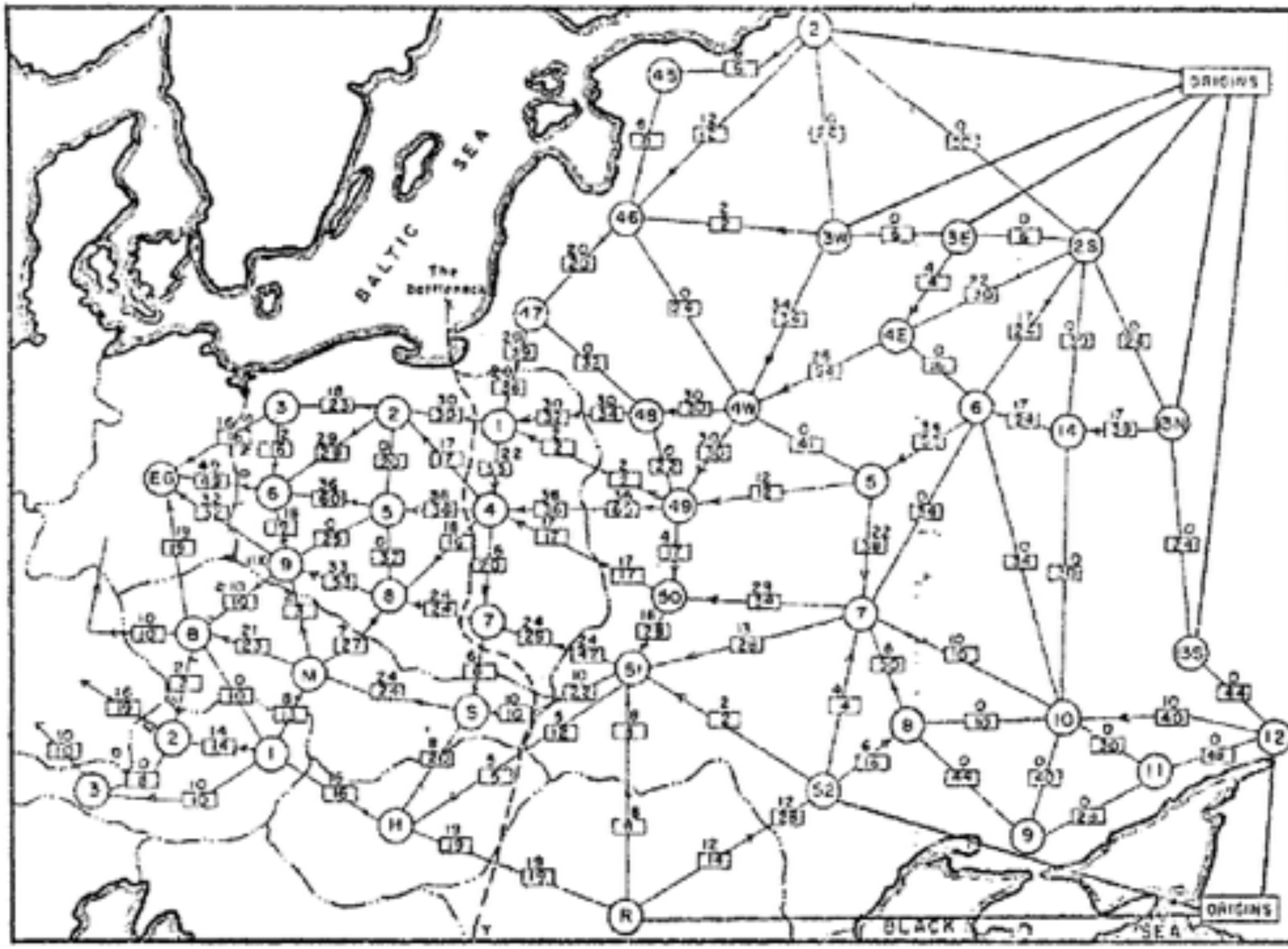
Le problème du flot maximum

Bibliographie

Algorithm Design, Jon Kleinberg, Eva Tardos

Transparents : traduction de ceux du livre, préparés par Kevin Wayne

Réseau ferroviaire Soviétique, 1955



Référence: On the history of the transportation and maximum flow problems.
Alexander Schrijver in Math Programming, 91: 3, 2002.

Flot maximum et Coupe minimum

- Problèmes algorithmiques très riches.
- Problèmes duaux.

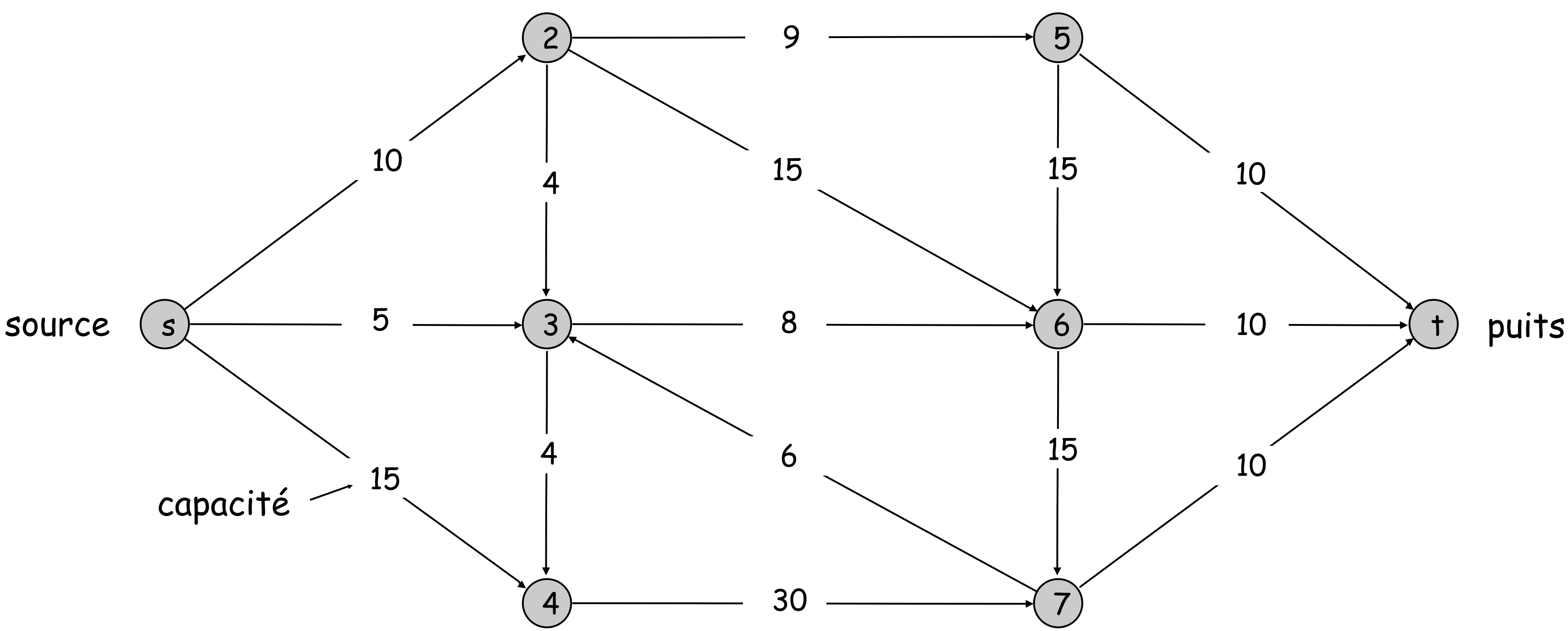
Applications / réductions.

- Data mining.
- Sélection de projets.
- Ordonnancement de vols.
- Couplage biparti.
- Segmentation d'image.
- Connexité dans les réseaux.

Le problème de la coupe minimum

Réseau de flot.

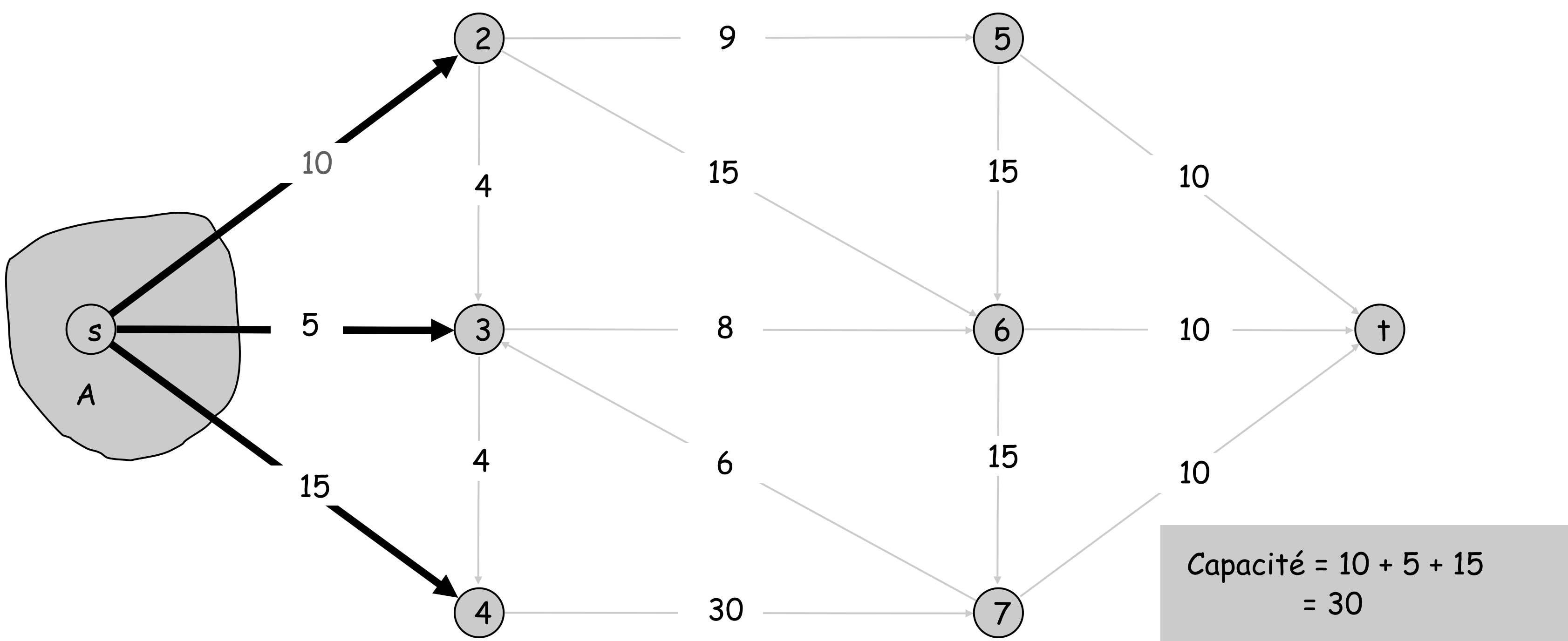
- Abstraction d'un liquide coulant à travers les tuyaux d'un réseau.
- G = (V, E) = graphe orienté, sans arêtes parallèles.
- Deux sommets particuliers : s = source, t = puits.
- c(e) = capacité de l'arête e.



Coupes

Déf. Une s-t coupe est une partition (A, B) de V avec s ∈ A et t ∈ B.

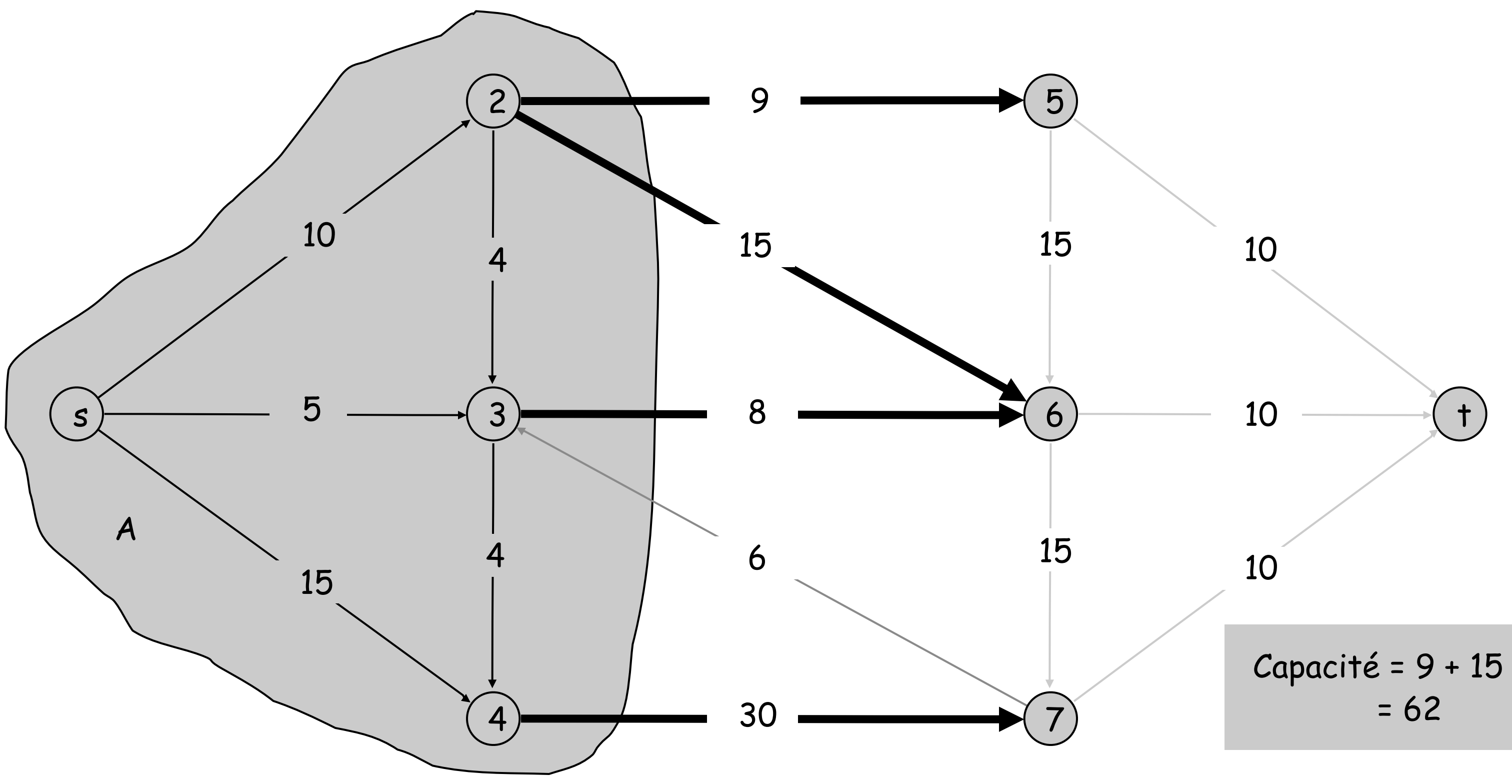
Déf. La capacité de la coupe (A, B) : $cap(A,B) = \sum_{e \text{ sortant de } A} c(e)$



Coupes

Déf. Une s-t coupe est une partition (A, B) de V avec s $\in$ A et t $\in$ B.

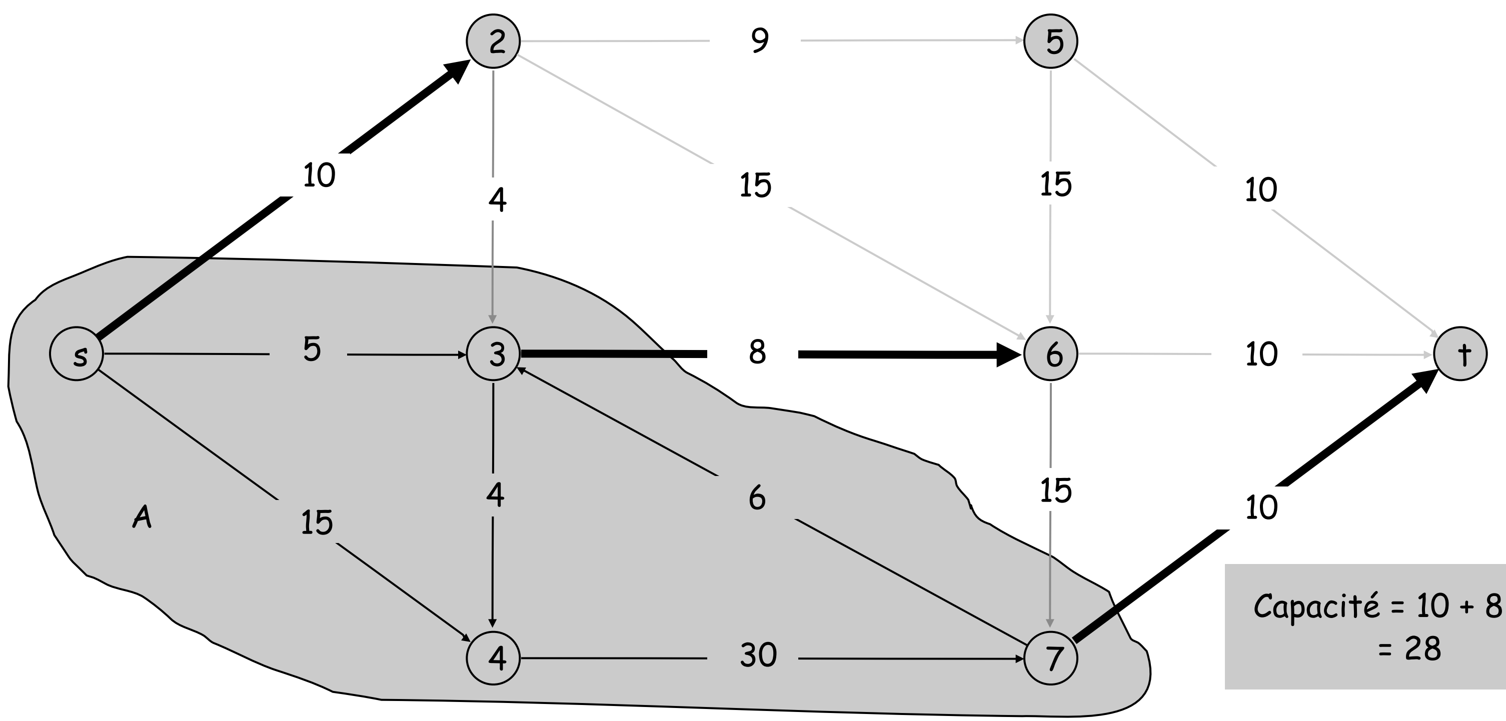
Déf. La capacité de la coupe (A, B) : $$cap(A, B) = \sum_{e \text{ sortant de } A} c(e)$$



Capacité = 9 + 15 + 8 + 30
= 62

Le problème de la coupe minimum

Le problème de la s-t coupe minimum.
Déterminer une s-t coupe de capacité minimum.



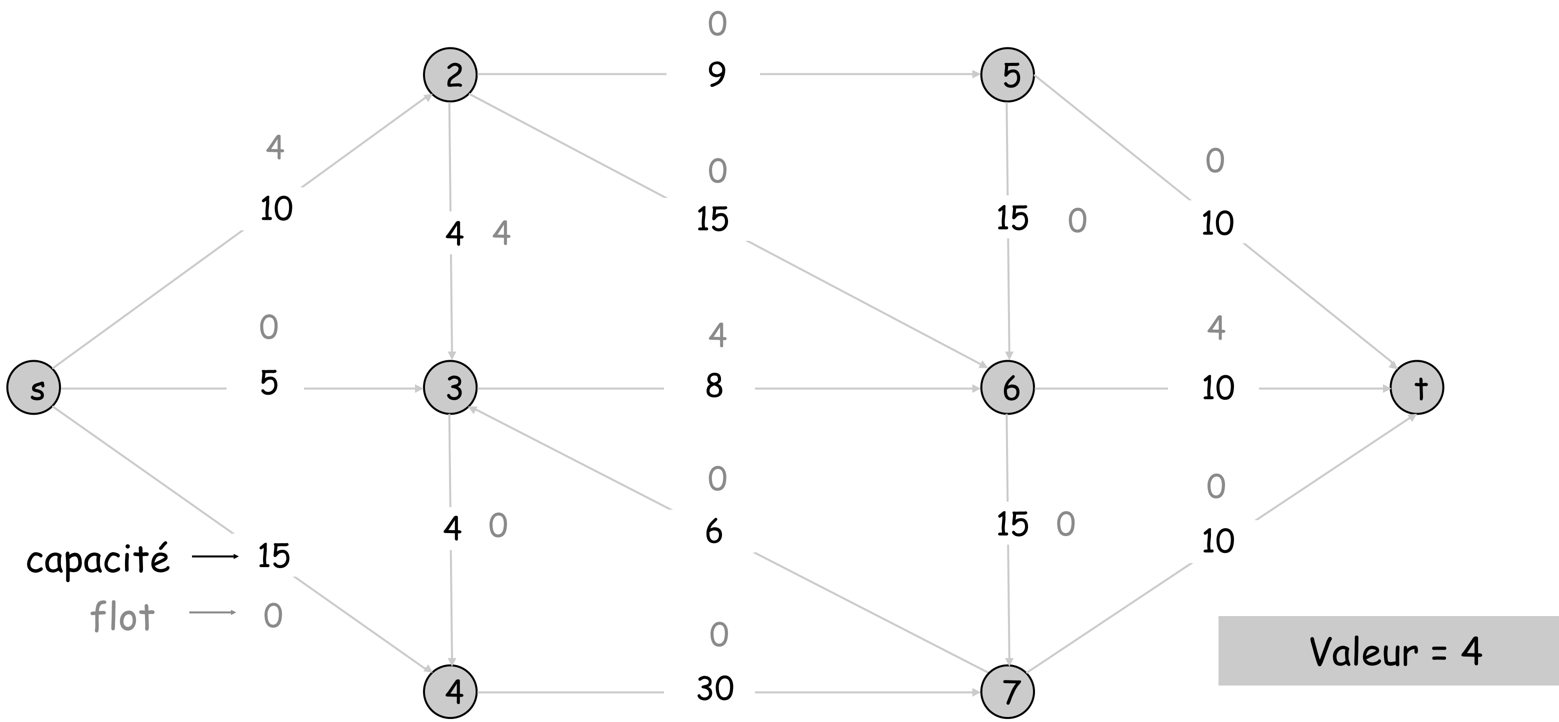
Capacité = 10 + 8 + 10
= 28

Flots

Déf. Un s-t flot est une fonction qui satisfait:

- Pour chaque $e \in E$: $0 \leq f(e) \leq c(e)$ (capacité)
- Pour chaque $v \in V - \{s, t\}$: $\sum_{e \text{ entrant à } v} f(e) = \sum_{e \text{ sortant de } v} f(e)$ (conservation)

Déf. La valeur d'un flot f est : $v(f) = \sum_{e \text{ sortant de } s} f(e)$

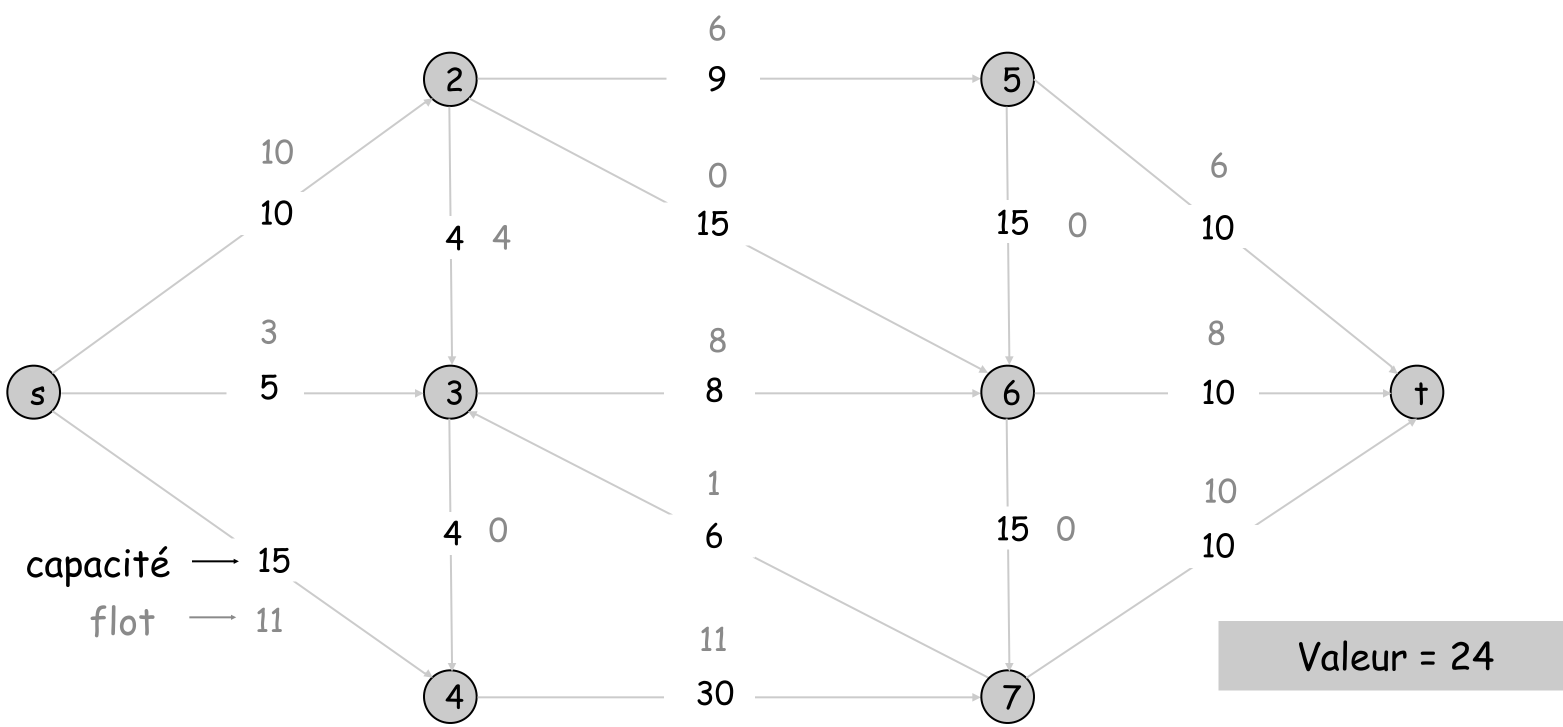


Flots

Déf. Un s-t flot est une fonction qui satisfait:

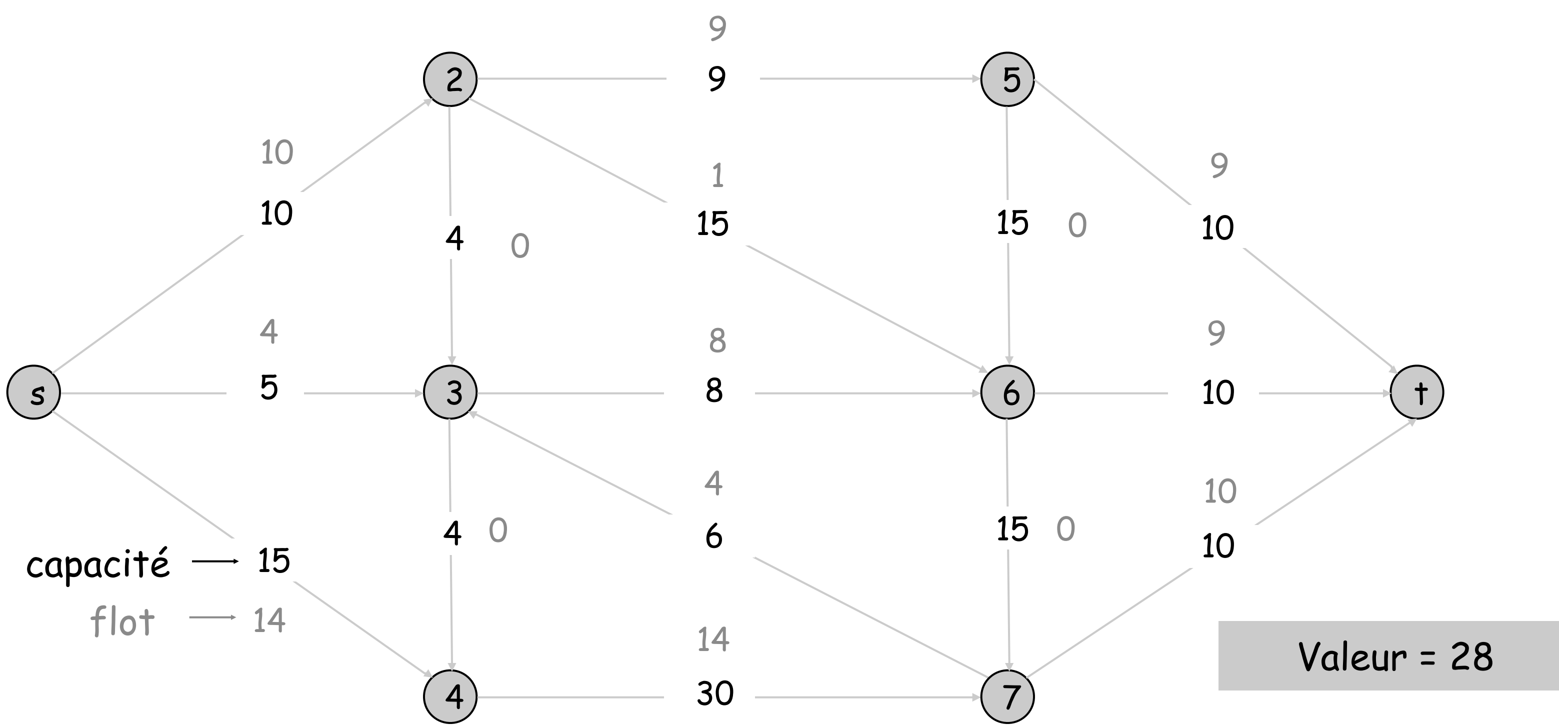
- Pour chaque $e \in E$: $0 \le f(e) \le c(e)$ (capacité)
- Pour chaque $v \in V - \{s, t\}$: $\sum_{e \text{ entrant à } v} f(e) = \sum_{e \text{ sortant de } v} f(e)$ (conservation)

Déf. La valeur d'un flot f est : $v(f) = \sum_{e \text{ sortant de } s} f(e)$



Le problème du flot maximum

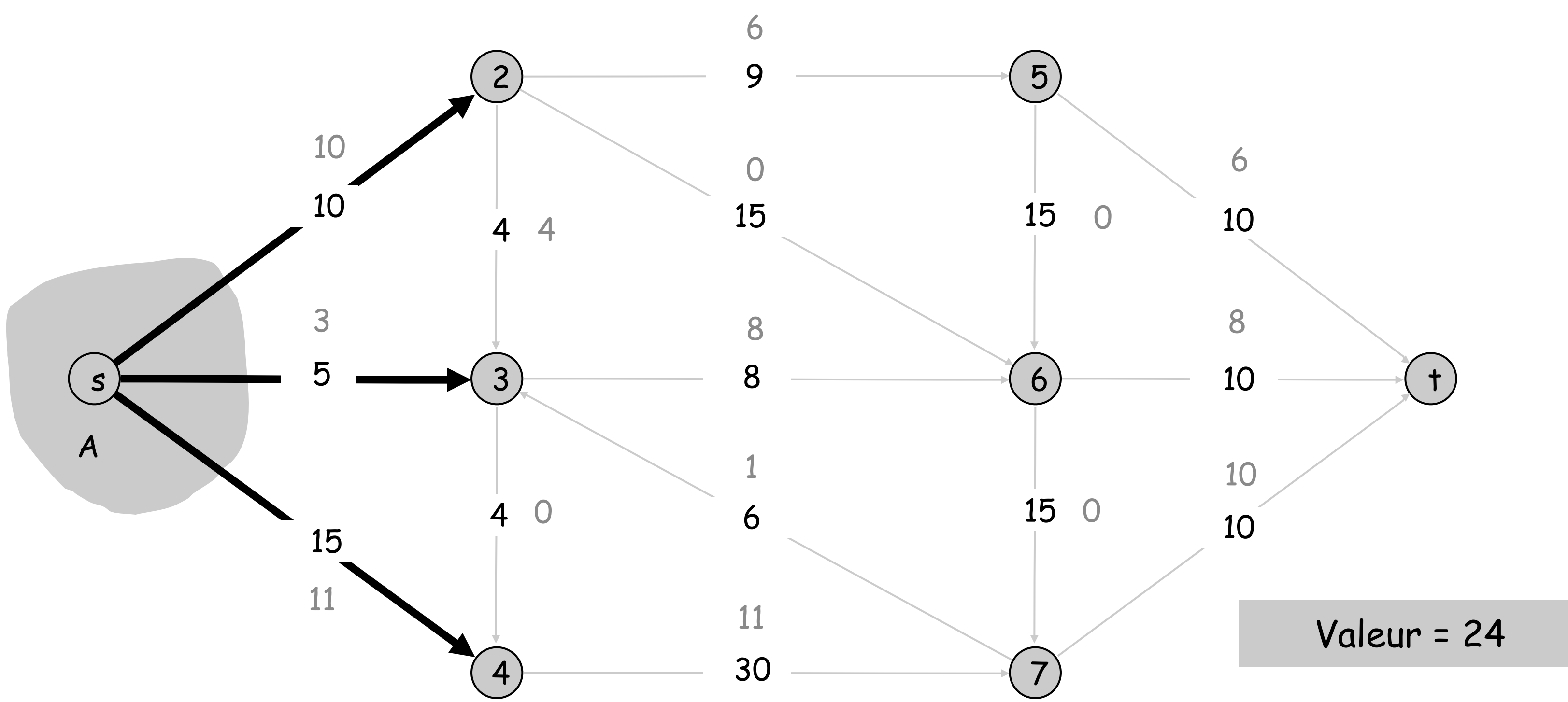
Le problème du flot max. Déterminer un s-t flot de valeur maximum.



Flots et Coupes

Lemme de la valeur du flot. Soit f un flot arbitraire, et soit (A, B) une s-t coupe arbitraire. Alors, le flot net envoyé à travers la coupe est égal à la quantité sortant de s.

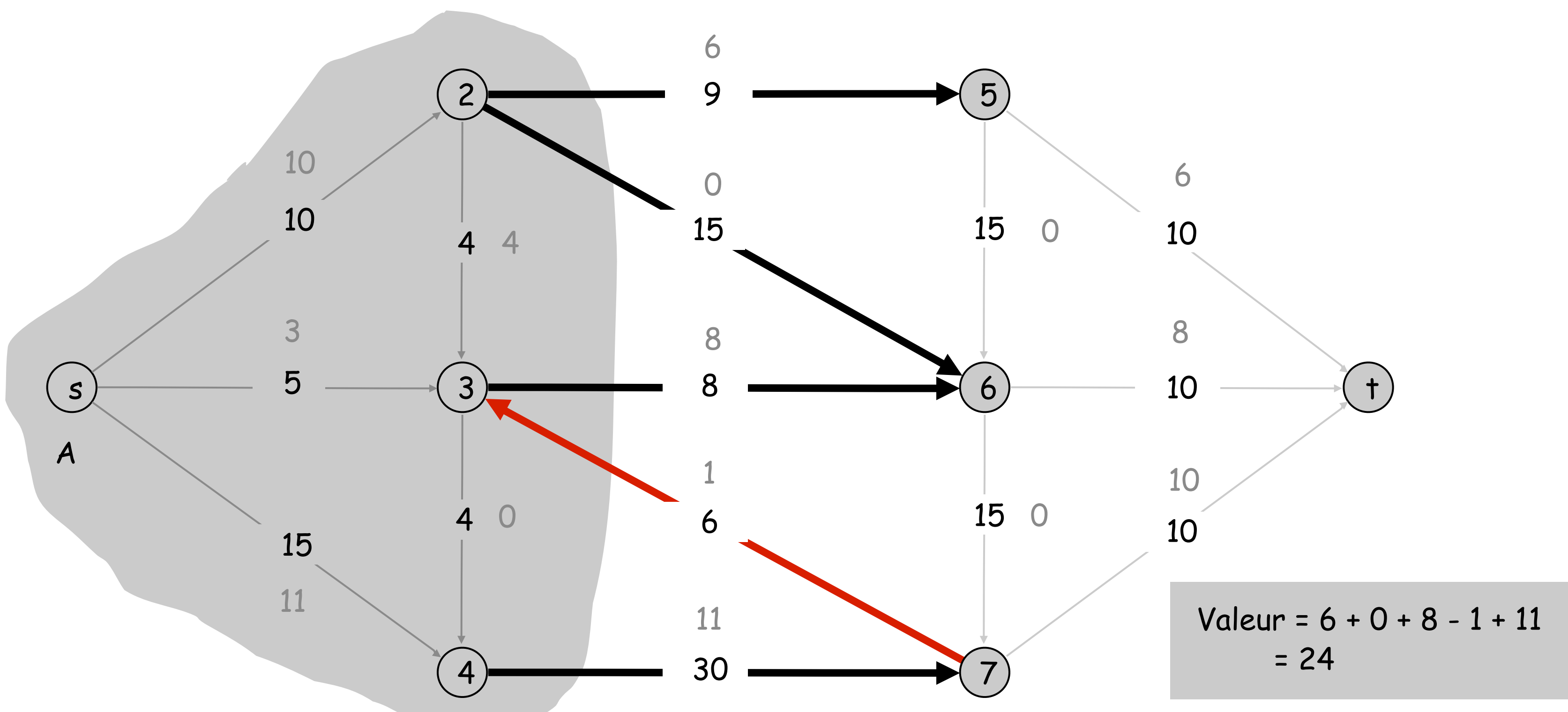
$$\sum_{e \text{ sortant de } A} f(e) - \sum_{e \text{ entrant à } A} f(e) = v(f)$$



Flots et Coupes

Lemme de la valeur du flot. Soit f un flot arbitraire, et soit (A, B) une s-t coupe arbitraire. Alors, le flot net envoyé à travers la coupe est égal à la quantité sortant de s.

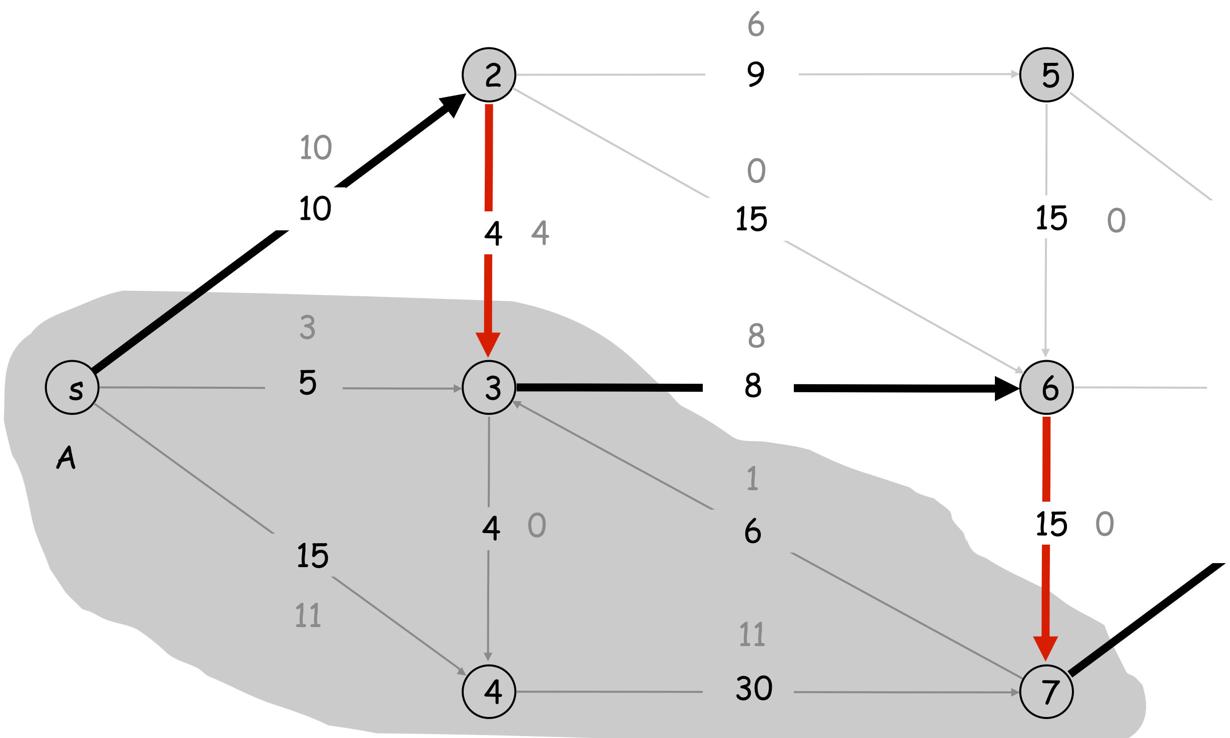
$$\sum_{e \text{ sortant de } A} f(e) - \sum_{e \text{ entrant à } A} f(e) = v(f)$$



Flots et Coupes

Lemme de la valeur du flot. Soit f un flot arbitraire, et soit (A, B) une s-t coupe arbitraire. Alors, le flot net envoyé à travers la coupe est égal à la quantité sortant de s.

$$\sum_{e \text{ out of } A} f(e) \; - \; \sum_{e \text{ in to } A} f(e) \; = \; v(f)$$



Valeur = 10 - 4 + 8 - 0 + 10
= 24

Flots et coupes

Lemme de la valeur du flot. Soit f un flot arbitraire, et soit (A, B) une s-t coupe arbitraire. Alors

$$\sum_{e \text{ out of } A} f(e) - \sum_{e \text{ in to } A} f(e) = v(f).$$

Preuve.

$$v(f) = \sum_{e \text{ out of } s} f(e)$$

par la conservation de flot, tous les termes mis à part v = s sont 0

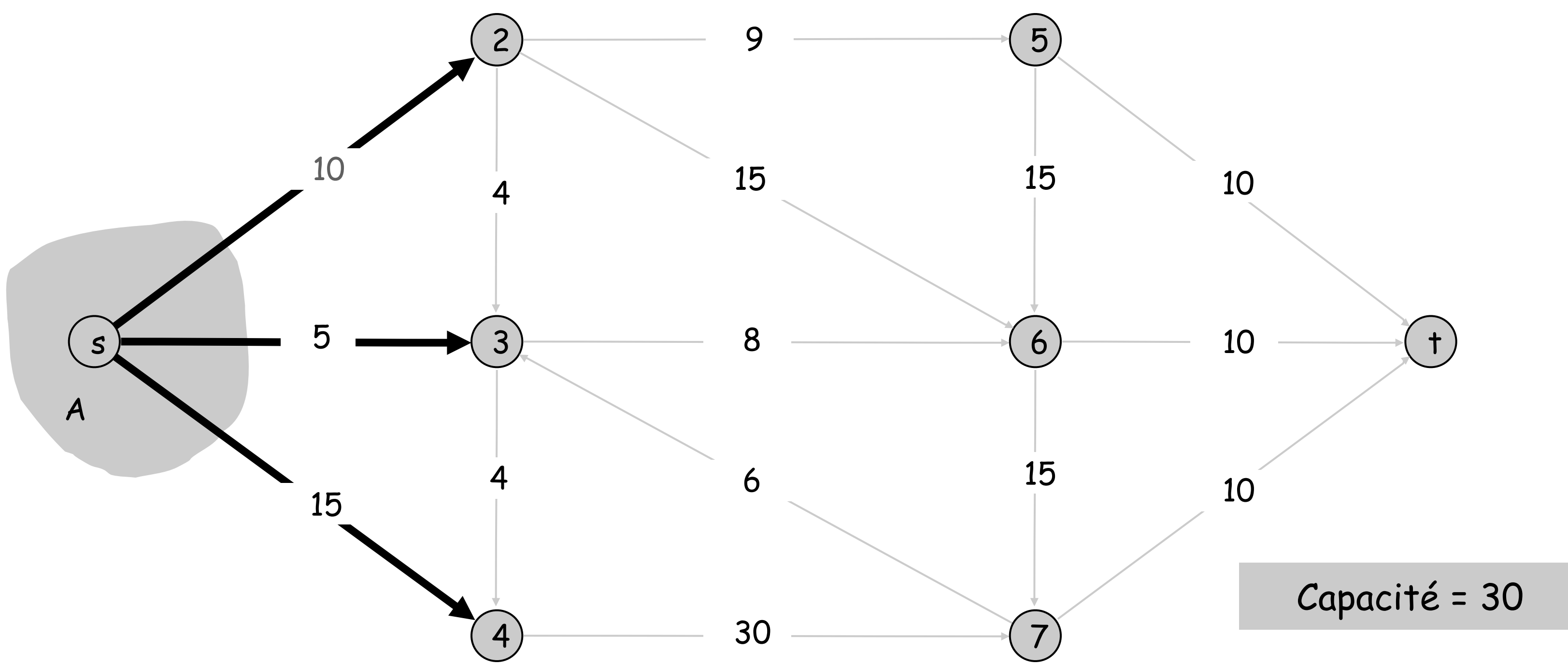
$$\longrightarrow \quad = \sum_{v \in A} \left(\sum_{e \text{ out of } v} f(e) - \sum_{e \text{ in to v}} f(e) \right)$$

$$= \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ in to A}} f(e).$$

Flots et Coupes

Dualité faible. Soit f un flot arbitraire, et soit (A, B) une s-t coupe arbitraire. Alors la valeur du flot est au plus la capacité de la coupe.

Capacité de la coupe = 30 $\Rightarrow$ Valeur du flot $\leq$ 30

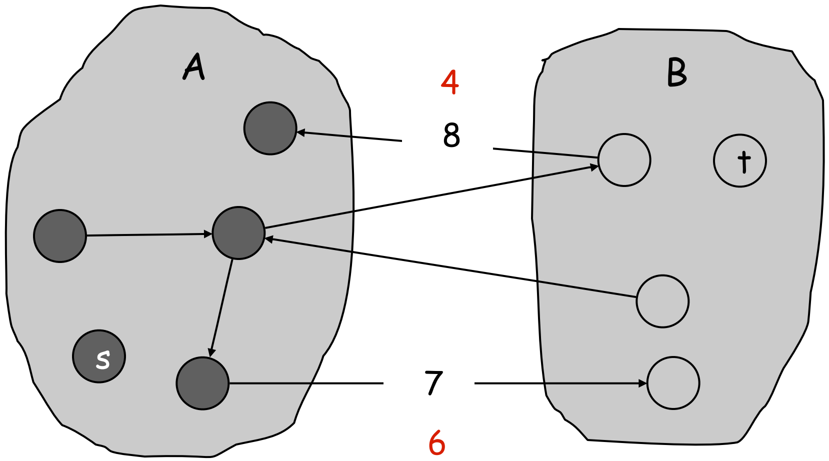


Flots et coupes

Dualité faible. Soit f un flot arbitraire. Alors, pour toute s-t coupe (A, B) nous avons $v(f) \leq cap(A, B)$.

Preuve.

$$\begin{aligned} v(f) &= \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ in to } A} f(e) \\ &\leq \sum_{e \text{ out of } A} f(e) \\ &\leq \sum_{e \text{ out of } A} c(e) \\ &= \operatorname{cap}(A, B) \quad \blacksquare \end{aligned}$$

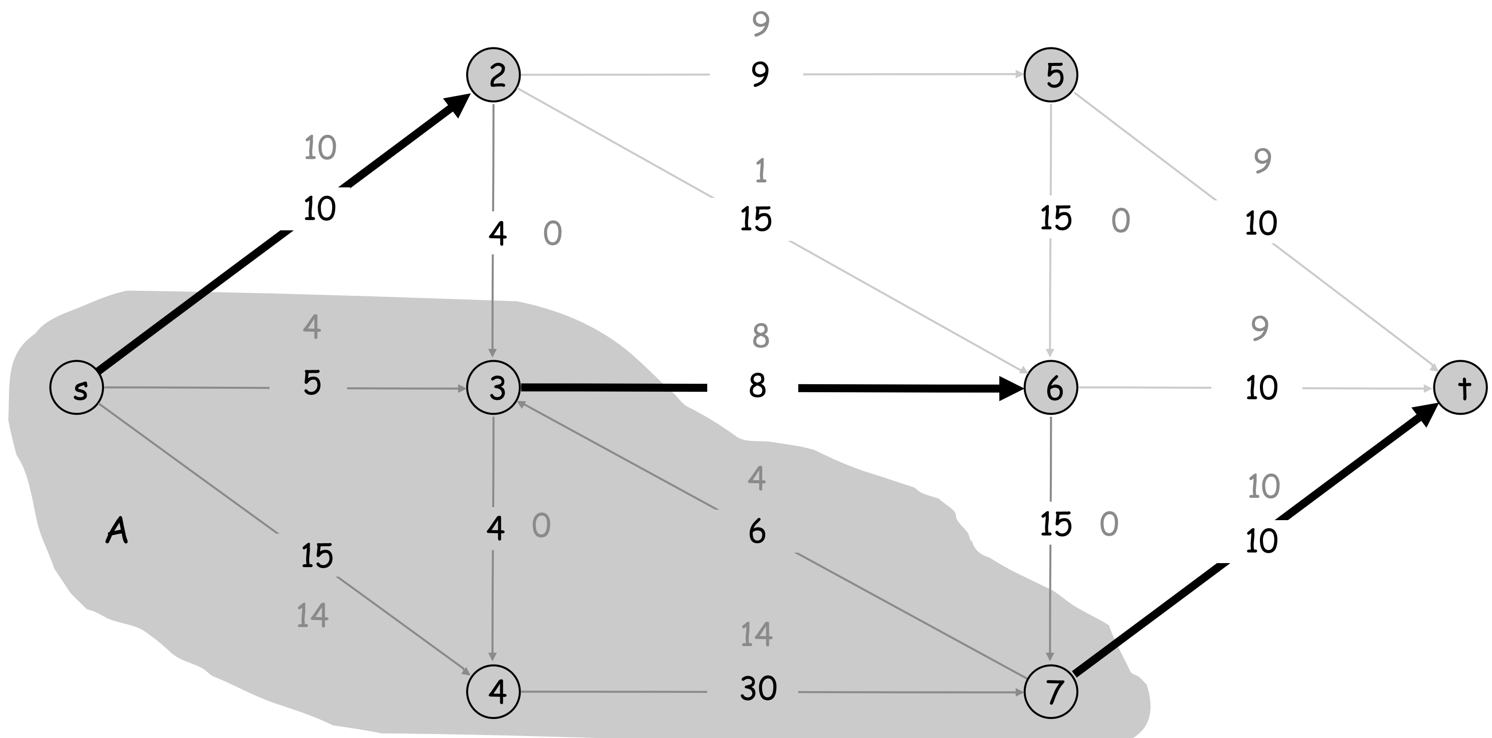


Certificat d'optimalité

Corollaire. Soit f un flot arbitraire, et soit (A, B) une coupe arbitraire. Si v(f) = cap(A, B), alors f est un flot max et (A, B) une coupe min.

Valeur du flot = 28

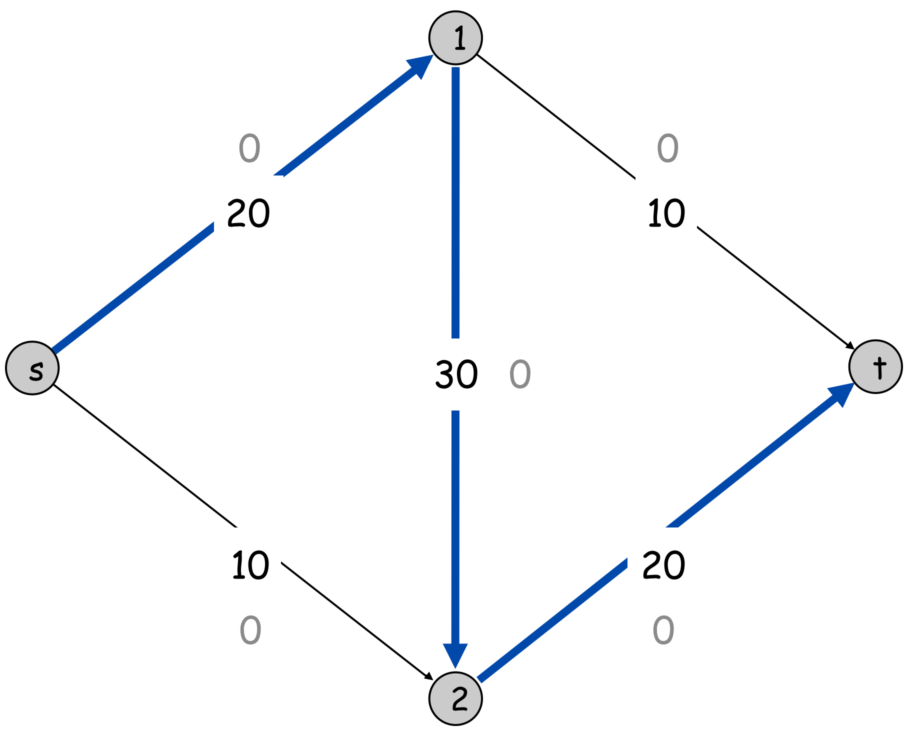
Capacité de la coupe = 28 $\Rightarrow$ Valeur du flot $\leq$ 28



Vers un algorithme de flot maximum

Algorithme glouton.

- On pose f(e) = 0 pour tous les arcs $e \in E$.
- Déterminer un s-t chemin P tel que pour tout arc $e \in P$, $f(e) < c(e)$.
- Augmenter le flot le long du chemin P.
- On répéte jusqu'à ce qu'on ne puisse plus trouver de chemin.

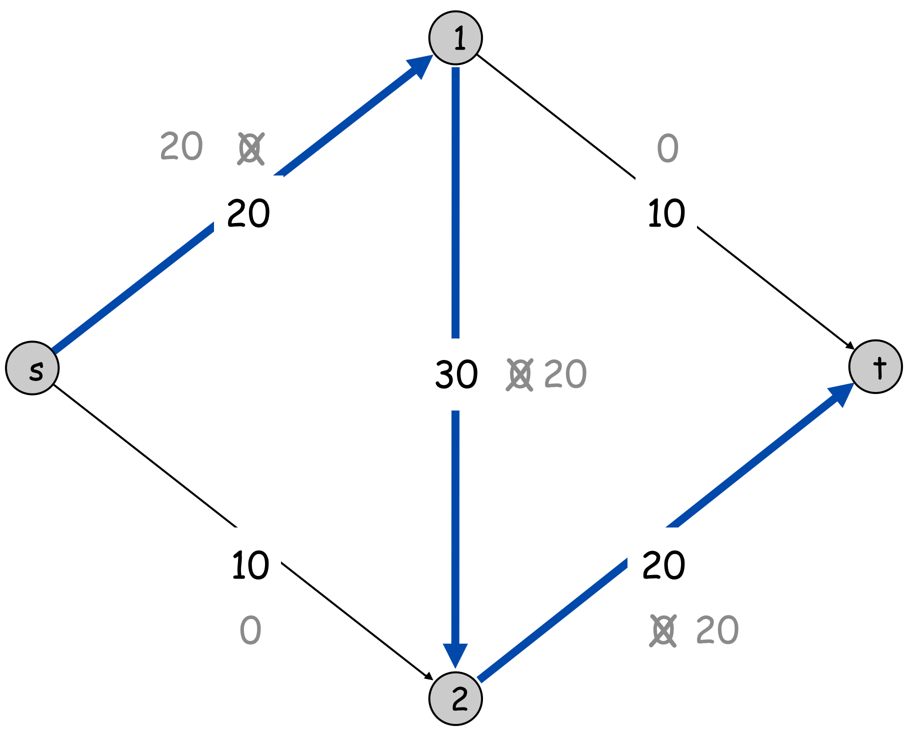


Valeur du flot = 0

Vers un algorithme de flot maximum

Algorithme glouton.

- On pose f(e) = 0 pour tous les arcs $e \in E$.
- Déterminer un s-t chemin P tel que pour tout arc $e \in P$, $f(e) < c(e)$.
- Augmenter le flot le long du chemin P.
- On répéte jusqu'à ce qu'on ne puisse plus trouver de chemin.



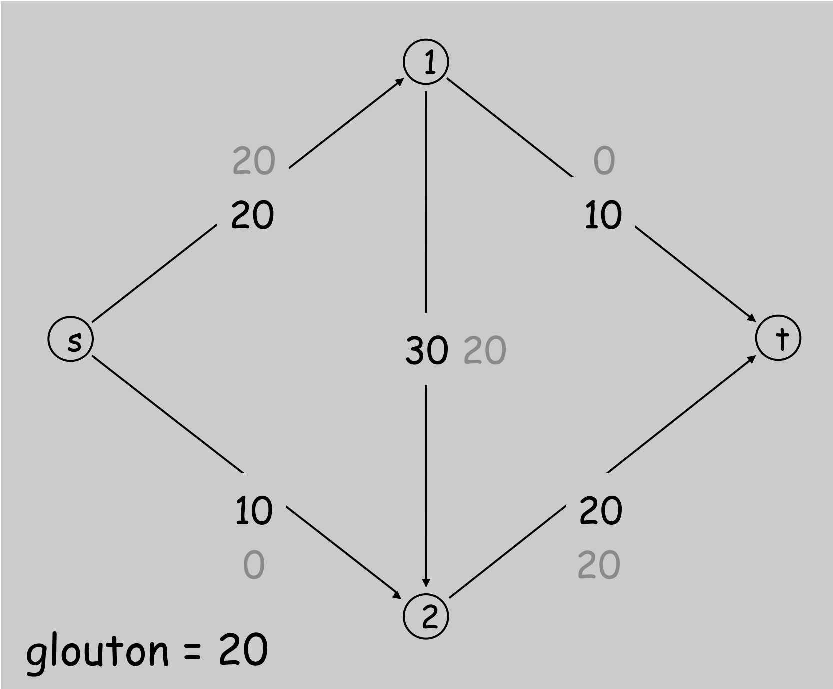
Valeur du flot = 20

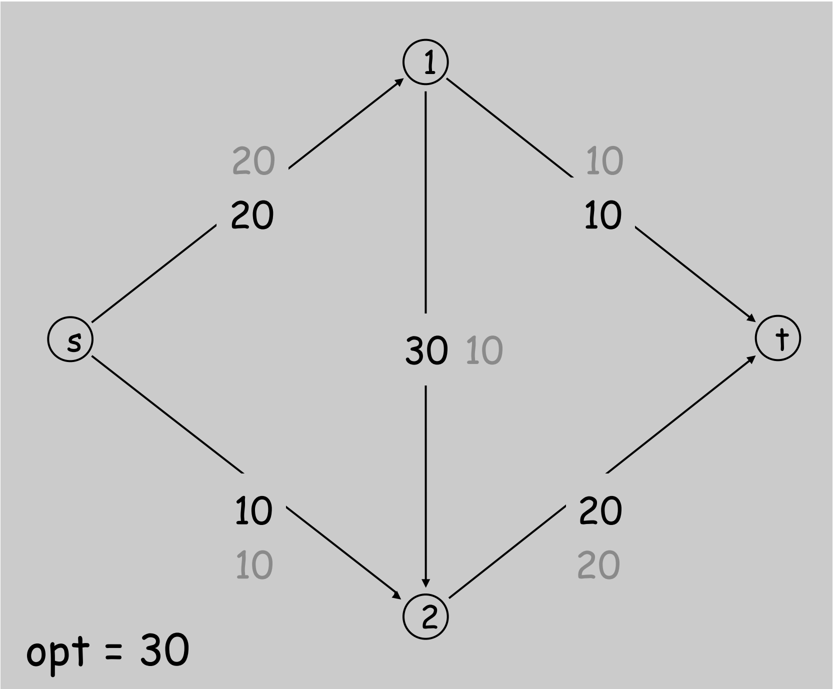
Vers un algorithme de flot maximum

Algorithme glouton.

- On pose f(e) = 0 pour tous les arcs $e \in E$.
- Déterminer un s-t chemin P tel que pour tout arc $e \in P$, $f(e) < c(e)$.
- Augmenter le flot le long du chemin P.
- On répéte jusqu'à ce qu'on ne puisse plus trouver de chemin.

optimalité locale $\not\Rightarrow$ optimalité globale

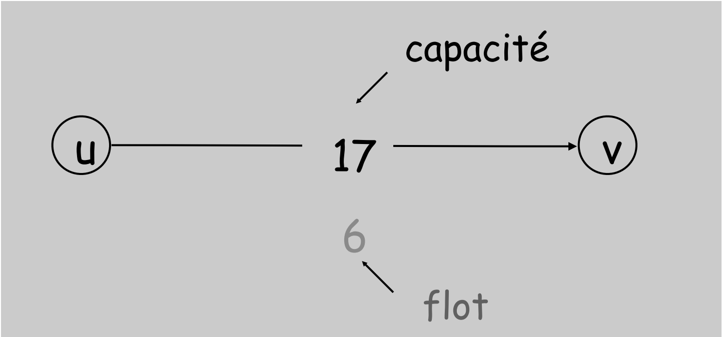




Graphe Résiduel

Arc original: $e = (u, v) \in E$.

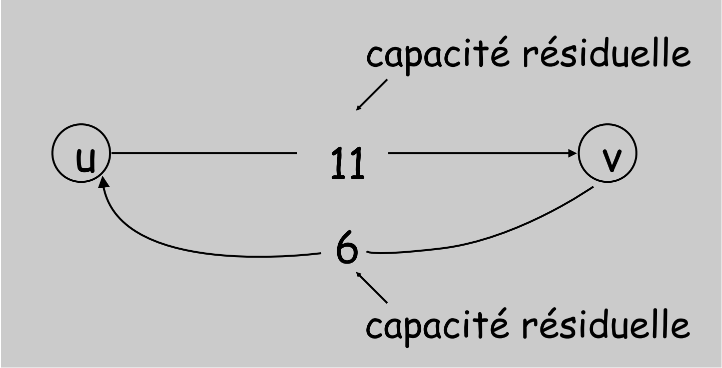
- Flot $f(e)$, capacité $c(e)$.



Arc résiduel

- "Refluer" le flot envoyé.
- $e = (u, v)$ et $e^R = (v, u)$.
- Capacité résiduelle :

$$c_f(e) = \begin{cases} c(e) - f(e) & \text{if } e \in E \\ f(e) & \text{if } e^R \in E \end{cases}$$



Graphe résiduel (ou graphe d'écart) : $G_f = (V, E_f)$.

- Arcs résiduels avec une capacité résiduelle positive.
- $E_f = \{e : f(e) < c(e)\} \cup \{e^R : f(e) > 0\}$.

Algorithme du chemin améliorant

```
Augmenter(f, c, P) {
   b ← goulot d'étranglement(P)
   pour chaque e ∈ P {
      si (e ∈ E) f(e) ← f(e) + b
      sinon      f(eᴿ) ← f(e) - b
   }
   retourner f
}
```

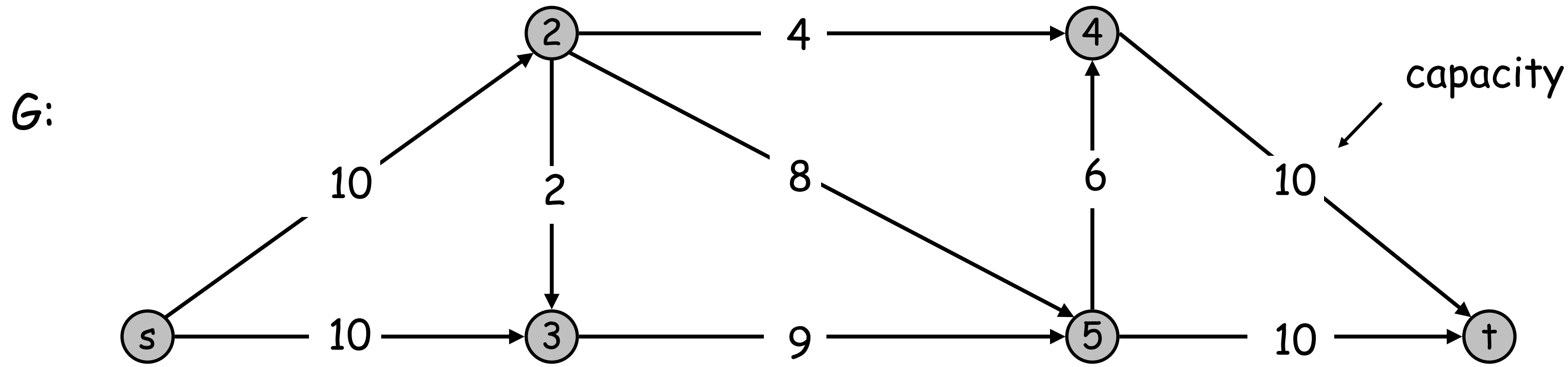
arc en avant

arc en arrière

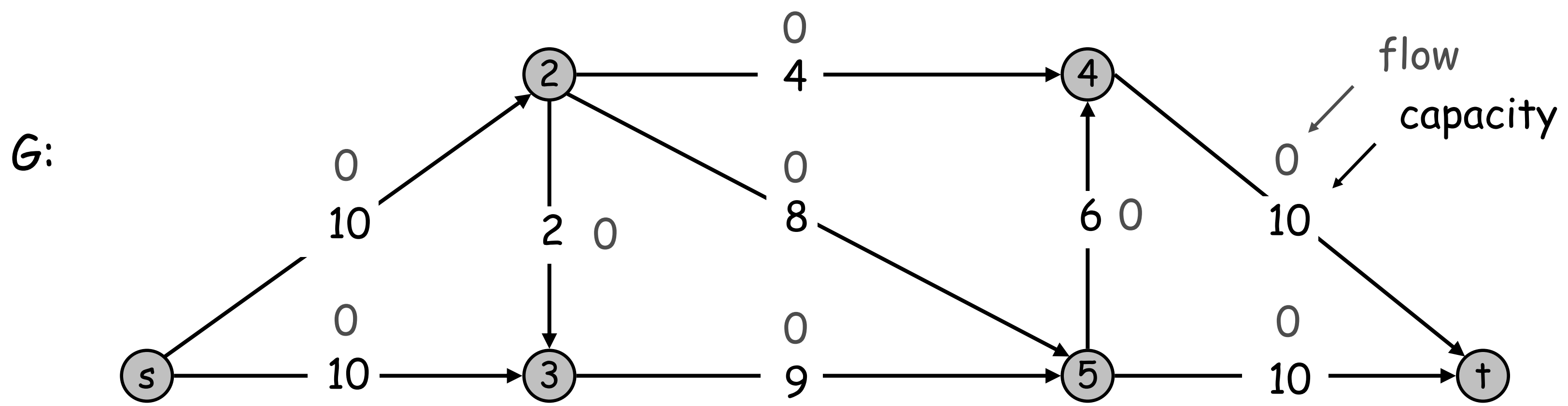
```
Ford-Fulkerson(G, s, t, c) {
  pour chaque e ∈ E  f(e) ← 0
  G_f ← graphe résiduel

  tant que (il existe chemin améliorant P) {
      f ← Augmenter(f, c, P)
      mettre à jour G_f
  }
  retourner f
}
```

Ford-Fulkerson Algorithm

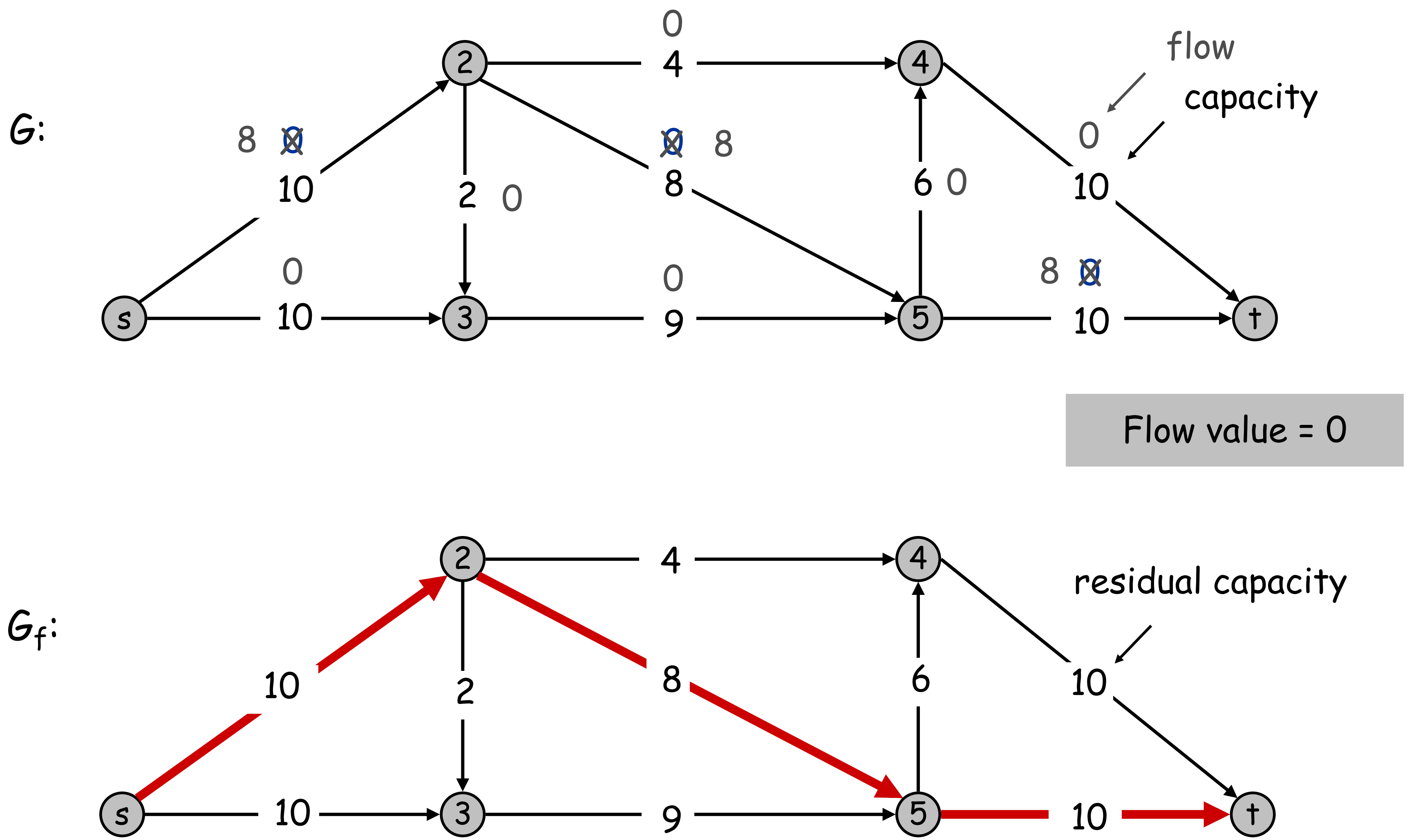


Ford-Fulkerson Algorithm

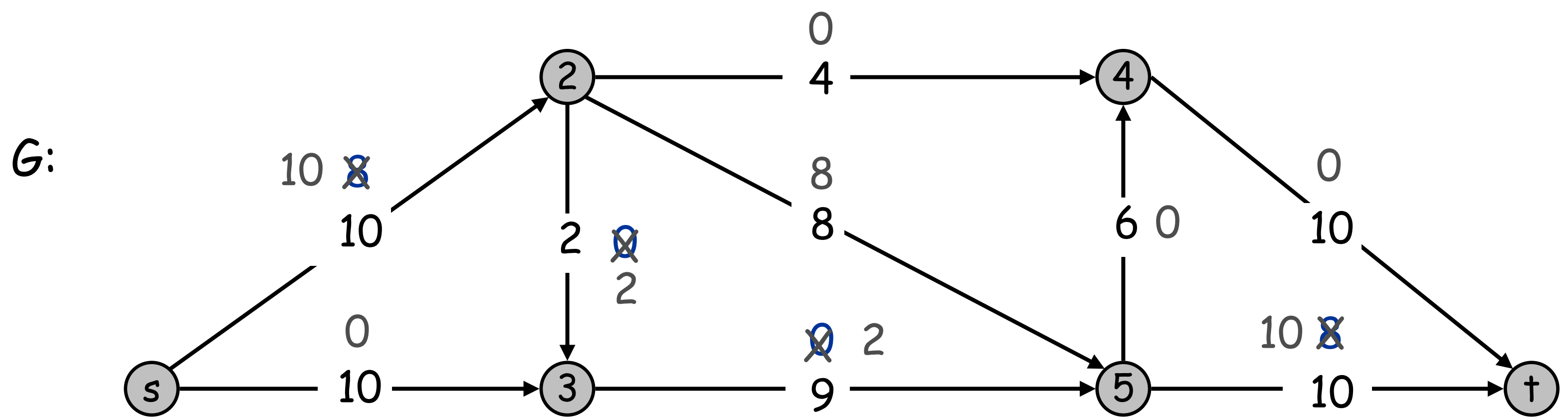


Flow value = 0

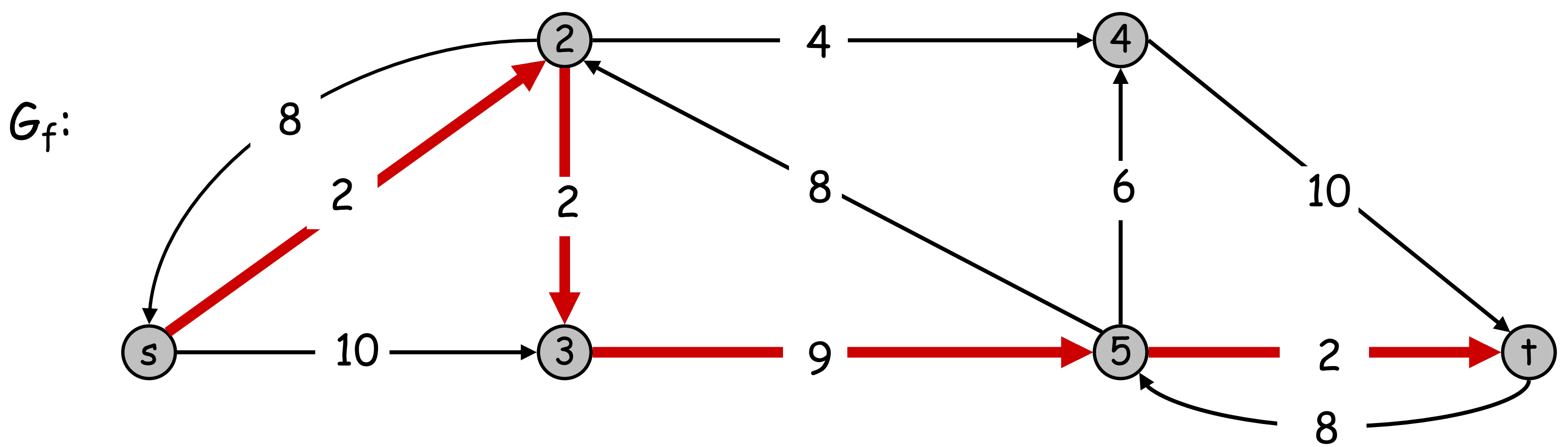
Ford-Fulkerson Algorithm



Ford-Fulkerson Algorithm

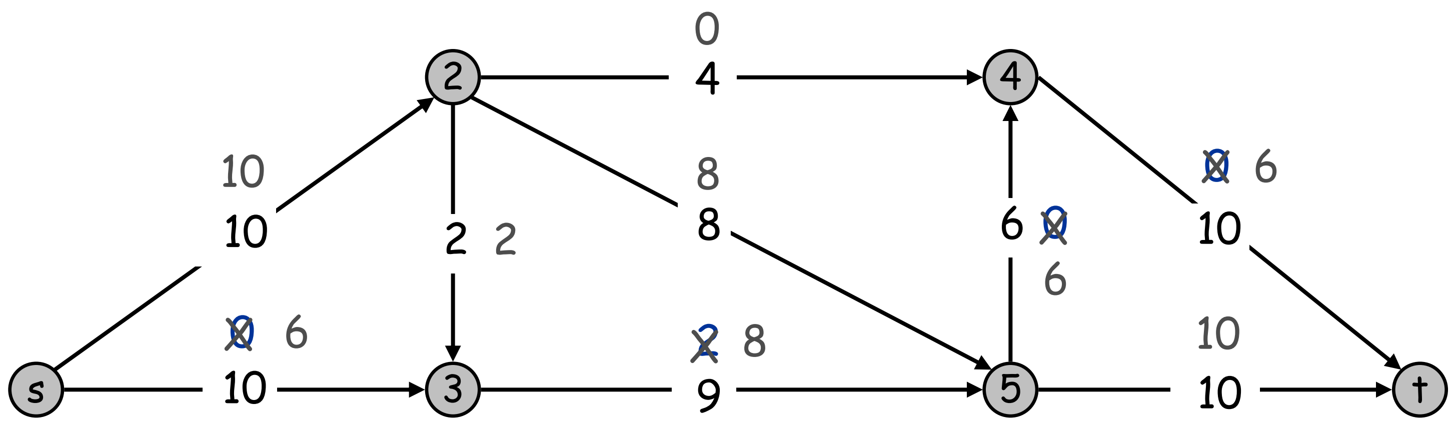


Flow value = 8



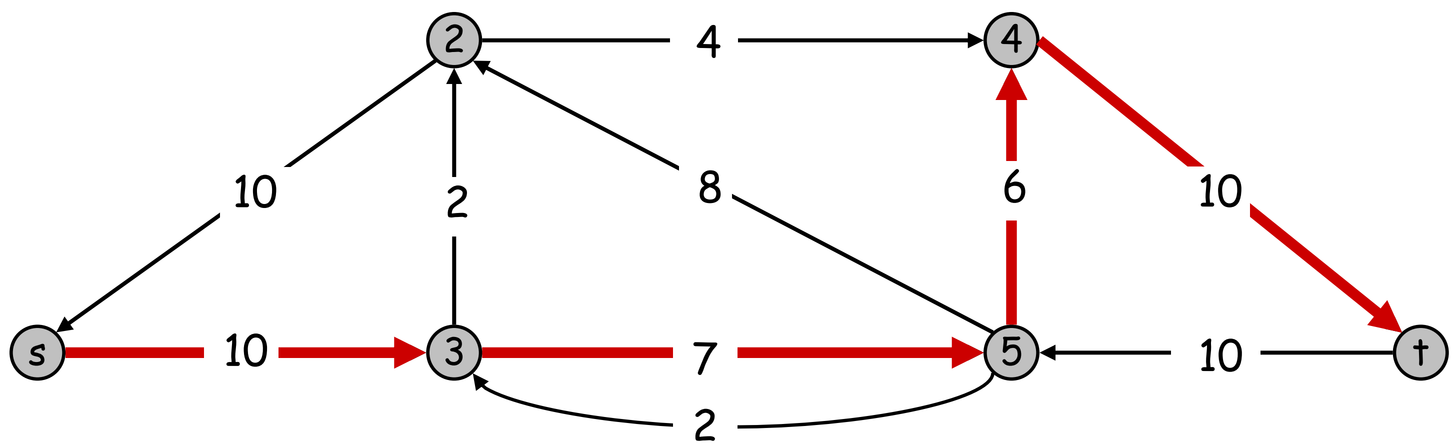
Ford-Fulkerson Algorithm

G:

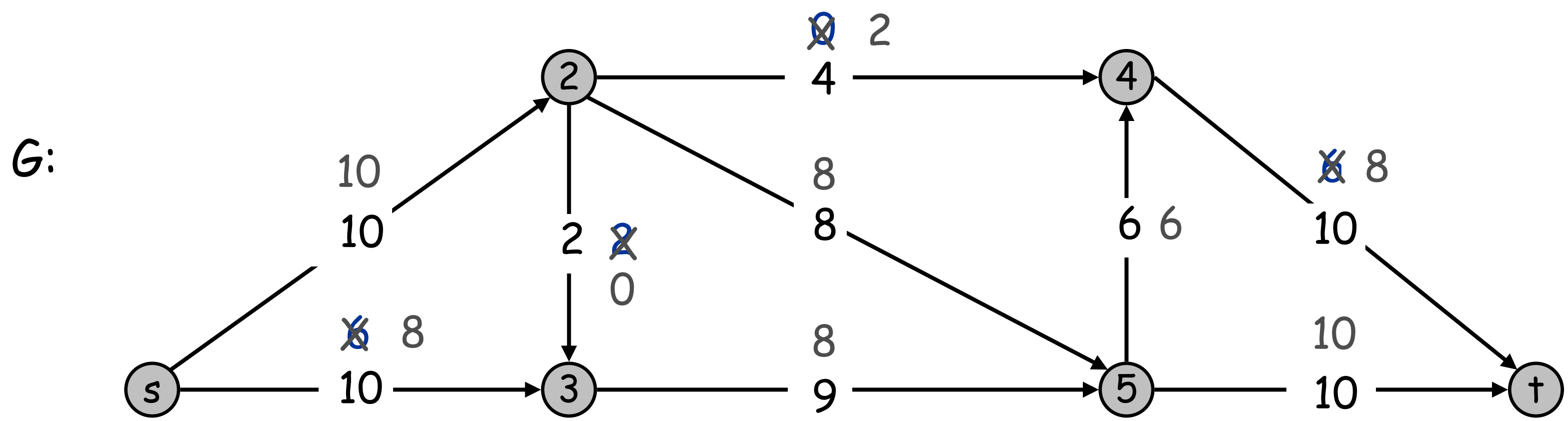


Flow value = 10

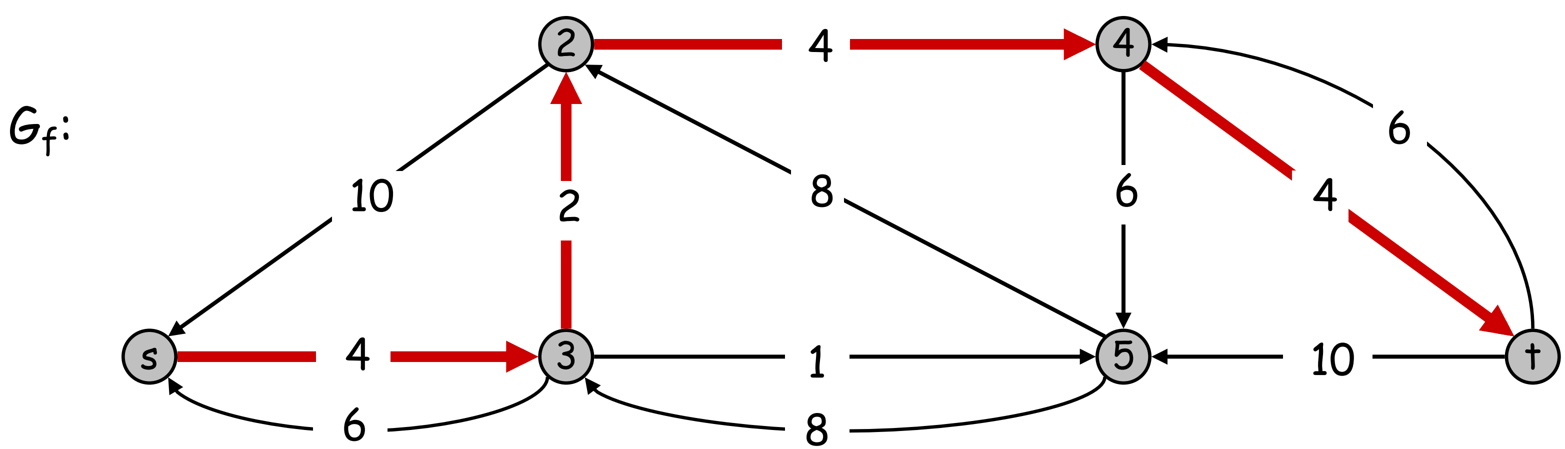
G_f:



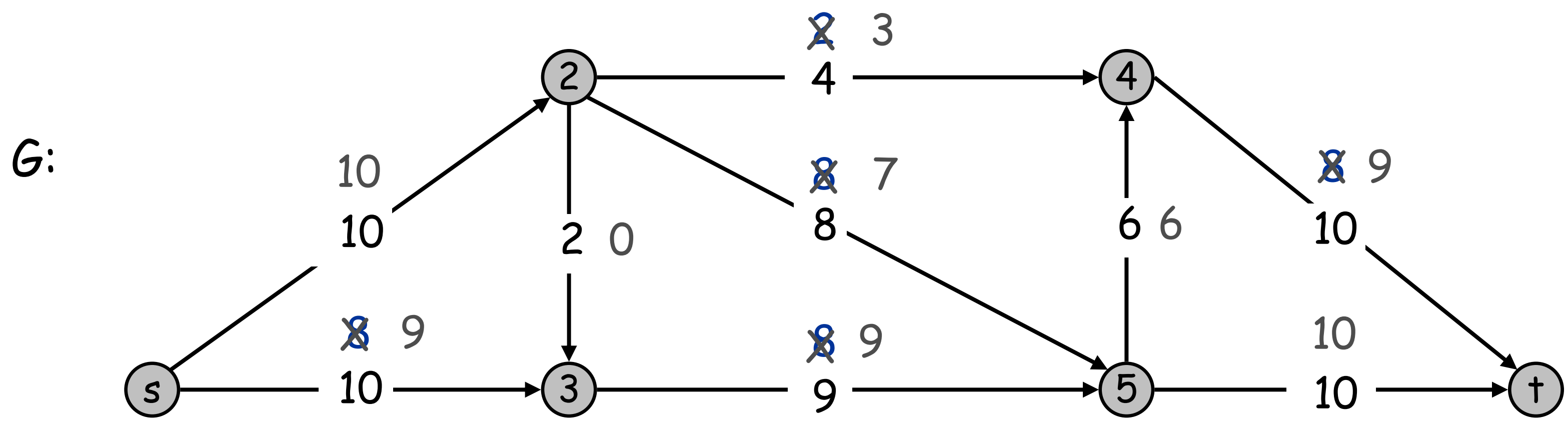
Ford-Fulkerson Algorithm



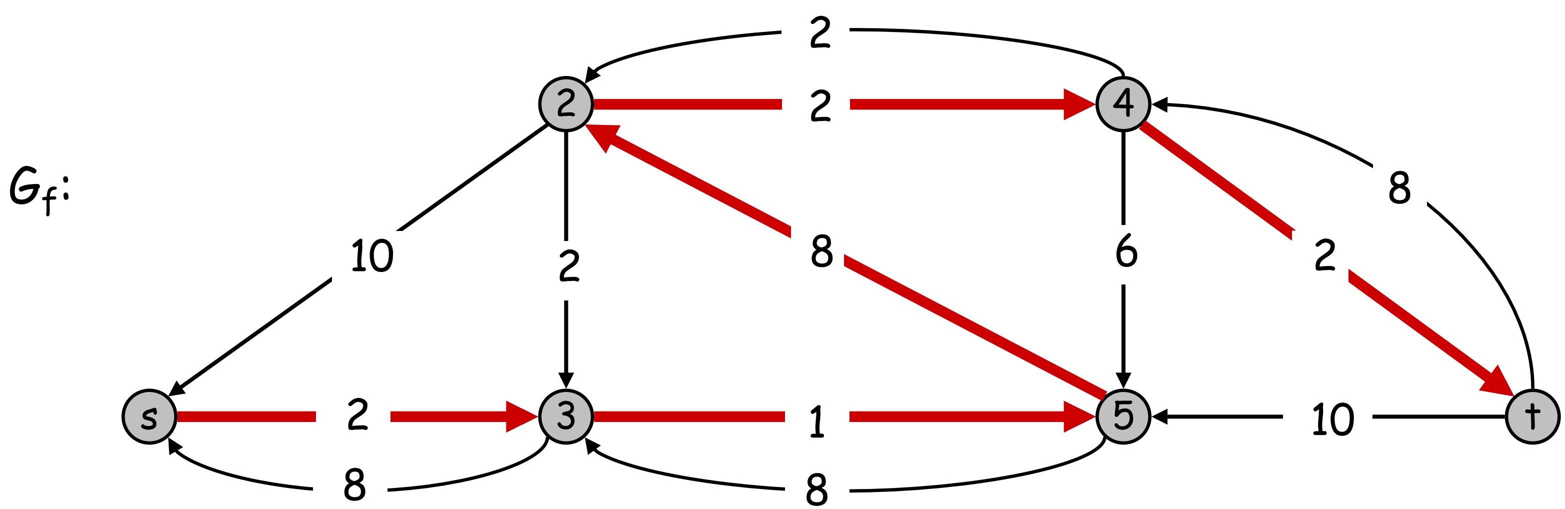
Flow value = 16



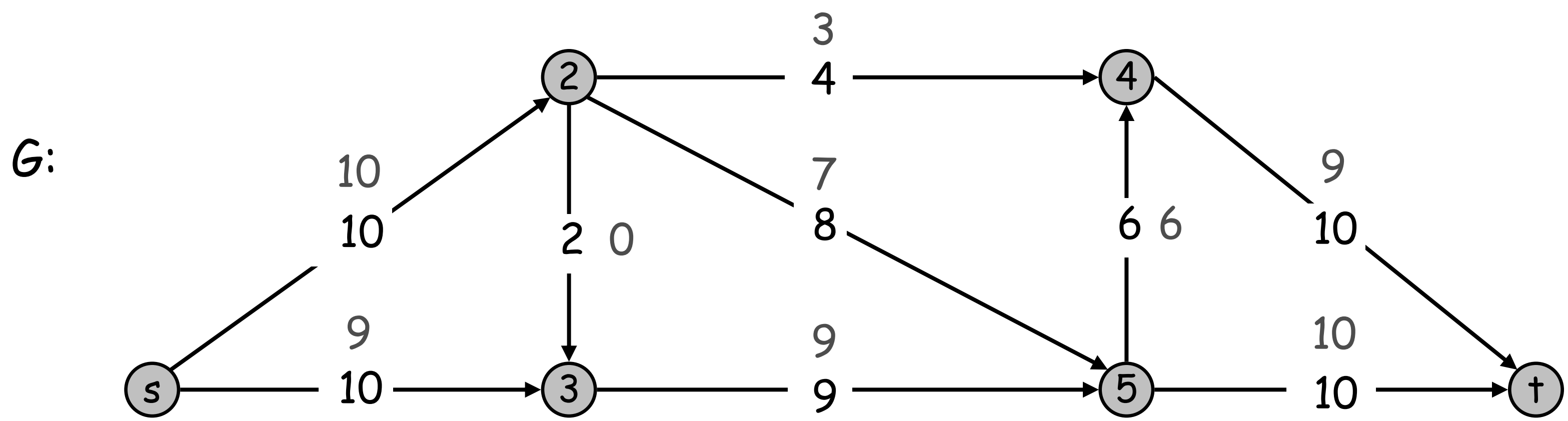
Ford-Fulkerson Algorithm



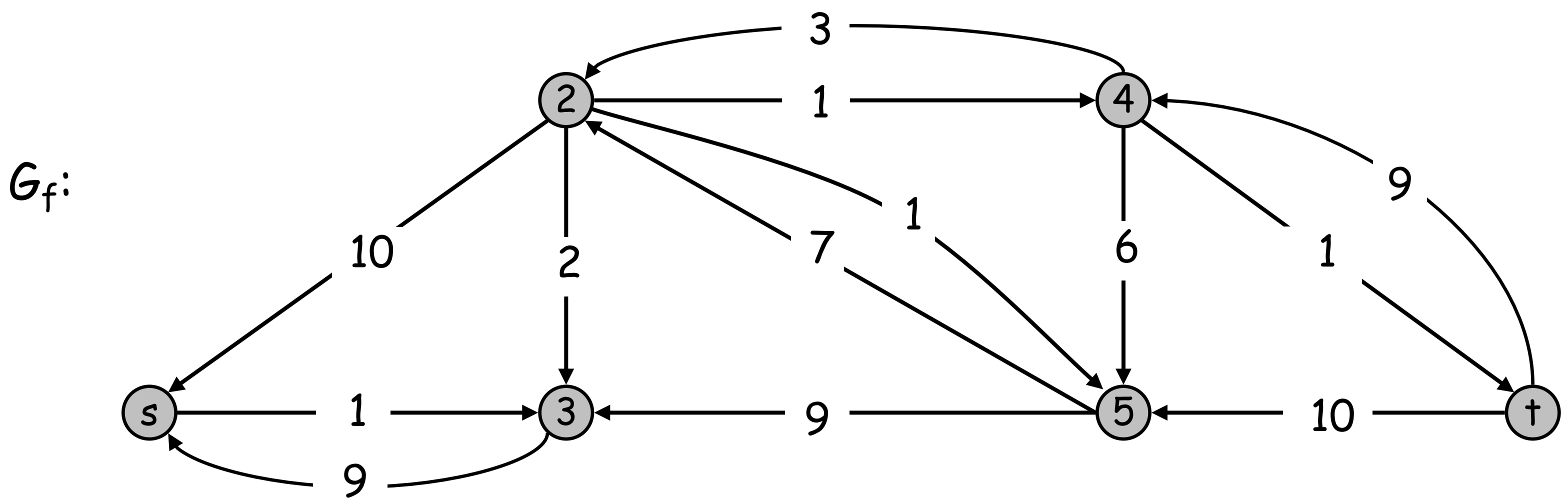
Flow value = 18



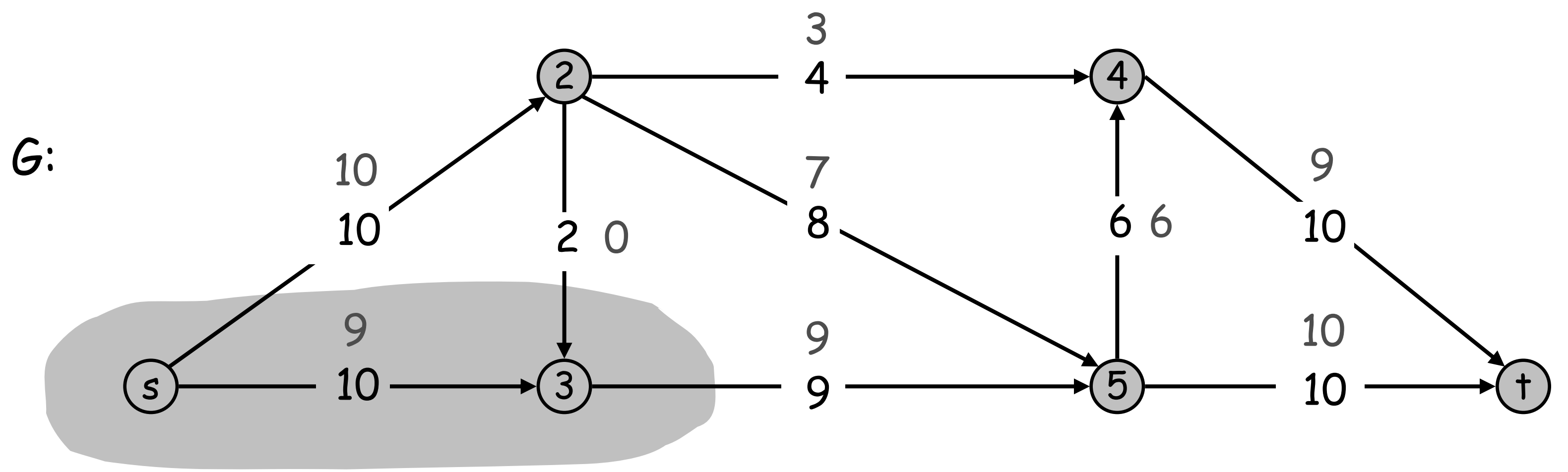
Ford-Fulkerson Algorithm



Flow value = 19

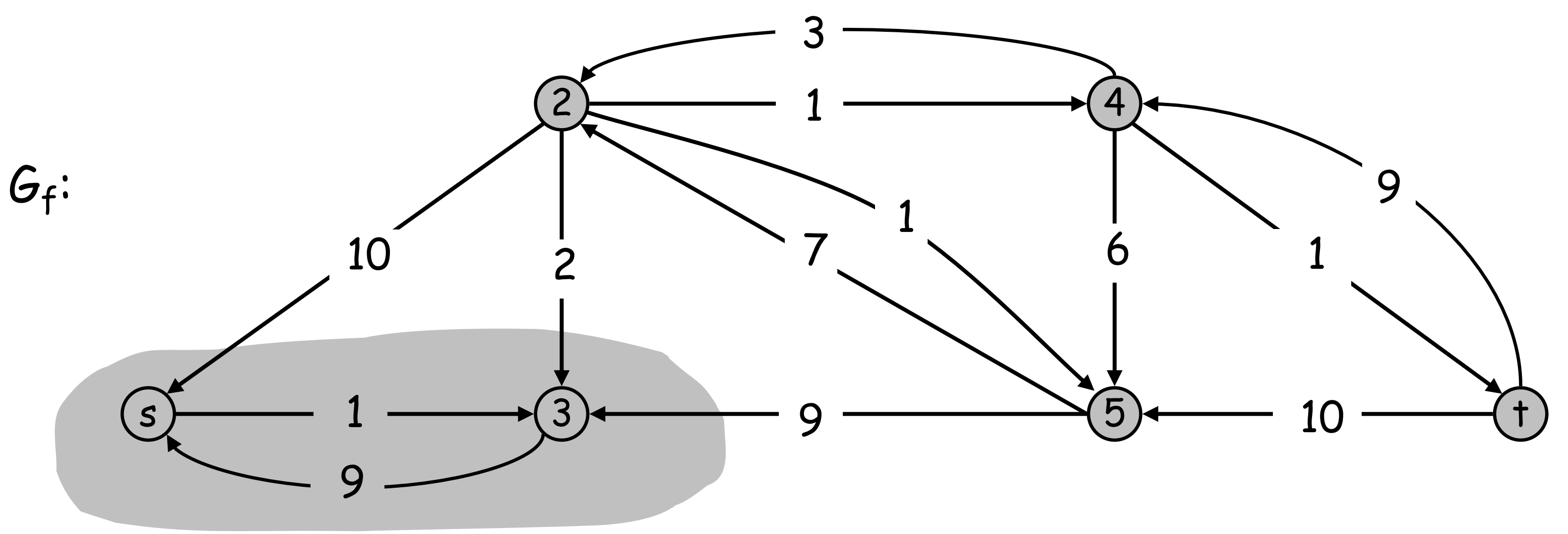


Ford-Fulkerson Algorithm



Cut capacity = 19

Flow value = 19



Théorème flot-max coupe-min

Théorème du chemin améliorant. Le flot f est un flot max ssi il n'existe plus de chemin améliorant.

Théorème flot-max coupe-min. [Ford-Fulkerson 1956] La valeur du flot max est égale à la valeur de la coupe min.

Idée de la preuve. On les montre simultanément en montrant que :

(i) Il existe une coupe (A, B) telle que v(f) = cap(A, B).

(ii) Le flot f est un flot max.

(iii) Il n'existe pas de chemin améliorant par rapport à f.

(i) $\Rightarrow$ (ii) Ceci vient du lemme de la dualité faible.

(ii) $\Rightarrow$ (iii) On montre la contrapositive.

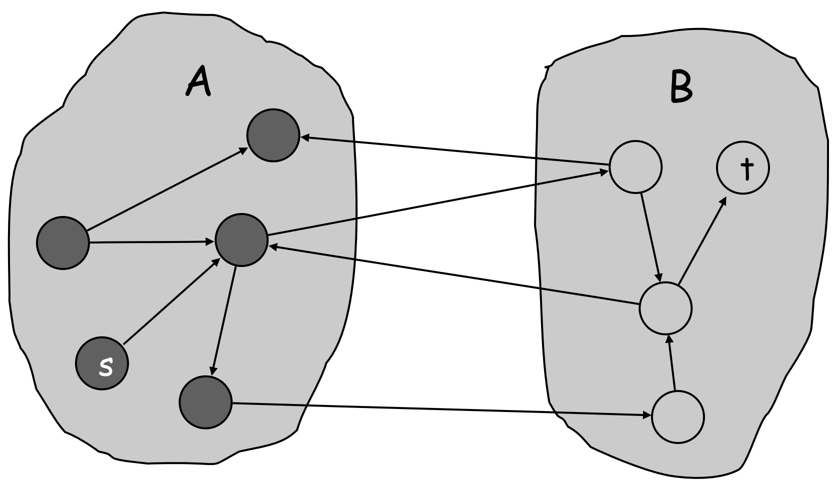
- Soit f un flot. Si il existe un chemin améliorant, alors on peut améliorer f en envoyant du flot le long de ce chemin.

Preuve du théorème de flot-max coupe-min

(iii) $\Rightarrow$ (i)

- Soit f un flot sans chemin améliorant.
- Soit A l'ensemble des sommets atteignables à partir de s dans le graphe résiduel.
- Par la définition de A, $s \in A$.
- Par la définition de f, $t \notin A$.

$$
\begin{aligned}
v(f) &= \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ in to A}} f(e) \\
&= \sum_{e \text{ out of } A} c(e) \\
&= cap(A,B) \quad \blacksquare
\end{aligned}
$$



réseau original

Complexité

Supposition. Toutes les capacités sont des entiers entre 1 et C.

Invariant. Chaque valeur du flot $f(e)$ et toutes les capacités résiduelles $c_f(e)$ sont des entiers durant l'exécution de l'algorithme.

Théorème. L'algorithme se termine au bout d'au plus $v(f^*) \leq nC$ itérations.

Preuve. Chaque augmentation augmente la valeur d'au moins 1. ▪

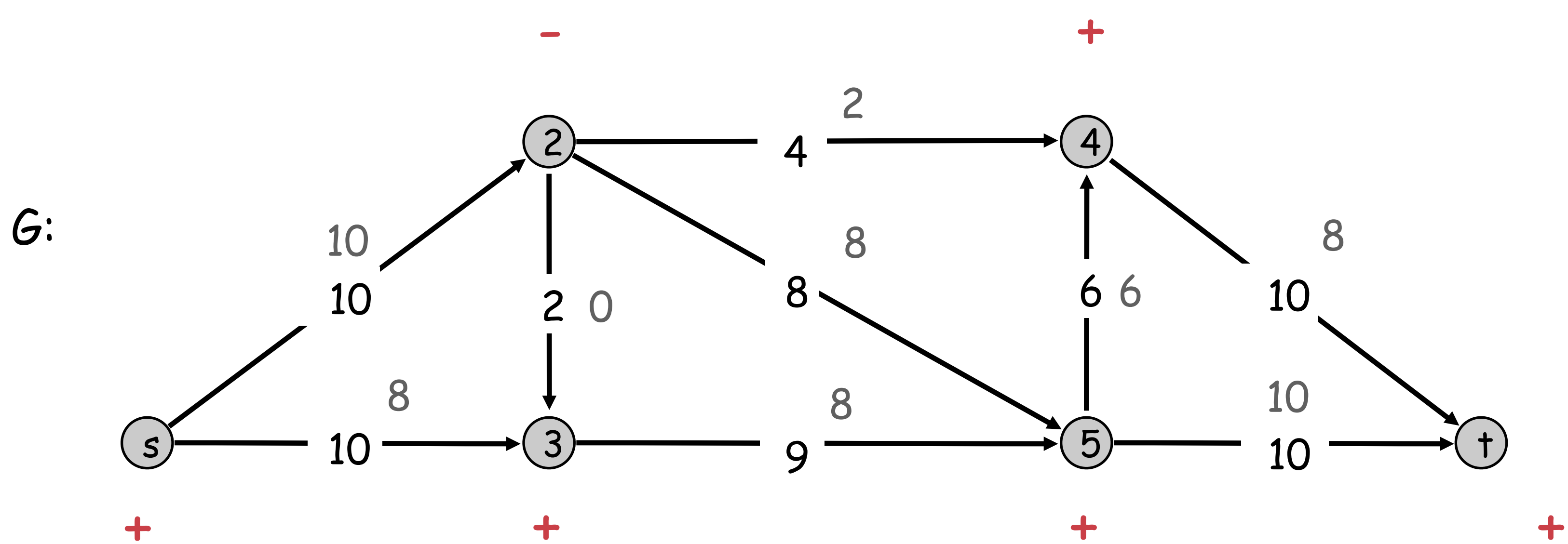
Corollaire. Si $C = 1$, Ford-Fulkerson tourne en temps $O(mn)$.

Théorème d'intégralité. Si toutes les capacités sont entières, alors il existe un flot max f pour lequel chaque valeur $f(e)$ est entière.

Preuve. Puisque l'algorithme se termine, le théorème est vrai grâce à l'invariant. ▪

Marquage de Ford-Fulkerson

```
Marquage de Ford-Fulkerson(G, s, t, c f) {
  marquer s d'un +
  repeat
      si il existe e=(u,v): u marqué, v non marqué et f(e)<c(e) alors
          marquer v d'un + ; père(v)← u
      sinon
         si il existe e=(u,v): v marqué, u non marqué et 0< f(e) alors
             marquer u d'un - ; père(u)← v
  jusqu'à ce que (il n'y a plus de marquage possible) ou (t est marqué)
  }
```



Intérêt du marquage

Deux cas de figure se présentent :

- On parvient à marquer t. Il existe alors une chaîne améliorante C sur laquelle on peut augmenter le flot de

$$b \leftarrow \min\{\min_{e \in C^+}\{c(e)-f(e)\}, \min_{e \in C^-}\{f(e)\}\}.$$

- On ne parvient pas à marquer t. Dans ce cas, la coupe minimale est donnée par la proposition ci-dessous :

Proposition

Si pour un flot réalisable f le marquage de Ford-Fulkerson ne permet pas de marquer t, alors la coupe (A,B) avec A l'ensemble des sommets marqués et B=V-A est une coupe de capacité minimum et le flot correspondant est maximum.

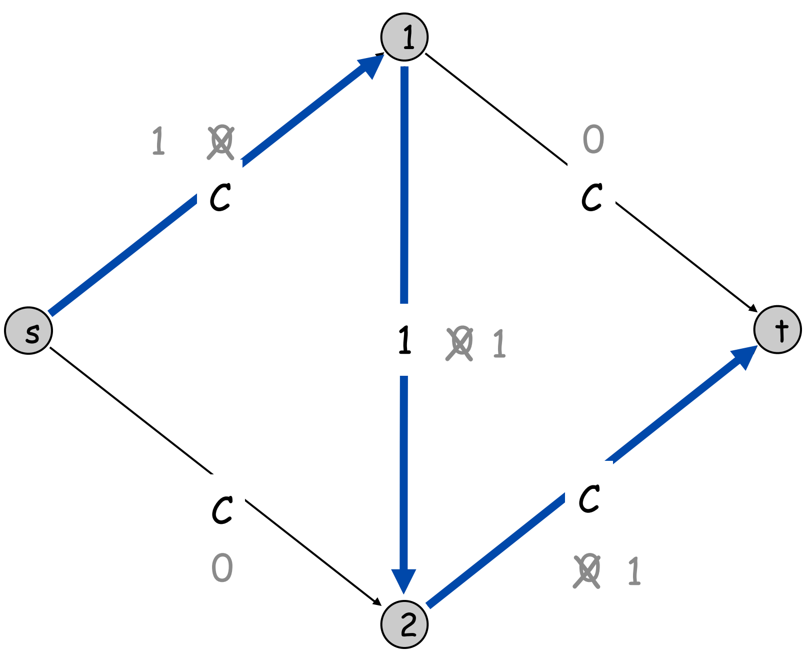
Choix de chemins améliorants

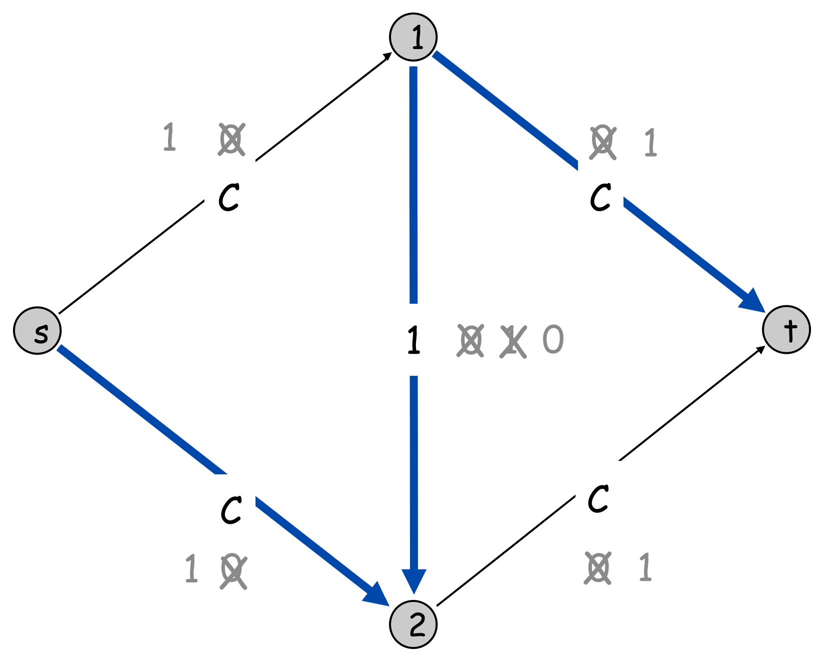
Ford-Fulkerson: Nombre exponentiel d'augmentations

Q. L'algorithme de Ford-Fulkerson est-il polynomial en la taille de l'entrée?

m, n, and log C

R. Non. Si la capacité max est C, alors l'algorithme peut faire C itérations.





Choix de chemins améliorants

Faire attention quand on choisit les chemins améliorants.

- Certains choix conduisent à des algorithmes exponentiels.
- Des choix intelligents donnent des algorithmes polynomiaux.
- Si les capacités sont irrationnelles, pas de garantie de terminaison !

Objectif : choisir des chemins améliorants tels que :

- on peut les déterminer de manière efficace.
- Peu d'itérations.

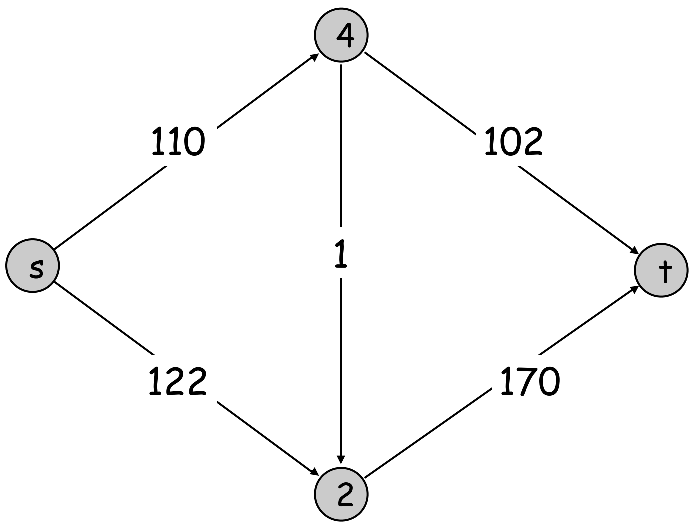
Choisir des chemins améliorants avec : [Edmonds-Karp 1972, Dinitz 1970]

- la capacité de goulot d'etranglement maximum.
- une capacité de goulot d'etranglement suffisament grande.
- le nombre minimum d'arcs.

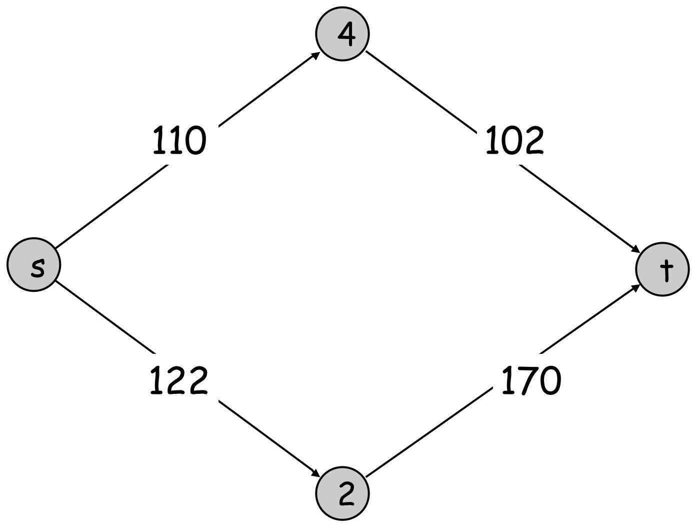
Mise à l'échelle de la capacité

Intuition. le choix du chemin avec la capacité maximum de goulot d'etranglement permet d'augmenter le plus la valeur du flot.

- Ne pas forcément calculer le chemin améliorant de capacité de goulot d'etraglement maximum.
- Maintenir un paramètre Δ de mise à l'échelle.
- Soit $G_f(\Delta)$ un sous-graphe du graphe résiduel contenant uniquement les arcs dont la capacité est au moins Δ.



G_f



$G_f(100)$

Mise à l'échelle de la capacité

```
Scaling-Flot-Max(G, s, t, c) {
   pour chaque e ∈ E  f(e) ← 0
   Δ ← plus grande puissance de 2 plus petite ou égale à C
   G_f ← graphe résiduel


   tant que (Δ ≥ 1) {
      G_f(Δ) ← Δ-graphe résiduel
      tant que (il existe un chemin augmentant P dans G_f(Δ)) {
         f ← augmenter(f, c, P)
         mettre à jour G_f(Δ)
      }
      Δ ← Δ / 2
   }
   retourner f
}
```

Pourquoi ça marche ?

Toutes les capacités sont des entiers entre 1 et C.

Invariant d'intégralité. les valeurs de flots et de capacités résiduelles sont entières.

Lemme. Si l'algorithme se termine, alors f est un flot max.
Preuve.
- Par l'invariant d'intégralité, quand $\Delta = 1 \Rightarrow G_f(\Delta) = G_f$.
- A la phase de terminaison $\Delta = 1$, il n'existe pas de chemins améliorants.

Complexité

Lemme 1. La boucle externe tant que se répète au plus $1 + \lceil \log_2 C \rceil$ fois.
Preuve. Δ diminue d'un facteur de 2 à chaque itération. ▪

Lemme 2. Soit f le flot à la fin d'une Δ-phase. Alors, la valeur de flot maximum est au plus $v(f) + m\,\Delta$.

← preuve à venir

Lemme 3. Il existe au plus 2m augmentations par phase de mise à l'échelle.

- Soit f le flot à la fin de la phase de mise à l'échelle précédente.

▪ Lemme 2 $\Rightarrow$ $v(f^*) \leq v(f) + m\,(2\Delta)$.
▪ Chaque augmentation dans une Δ-phase augmente v(f) d'au moins Δ. ▪

Théorème. L'algorithme scaling-flot-max détermine un flot maximum en $O(m \log C)$ augmentations. Sa complexité en temps est en $O(m^2 \log C)$. ▪

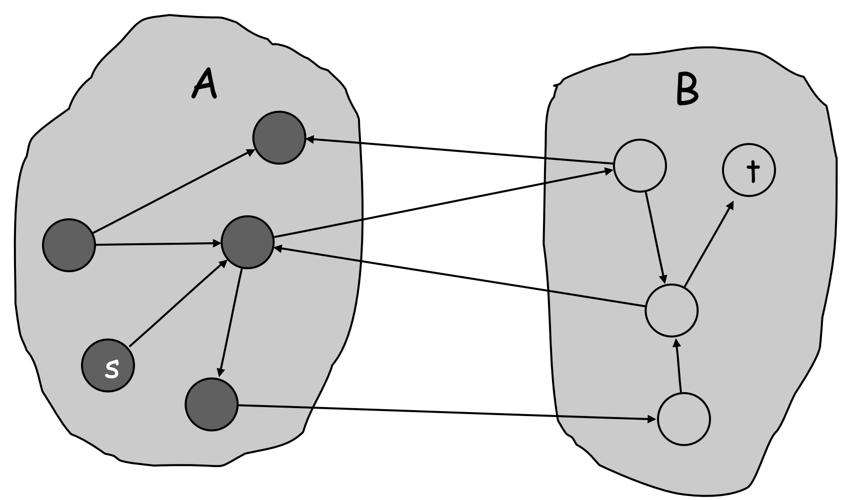
Complexité

Lemme 2. Soit f le flot à la fin d'une Δ-phase. Alors la valeur du flot maximum est au plus v(f) + m Δ.

Preuve.

- Montrons qu'à la fin d'une Δ-phase, il existe une coupe (A, B) telle que cap(A, B) $\leq$ v(f) + m Δ.
- Choisir A l'ensemble des sommets atteignables à partir de s dans $G_f(\Delta)$.
- Par la définition de A, $s \in A$.
- Par la définition de f, $t \notin A$.

$$\begin{aligned} v(f) &= \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ in to } A} f(e) \\ &\geq \sum_{e \text{ out of } A} (c(e) - \Delta) - \sum_{e \text{ in to } A} \Delta \\ &= \sum_{e \text{ out of } A} c(e) - \sum_{e \text{ out of } A} \Delta - \sum_{e \text{ in to } A} \Delta \\ &\geq cap(A, B) - m\Delta \quad \blacksquare \end{aligned}$$



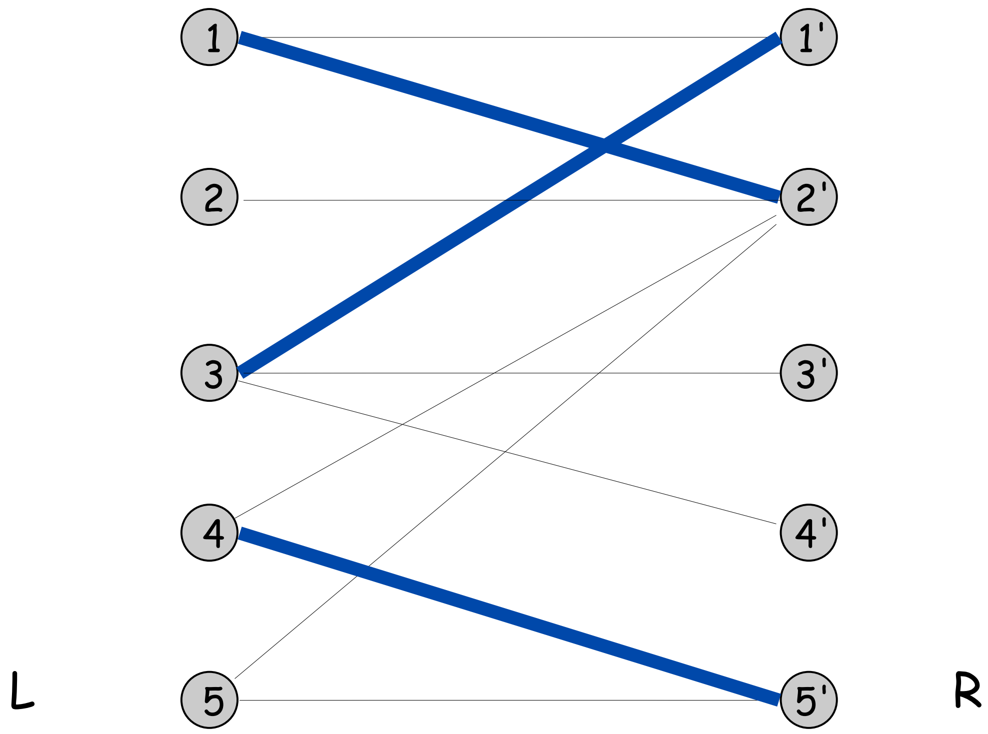
réseau original

IV. Application au problème du couplage

1. Couplage dans un graphe biparti

Déf. Dans un graphe $G=(V,E)$, un couplage est un ensemble d'arêtes M inclus dans E tel que tout sommet est l'extrémité d'au plus une arête de M

Problème. Etant donné un graphe (biparti) $G=(V,E)$, trouver un couplage de taille maximale.

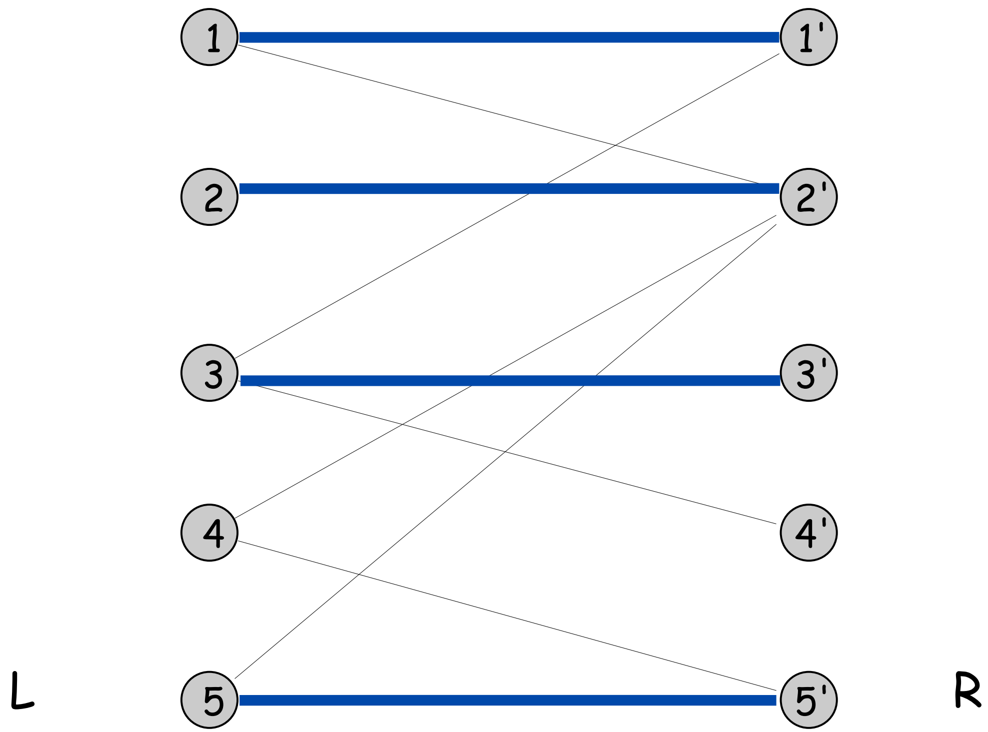


IV. Application au problème du couplage

1. Couplage dans un graphe biparti

Déf. Dans un graphe G=(V,E), un couplage est un ensemble d'arêtes M inclus dans E tel que tout sommet est l'extrémité d'au plus une arête de M

Problème. Etant donné un graphe (biparti) G=(V,E), trouver un couplage de taille maximale.



IV. Application au problème du couplage

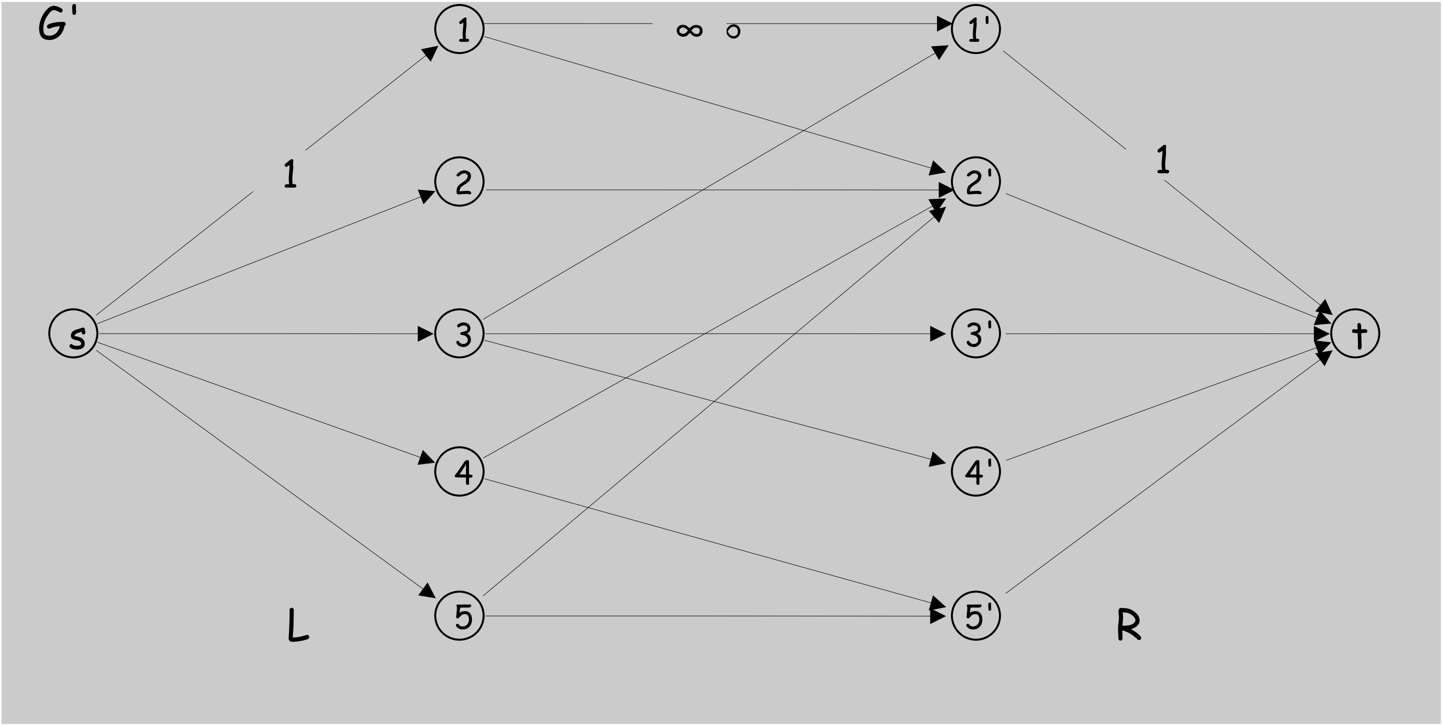
2. Modélisation par un flot maximum

Créer un graphe orienté $G' = (L \cup R \cup \{s, t\},\ E')$.

Orienter toutes les arêtes de L à R, et affecter une capacité infinie (ou unitaire).

Ajouter s (source), et des arêtes de capacité 1 de s vers tout sommet de L.

Ajouter t (puits), et des arêtes de capacité 1 de tout sommet de R vers t.



IV. Application au problème du couplage

2. Modélisation par un flot maximum