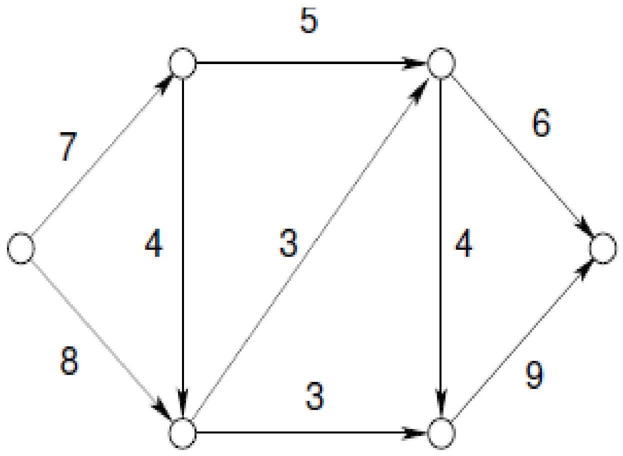
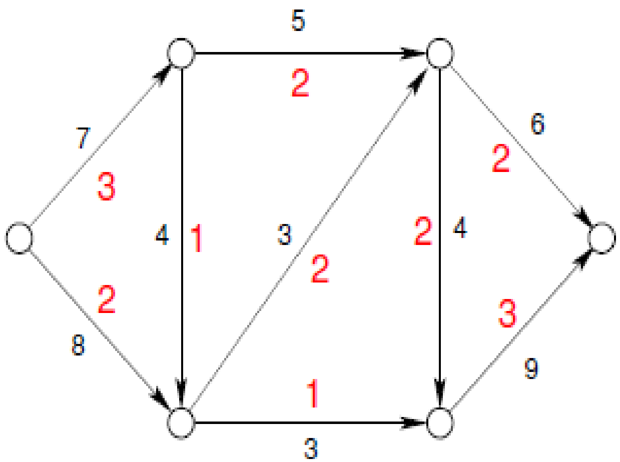
Algorithme de Ford-Fulkerson

Un réseau



un flot réalisable pour ce réseau



Connectivité

Pour un graphe connexe, la connectivité est définie comme le nombre minimum de sommets dont la suppression entraîne la perte de connexité.

Théorème de Menger (1927) : Pour deux sommets non adjacents quelconques d'un graphe connexe G, le nombre minimum d'arêtes dont la suppression entraîne leur déconnexion est égal au nombre maximum de chaînes deux-à-deux arêtes disjointes les connectant dans G.

Ce résultat a été re-formulé en termes de réseaux en 1956 par Ford et Fulkerson et est connu sous le nom de théorème maximum flow/minimum cut ou max flow-min cut.

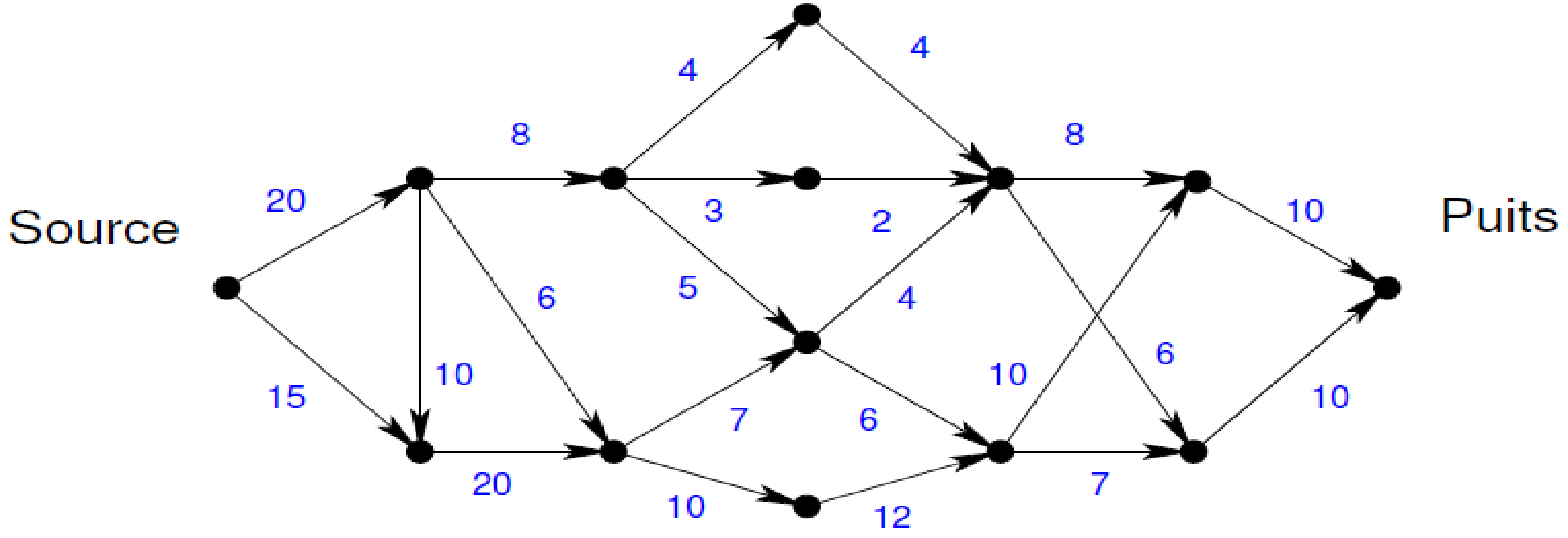
Problème :

Question : quel est le flot maximum qu'il est possible de faire passer dans ce réseau depuis la source vers le puits ?

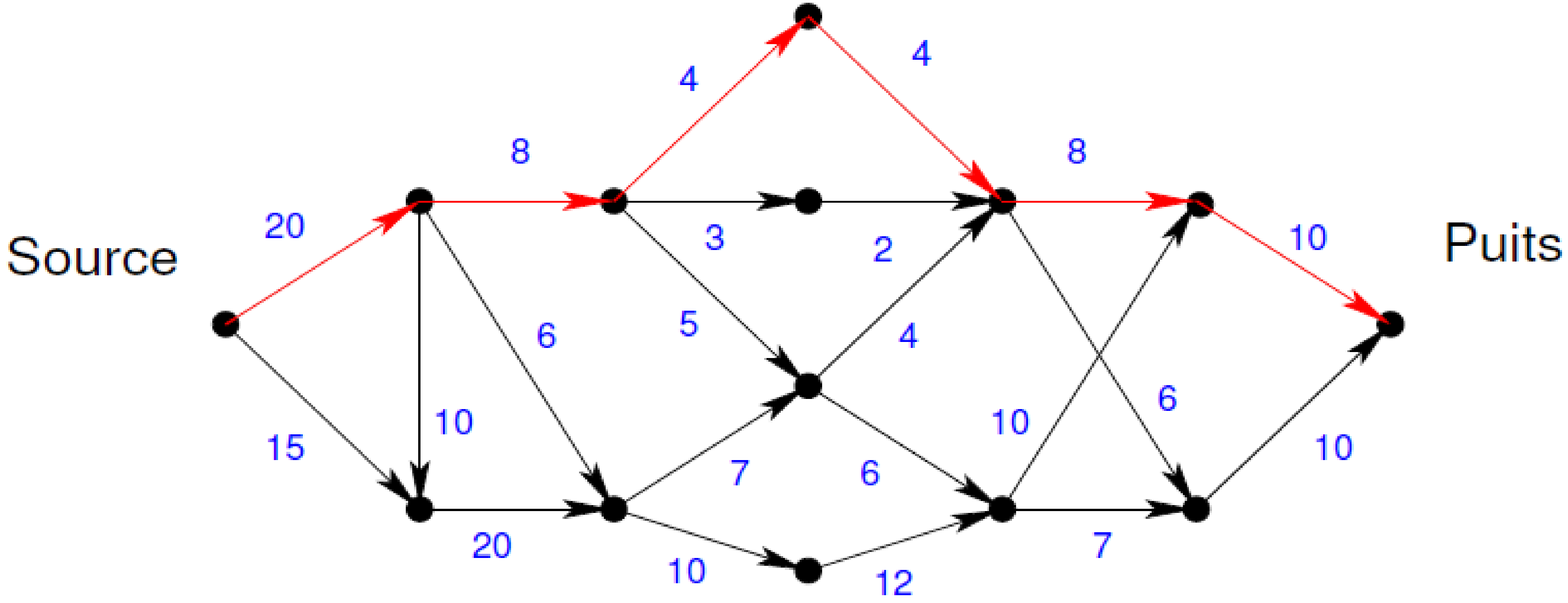
Intuitivement cela fait référence aux problèmes de plomberie ou de trafic routier.

On dispose d'une source et d'un puits et le flot doit s'écouler de la source vers le puits et il doit être maximum en fonction de la capacité des arcs.

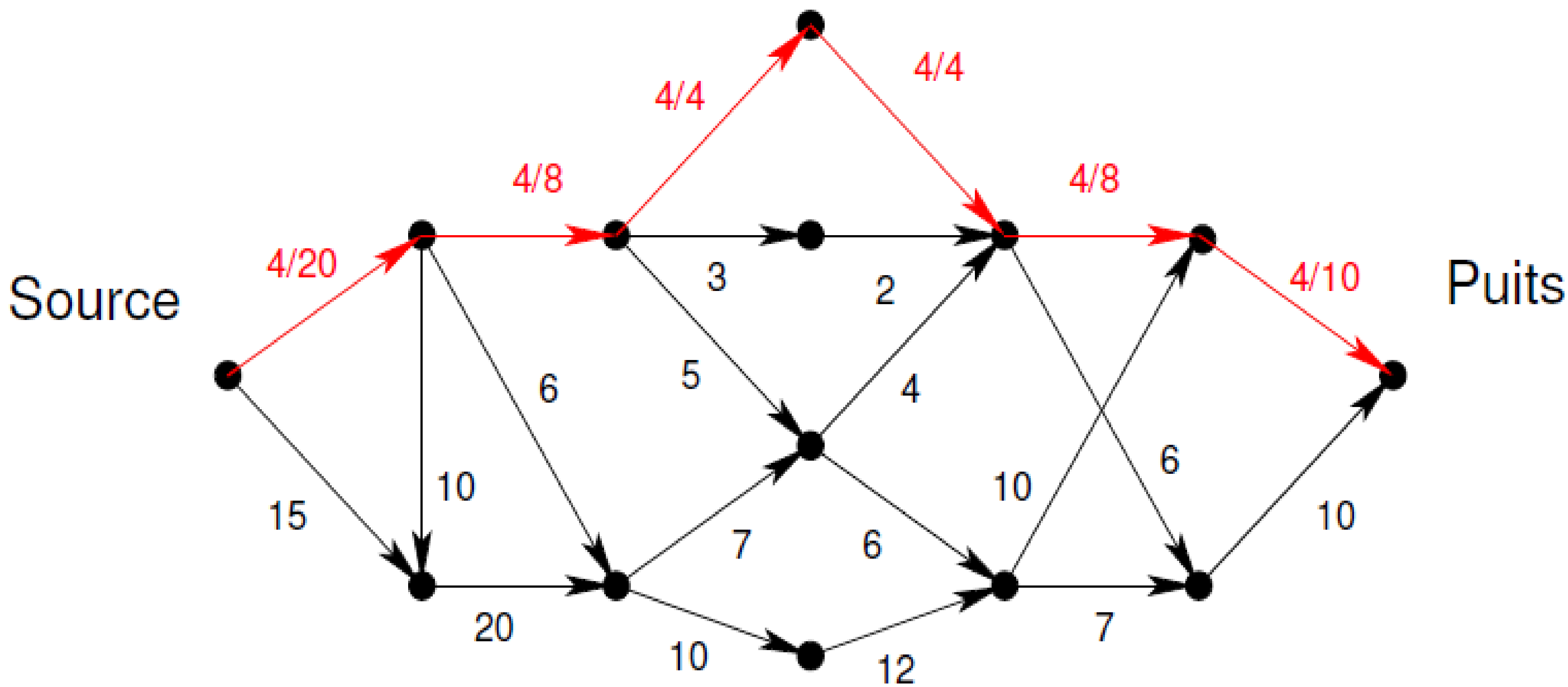
Exemple :



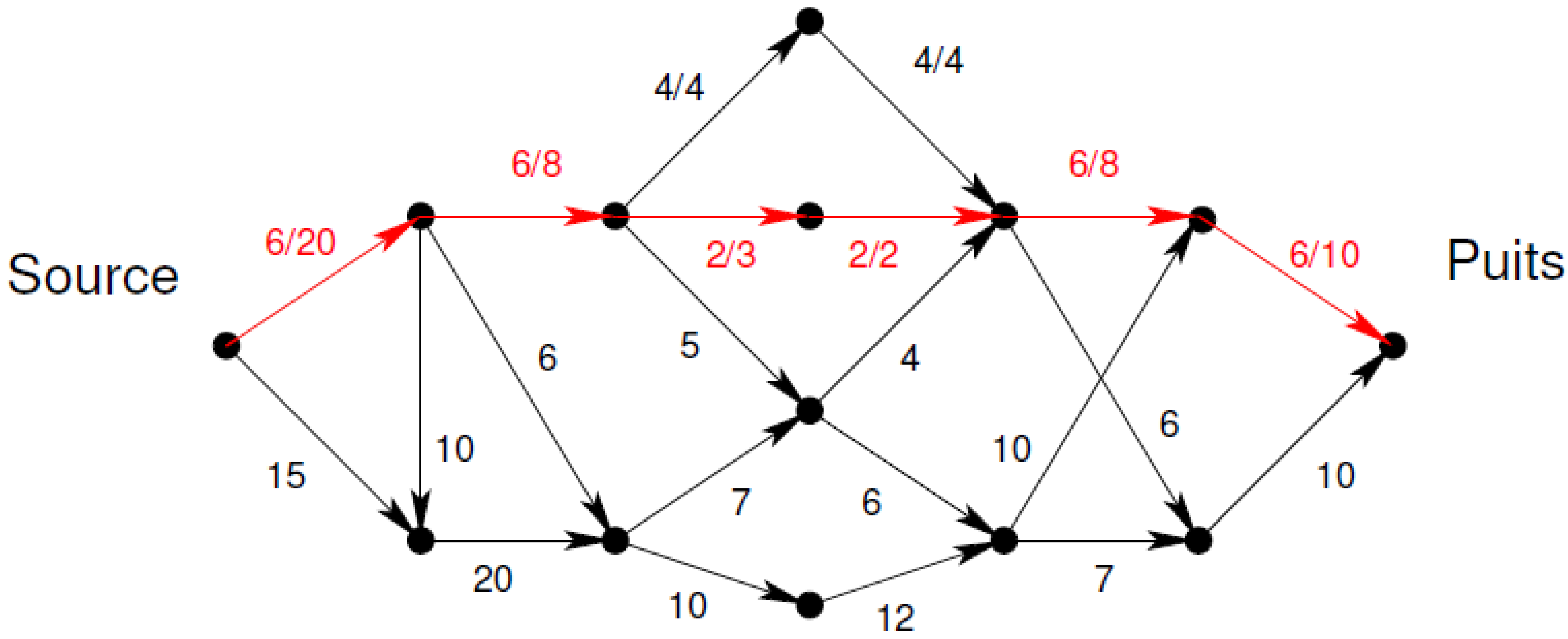
Exemple :



Exemple :

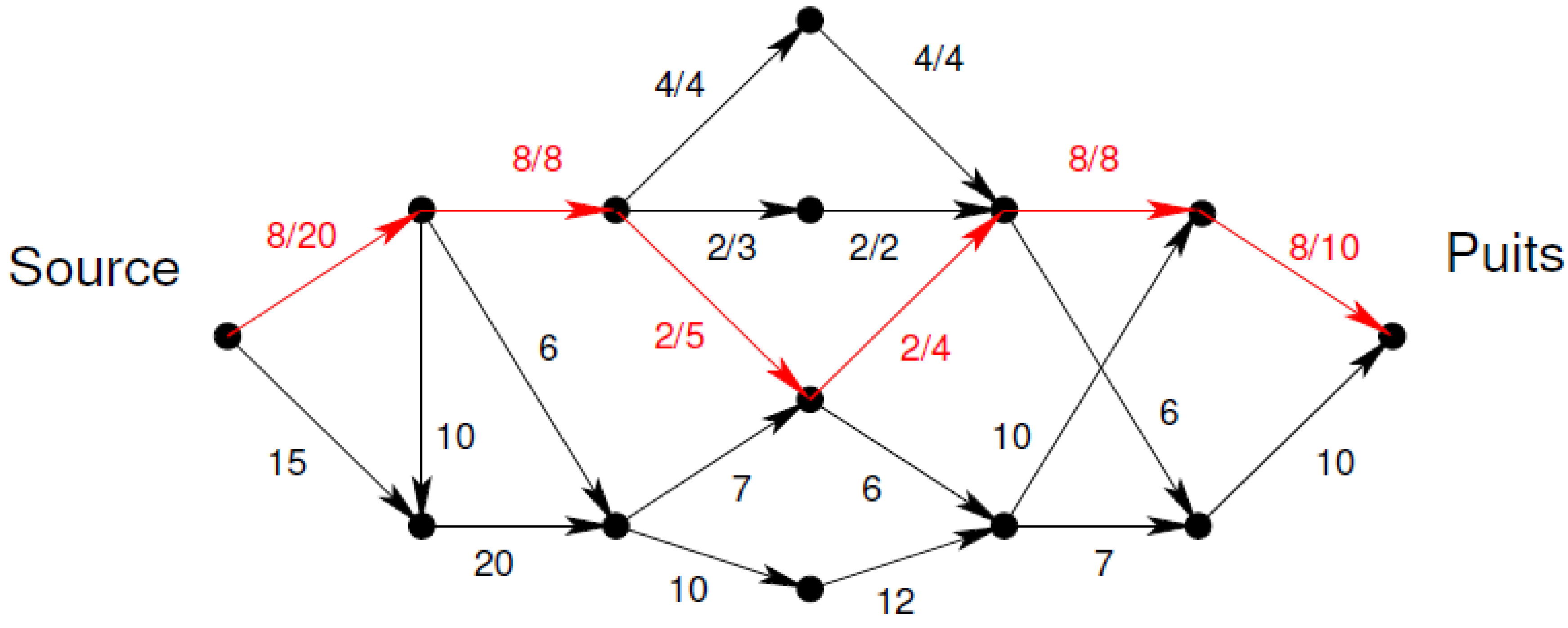


Exemple :



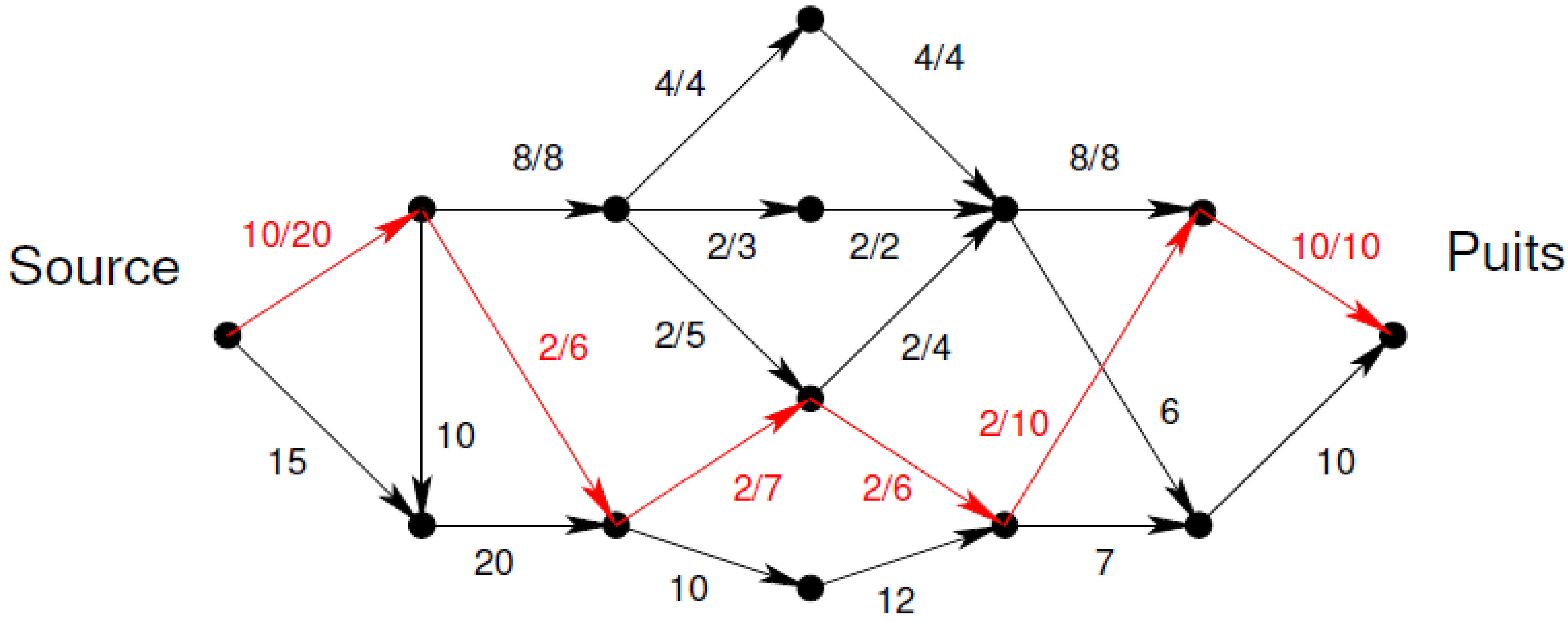
4/4
4/4
6/8
6/8
Source
6/20
2/3
2/2
6/10
Puits
6
5
4
10
15
6
10
7
6
20
10
12
7
10

Exemple :



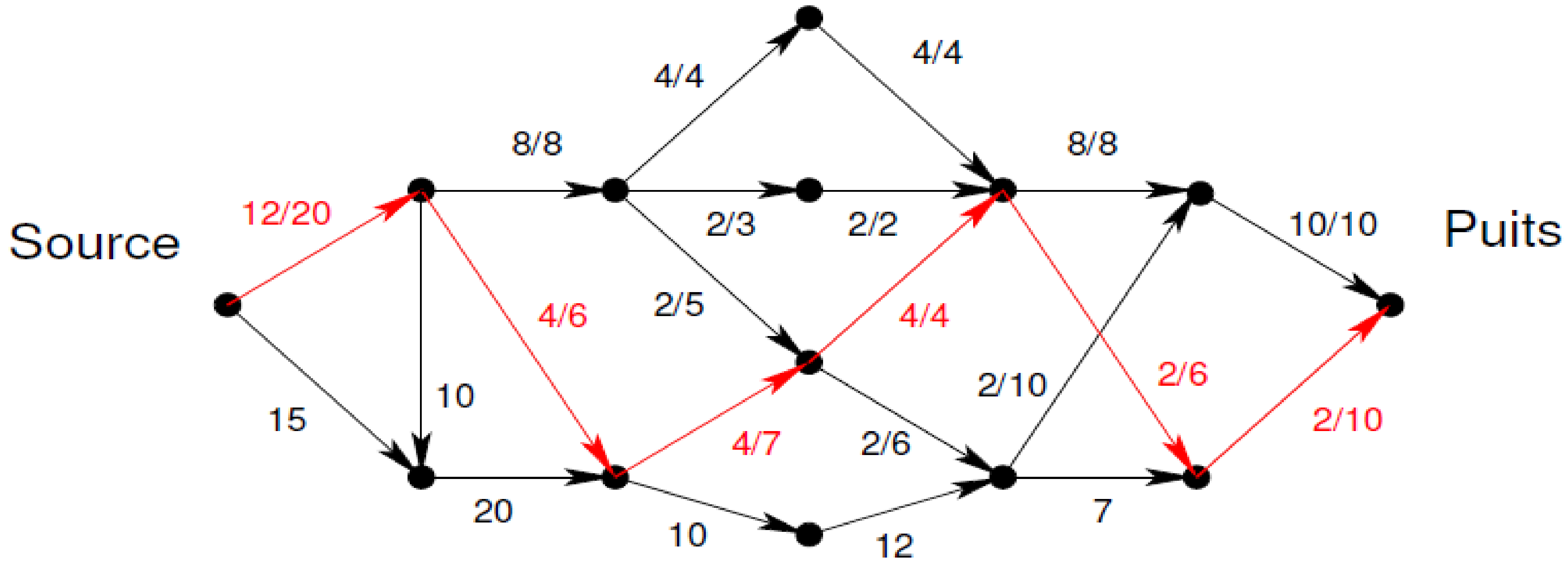
Source
Puits
8/20
15
8/8
10
6
20
4/4
2/3
2/5
7
10
4/4
2/2
2/4
6
12
8/8
10
6
7
8/10
10

Exemple :



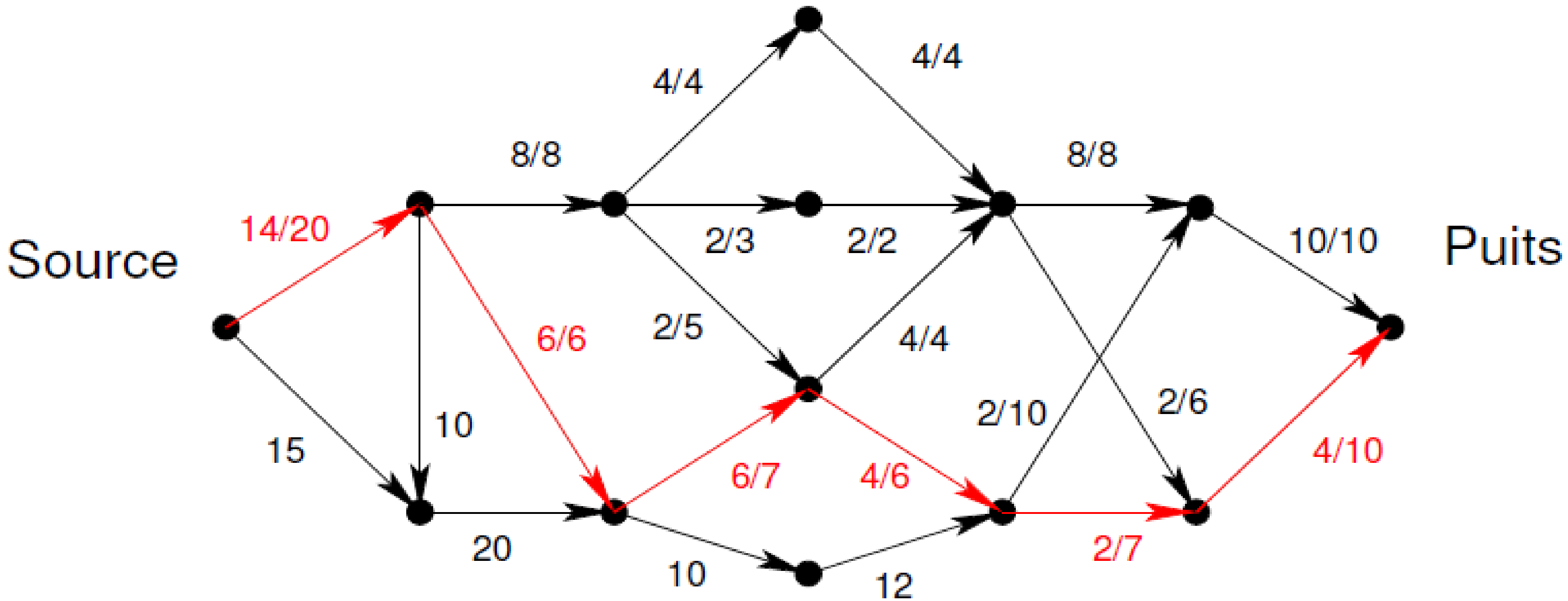
Source
Puits
10/20
15
8/8
10
2/6
20
4/4
2/3
2/5
2/7
10
4/4
2/2
2/4
2/6
12
8/8
6
2/10
7
10/10
10

Exemple :



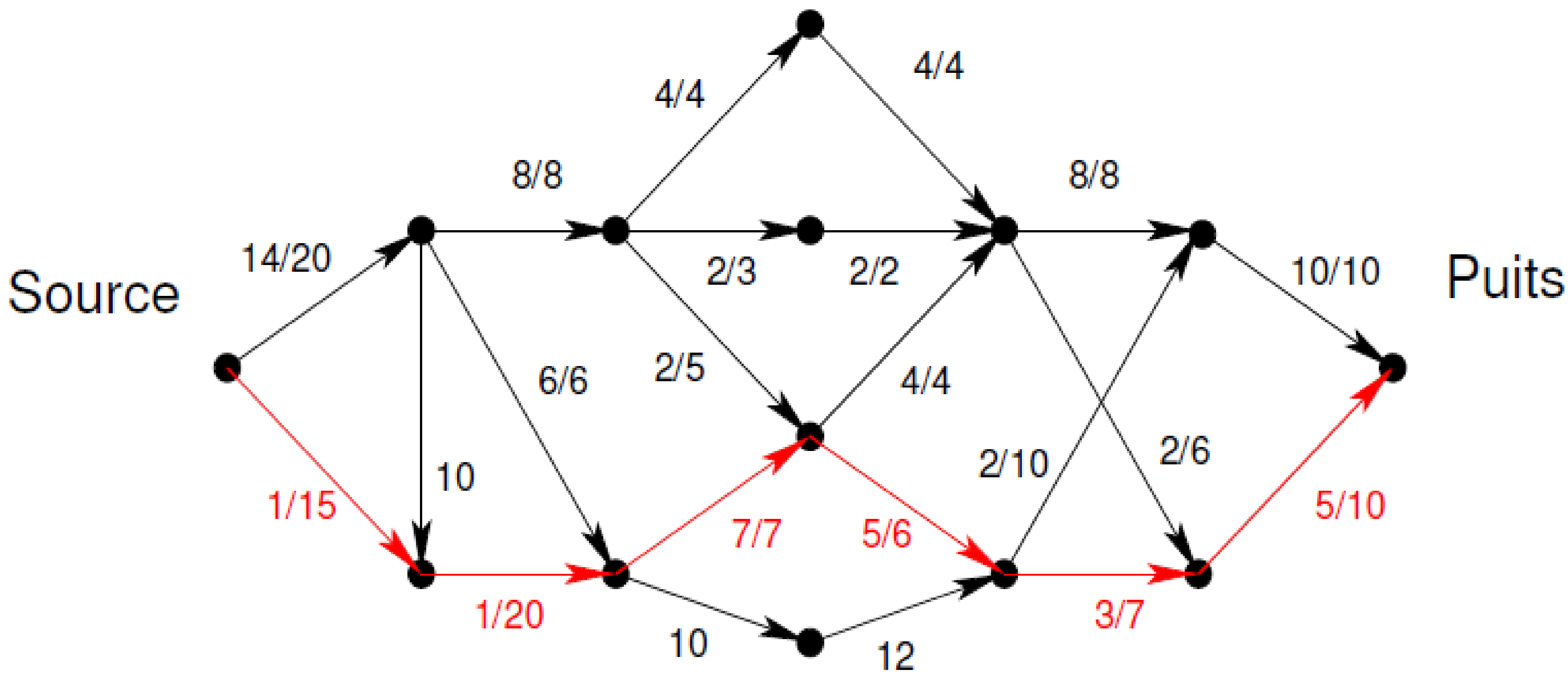
Source
Puits
12/20
15
8/8
10
4/6
20
4/4
2/3
2/5
4/7
10
4/4
2/2
4/4
2/6
12
8/8
2/10
7
2/6
10/10
2/10

Exemple :



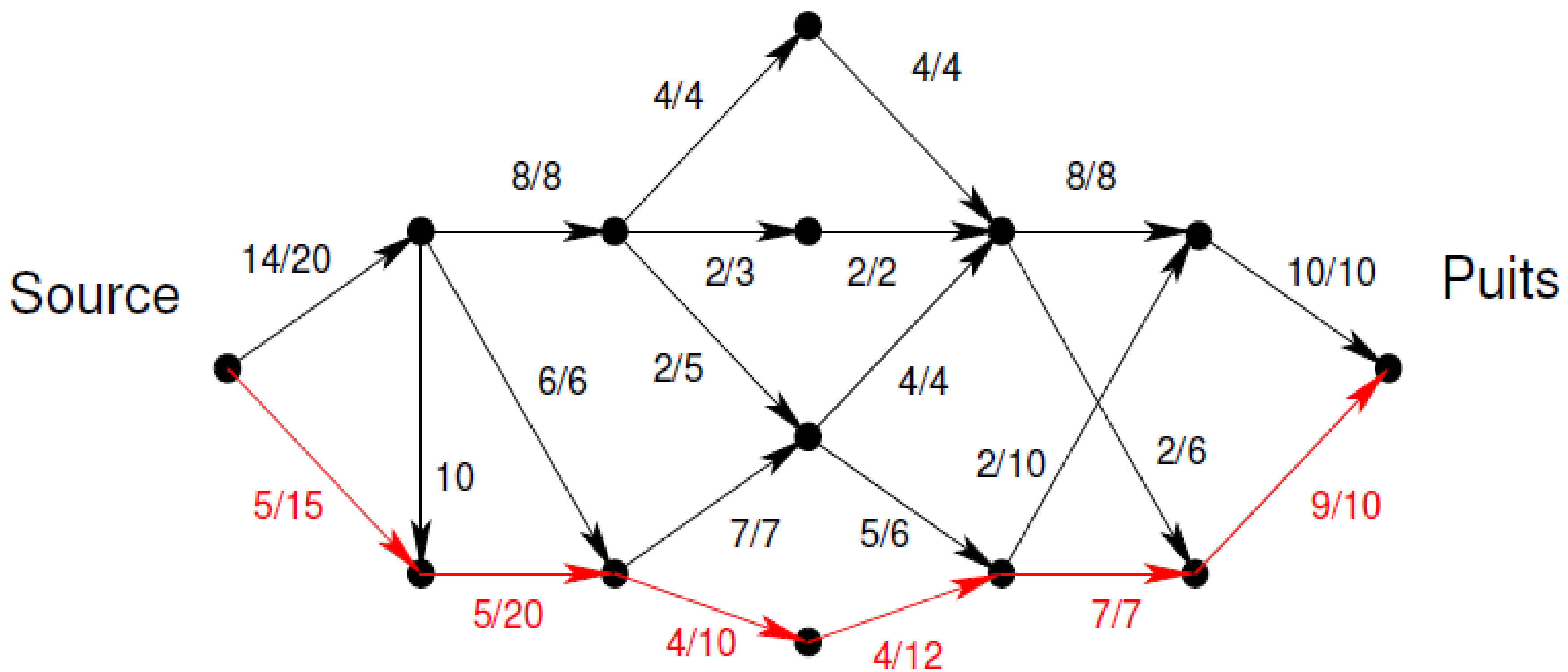
Source
Puits
14/20
15
8/8
10
6/6
20
4/4
2/3
2/5
10
4/4
2/2
4/4
6/7
4/6
12
8/8
2/10
2/6
2/7
10/10
4/10

Exemple :



Source
Puits
14/20
1/15
8/8
4/4
4/4
10
6/6
2/5
2/3
2/2
8/8
10/10
4/4
2/10
2/6
7/7
5/6
1/20
10
12
3/7
5/10

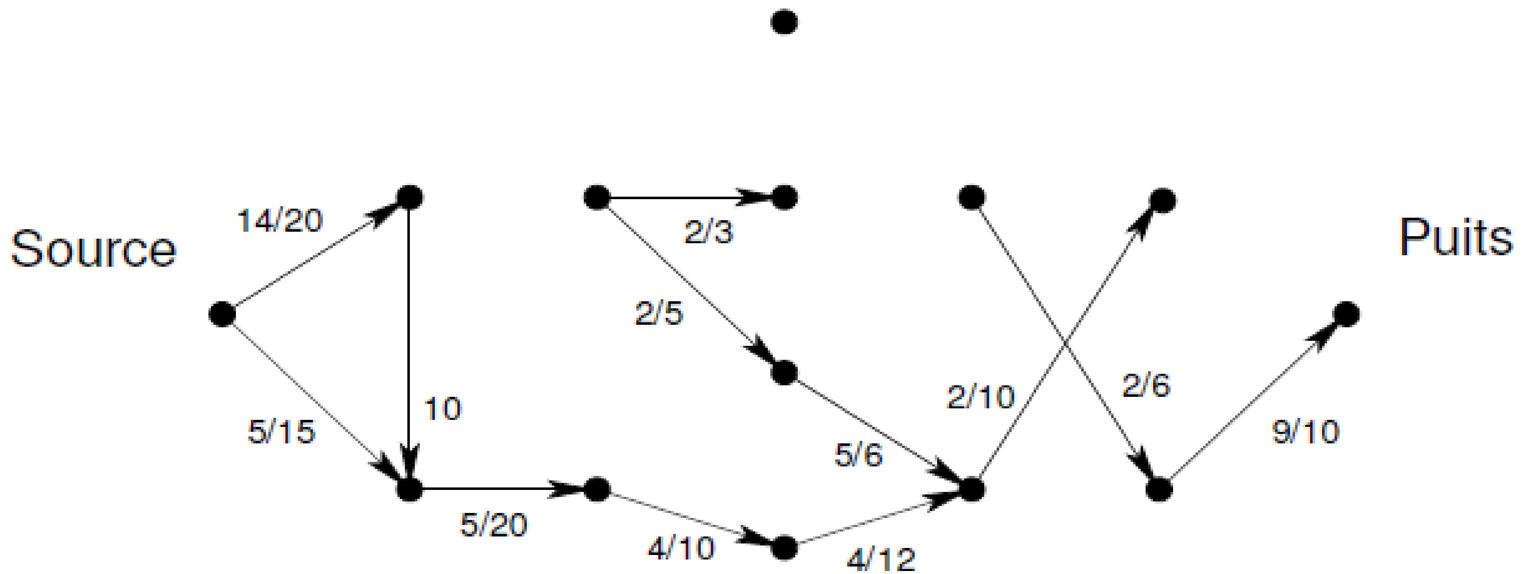
Exemple :



Source
Puits
14/20
5/15
10
8/8
6/6
5/20
4/4
2/3
2/5
4/10
4/4
2/2
4/4
7/7
5/6
4/12
8/8
2/10
2/6
7/7
10/10
9/10

Exemple : Rapport avec la connexité

Il suffit de supprimer les arcs pour lesquels la valeur du flot est égale à la capacité :



Définitions :

Capacité résiduelle :

Cr(si ;sj) = c(si ;sj) - f ((si ; sj)). La capacité résiduelle (Cr), pour un arc (si ; sj) donné, représente la quantité de flot pouvant encore passer par cet arc.

Réseau résiduel :

G_résiduel = (S;A_résiduel) tel que :

A_résiduel = {(si ; sj) de A; Cr(si ;sj) > 0}

le **réseau résiduel**, est le graphe partiel obtenu à partir du graphe d'origine mais dont ont été retiré tous les arcs dont la capacité résiduelle est nulle.

Définitions :

Chemin améliorant ou augmentant :
Un chemin améliorant ou chemin augmentant, est un chemin de G_résiduel, allant de s à p et sans circuit

Capacité résiduelle d’un chemin améliorant :
La capacité résiduelle d’un chemin améliorant est le minimum des capacités résiduelles des arcs appartenant au chemin

Arc saturé :
Un arc est dit saturé si sa capacité résiduelle est nulle.

Algorithme de Ford-Fulkerson

Principe :

Tant qu'il existe un chemin augmentant dans le graphe, on ajoute un flot le long de ce chemin.

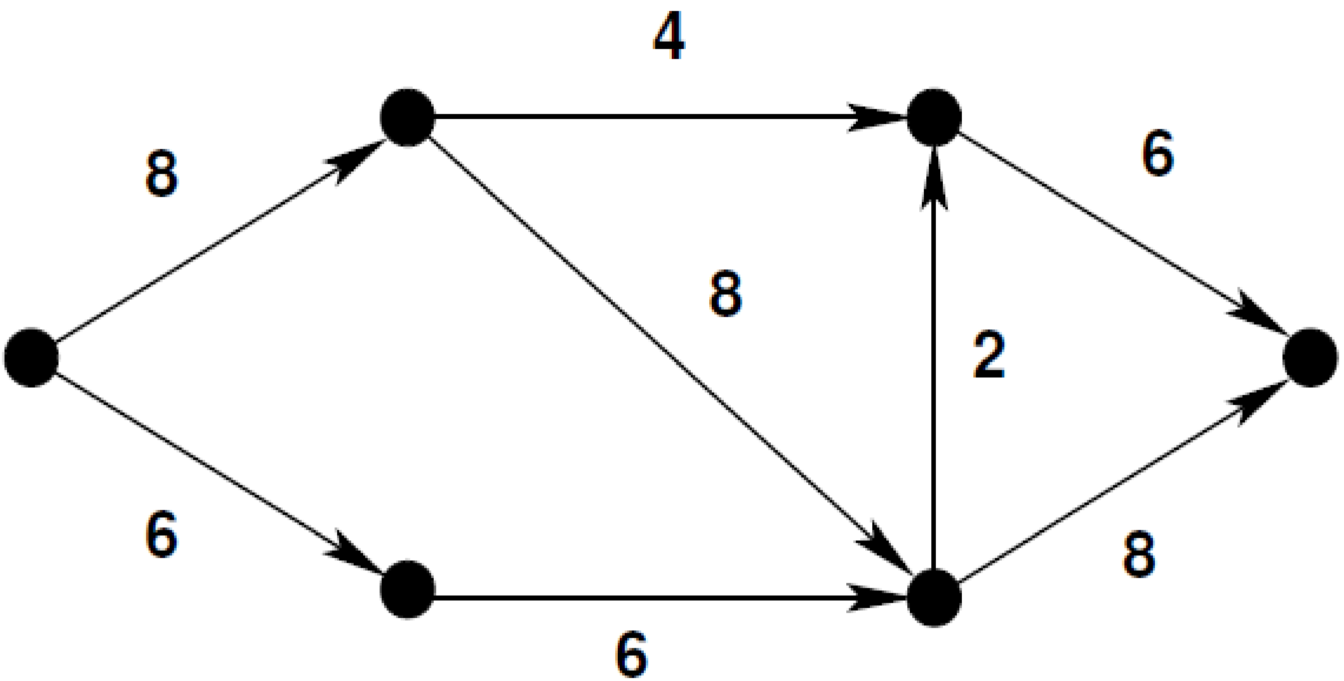
* Créer un graphe résiduel G_résiduel = G
* Initialement tous les arcs sont valués par leur capacité
* Déterminer un chemin augmentant
***TantQue** il existe un chemin augmentant de s vers p **Faire** mettre à jour le graphe résiduel
* Chercher un nouveau chemin augmentant
* **Fin** Tant Que

Remarque :

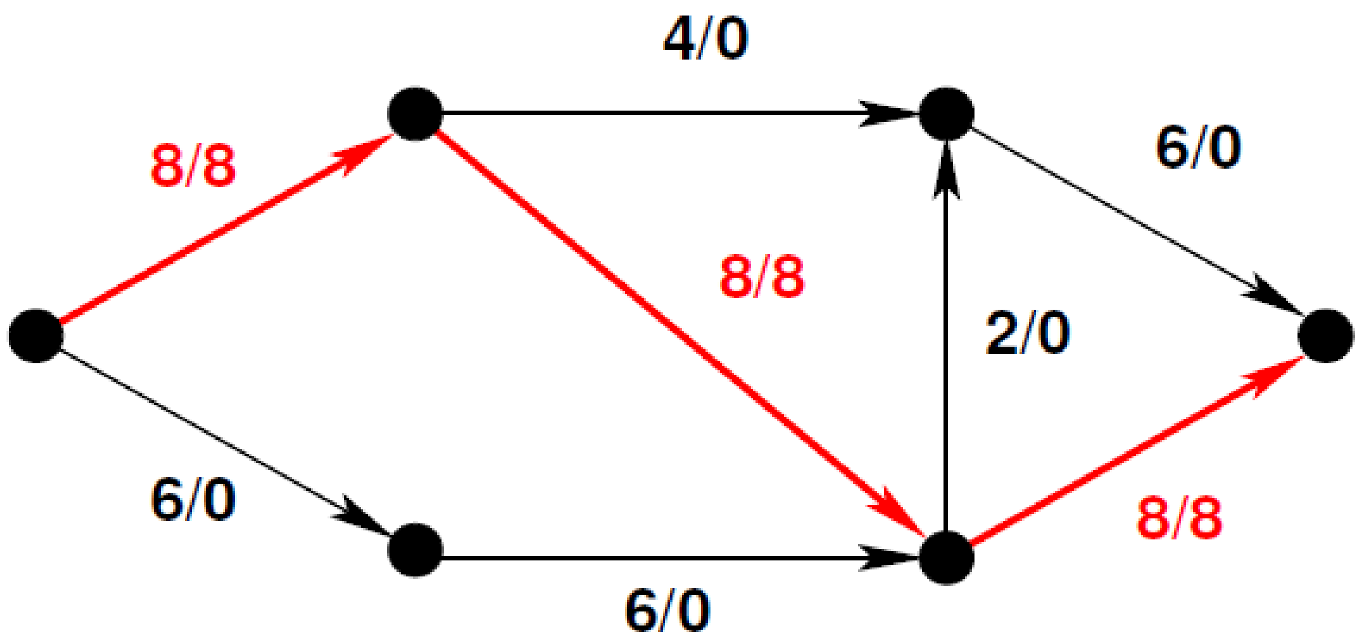
----> déterminer un chemin augmentant
----> mettre à jour le graphe résiduel:
----> Soit c un chemin augmentant
----> Soit Cr la capacité résiduelle de ce chemin
----> Pour tous les arcs (si ; sj) appartenant au chemin Faire mettre à jour la capacité résiduelle de l'arc. finPour

Détermination d'un chemin augmentant:
----> Plus délicat.
----> Examen des chemins du graphe résiduel ! parcours du graphe résiduel.
----> Il peut y avoir remise en question de flots déjà validés.

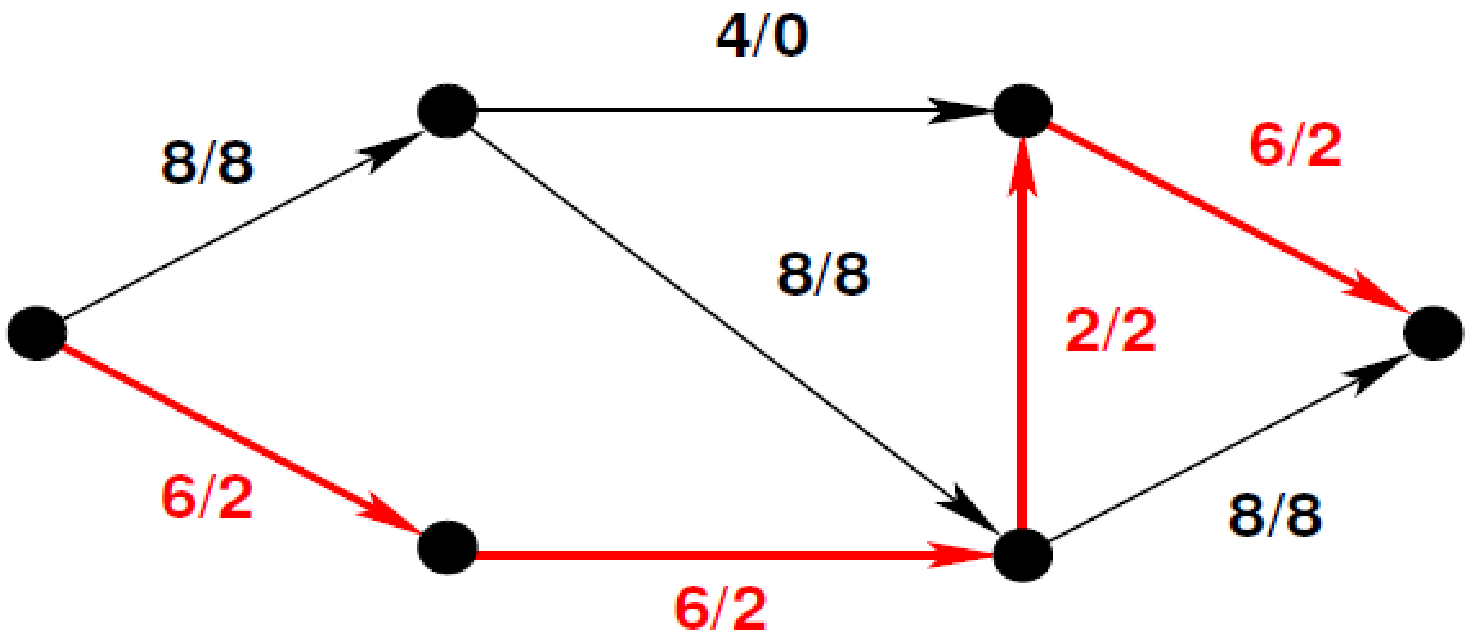
Exemple :



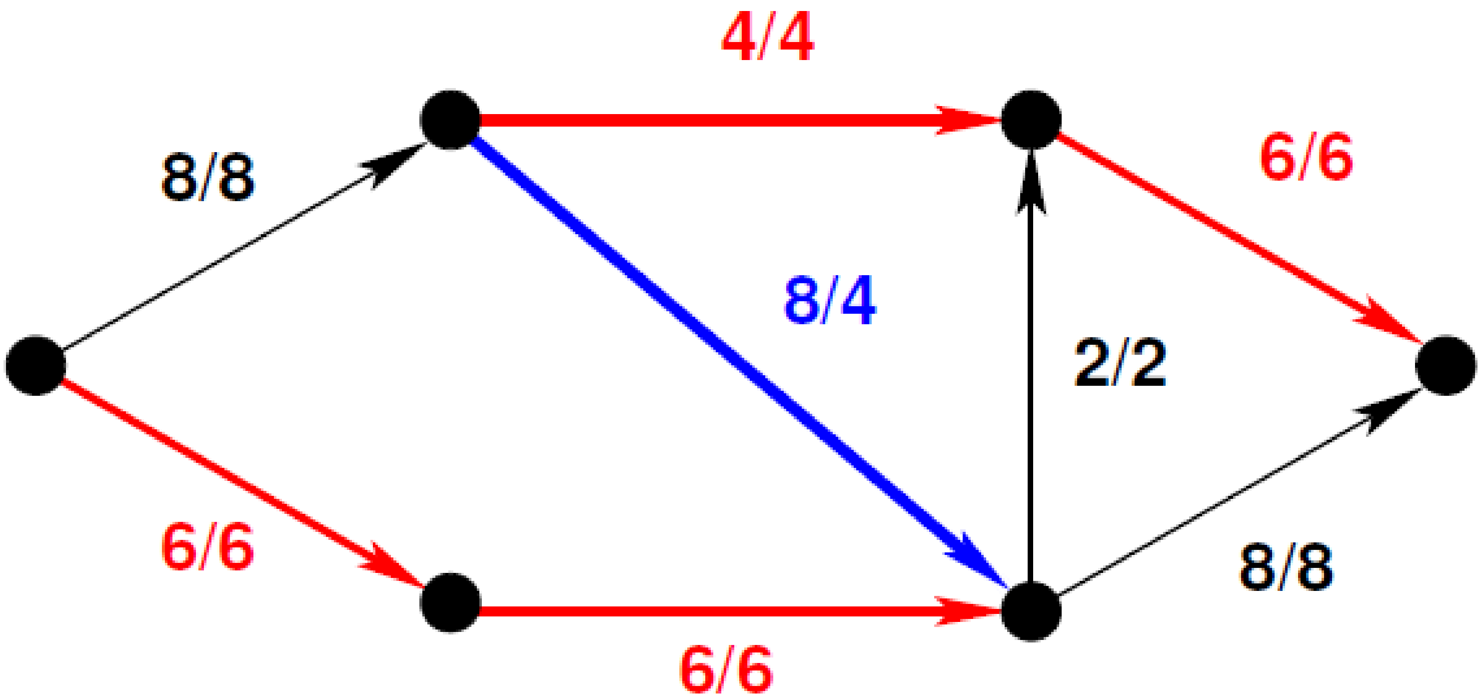
Exemple :



Exemple :



Exemple :



la méthode Hongroise

Introduction :

Les problèmes d'affectation sont des cas spéciaux du problème de transport où la demande associée à chaque destination est égale à 1.

Il existe plusieurs méthodes de résolution, dans cette partie du cours, on présente la "méthode hongroise" qui simplifie la résolution du problème d'affectation.

Algorithmes

Étape 1:
Réduction des lignes : créer une nouvelle matrice des coûts en choisissant le coût minimal sur chaque ligne et en le soustrayant de chaque coût sur la ligne.
Étape 2:
Réduction des colonnes : créer une nouvelle matrice des coûts en choisissant le coût minimal dans chaque colonne et en le soustrayant de chaque coût dans la colonne.
Étape 3:
Déterminer le nombre minimal de lignes nécessaires sur les lignes et les colonnes pour couvrir tous les zéros. Si ce nombre est égal au nombre de lignes (ou colonnes), la matrice est réduite; aller à l'étape 5. Si ce nombre est inférieur au nombre de lignes (ou colonnes), aller à l'étape 4.
Étape 4:
Trouver la cellule de valeur minimum non-couverte par une ligne. Soustraire cette valeur de toutes les cellules noncouvertes. Ajouter cette valeur aux cellules situées à l'intersection de deux lignes. Retourner à l'étape 3.
Étape 5:
Déterminer la solution optimale.

Exemple :

Matrice des coûts

	USINE			
V.P.	**1**	**2**	**3**	**4**
F	24	10	21	11
M	14	22	10	15
O	15	17	20	19
P	11	19	14	13

Ex. Le coût d'affecter le V.P. «P» à l'usine 4 est de 13.

Exemple :

La première ligne devient: 24-10=14, 10-10=0, 21-10=11, 11-10=1

	USINE				
V.P.	**1**	**2**	**3**	**4**	**RÉDUIT DE:**
F	14	0	11	1	10
M	4	12	0	5	10
O	0	2	5	4	15
P	0	8	3	2	11

Exemple :

	USINE			
V.P.	**1**	**2**	**3**	**4**
F	14	0	11	0
M	4	12	0	4
O	0	2	5	3
P	0	8	3	1
RÉDUIT DE:	0	0	0	1

Exemple :

Dans ce cas, le nombre minimal de lignes est de 3.

	USINE			
V.P.	1	2	3	4
F	14	0	11	0
M	4	12	0	4
O	0	2	5	3
P	0	8	3	1

Donc, on va à l'étape 4.

Exemple :

	USINE			
V.P.	**1**	**2**	**3**	**4**
~~F~~	~~14~~	~~0~~	~~11~~	~~0~~
M	4	12	0	4
O	0	2	5	3
P	0	8	5	1

Valeur minimum

Exemple :

+ 1

	USINE			
V.P	1	2	3	4
F	15	0	12	0
M	4	11	0	3
O	0	1	5	2
P	0	7	1	0

- 1

Exemple :

Maintenant, le nombre minimal de lignes est de 4.

	USINE			
V.P.	1	2	3	4
F	15	0	12	0
M	4	11	0	3
O	0	1	5	2
P	0	7	3	0

Donc, on passe à l'étape 5.

Résultat :

<u>Note</u>: *P1 ne pourrait pas être choisi car l'affectation de «O» ne serait pas de coût minimal.*

	USINE			
V.P.	**1**	**2**	**3**	**4**
F	15	~~0~~	12	0
M	4	11	~~0~~	3
O	~~0~~	1	5	2
P	0	7	3	~~0~~

AFFECTATION		
V.P.	USINE	COÛT
F	2	10
M	3	10
O	1	15
P	4	13
COÛT TOTAL		48

Exercice : problème d'affectation

- Sur un chantier, il faut affecter 4 ouvriers à 4 tâches.
- L'ouvrier i effectue la tâche j en c_{ij} heures :

Ouvrier	Tâche 1	Tâche 2	Tâche 3	Tâche 4
A	9	2	7	8
B	6	4	3	7
C	5	8	1	8
D	7	6	9	4

- Comment répartir les tâches pour que le nombre total d'heures soit le plus petit possible ?