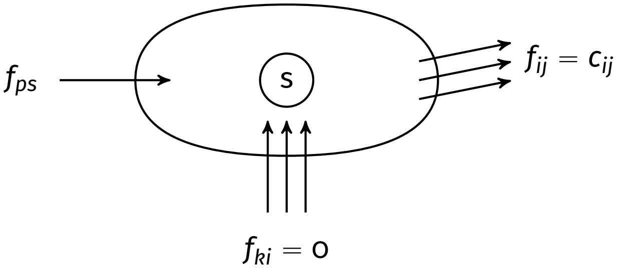
Théorème [Ford-Fulkerson]

La valeur d'un **flot maximal** est **égale** à la valeur d'une **coupe minimale**.

max flot $\leqslant$ min coupe : Soit f un flot quelconque, et Y une coupe quelconque. L'arc (p, s) entre dans la coupe Y, donc f_{ps} est inférieur au flot entrant en Y. Or ce flot entrant est égal au flot sortant, qui est inférieur à la capacité de la coupe, d'où $f_{ps} \leqslant C(Y)$. Comme f et Y sont quelconques, on a en particulier $\max f_{ps} \leqslant \min C(Y)$.

Flot max, Coupe min

max flot $\geqslant$ min coupe : Soit M la coupe formée des sommets marqués quand FF se termine, et f^* le flot résultant. On sait alors que tous les arcs sortants de M sont tels que $f^*_{ij} = c_{ij}$ et tous les arcs entrants hormis (p, s) sont tels que $f^*_{ij} = 0$. Kirchoff nous donne donc $f_{ps} = C(M)$. Or $\max f_{ps} \geqslant f^*_{ps}$ et $\min C(Y) \leqslant C(M)$, d'où $\max f_{ps} \geqslant \min C(Y)$.



Corollaires

Corollaire

Un flot est **maximal** si et seulement si il n’existe **pas de chaîne améliorante**.

Corollaire

L’algorithme de **Ford-Fulkerson** construit un **flot maximal**.

Graphe d'écart

Définition

Soit $G = (X, U, c)$ un réseau de transport et f un flot sur G. On appelle graphe d'écart et on note $G_e(f)$ le réseau de transport formé comme suit :

- G_e a les mêmes sommets X que G
- $\forall (i, j) \in U$, on ajoute un arc (i, j) de capacité $c_{ij} - f_{ij}$
- $\forall (i, j) \in U$, on ajoute un arc (j, i) de capacité f_{ij}
- les arcs de capacité nulle sont omis dans G_e

Algorithme du graphe d'écart

Algorithme : Algorithme du graphe d'écarts

tant que il existe un chemin c **allant de s à p dans** $G_e(f)$ **faire**

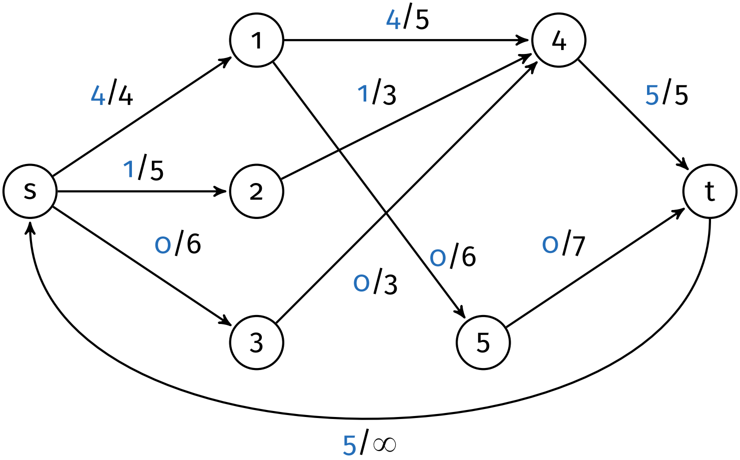
  Améliorer f en saturant le chemin c

  Recalculer $G_e(f)$

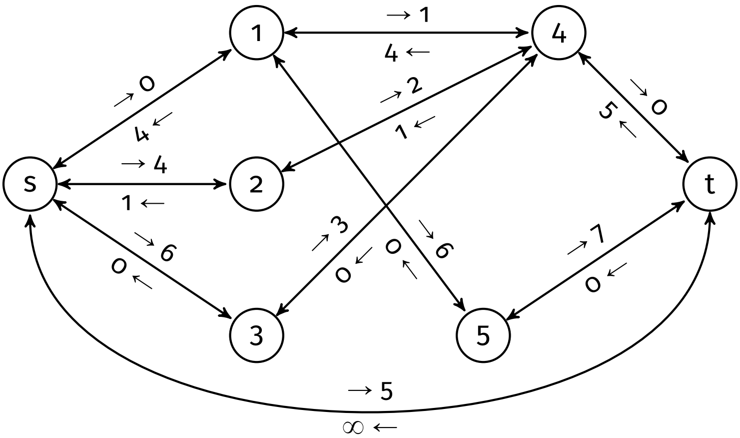
Remarque

chaîne améliorante dans G = chemin améliorant dans G_e

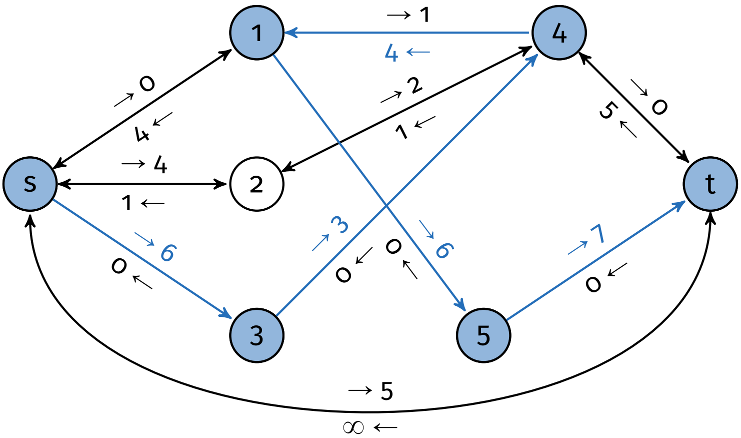
Exemple



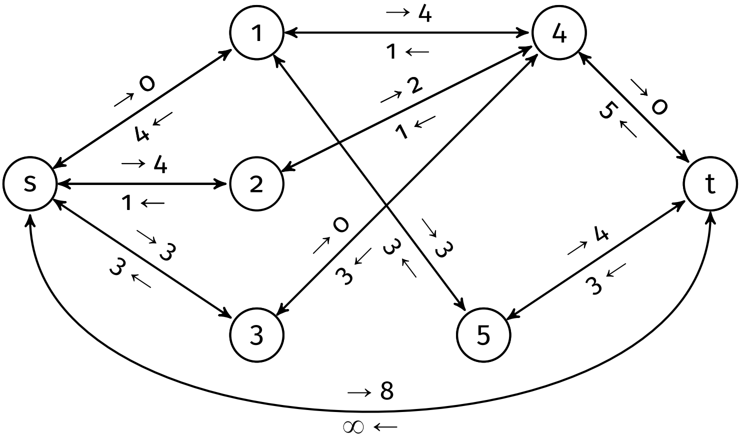
Exemple



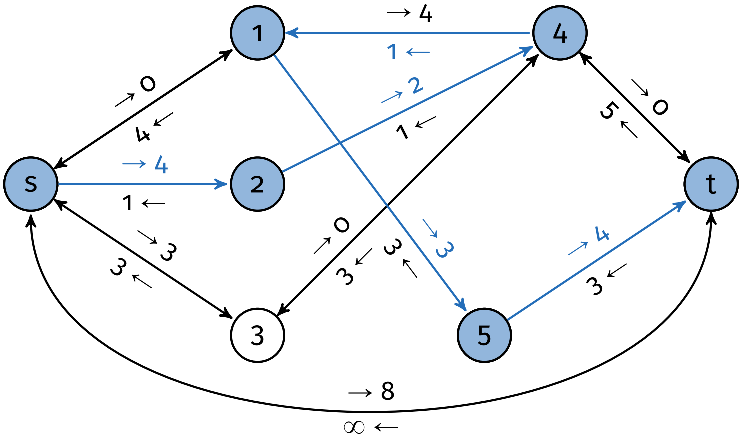
Exemple



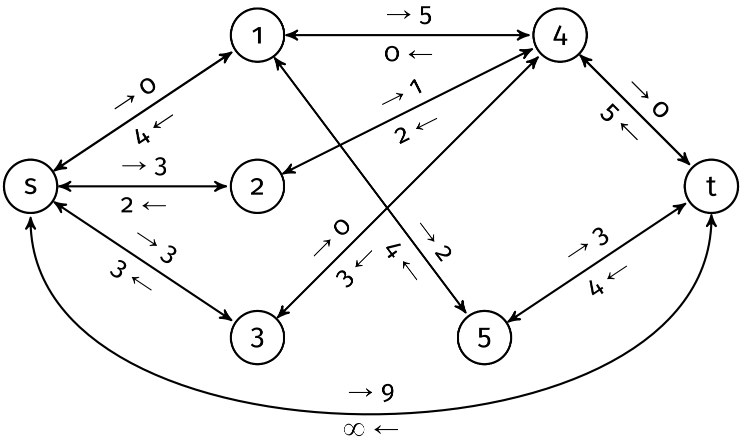
Exemple



Exemple



Exemple



Flot maximal à coût minimal

Flot maximal à coût minimal

On suppose désormais que les arcs (i, j) du réseau de transport sont également dotés d'un **coût unitaire** d_{ij}.

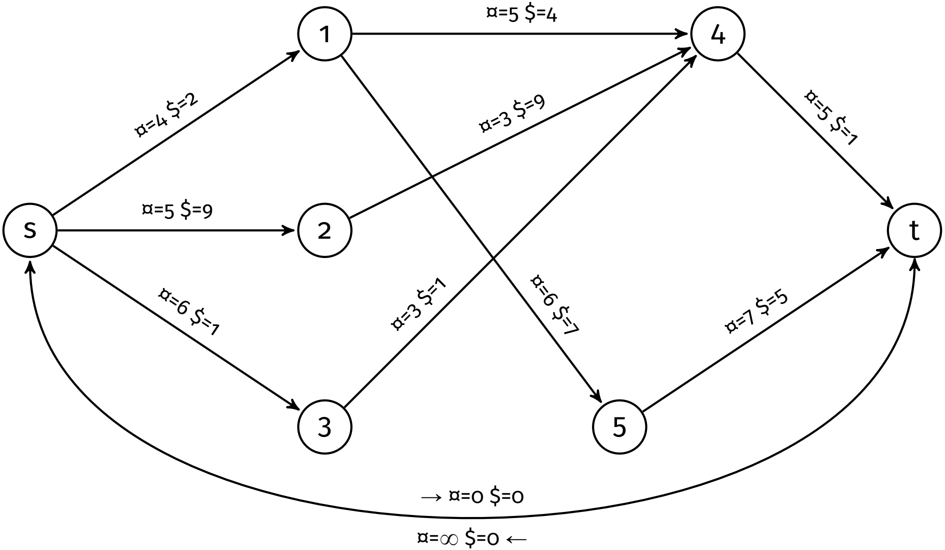
On cherche alors un **flot maximal à coût minimal** (*i.e.* qui minimise $\sum_{(i,j)\in U} d_{ij}f_{ij}$).

Algorithme de Roy

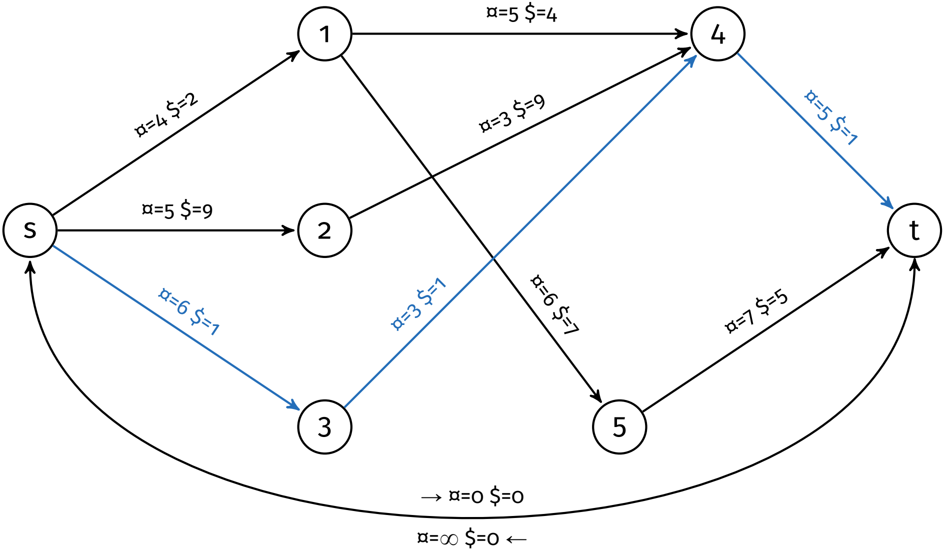
Algorithme : Algorithme de Roy

$f \leftarrow 0$
Construire $G_e(f)$
tant que il existe un chemin de s à p dans $G_e(f)$ **faire**
    Calculer un pcc (selon les coûts) C entre s et p dans $G_e(f)$
    Améliorer f en saturant C
    Reconstruire $G_e(f)$

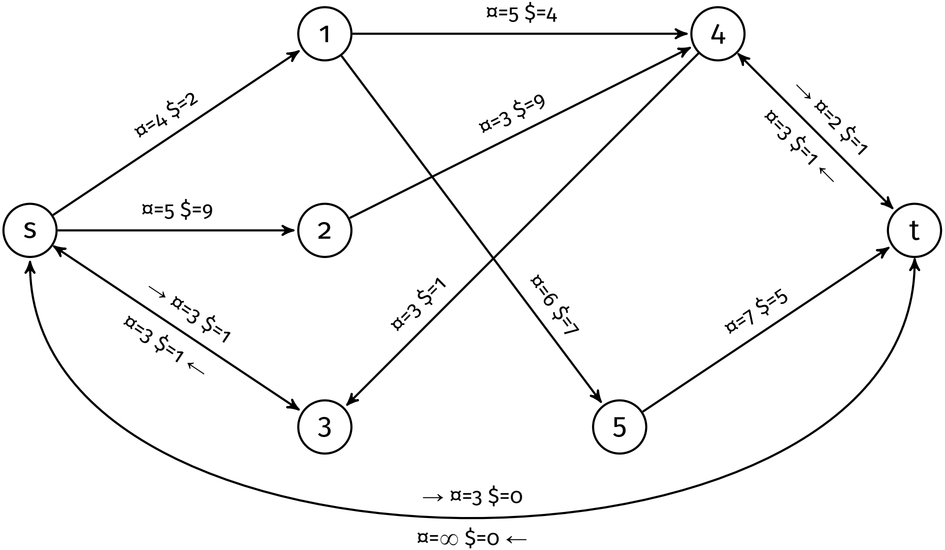
Exemple



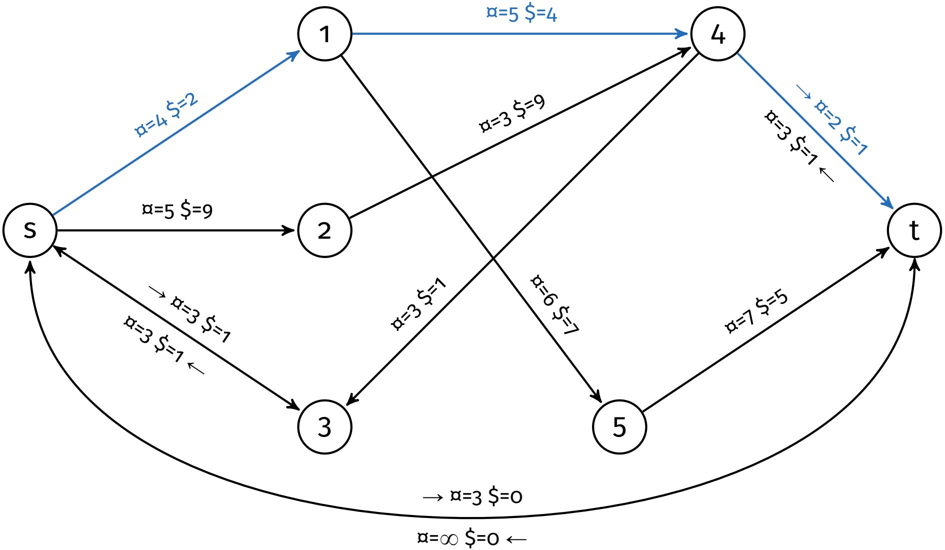
Exemple

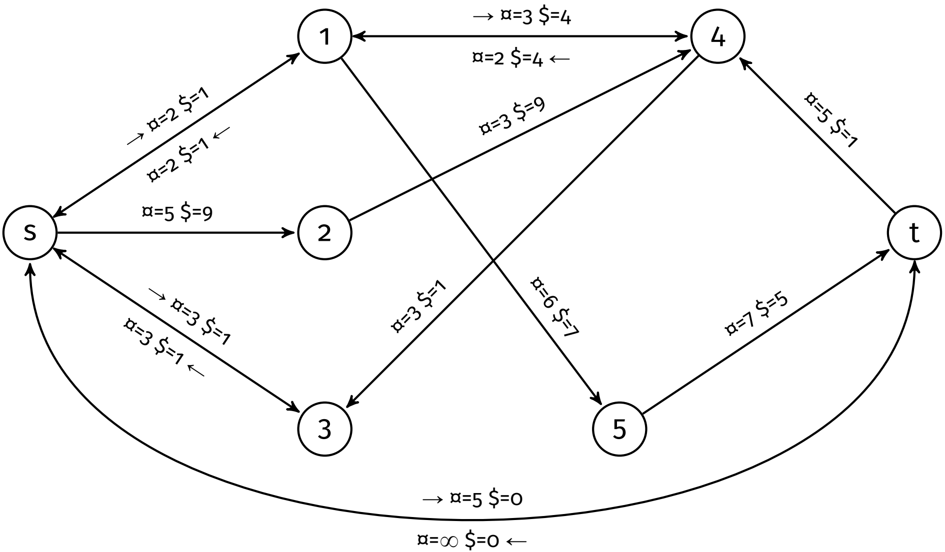


Exemple

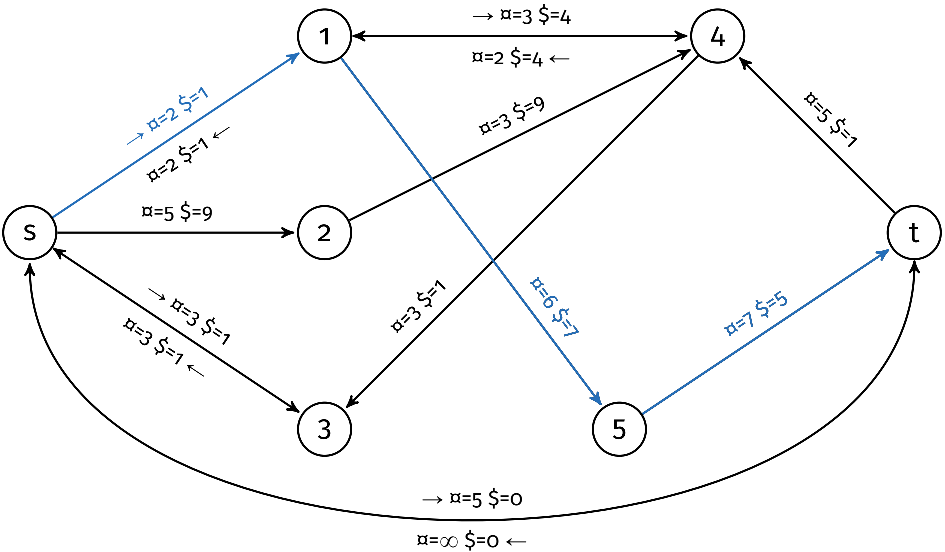


Exemple

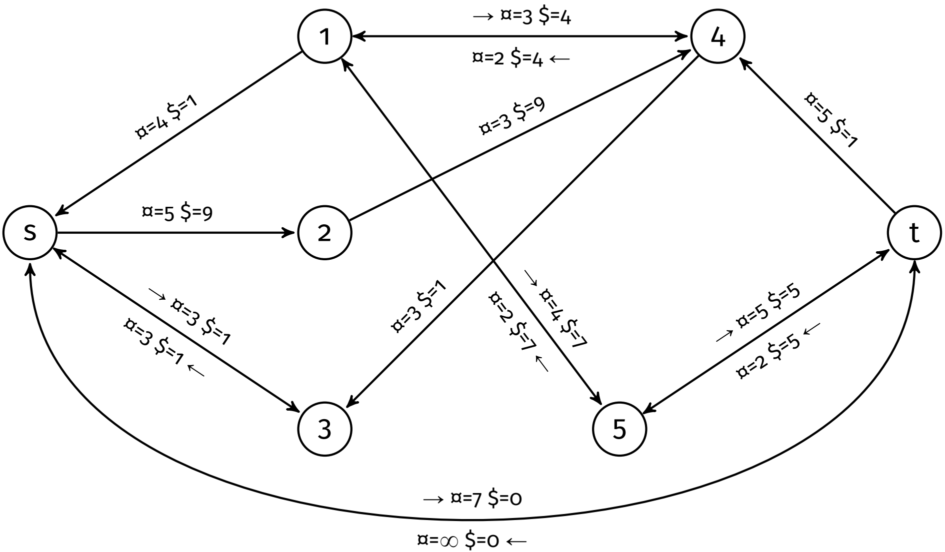



→ ¤=3 $=4
¤=2 $=4 ←
→ ¤=2 $=1
¤=2 $=1 ←
¤=3 $=9
¤=5 $=1
¤=5 $=9
→ ¤=3 $=1
¤=3 $=1 ←
¤=3 $=1
¤=6 $=7
¤=7 $=5
→ ¤=5 $=0
¤=∞ $=0 ←
1
2
3
4
5
s
t

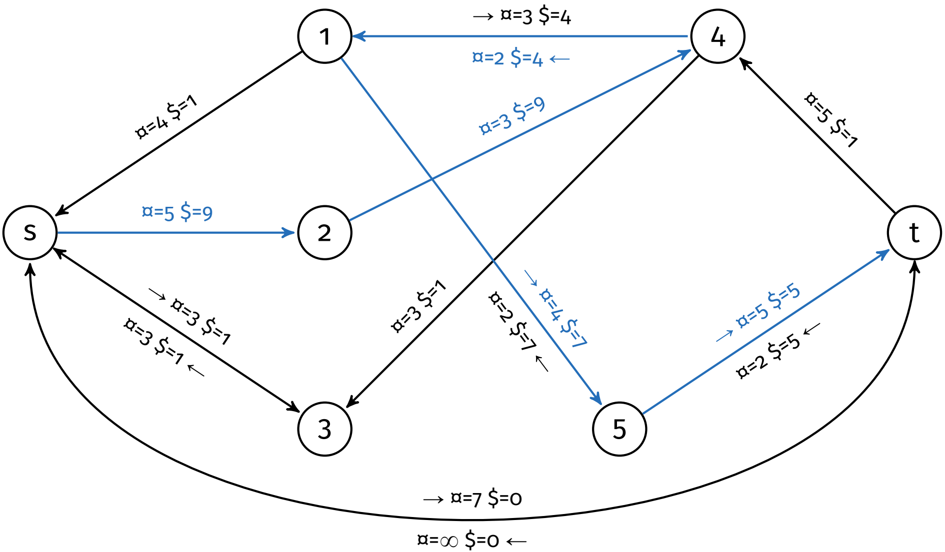
Exemple



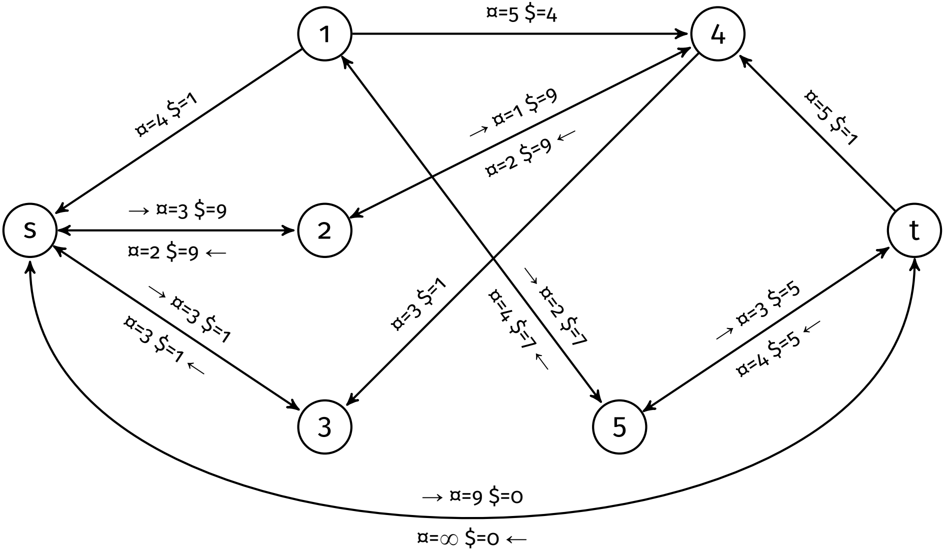
Exemple



Exemple



Exemple



Flot canalisé

Flot canalisé :

Définition

Une **borne** b_{ij} est une valeur minimale que doit respecter tout flot sur un arc (i, j).

Définition

Un **flot canalisé** sur un réseau $G = (X, U, b, c)$ est un flot qui satisfait en plus les contraintes de bornes en chaque arc.

Flot canalisé : théorème d'existence

Théorème [Hoffman]

Il existe un flot canalisé sur un réseau $G = (X, U, b, c)$ si et seulement si pour tout ensemble de sommets Y, la somme des bornes entrantes en Y est inférieure ou égale à la somme des capacités sortantes de Y, soit :

$$\forall Y \subset X, \sum_{u \in U^-(Y)} b(u) \leqslant \sum_{u \in U^+(Y)} c(u)$$

Flot canalisé : algorithme de recherche

Algorithme : Recherche d'un flot canalisé

$f \leftarrow 0$

tant que il existe un arc (i, j) **tel que** $f_{ij} < b_{ij}$ **faire**

  Poser $s \leftarrow i$ et $p \leftarrow j$

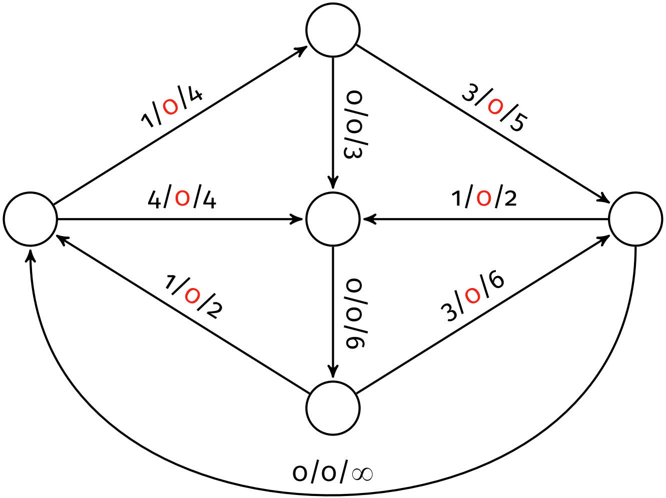
  Chercher une chaîne améliorante C de s à p avec les arcs dans le sens positif insaturés et les arcs dans le sens négatif avec un flot strict. supérieur à la borne.

  si une telle chaîne n'existe pas alors Terminer avec FIN2;

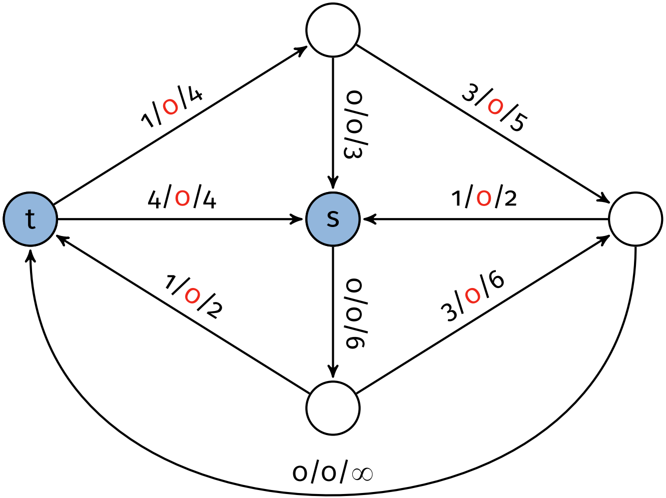
  Améliorer F avec C

Terminer avec FIN1

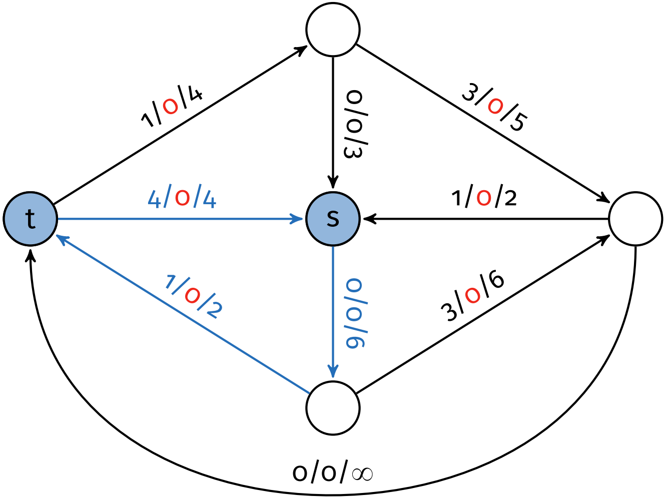
Exemple



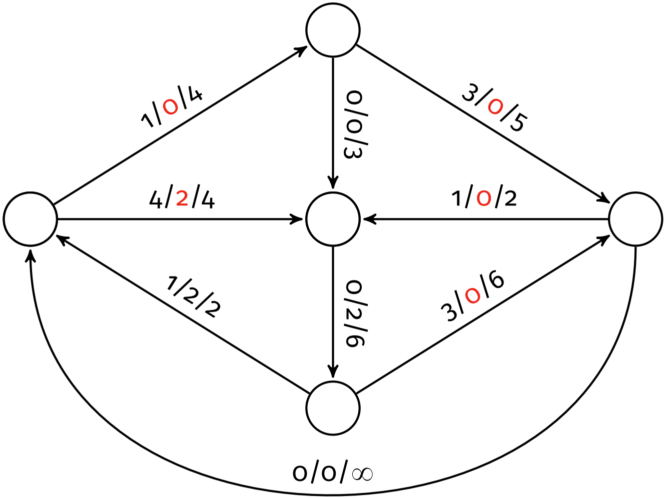
Exemple



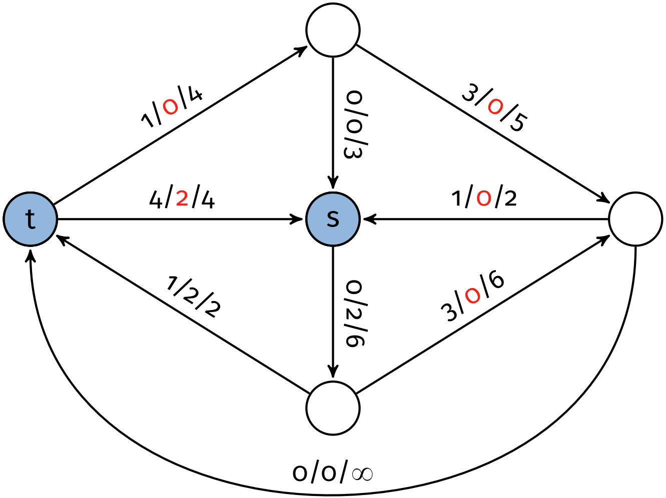
Exemple



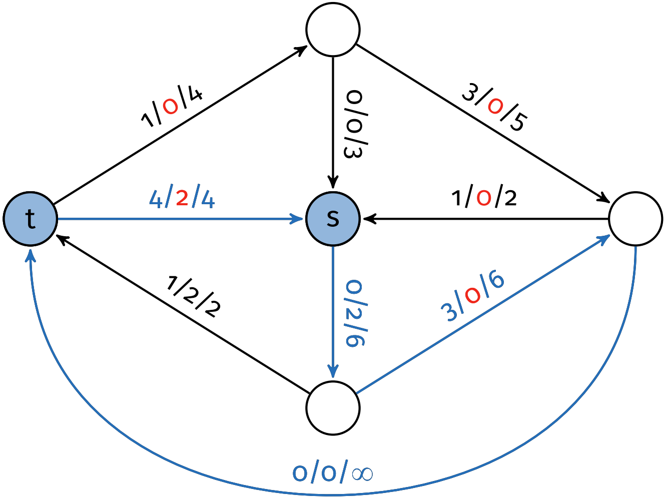
Exemple



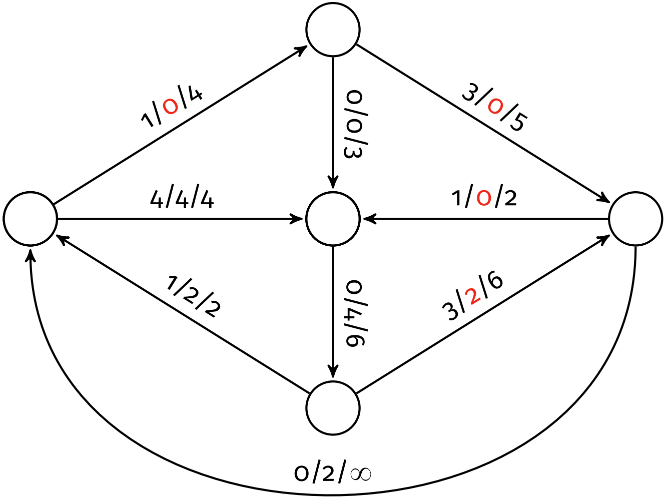
Exemple



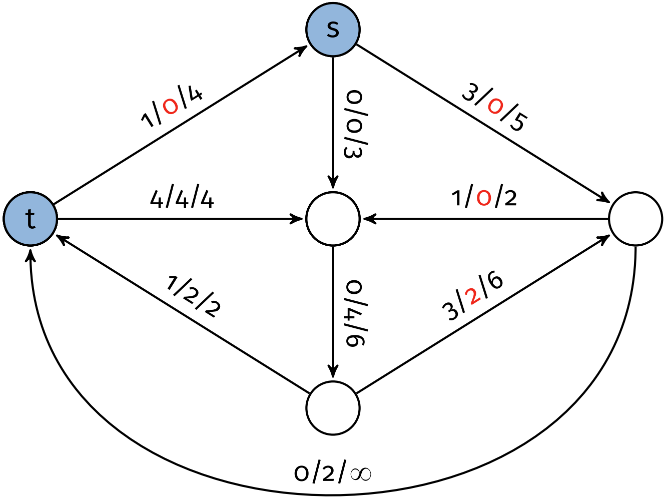
Exemple



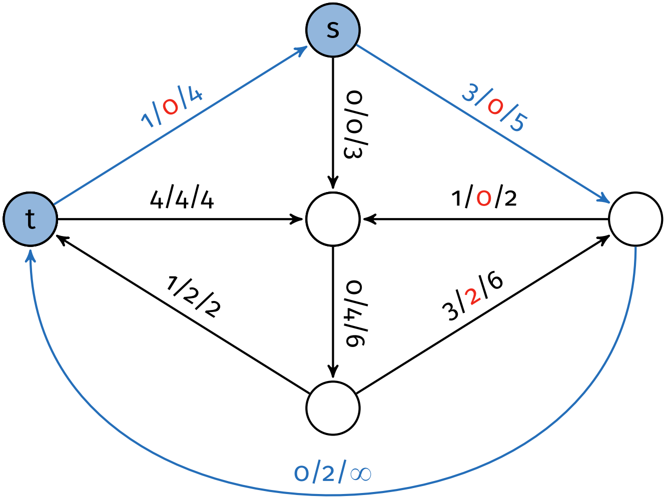
Exemple



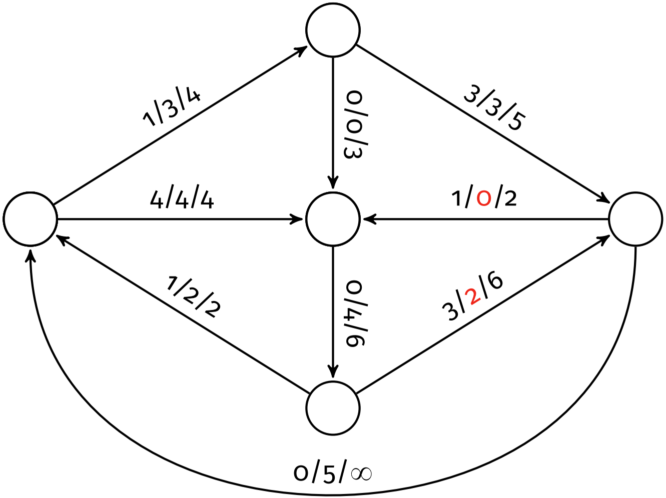
Exemple



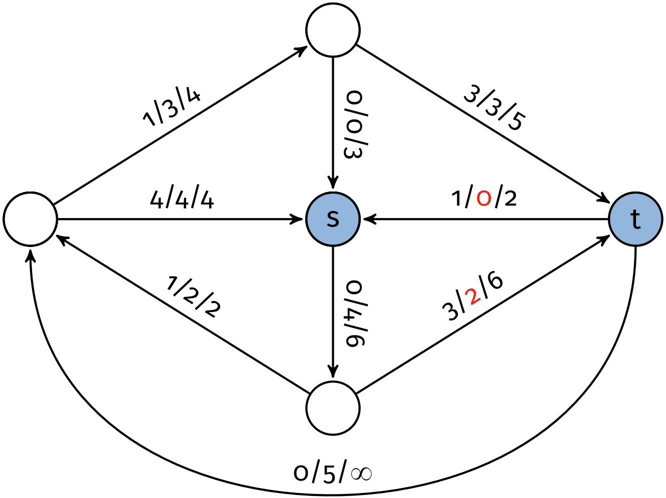
Exemple



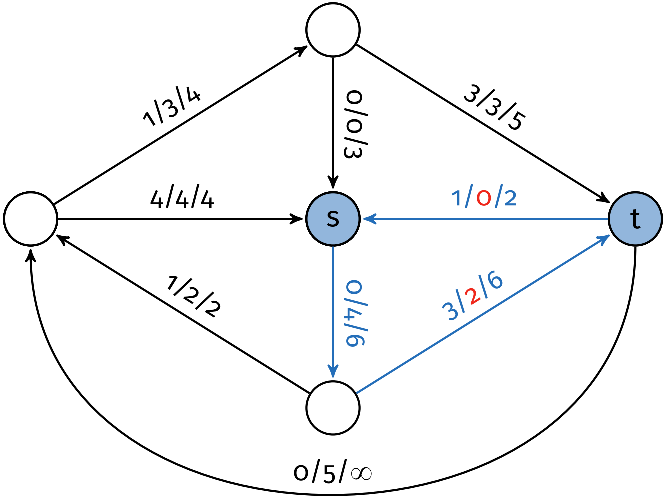
Exemple



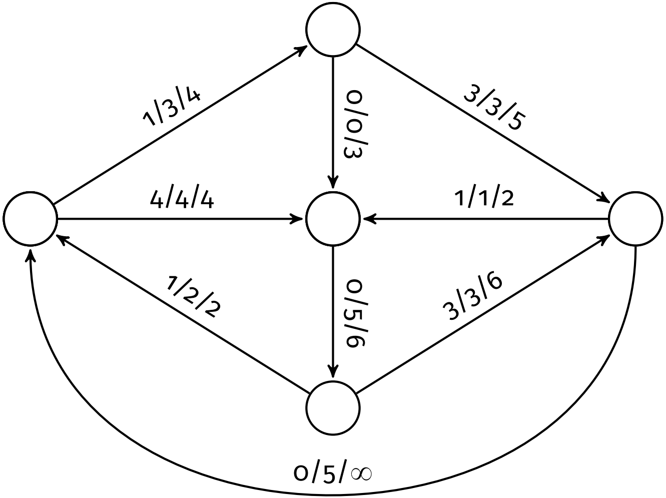
Exemple



Exemple



Exemple



Flot canalisé à coût minimal

Décomposition d'un flot

Théorème

Pour tout flot f positif, il existe une famille de circuits $\mu_1, \mu_2, \ldots, \mu_p$ et des entiers positifs $\lambda_1, \lambda_2, \ldots, \lambda_p$ tels que :

$$f = \lambda_1 \mu_1 + \lambda_2 \mu_2 + \ldots + \lambda_p \mu_p$$

Preuve : On choisit un arc avec un flot positif. Kirchoff nous indique qu'il y a à la suite de cet arc un autre arc nécessairement positif. De proche en proche, on peut ainsi construire un circuit. On lui affecte pour coefficient la plus grande valeur de flot qui n'excède pas les capacités résiduelles le long de ce circuit. On recommence jusqu'à atteindre un flot nul.

Graphe d'écart d'un flot canalisé

Définition

Soit $G = (X, U, c, b)$ un réseau de transport et f un flot sur G. On appelle graphe d'écart et on note $G_e(f)$ le réseau de transport formé comme suit :

- G_e a les mêmes sommets X que G
- $\forall (i, j) \in U$, on ajoute un arc (i, j) de capacité $c_{ij} - f_{ij}$
- $\forall (i, j) \in U$, on ajoute un arc (j, i) de capacité $f_{ij} - b_{ij}$
- les arcs de capacité nulle sont omis dans G_e

Amélioration d'un flot canalisé

Soit f_0 un flot canalisé et $G_e(f_0)$ le graphe d'écart associé. Soit φ un flot sur $G_e(f_0)$.

On définit un nouveau flot f canalisé (on note $f = f_0 \oplus \varphi$) en suivant la formule $(f)_u = (f_0)_u + (\varphi)_{u^+} + (\varphi)_{u^-}$.

Le coût de f est égal au coût de f_0 plus le coût de φ.

Théorème

Un **flot canalisé** f_0 est **de coût minimal** si et seulement si $G_e(f_0)$ ne contient pas de circuit de coût strictement négatif.

$\Rightarrow$ S'il existe un circuit de coût strictement négatif, alors on peut construire un flot strictement meilleur que f.

$\Leftarrow$ Soit $G_e(f_0)$ sans circuit négatif et f un autre flot canalisé. Alors il existe φ tel que $f = f_0 \oplus \varphi$, avec $\varphi = \lambda_1\mu_1 + \lambda_2\mu_2 + \ldots + \lambda_p\mu_p$. Comme $G_e(f_0)$ sans circuit négatif, tous les μ_j ont un coût positif donc le coût de φ est positif. Donc f_0 est de coût inférieur à tout autre flot, il est donc de coût minimal.

Retour sur l'algorithme de Roy

On peut désormais prouver que :

Théorème

L'algorithme de Roy détermine un flot maximal à coût minimal.

Idée de la preuve : On montre par récurrence que tous les flots f_k que construit l'algorithme sont de coût minimal parmi les flots de valeur v_k. Ainsi le dernier flot est bien maximal de coût minimal.