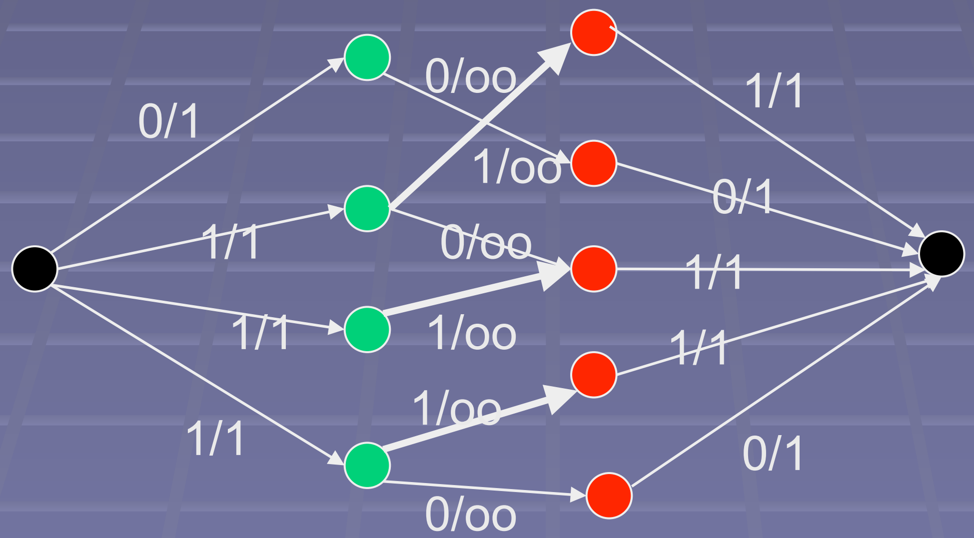
Problèmes de Flots, Problèmes de couplage, Problèmes d'affectation



Jean-Philippe Préaux

http://www.i2m.univ-mrs.fr/~preaux

Plan :

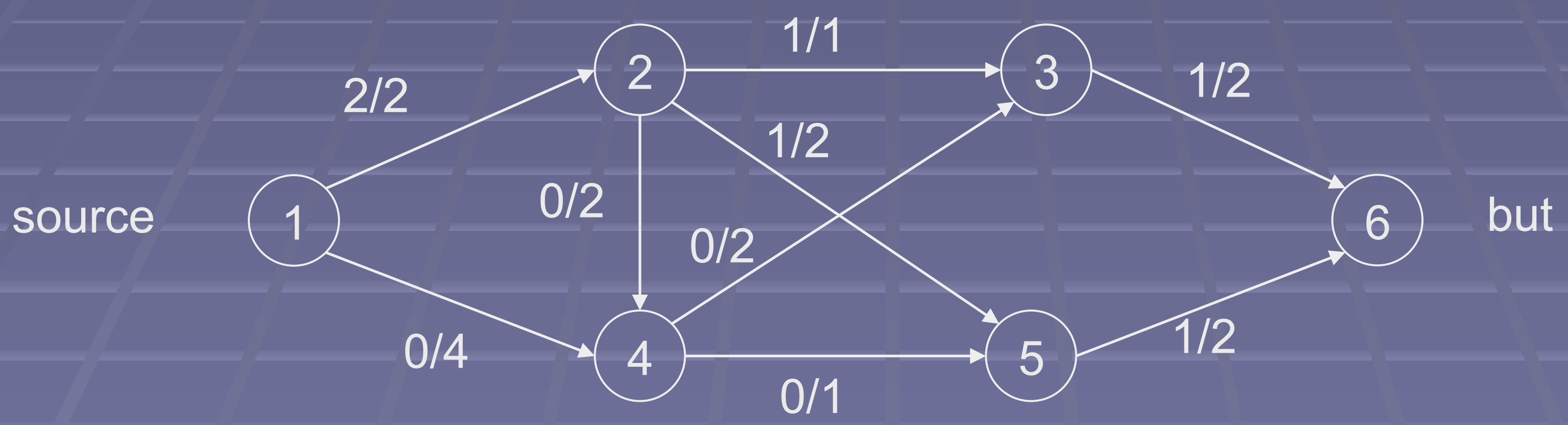
- Maximisation de flot : Algorithme de Ford-Fulkerson
- Problème de couplage
- Problème de mise en boîte
- Couplage dans un graphe biparti
- Couplage de coût minimal
- Maximisation de flot à coût : Algorithme de Busacker-Gowen
- Problème de Transport
- Affectation de coût minimal
- Méthode Hongroise

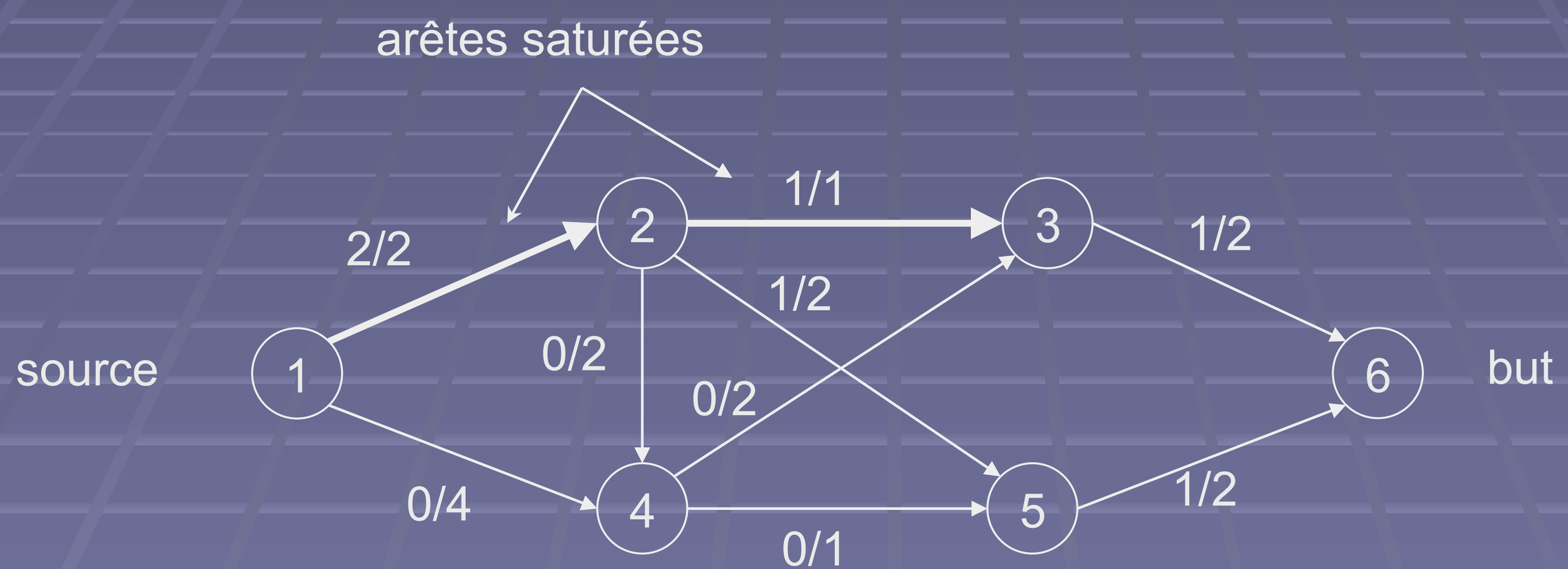
Problème de flot maximal

- **<u>Définition</u>** : Un <u>*réseau de transport*</u> est un graphe :
  - connexe et <u>orienté</u>,
  - sans cycle,
  - un unique sommet n'a pas de prédécesseur : c'est la <u>*source*</u>,
  - un unique sommet n'a pas de successeur, c'est le <u>*but*</u>,
  - chaque arête est bi-valuée : f / c ,
    avec f,c des entiers positifs, (*flot* et *capacité*)
    $c > 0$, et $0 \leq f \leq c$.
  - En tout sommet

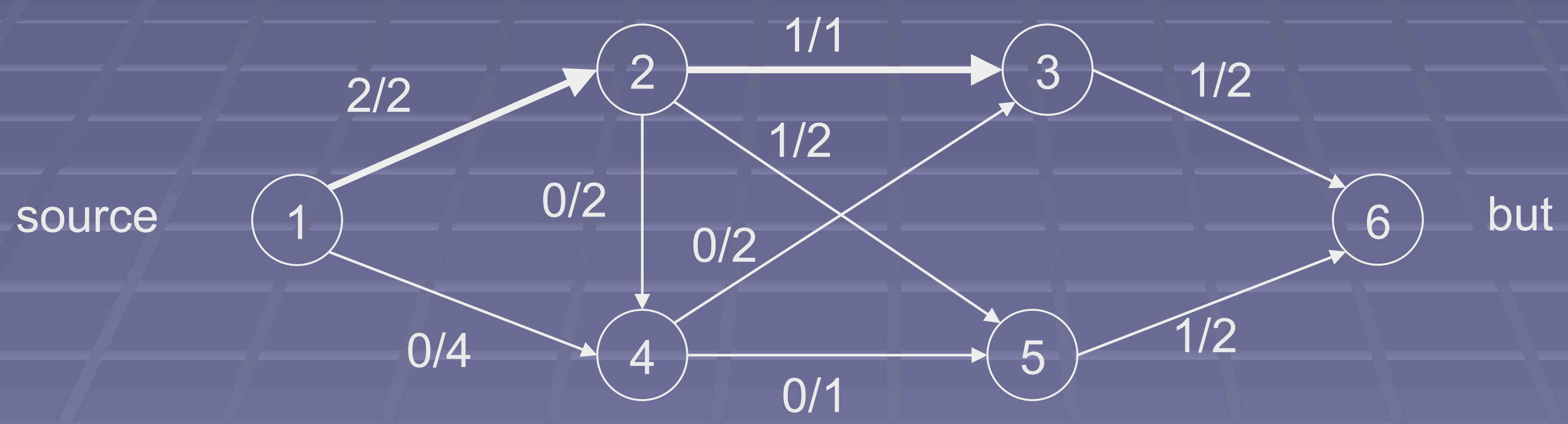
$$\sum \text{flots entrants} = \sum \text{flots sortants}$$

- Un réseau de transport peut être représenté ainsi (par exemple), aucune arête ne va de droite à gauche :



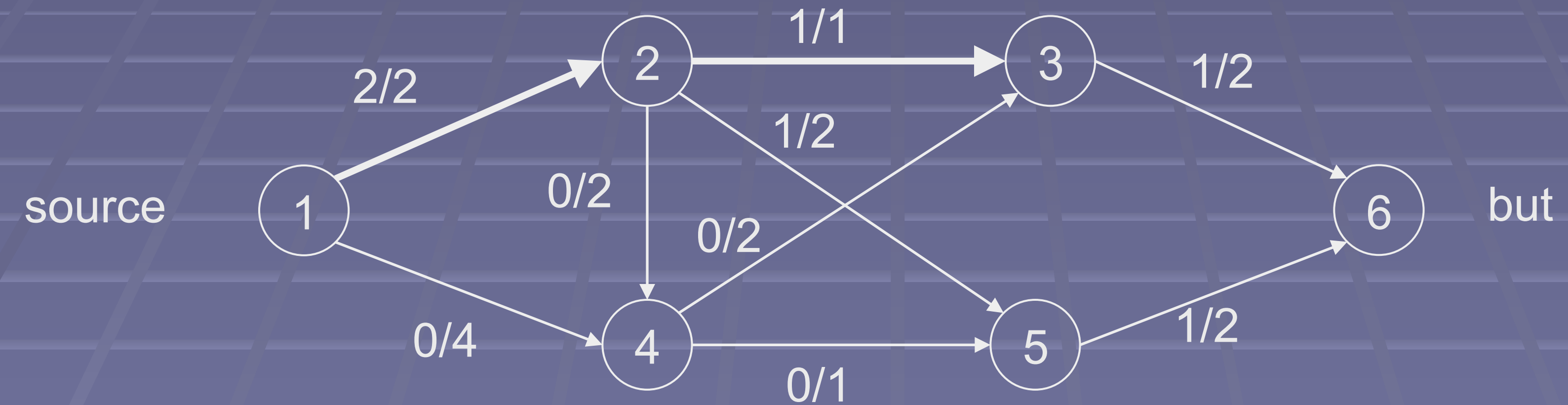


Une arête est *saturée* si flux=capacité. Sinon elle est *insaturée.*



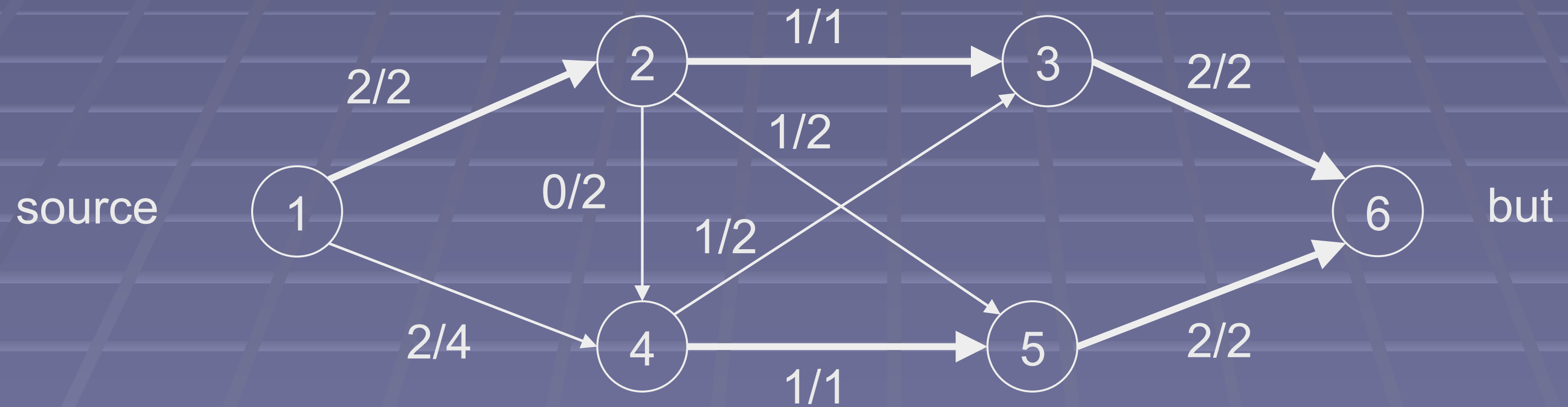
On peut facilement montrer que le *flux entrant* égale le *flux sortant.* (ici 2)

Le problème du flot maximal consiste à maximiser le débit.



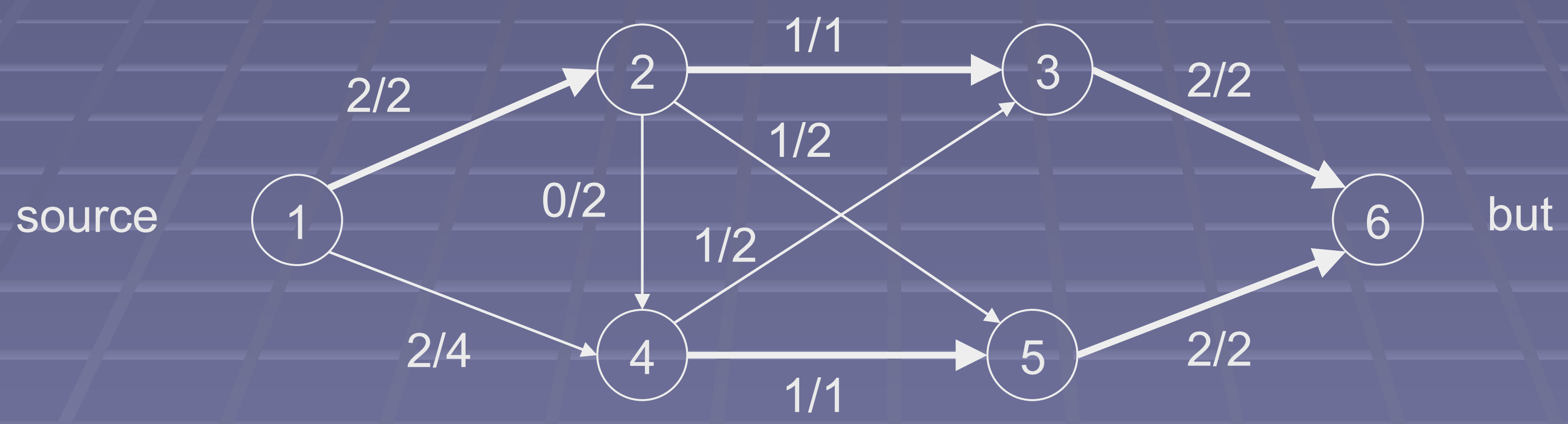
On peut facilement montrer que le *flux entrant* égale le *flux sortant.* (ici 2). C'est le *débit*.

Le problème du flot maximal consiste à maximiser le débit.



Par exemple, ici, le débit peut-être accru à 4.

Le problème du flot maximal consiste à maximiser le débit.



Attention !! il serait <u>trompeur</u> de croire que le flot est maximal dès que tout chemin de la source au but contient une arête saturée !!

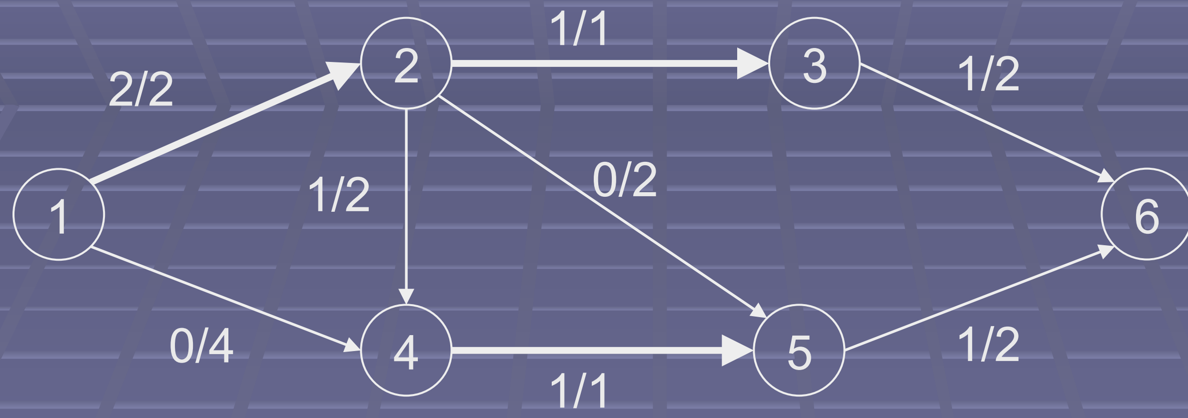
Graphe d'écart

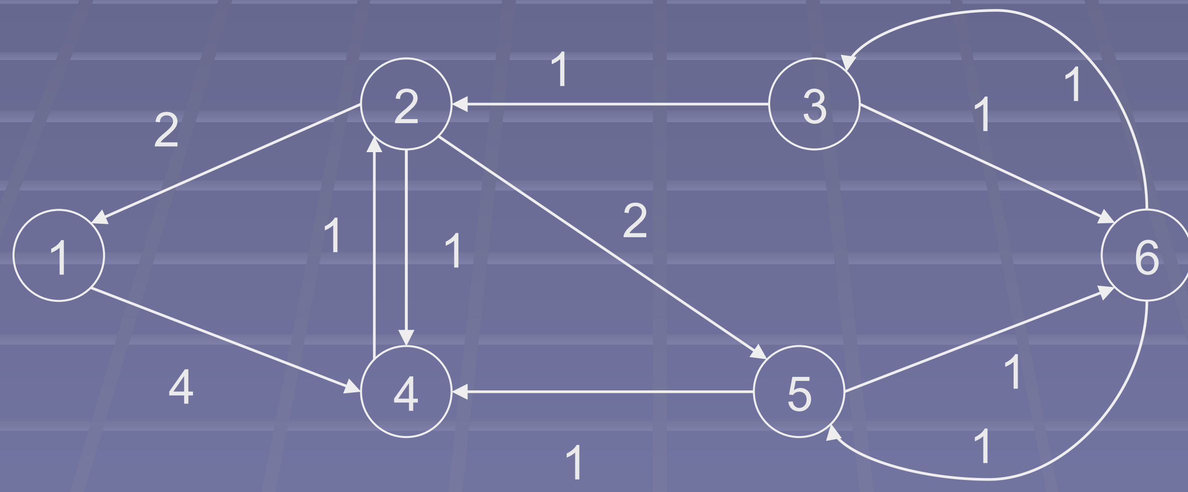
- <u>Définition :</u>

  Etant donné un réseau de transport *G*, son <u>*graphe d'écart*</u> a :

  - même sommets que *G*,
  - toute arête *e* insaturée de *G ;*
    valuée par $c(e) - f(e)$.
  - pour toute arête *e* de *i* à *j*, de flux non nul,
    l'arête $-e$ de *j* à *i*, valuée par $f(e)$.

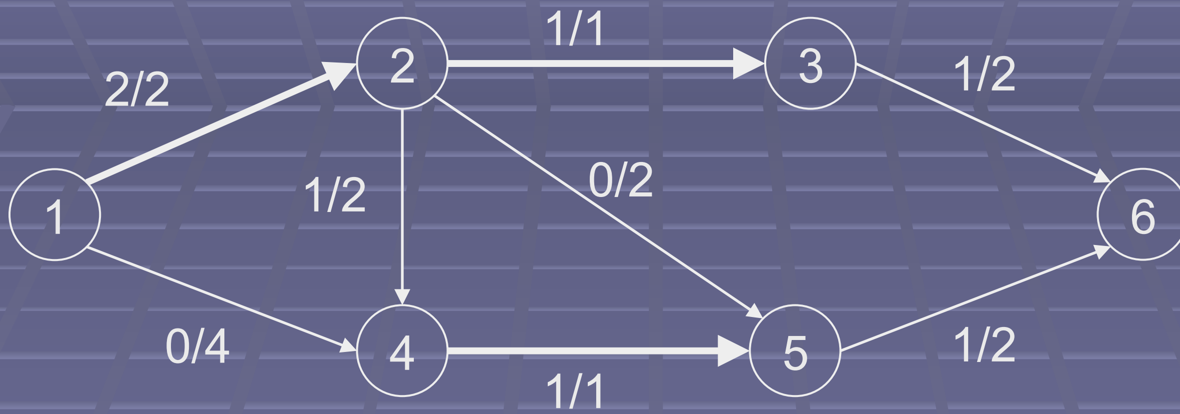
réseau de transport

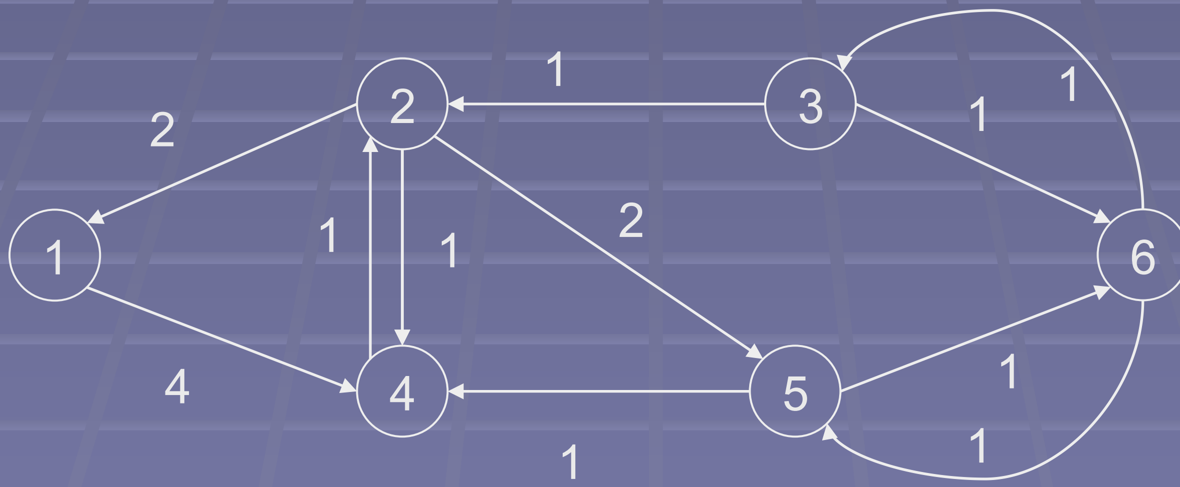




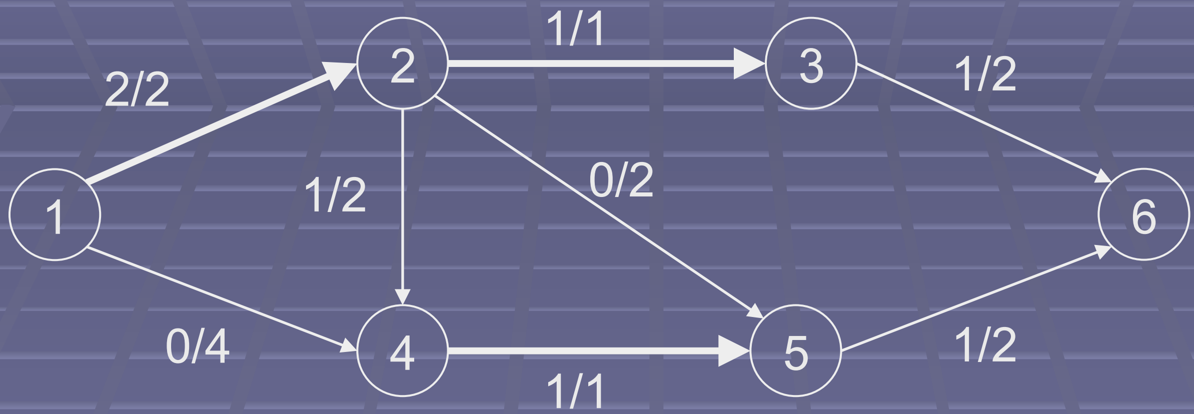
graphe d’ écart

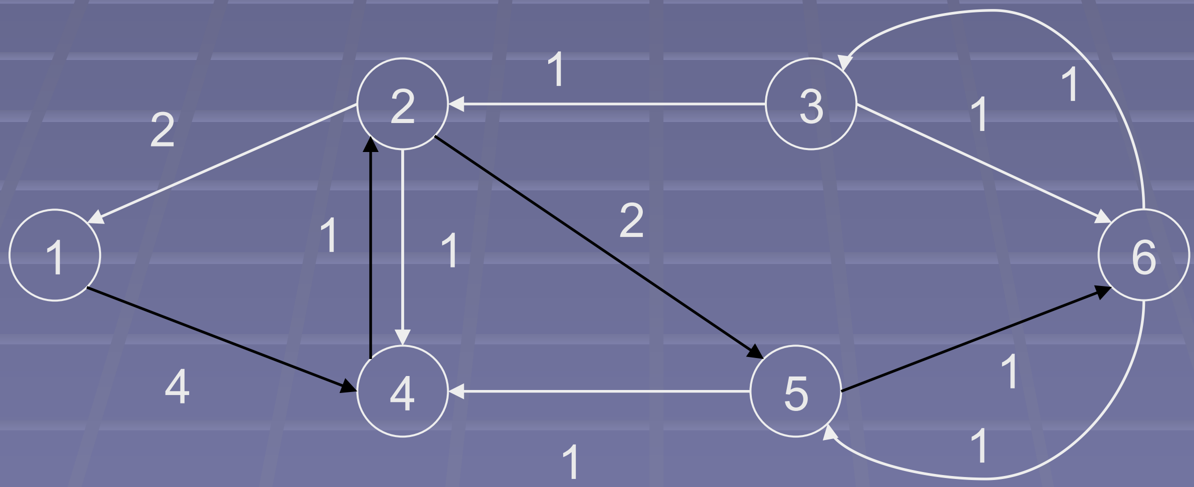
Le flot peut être augmenté si et seulement si le graphe d’écart contient un chemin de la source au but …





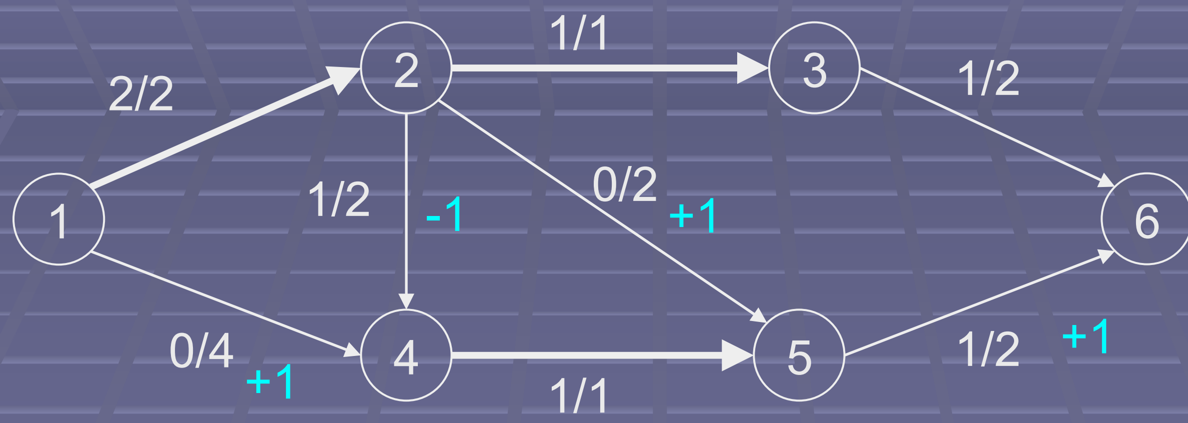
Le flot peut être augmenté si et seulement si le graphe d'écart contient un chemin de la source au but …

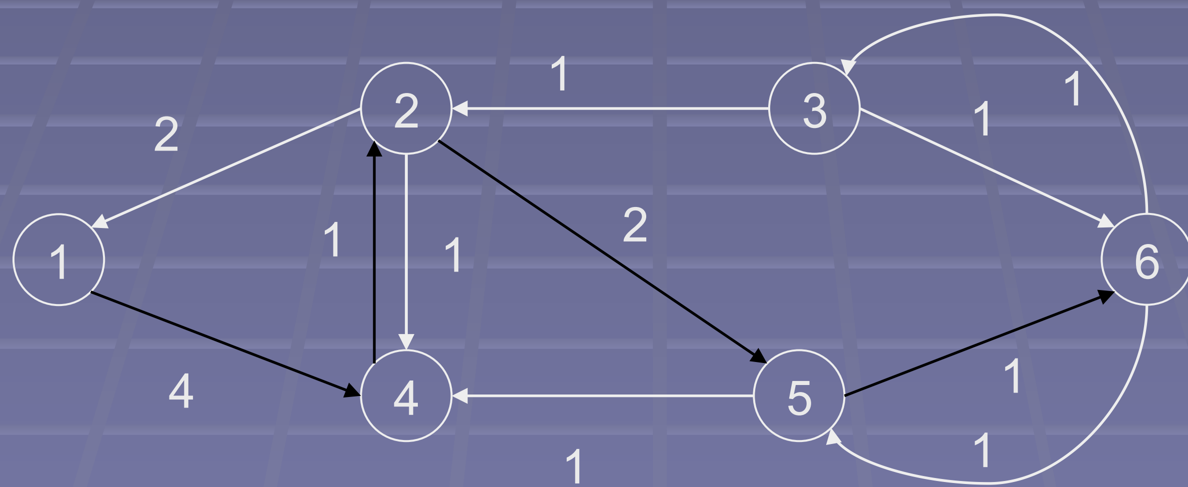




Le plus petit poids des arête du chemin est l'augmentation. Ici 1.

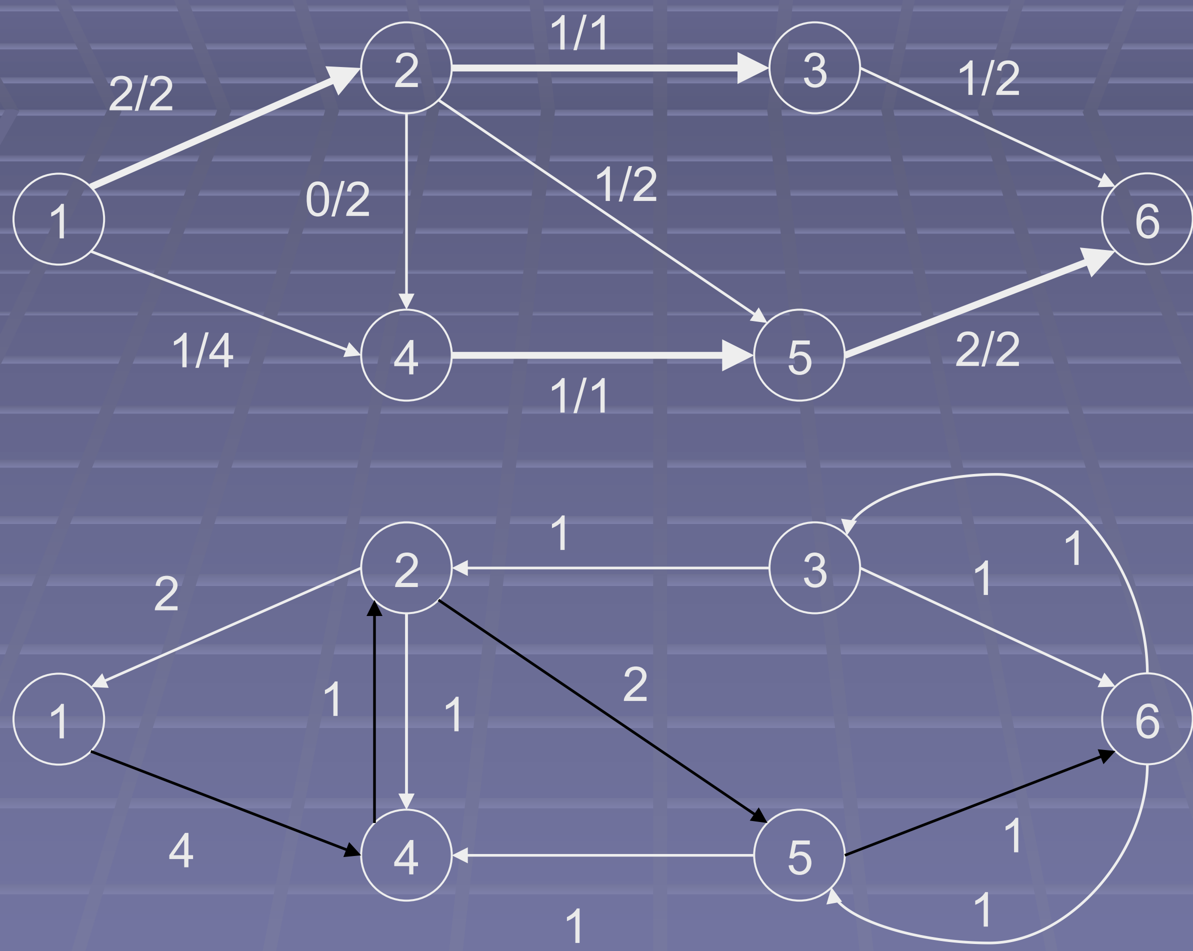
On l’ajoute au flot des arêtes ‘positives’, et on le retranche au flot des arêtes ‘négatives’…





Le plus petit poids des arête du chemin est l’augmentation. Ici 1.

On l’ajoute au flot des arêtes ‘positives’, et on le retranche au flot des arêtes ‘négatives’…



Le plus petit poids des arêtes du chemin est l’augmentation. Ici 1.

Algorithme de Ford-Fulkerson

- Répéter
  - Construire le graphe d'écart *Ge*,
  - Chercher dans *Ge* un chemin *C* de la source au but
  - Calculer l'augmentation $\Delta(C)$
  - Augmenter les arcs + de *C* dans *G* de $\Delta(C)$
  - Diminuer les arcs - de *C* dans *G* de $\Delta(C)$
- Tant que *Ge* contient un tel chemin.

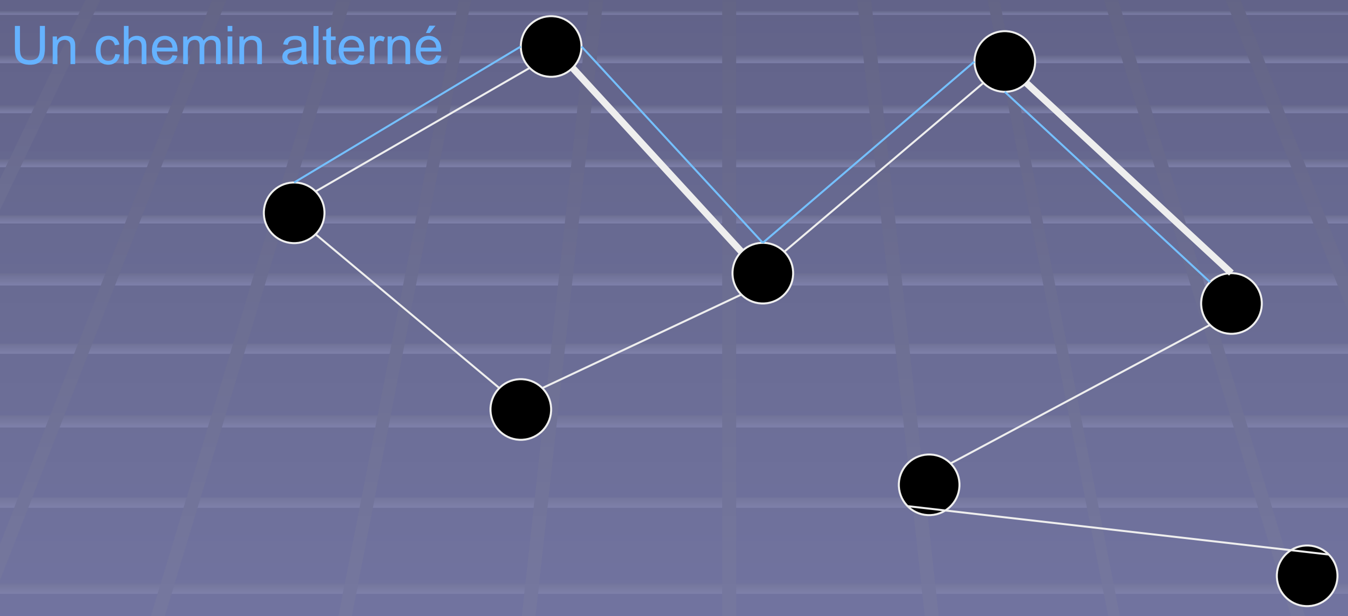
Dans la pratique on pourra ne pas construire *Ge* mais marquer + ou (et) – les sommets successifs en partant de la source (…).

Problèmes de couplage

- Dans un graphe simple, un *couplage* est un ensemble d'arêtes deux à deux non incidentes.
- Un couplage est *maximal* si son nombre d'arêtes est maximal (un couplage parfait est maximal)
- La recherche de couplage maximal est un problème générique auquel de nombreux problèmes peuvent se ramener.
- Par exemple : si un couplage parfait existe (comme dans le cas d'un graphe simple complet ayant $2n$ sommets), pour le trouver il suffit de trouver un couplage maximal.

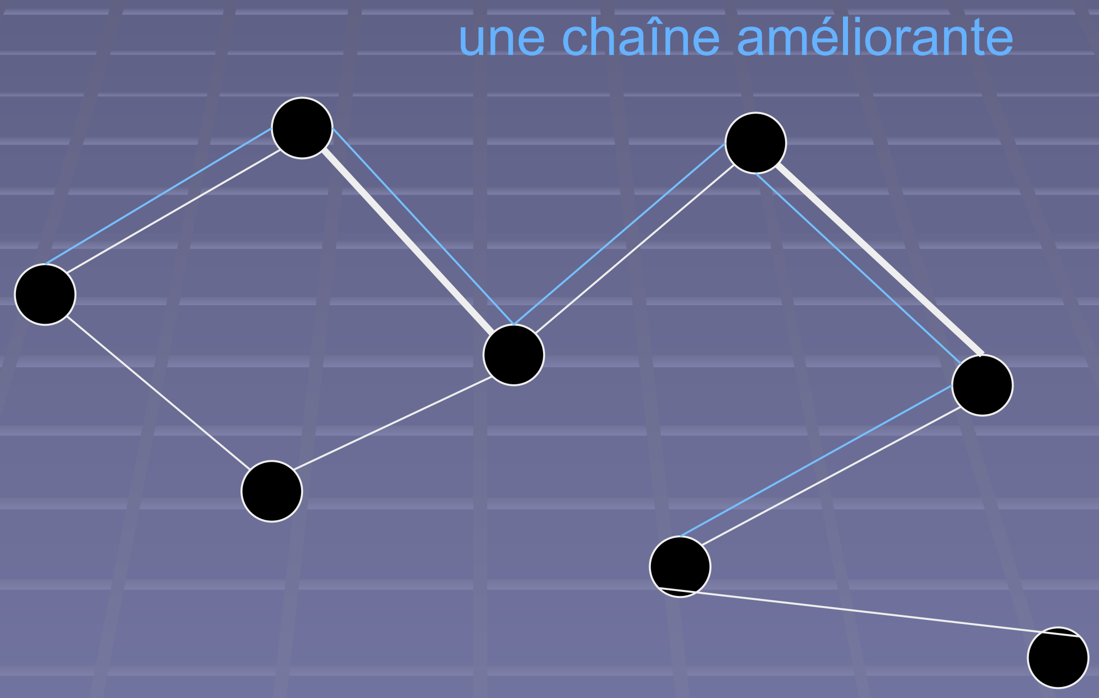
A rapprocher du problème du camion d'épandage,

- Pour un couplage *C*, un chemin *U* est <u>*alterné*</u> s'il emprunte alternativement des arêtes dans *C* et hors de *C*.

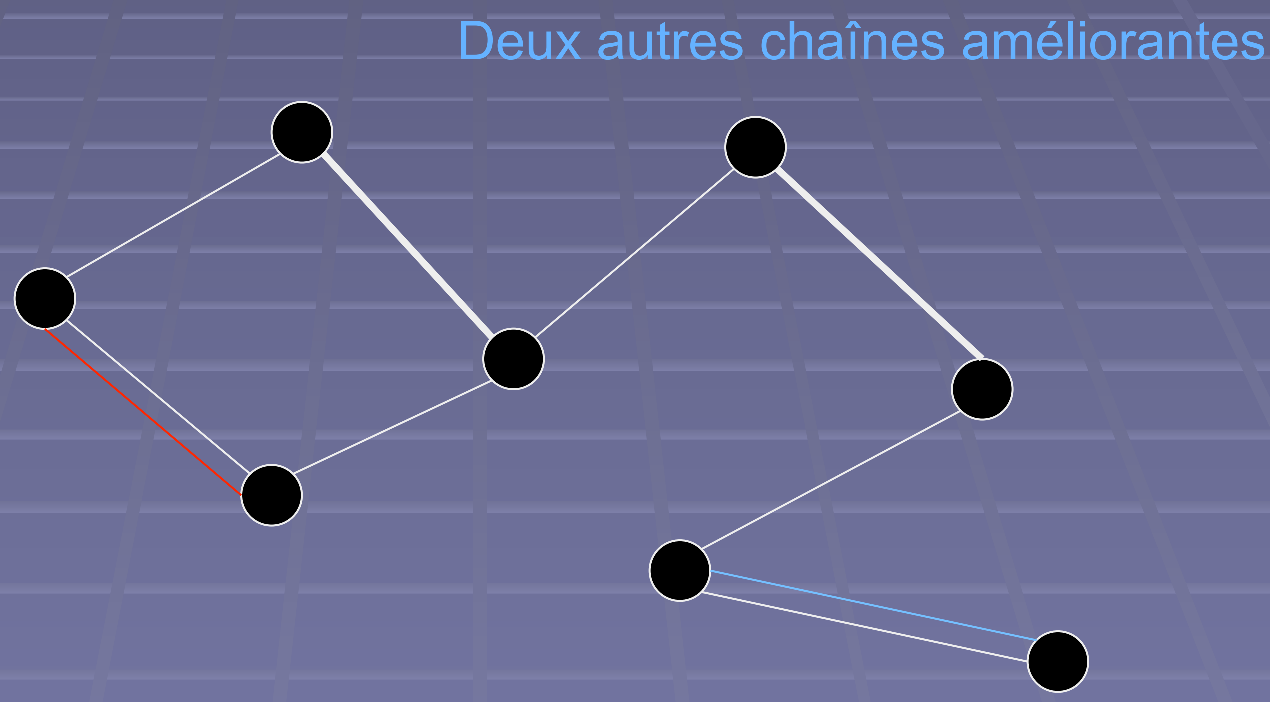


Par définition tout chemin réduit à une arête est alterné…

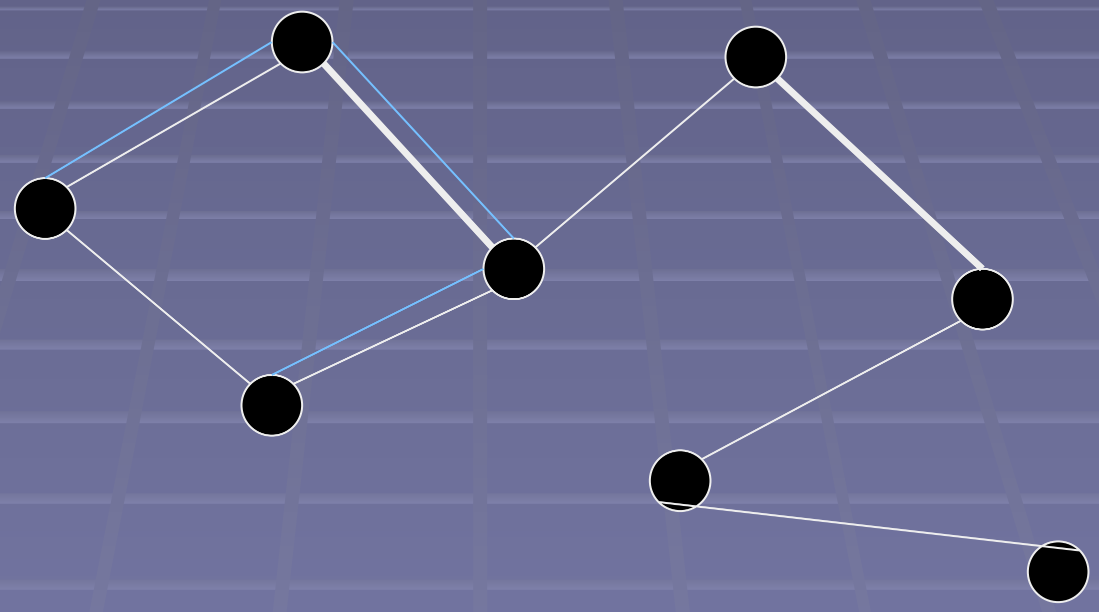
- Une <u>*chaîne améliorante*</u> est un chemin alterné dont les deux sommets extrémités sont 'hors de *C*'.



- Une <u>*chaîne améliorante*</u> est un chemin alterné dont les deux sommets extrémités sont ‘hors de *C*’.

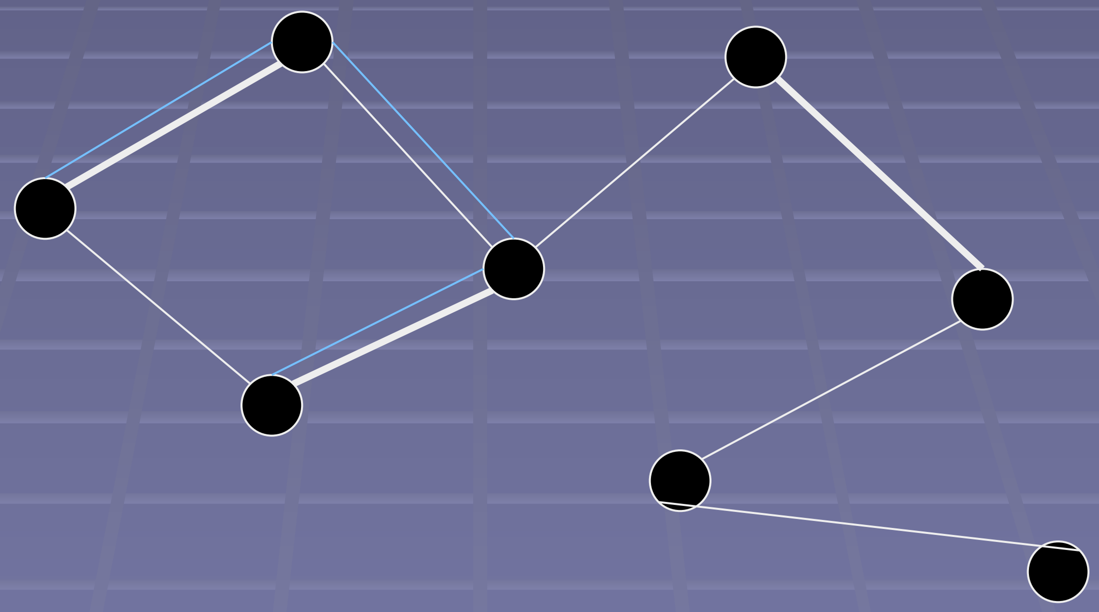


- Une <u>*chaîne améliorante*</u> est un chemin alterné dont les deux sommets extrémités sont ‘hors de *C*’.



La dernière… Il n’y en a pas d’autres

- Si *U* est une chaîne améliorante, un <u>*transfert*</u> de *C* par *U* consiste à retirer de *C* ses arêtes qui sont dans *U* à rajouter les arêtes de *U* qui ne sont pas dans *C*.



On montre facilement que l'on a construit un nouveau couplage ; et que sa cardinalité a augmenté de +1.

Théorème :

- Un couplage est maximal si et seulement si il n'admet aucune chaîne améliorante.

- Un transfert de C est un nouveau couplage C' vérifiant : $\#C' = \#C + 1$.

- Ainsi déterminer un couplage maximal se ramène à la recherche de chaînes améliorantes.
- On commence initialement avec le couplage vide.
- Chercher une telle chaîne peut s'implémenter avec des algorithmes de recherches classiques; polynomiaux. On obtient un algorithme de complexité cubique.
- Dans la pratique, lorsque le graphe n'est pas trop 'gros' on peut facilement chercher des chaînes améliorantes 'à la main', en remarquant d'abord qu'une arête ne rencontrant pas C donne une chaîne améliorante

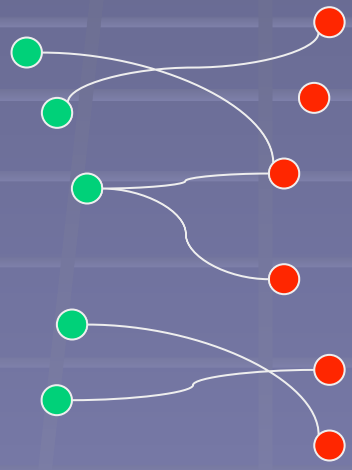
Exemple : Problème de 'mise en boîte'.

- On dispose en quantité abondante de sacs (divers conteneurs) tous de capacités $K > 0$.
- On a un ensemble d'objets chacun d'un poids inférieur à K.
- On veut ranger tous les objets dans des sacs en minimisant le nombre de sacs employés.
- On convient que :
  - On peut mettre autant d'objets que l'on désire dans un sac, tant que la somme de leur poids n'excède pas la capacité du sac.

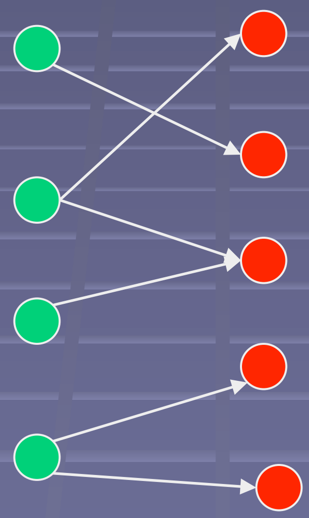
- En général il s'agit d'un problème NP-difficile.
- Si par contre on ne peut mettre qu'au plus 2 objets dans un même sac (si par exemple le poids de chaque objet est > *K*/3), alors il se ramène à un problème de couplage maximal
- Construire un graphe non orienté ayant un sommet pour chaque objet, et où une arête relie deux objets dès que la somme de leur poids n'excède pas *K*.
- Minimiser le nombre de sacs revient à trouver un couplage maximal de ce graphe.

Couplage dans un graphe biparti, et problème de flot.

- <u>*Définition :*</u> Un graphe est <u>*biparti*</u> si l'on peut partitionner ses sommets en deux ensembles S_1 et S_2, tels que chaque arête a une extrémité dans S_1 et l'autre dans S_2.

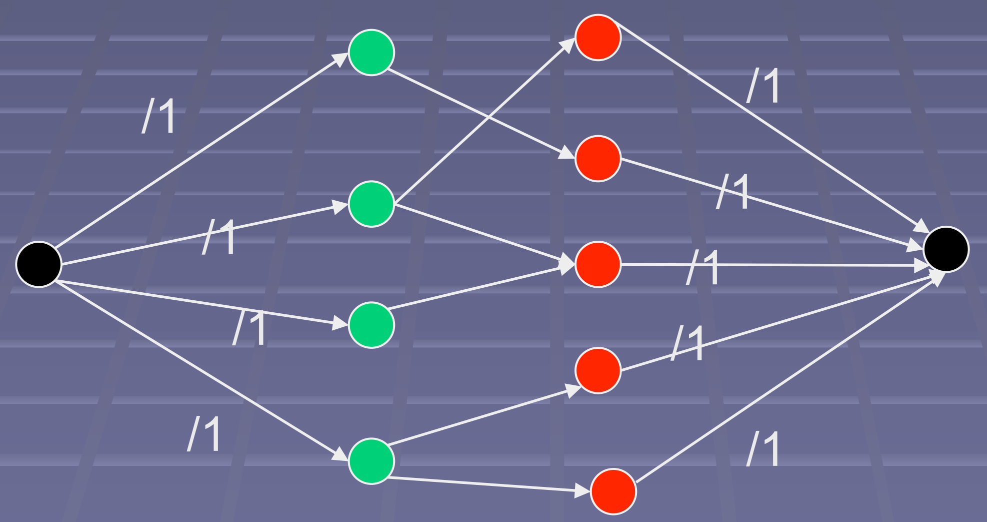


Un problème de couplage dans un graphe biparti se ramène à un problème de flot de la façon suivante :



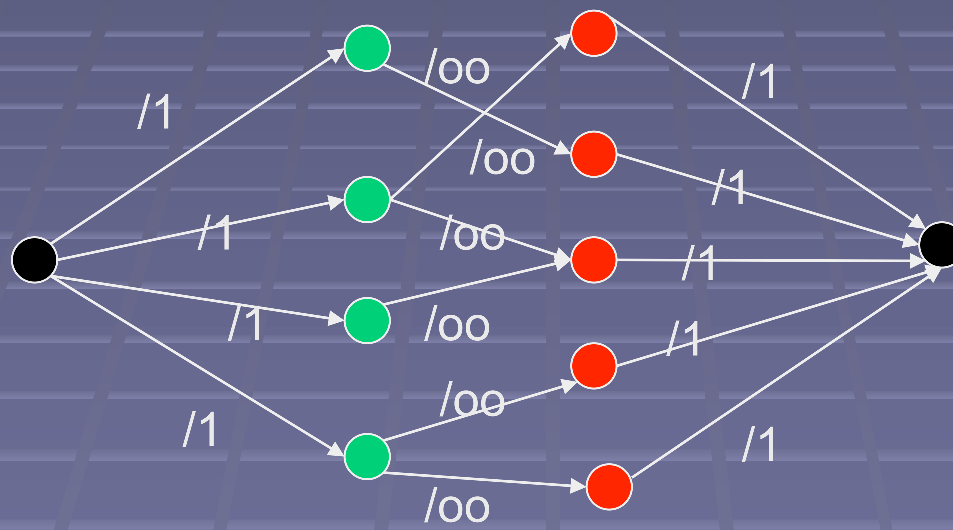
- On oriente toutes les arêtes de S_1 vers S_2.

Un problème de couplage dans un graphe biparti se ramène à un problème de flot de la façon suivante :



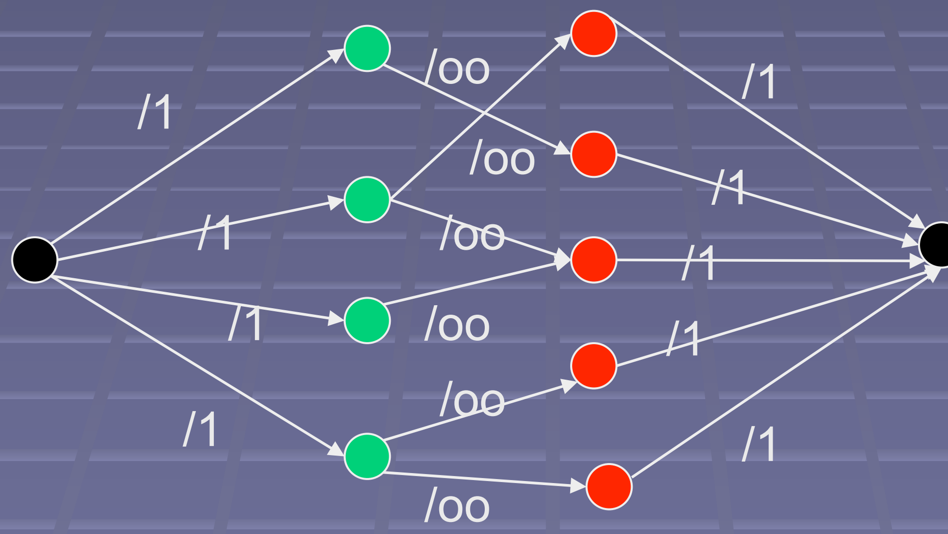
- On ajoute une source et des arêtes de capacité 1 la reliant aux sommets de S_1.
- On ajoute un but et des arêtes de capacité 1 qui relient les éléments de S_2 au but.

Un problème de couplage dans un graphe biparti se ramène à un problème de flot de la façon suivante :



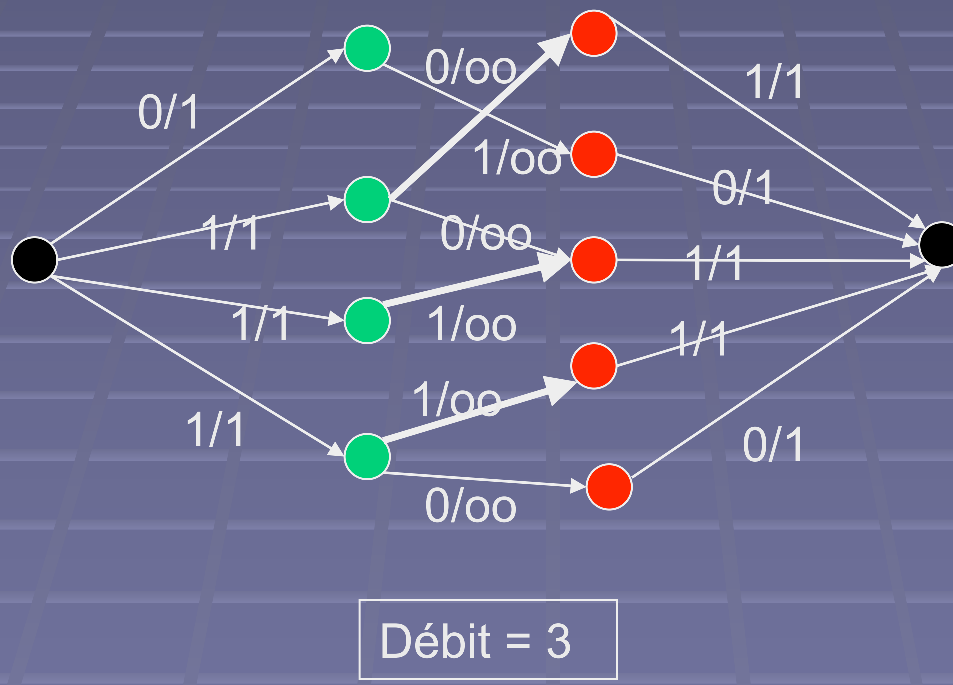
- On donne une capacité infinie aux arêtes reliant S_1 à S_2.

Un problème de couplage dans un graphe biparti se ramène à un problème de flot de la façon suivante :



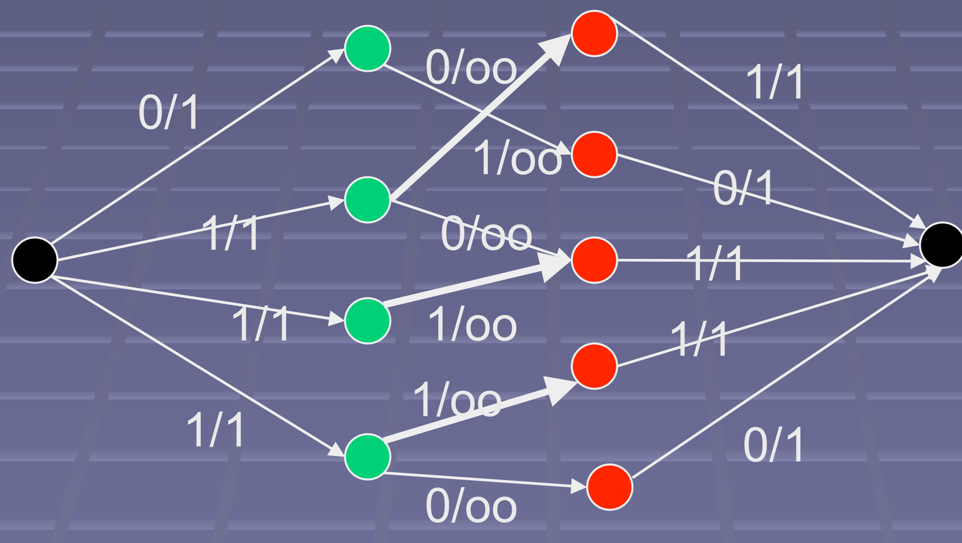
- A un couplage C du graphe biparti, correspond un flot de débit #C.

Un problème de couplage dans un graphe biparti se ramène à un problème de flot de la façon suivante :



- A un couplage C du graphe biparti, correspond un flot de débit #C.

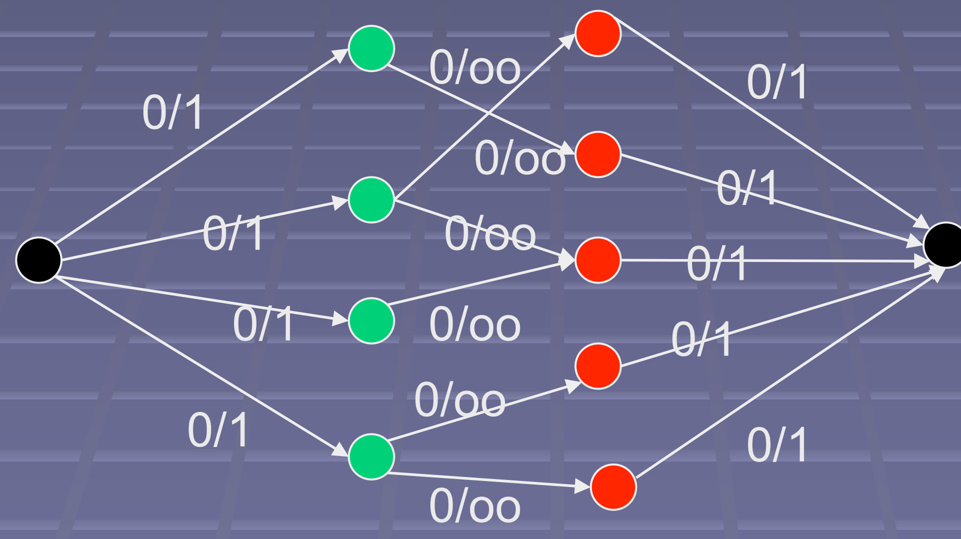
Un problème de couplage dans un graphe biparti se ramène à un problème de flot de la façon suivante :



Débit = 3

- Réciproquement sous ces hypothèses (tous les flots sont entiers !!), un flot de débit D donne un couplage de cardinalité D.
- Une chaîne augmentante du couplage correspond à un chemin de la source au but dans le graphe d'écart, et donc à une augmentation du flot.

Un problème de couplage dans un graphe biparti se ramène à un problème de flot de la façon suivante :

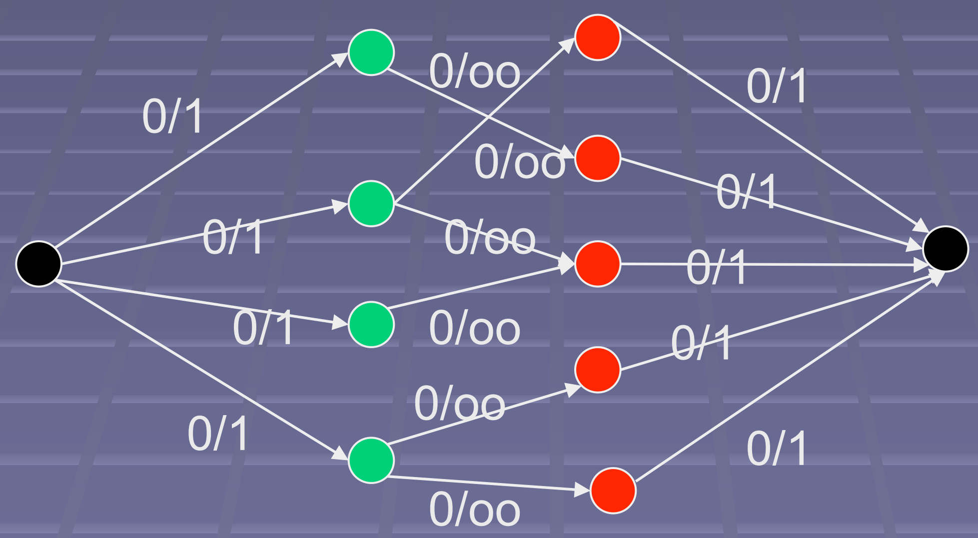


Pour chercher un couplage maximal dans le graphe biparti, on applique l'algorithme de Ford-Fulkerson (en partant d'un flot nul) à ce réseau de transport.

Couplage de coût minimal

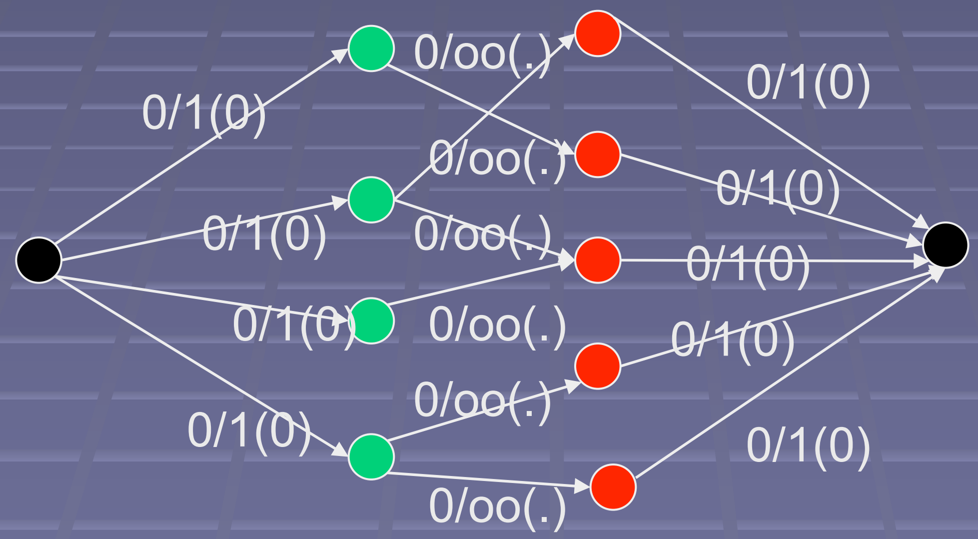
- Si le graphe est valué par des entiers non négatifs (ce sont les coûts),
- un *couplage maximal de coût minimal* est un couplage maximal, dont la somme des coûts est minimale parmi tous les couplages maximaux.

Couplage de coût minimal sur un graphe biparti et problème de flot de coût minimal



Comme précédemment on ramène le problème de couplage dans un graphe biparti à un problème de flot.

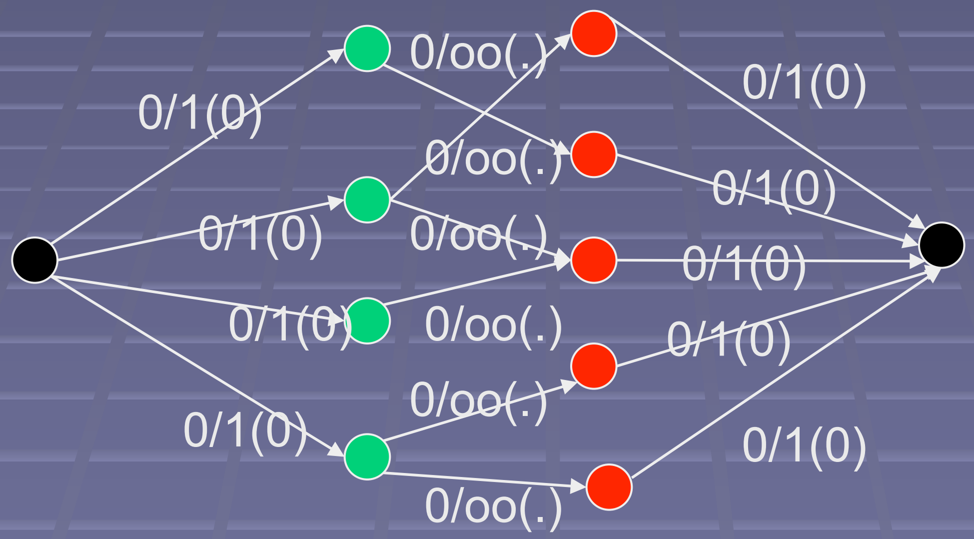
Couplage de coût minimal sur un graphe biparti et problème de flot de coût minimal



Sauf qu'en plus les arcs sont munis d'un coût :

- nul pour ceux incidents à la source ou au but.
- pour les autres, c'est leur coût dans le graphe biparti.

Couplage de coût minimal sur un graphe biparti et problème de flot de coût minimal



Chercher un couplage maximal de coût minimal revient à : chercher un flot de débit maximal, et de coût minimal (avec ce qui va suivre …).

Exemple d'applications : encore un problème d'affectation.

- *N* personnes à affecter à *M* machines.
- Chaque affectation a un coût. Minimiser le coût total.
- Pour résoudre le problème on constitue un graphe biparti, où les sommets de la partition correspondent respectivement aux travailleurs et aux machine.
- Un arête relie chaque travailleur à chaque machine ; elle est valuée par le coût de l'affectation.
- Résoudre le problème revient à chercher un couplage maximal de coût minimal dans le graphe biparti. La méthode s'applique.

Problèmes de flot de coût minimal

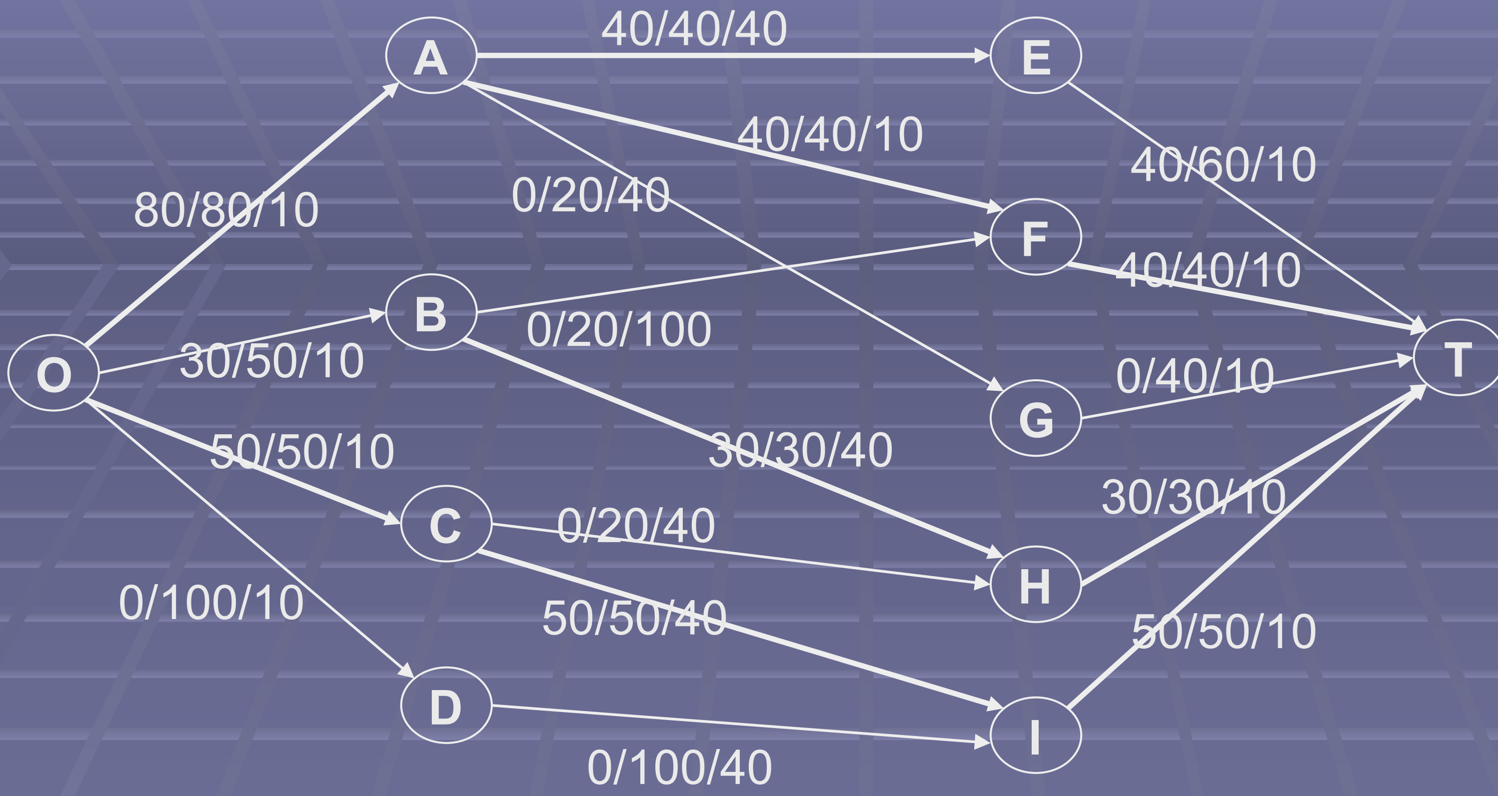
- On considérera un réseau de transport avec coût.
- Il s’agit d’un réseau de transport où les arêtes sont en plus munies d’un coût : un entier non négatif. C’est le coût de passage d’une unité.
- Le problème du flot de coût minimal consiste à acheminer un flot de débit fixé D, pour un coût total (Σ coût x flot) minimal.

Graphe d'écart avec coût

- Comme pour un problème de flot sans coût on constitue le graphe d'écart.
- Sauf qu'en plus les arêtes du graphe d'écart sont munies d'un 'pseudo-coût' :
  - Les arêtes 'positives' gardent leur coût $C_i \geq 0$
  - Les arêtes 'négatives' ont le coût opposé $-C_i \leq 0$.

Algorithme : trouver un flot de coût minimal

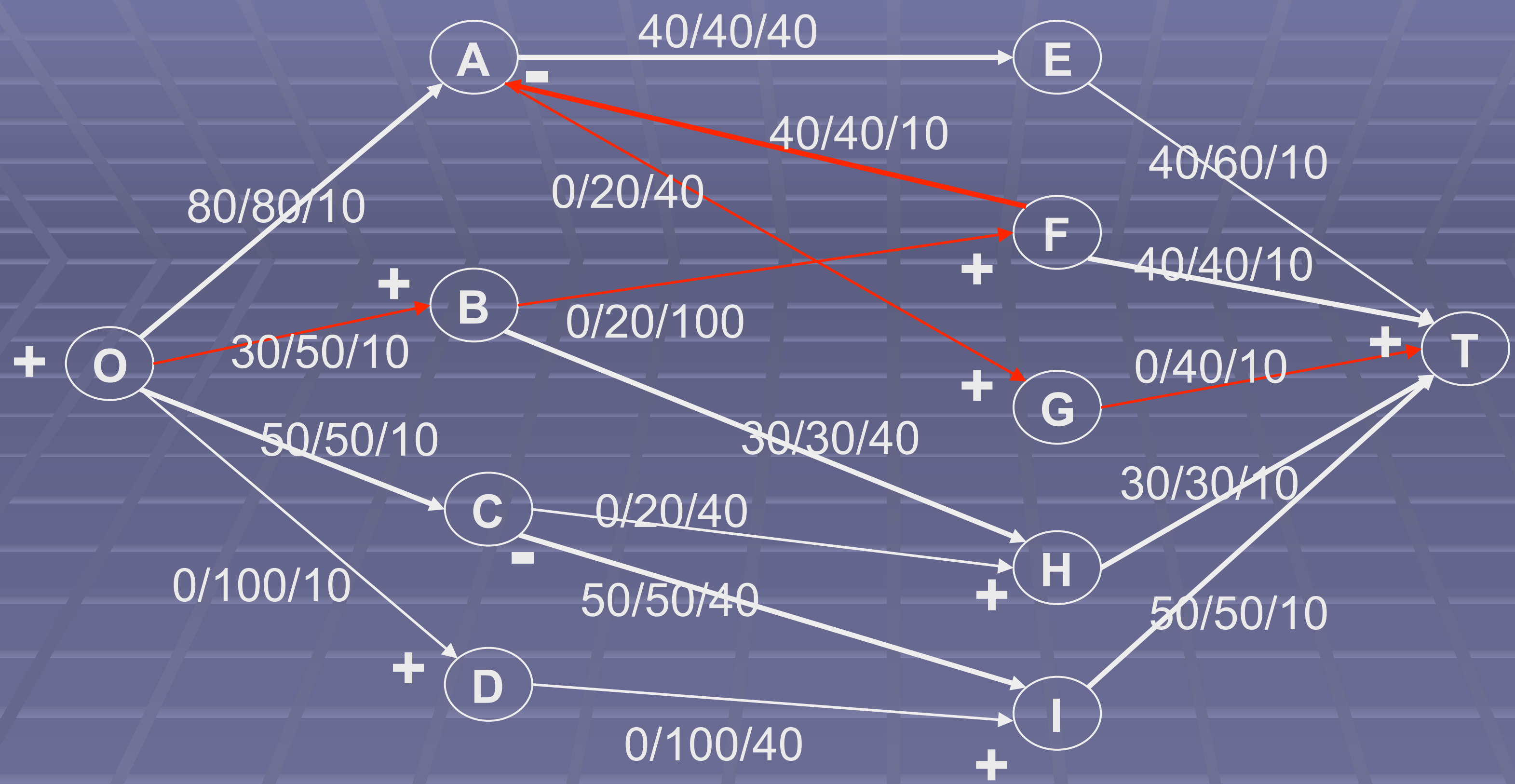
- Paramètre initial M (débit souhaité)
- Initialement le débit D et le coût K sont = 0
- Répéter
  - Chercher une chemin se S à B dans le graphe d'écart, <u>de coût minimal c.</u>
  - Calculer son augmentation Δ.
  - Δ=min (Δ , M-D)
  - Dans le réseau de transport augmenter les flux des arêtes + de Δ, diminuer le flux des arêtes – de Δ, et augmenter D de Δ .
  - K= K+c x Δ
- Tant que l'on trouve un tel chemin et D<M.
- A la fin le débit est min(M, débit maximal), et le flot est de coût minimal parmi de tels flots.



Trouver un flot maximal de coût minimal

1ère Méthode : Partir d'un flot non nul

D'abord maximiser le flot puis minimiser son coût.

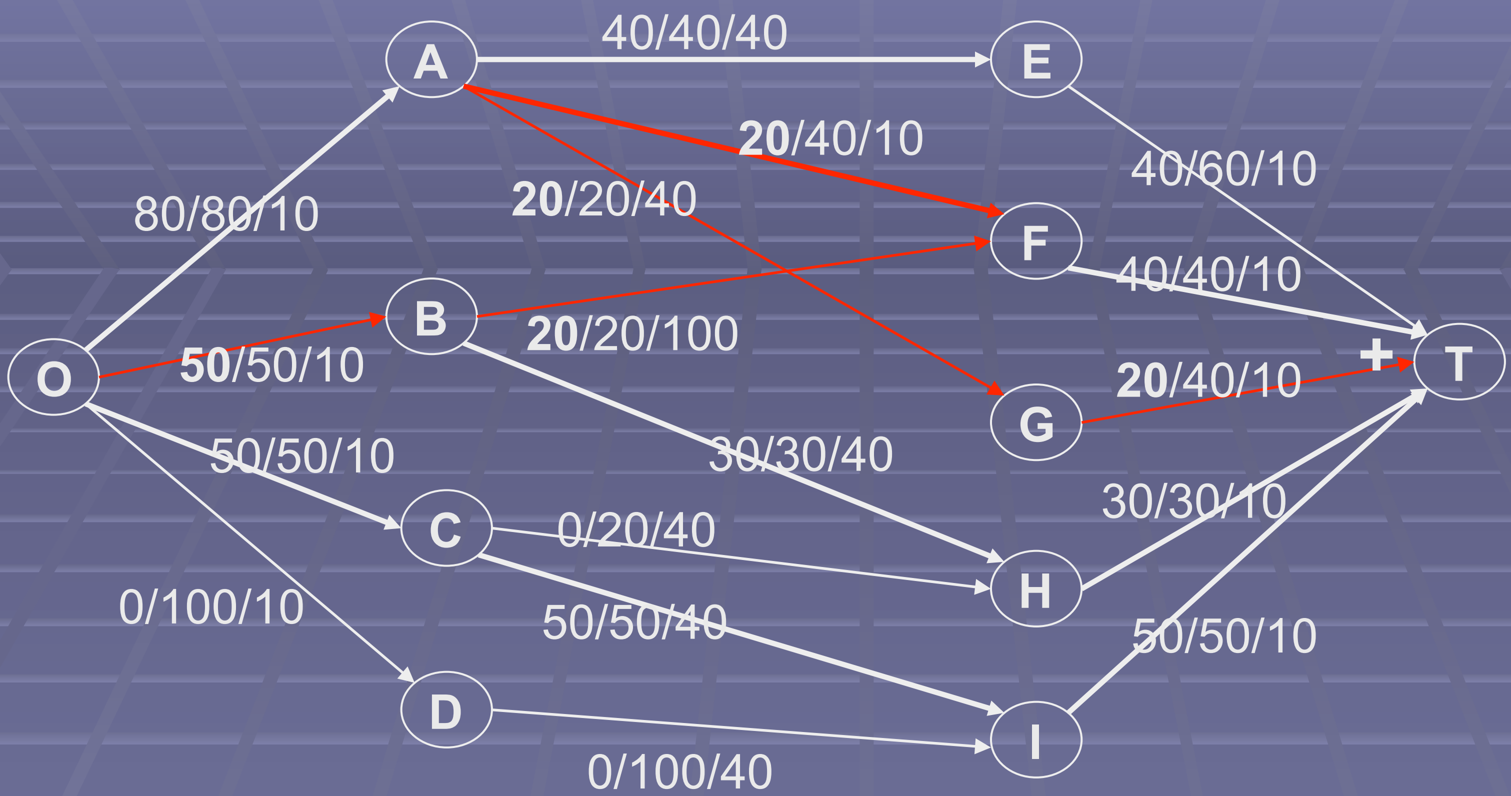


Recherche de chaînes augmentantes

Une seule : **OBFAGT**

Capacité : 20

Sur-coût unitaire : 10+100-10+40+10=150

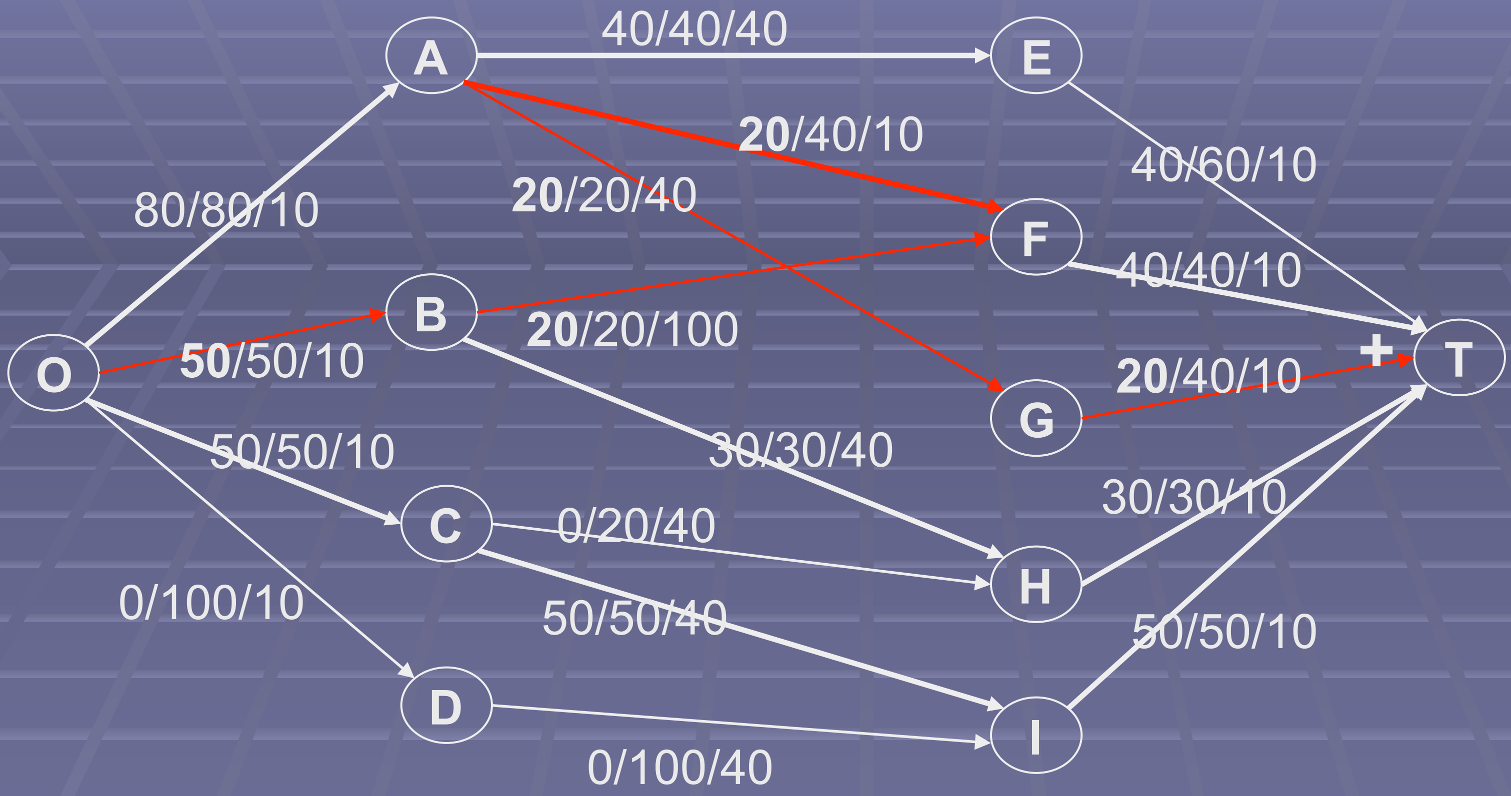


Flot augmente de 20 → 180

Coût augmente de 20*150=3000 → 11400

Le flot est maximal

Le coût n'est pas nécessairement minimal (parmi ceux maximaux)

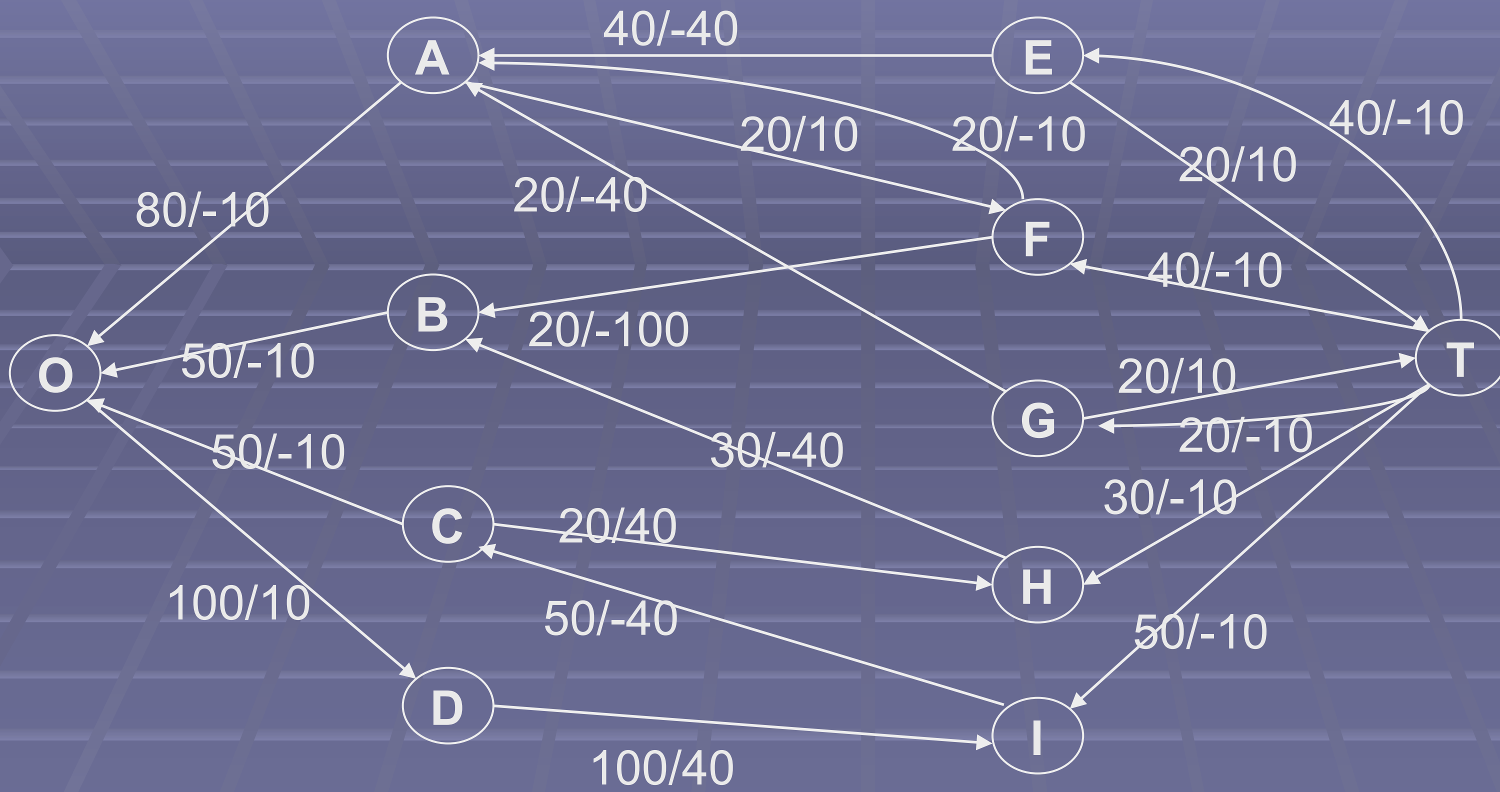


Comment optimiser le coût à débit constant ?

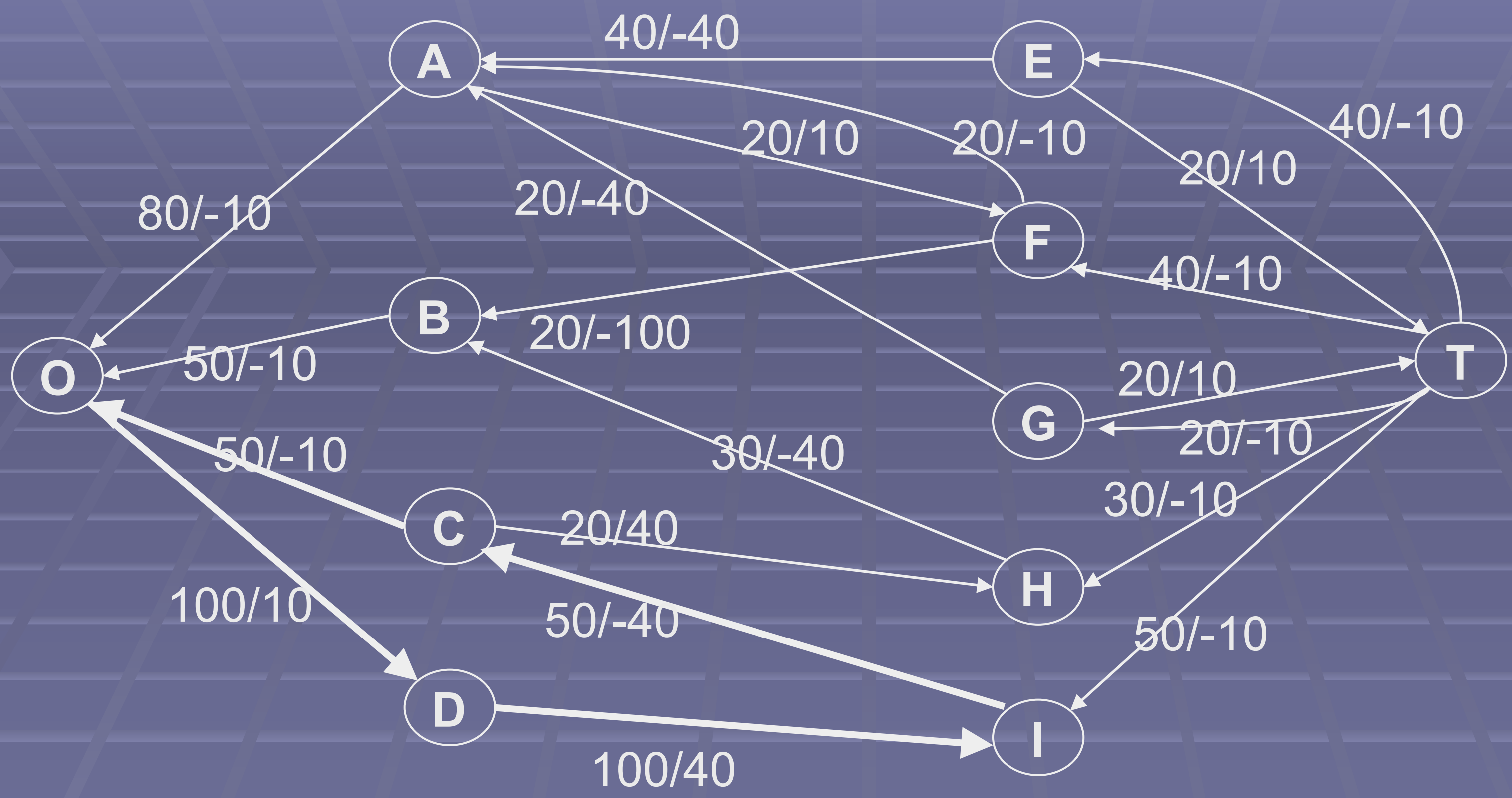
Le flot est maximal

Le coût n'est pas nécessairement minimal
(parmi ceux maximaux)

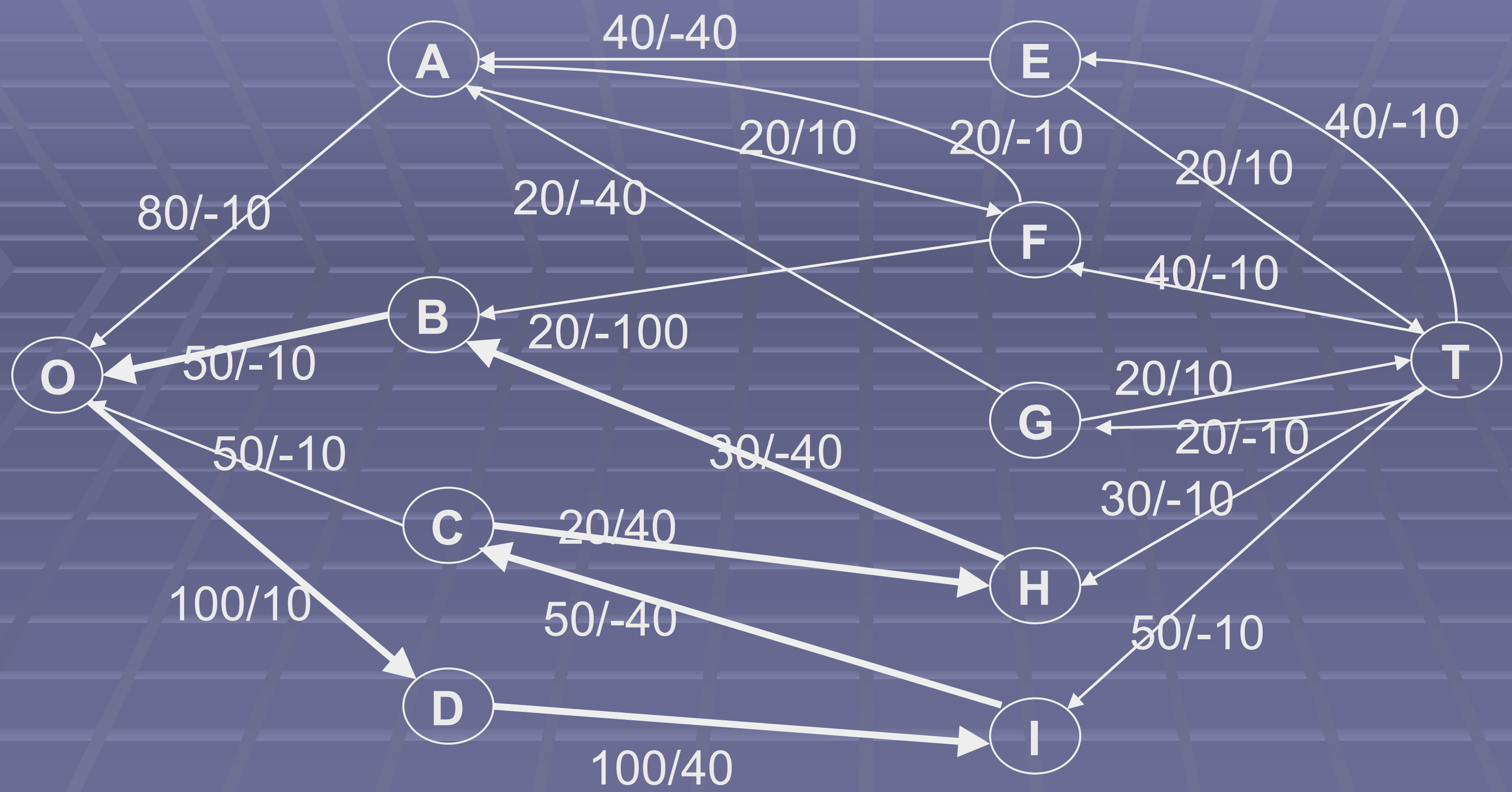
- Réponse :
- On construit le graphe d'écart (avec coût), et on cherche un cycle de coût <0. On modifie le flot puis on continue tant que cela est possible. Car :
- **Théorème :** à débit fixé un flot n'est pas de coût minimal si et seulement si son graphe d'écart contient un cycle négatif.



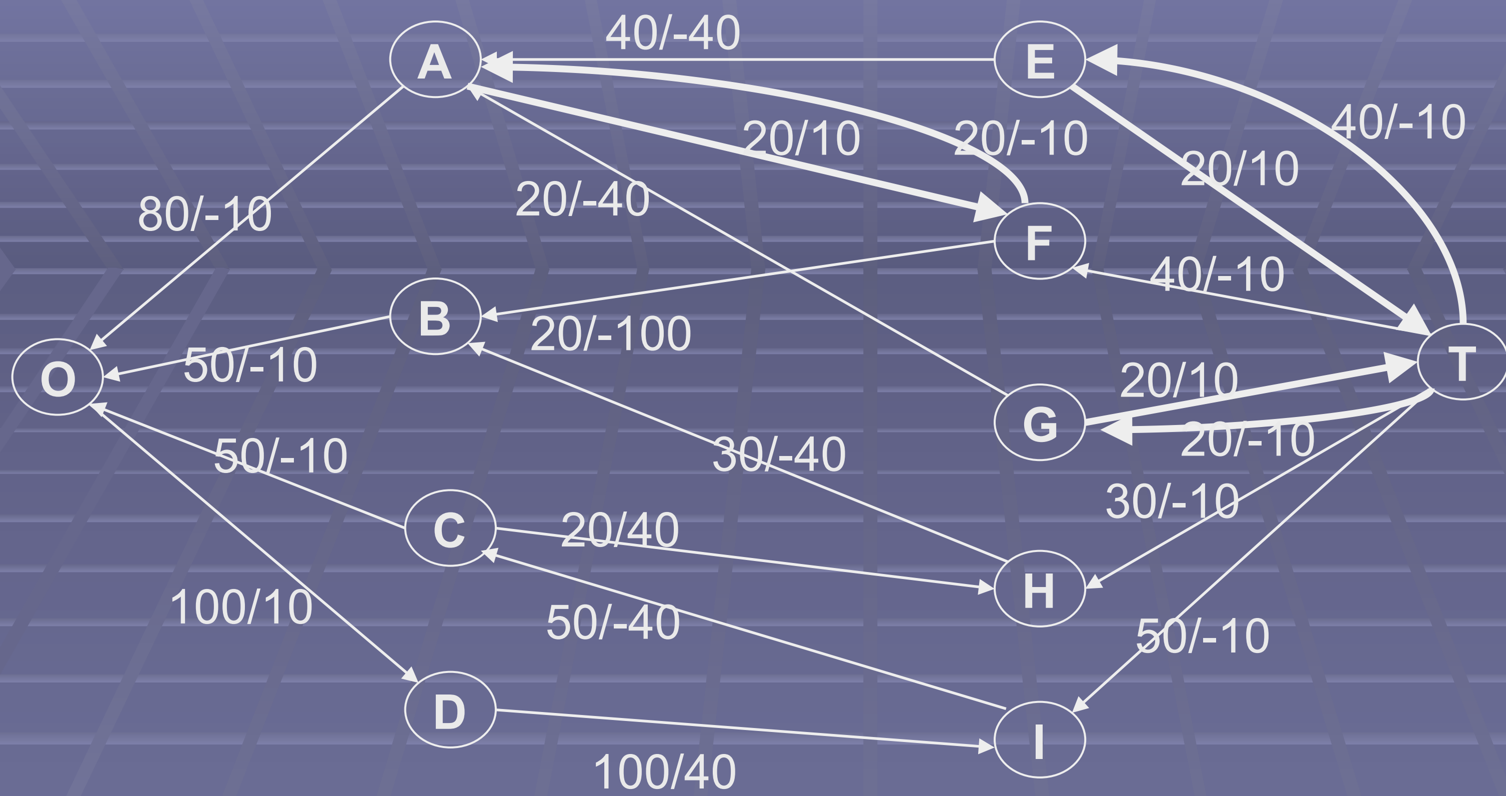
Le graphe d' écart



Un cycle de coût 10+40-40-10=0

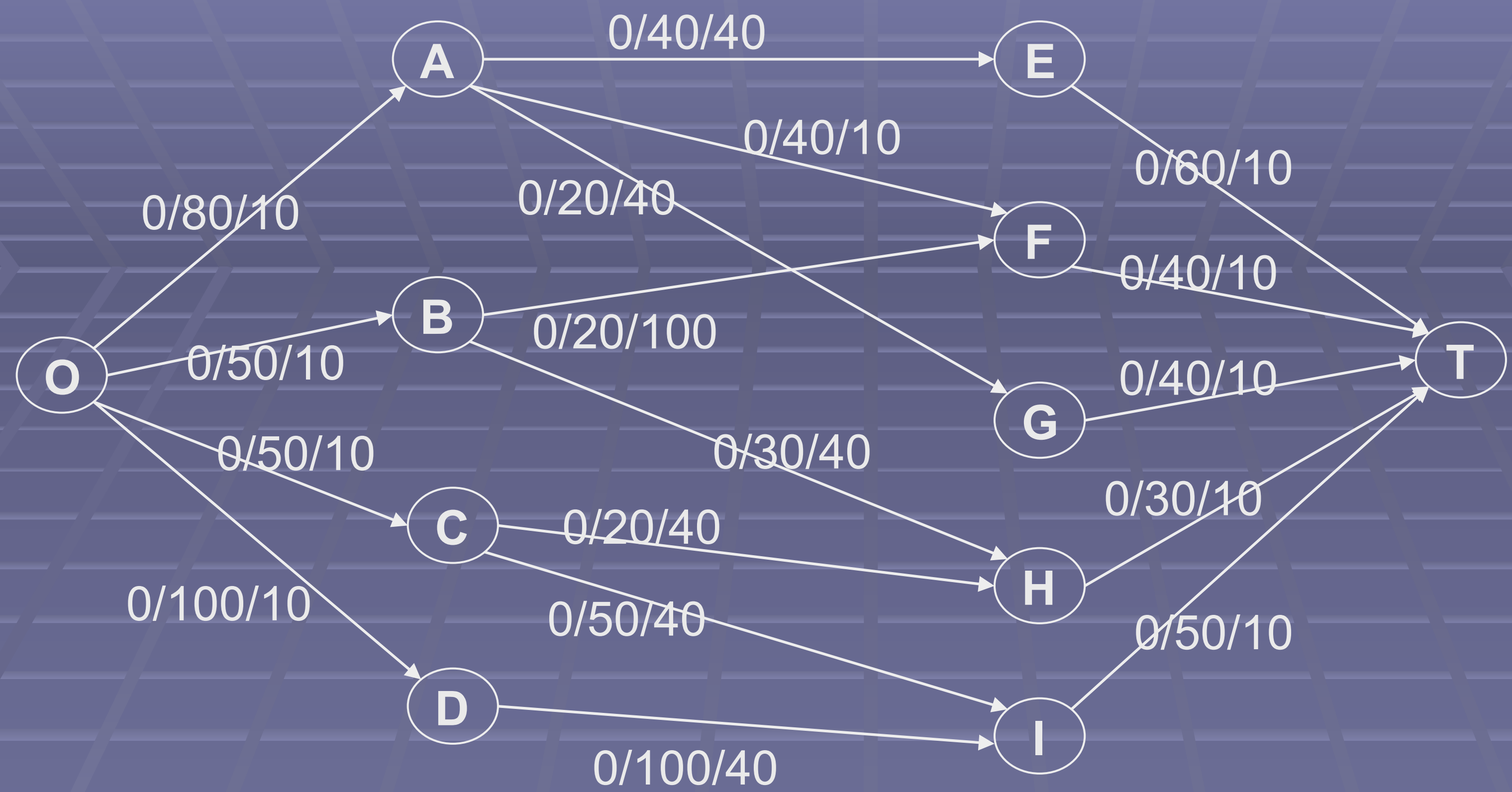


Un cycle de coût 10+40-40+40-40-10=0



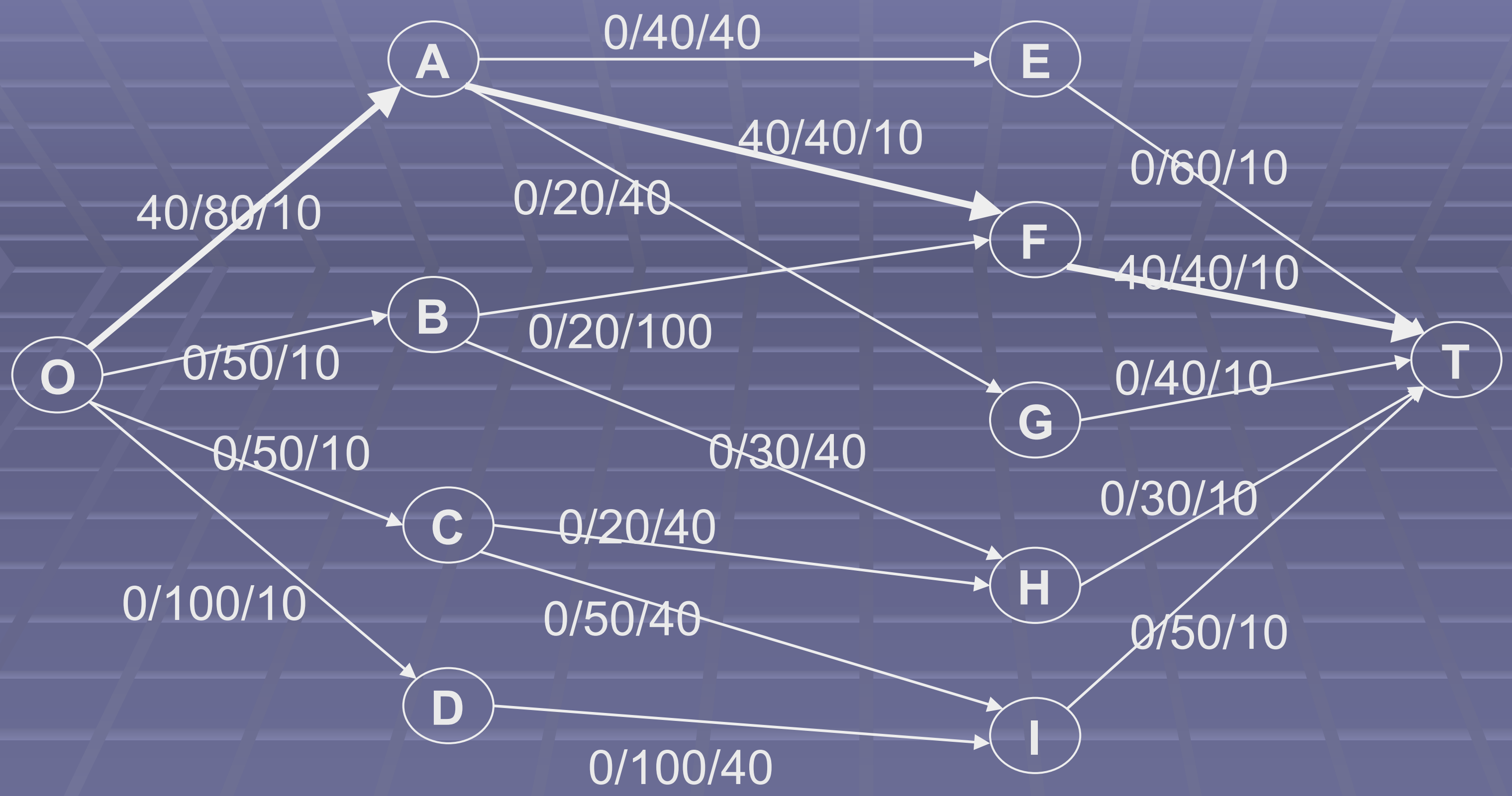
Les cycles ʻaller-retourʼ sont toujours de coût =0

Il nʼy a pas de cycles de coût négatifs : le coût trouvé est minimal. CQFD



<u>2ème Méthode</u> : algorithme de Busacker-Gowen

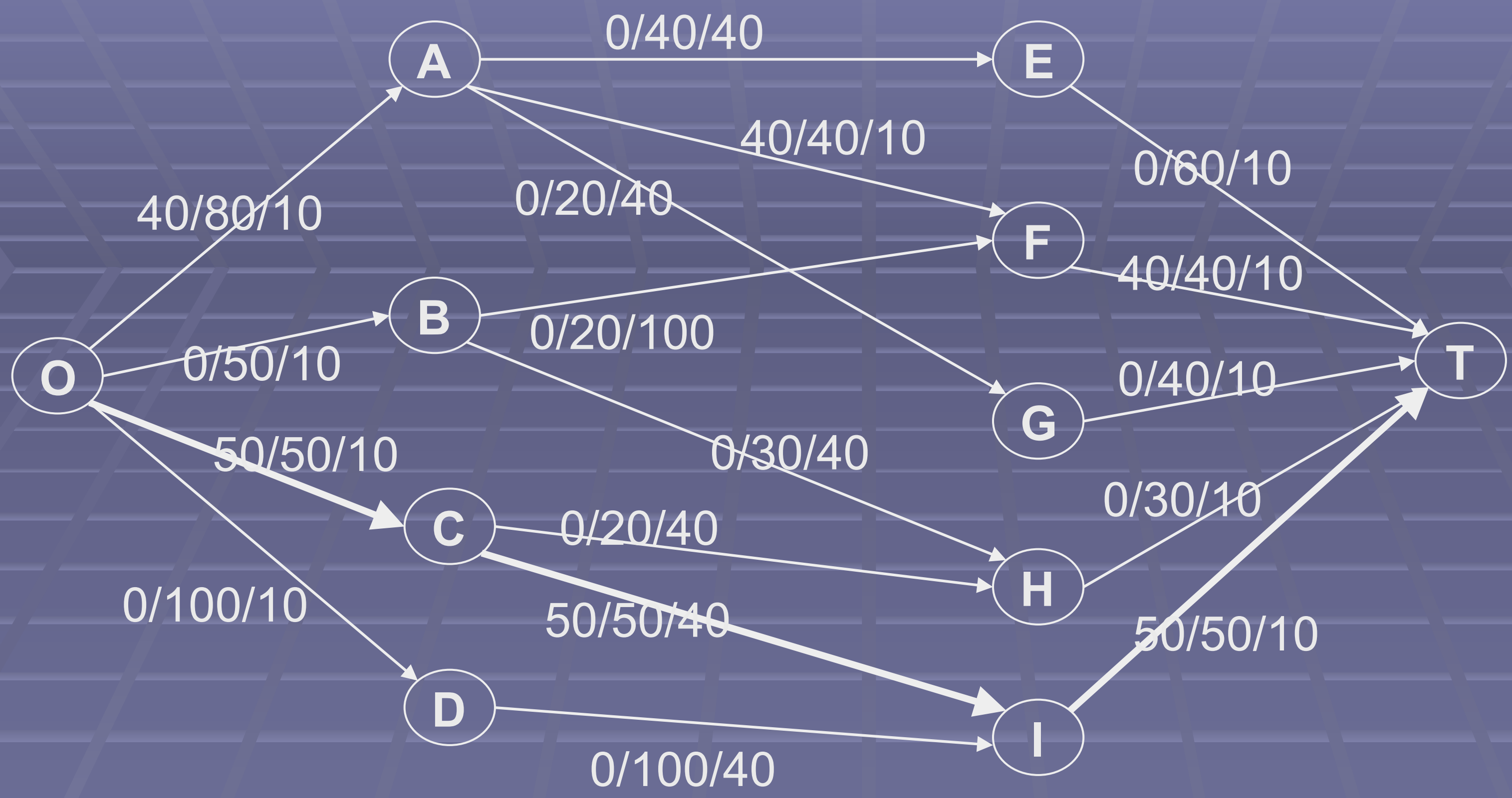
On part du flot nul !!



On Cherche une chaîne augmentante de coût minimal

OAFT : capacité 40, coût 30

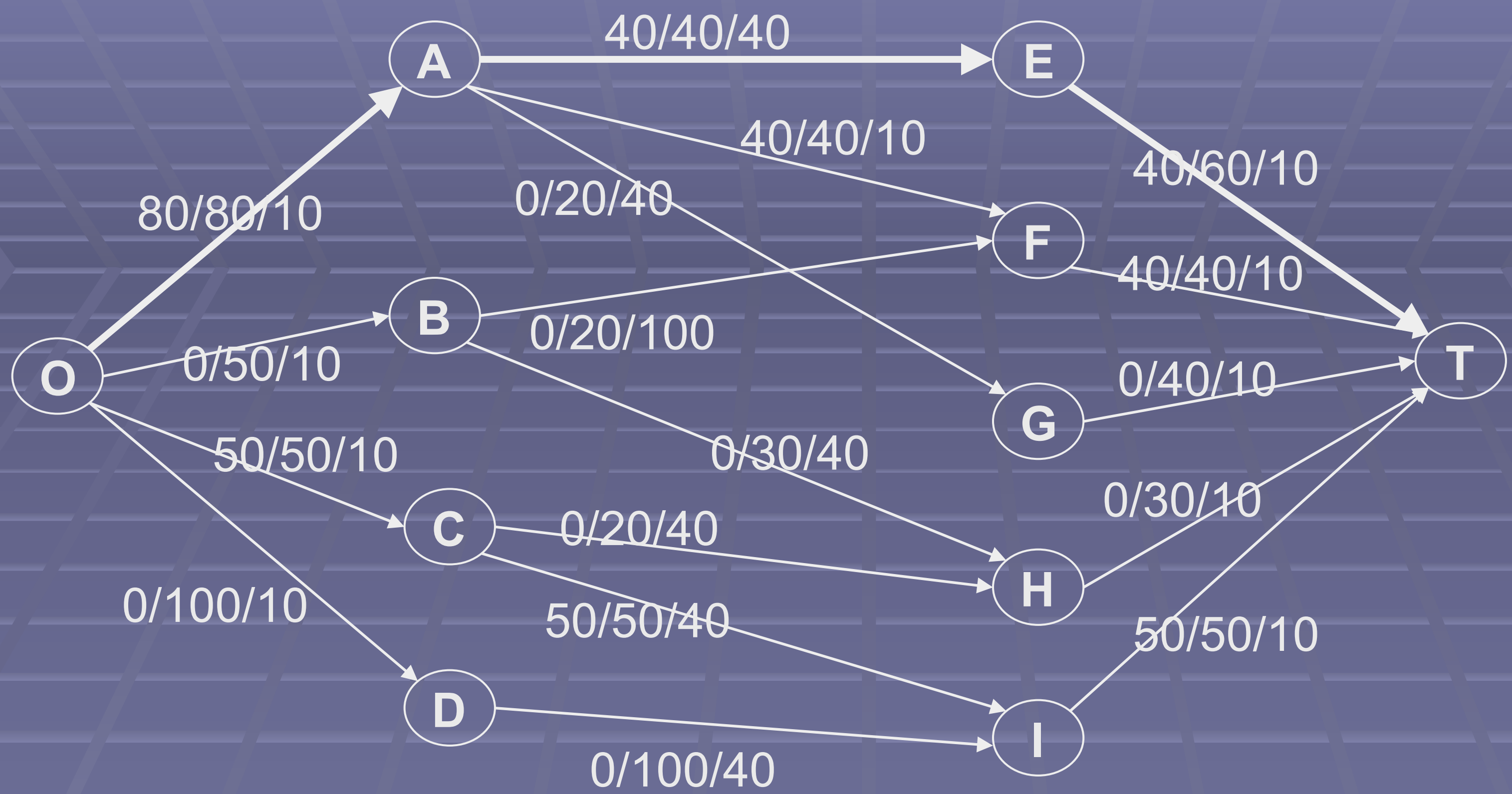
Débit = 40, coût= 1200 (minimal parmi ceux de débit 40)



On Cherche une chaîne augmentante de coût minimal

OCIT : capacité 50, coût 60

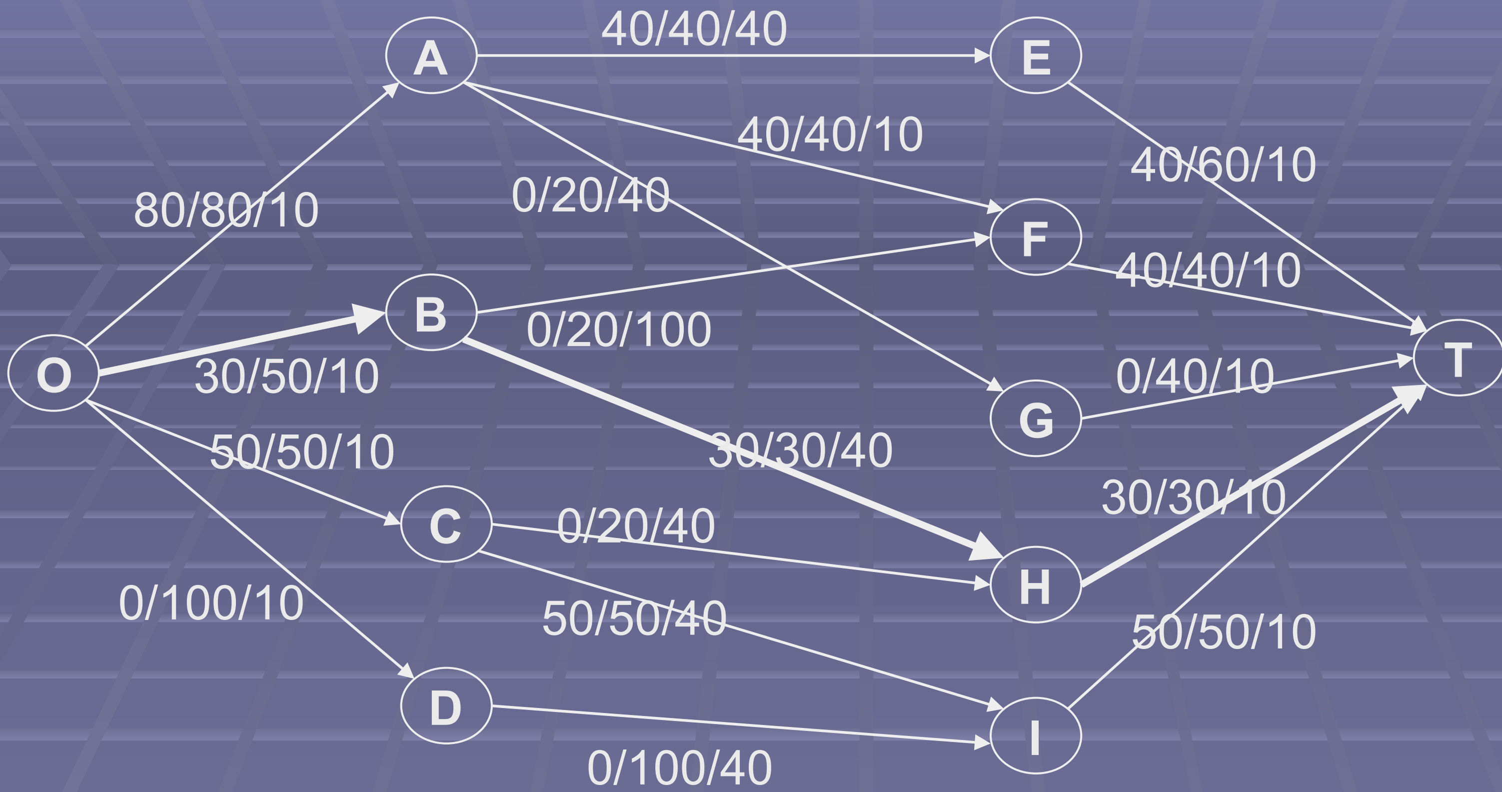
Débit = 90, coût= 4200 (minimal parmi ceux de débit 90)



On Cherche une chaîne augmentante de coût minimal

OAET : capacité 40, coût 60

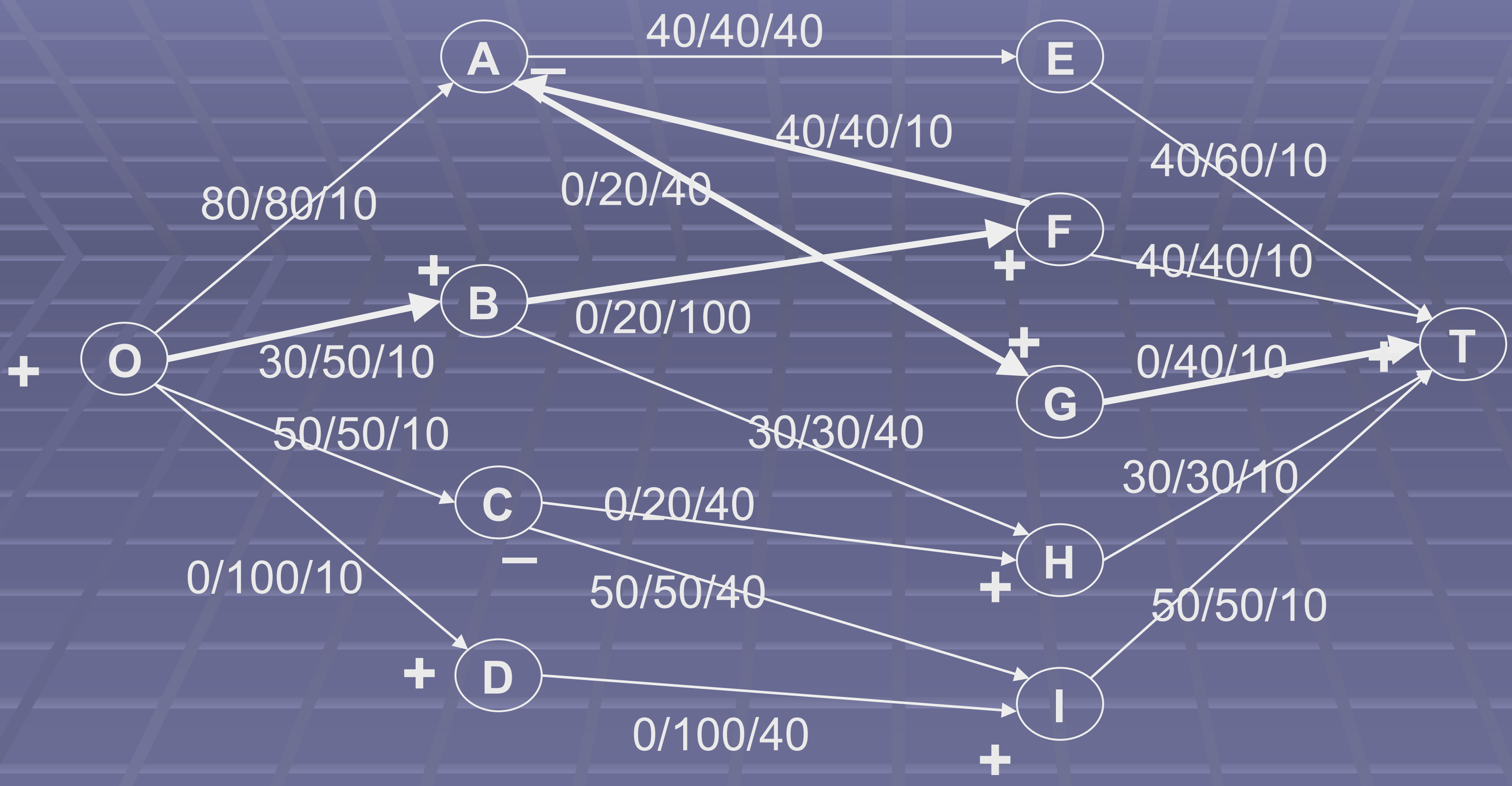
Débit = 130, coût= 6600 (minimal parmi ceux de débit 130)



On Cherche une chaîne augmentante de coût minimal

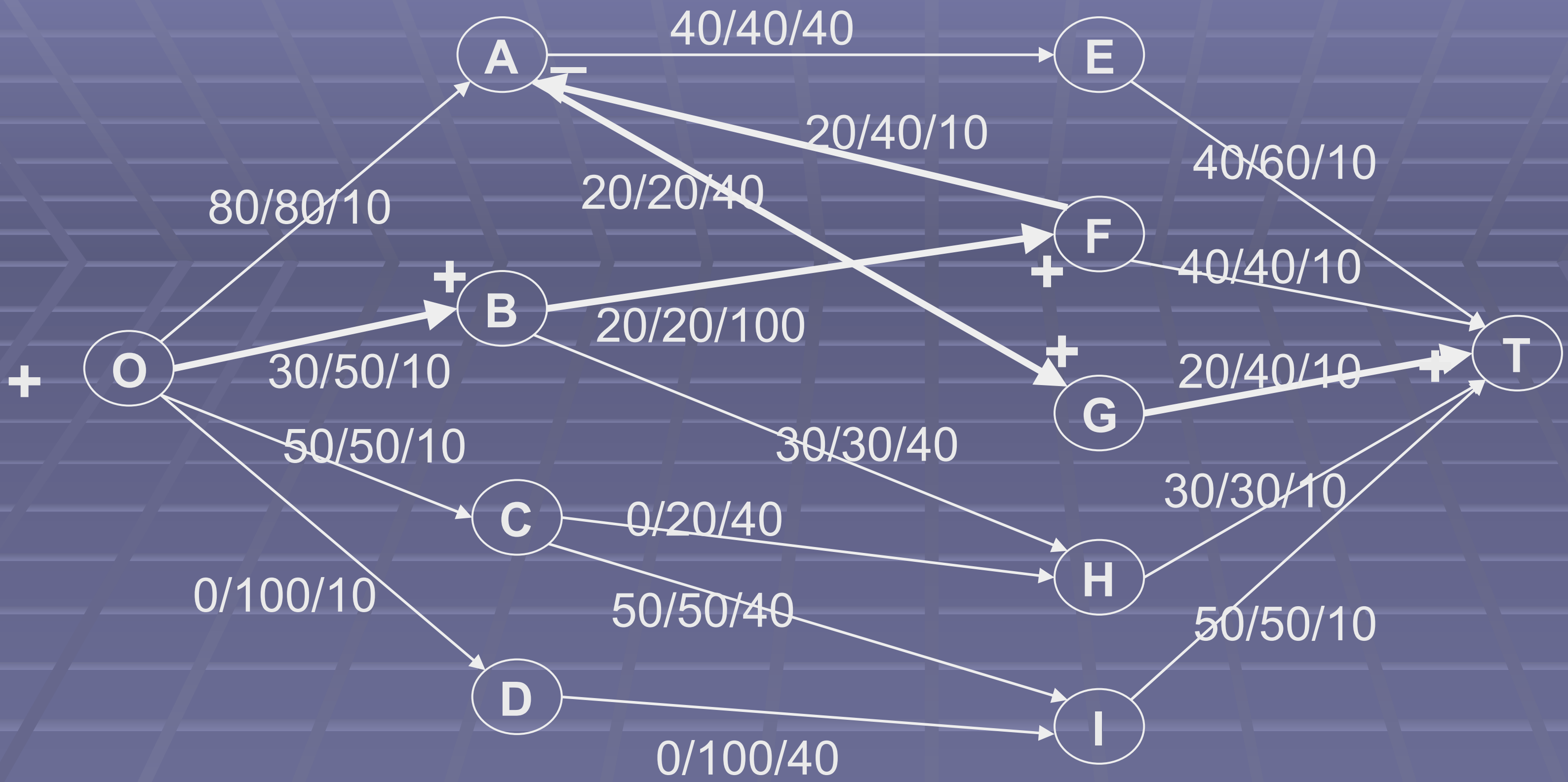
OBHT : capacité 30, coût 60

Débit = 160, coût= 8400 (minimal parmi ceux de débit 160)



On Cherche une chaîne augmentante de coût minimal

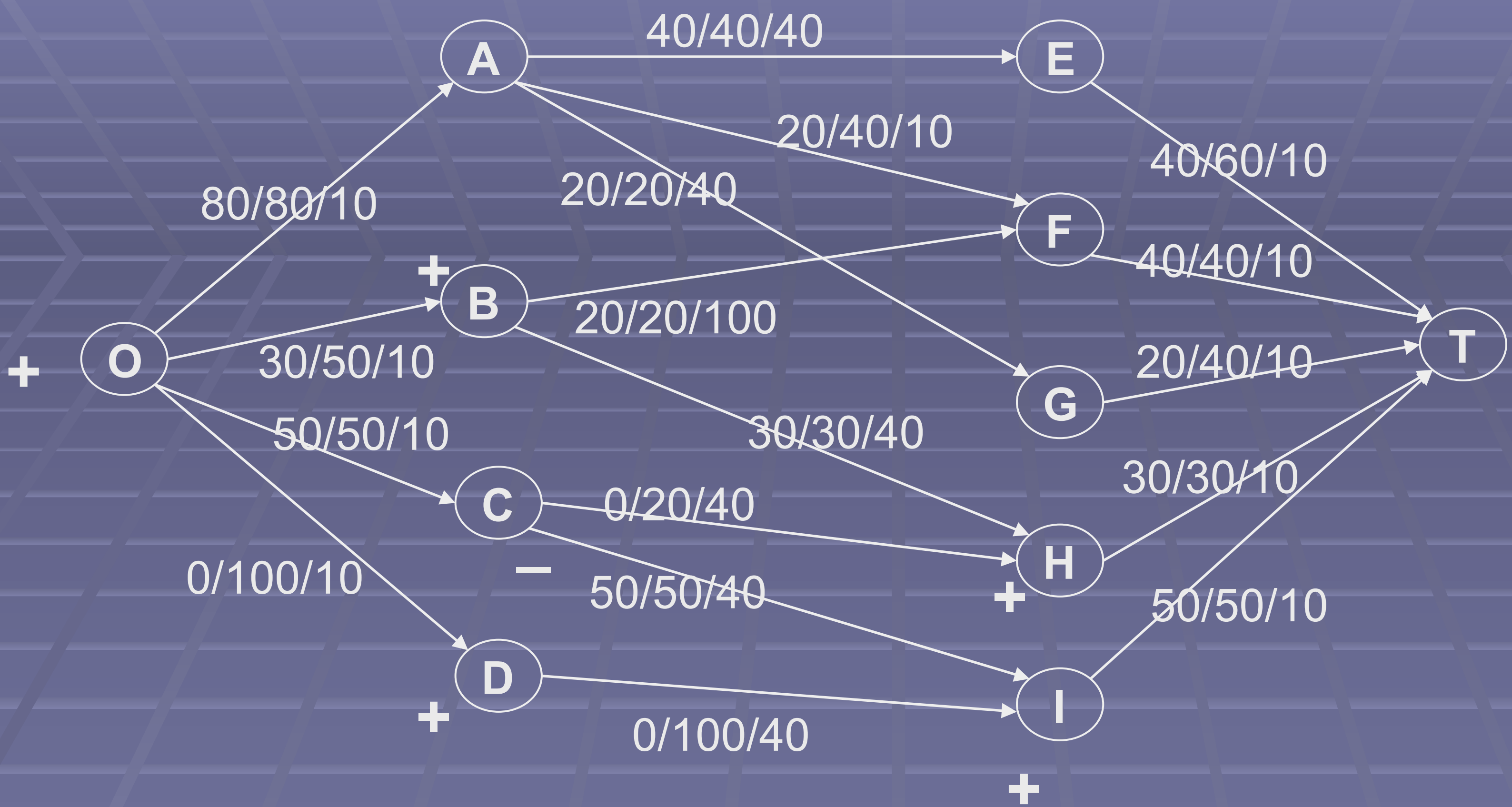
Une seule : OBFAGT



On Cherche une chaîne augmentante de coût minimal

OBFAGT : capacité 20, coût 150

Débit=180, coût=11400 (minimal parmi ceux de débit 180)



On Cherche une chaîne augmentante de coût minimal

Plus aucune… L'algorithme s'arrête

Débit=180　Coût=11400

Exemple : Problème de transport

- Supposons que l'on doive organiser des acheminements (en bateau, avion) entre des base de dépôts et des lieux d'expédition.
- Toute base i dispose d'un stock A_i.
- Chaque lieu d'expédition j a une demande B_j.
- Stock total = demande totale
- Pour chaque acheminement entre un dépôt A_i et un lieu B_j on a un coût C_{ij} par tonne de fret.
- Problème : Acheminer tout le stock avec un coût total minimal.

Pour le résoudre

- Considérer un graphe biparti (bases/lieux) où une arête relie une base à tous les lieux où il est possible et demandé d'envoyer des stocks. Elle est munie du coût d'acheminement (pour 1t) et d'une capacité infinie.
- Ajouter une source reliée à chaque base i par un arc de coût nul et de capacité A_i
- Ajouter un but relié à chaque lieu d'expédition j par un arc de coût nul et de capacité B_j.
- Si M est la capacité en stock, le problème revient à chercher dans ce réseau de transport avec coût un flot de débit M et de coût minimal.

Affectation de coût minimal

- On a n tâches à effectuer et n travailleurs.
- L'affectation d'un travailleur i à la tâche j a un coût C_{ij}.
- Comment effectuer toutes ces tâches pour un coût minimum ?

Méthode hongroise

- On considère la matrice :
- Il s'agit de choisir n éléments de la matrice, tels que chaque ligne et chaque colonne ne contienne qu'un seul de ces éléments, et que leur somme soit minimale.
- Le problème est inchangé si on ajoute un même nombre à une colonne ou à une ligne.

C_{11}	C_{12}	…	C_{1n}
C_{21}	C_{22}	…	C_{2n}
		…	
C_{n1}	C_{n2}	…	C_{nn}

- Une *réduction* consiste à soustraire de chaque ligne son plus petit élément, puis de soustraire de chaque colonne son plus petit élément.
- Ex :

1	3	2
3	2	2
2	1	3

0	2	1
1	0	0
1	0	2

Algorithme hongrois

- Réduire la matrice
- Tant que on ne peut pas affecter n zéros, 1 par ligne, 1 par colonne,
  - Chercher un support (lignes & colonnes) minimal des zéros, et barrer ses rangées.
  - Soit k le + petit terme non barré. L'ajouter aux termes barrés 2 fois, et le retrancher aux termes non barrés.
- Fin Tant que.

réduction puis au + 1 affectation

1	2	3
1	4	4
1	3	5

0	0	0
0	2	1
0	1	2

0	0	0
0	2	1
0	1	2

k=1

1	0	0
0	1	0
0	0	1

1	0	0
0	1	0
0	0	1

fini.

Algorithme de recherche d'un support minimal des zéros

Essayer d'encadrer un zéro par ligne et par colonne.

Si c'est possible la procédure est finie.

Sinon :

- Marquer toute ligne n'ayant pas de zéro encadré.
- **Tant que c'est possible :**
  - Marquer toute colonne ayant un zéro non encadré sur une ligne marquée.
  - Marquer toute ligne ayant un zéro encadré dans une colonne marquée
- **Fin tant que.**
- Barrer chaque colonne marquée et chaque ligne non marquée. **C'est un support minimal des zéros.**