Algorithmique et Complexité
Cours 4/7 : Graphes de flots

CentraleSupélec – Gif

ST2 – Gif

Plan

1. Problème pratique
2. Modélisation du problème
3. Ford-Fulkerson
4. Graphe “à flot” comme modèle pour d’autres problèmes
5. Conclusion

Problème de l'opérateur d'un réseau de télécommunication

Que veut-il faire ?

Un opérateur d'un réseau de télécommunication gère sa propre infrastructure et connaît la capacité de chaque lien. Il perçoit la rémunération pour le trafic d'autres opérateurs qui transite par son réseau entre le routeur d'entrée s et le routeur de sortie t, $s \neq t$.

Problème de l'opérateur d'un réseau de télécommunication

Que veut-il faire ?

Un opérateur d'un réseau de télécommunication gère sa propre infrastructure et connaît la capacité de chaque lien. Il perçoit la rémunération pour le trafic d'autres opérateurs qui transite par son réseau entre le routeur d'entrée s et le routeur de sortie t, $s \neq t$.

Quel est son but ?

Connaître le débit maximal que son infrastructure peut assurer entre le point d'entrée dans son réseau s et le point de sortie t.

Problème de l'opérateur d'un réseau de télécommunication

Que veut-il faire ?

Un opérateur d'un réseau de télécommunication gère sa propre infrastructure et connaît la capacité de chaque lien. Il perçoit la rémunération pour le trafic d'autres opérateurs qui transite par son réseau entre le routeur d'entrée s et le routeur de sortie t, $s \neq t$.

Quel est son but ?

Connaître le débit maximal que son infrastructure peut assurer entre le point d'entrée dans son réseau s et le point de sortie t.

Quelle est la nature de son problème ?

C'est un problème d'optimisation.

Autres problèmes pratiques . . .

- Quel est le débit **maximal** entre s et t d'un système hydraulique composé de tuyaux ?

Autres problèmes pratiques . . .

- Quel est le débit **maximal** entre s et t d'un système hydraulique composé de tuyaux ?
- Quel est le nombre **maximal** de camions par jour que l'on peut acheminer entre s et t dans un réseau routier ?

Autres problèmes pratiques . . .

- Quel est le débit **maximal** entre s et t d'un système hydraulique composé de tuyaux ?
- Quel est le nombre **maximal** de camions par jour que l'on peut acheminer entre s et t dans un réseau routier ?
- L'ennemi transporte l'acier produit dans l'usine sidérurgique localisée en s à l'usine de fabrication de chars localisée en t par un réseau ferroviaire. Quels tronçons de voie ferrée doivent être détruits pour impacter au maximum la production des chars ?
- etc.

Autres problèmes pratiques . . .

- Quel est le débit **maximal** entre s et t d'un système hydraulique composé de tuyaux ?
- Quel est le nombre **maximal** de camions par jour que l'on peut acheminer entre s et t dans un réseau routier ?
- L'ennemi transporte l'acier produit dans l'usine sidérurgique localisée en s à l'usine de fabrication de chars localisée en t par un réseau ferroviaire. Quels tronçons de voie ferrée doivent être détruits pour impacter au maximum la production des chars ?
- etc.

Ce sont des problèmes de flot de valeur maximale

Plan

1 Problème pratique

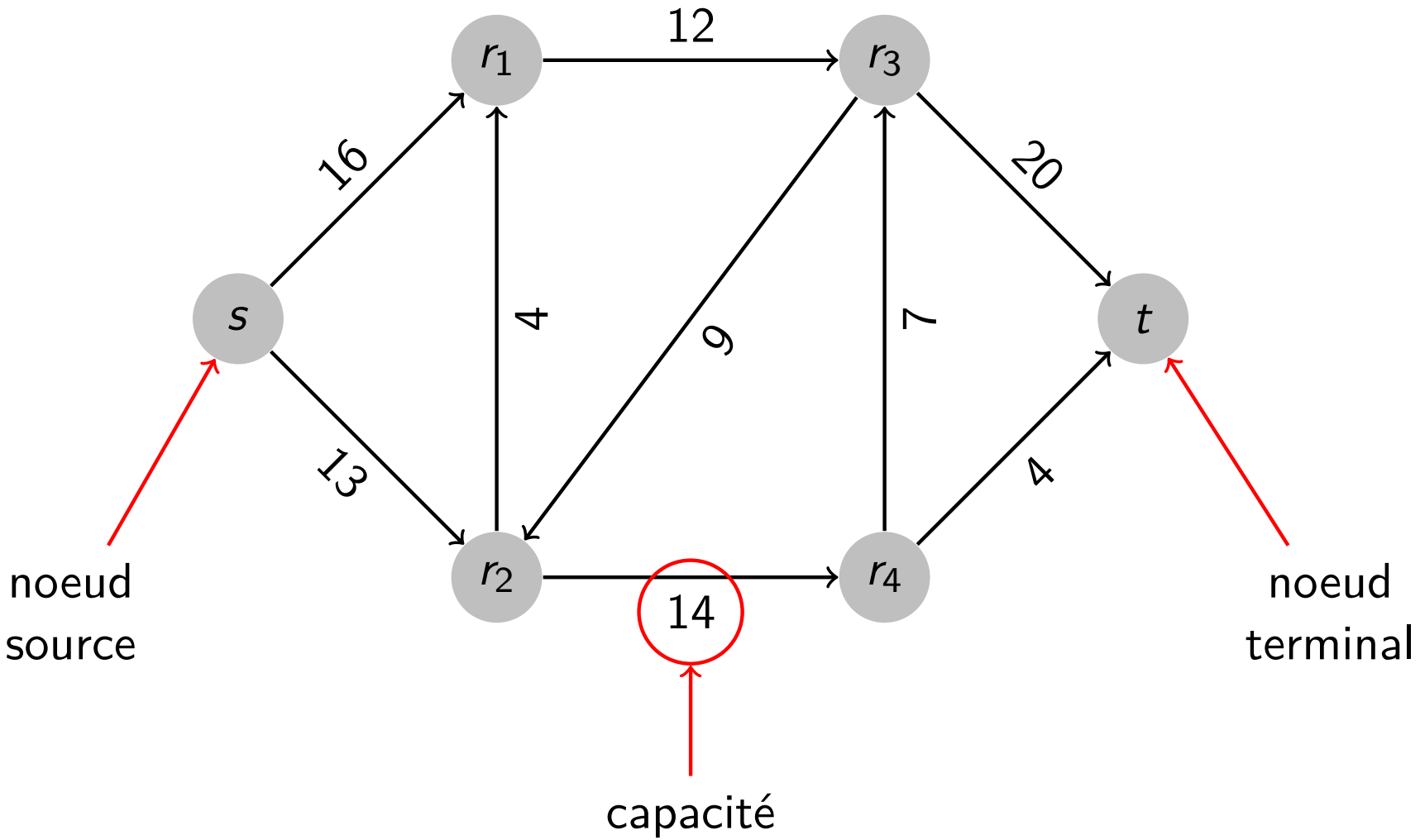
2 Modélisation du problème
- Exemple
- Problème

3 Ford-Fulkerson

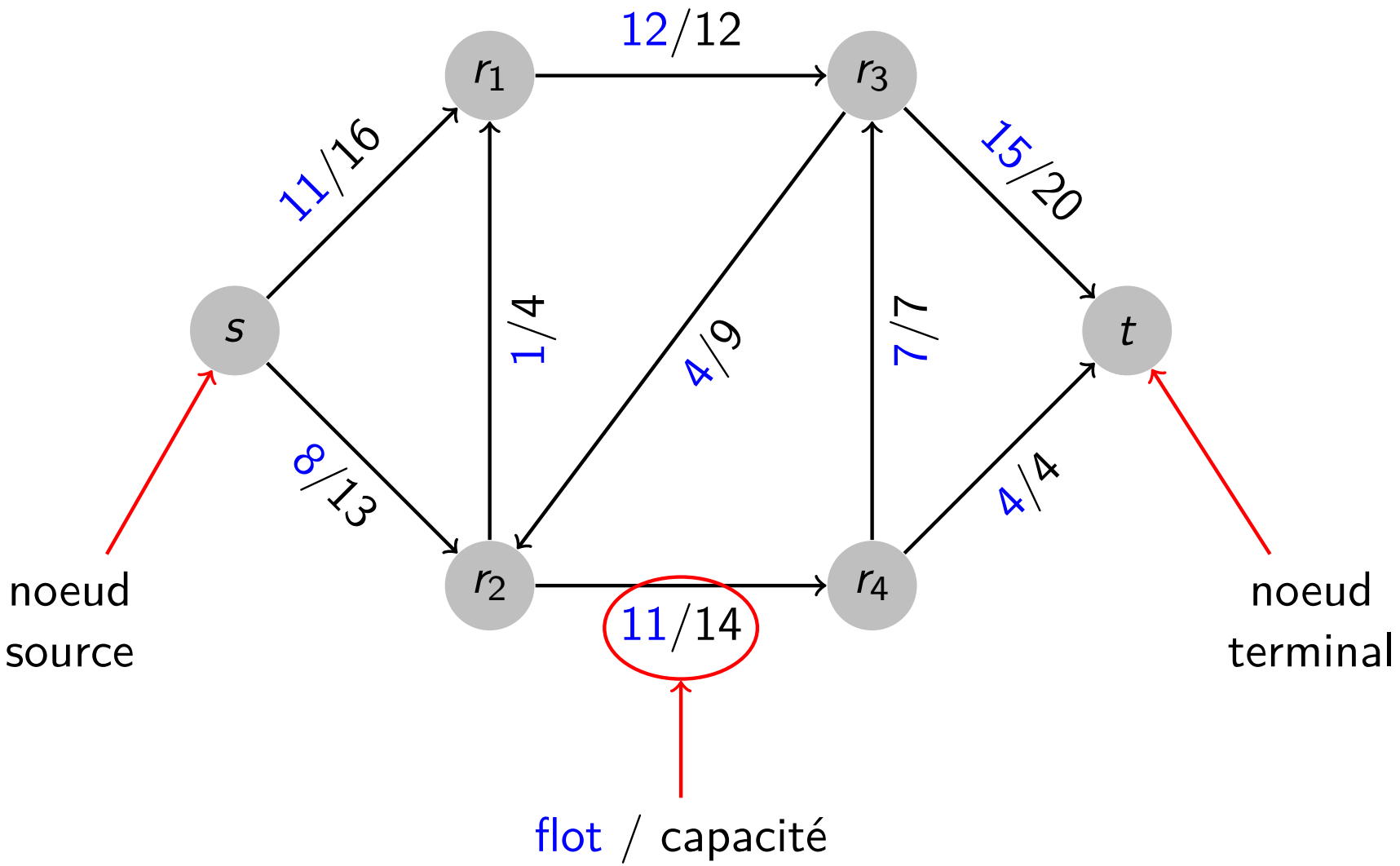
4 Graphe “à flot” comme modèle pour d’autres problèmes

5 Conclusion

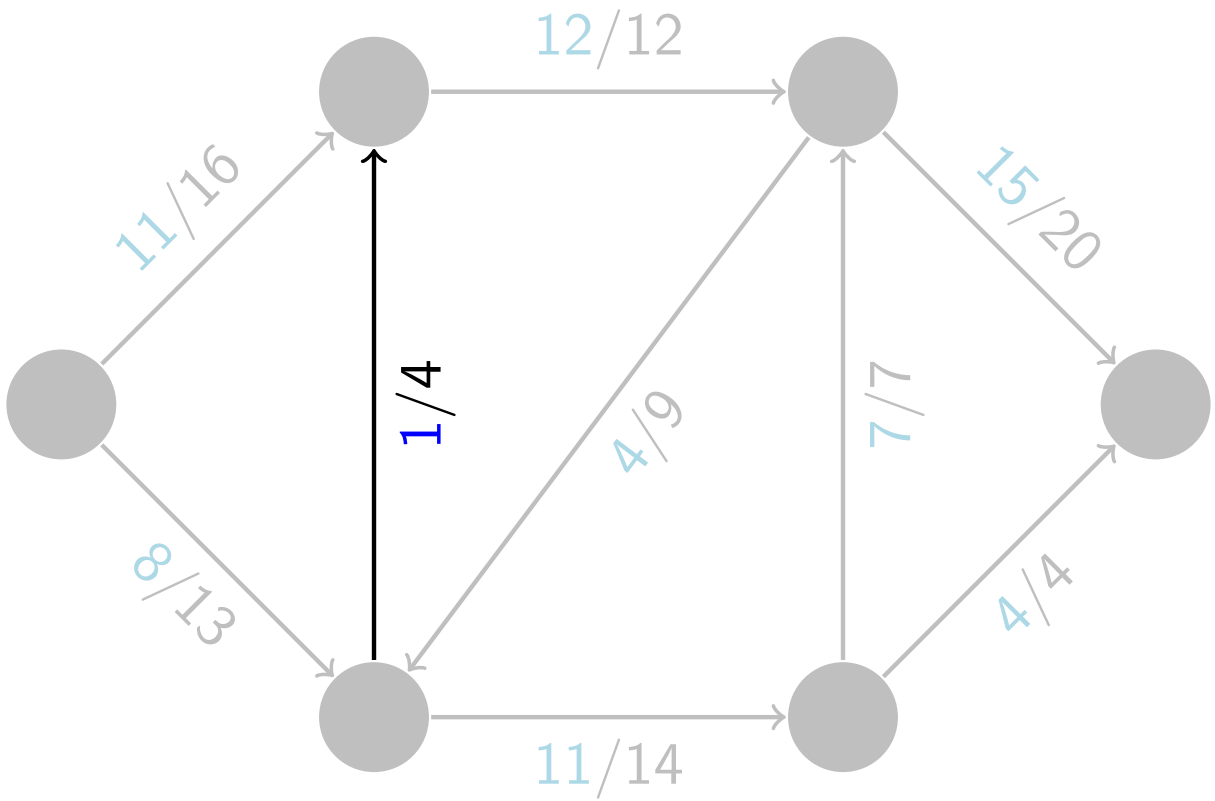
Exemple graphe à flot : Capacités



Exemple graphe à flot : Flots

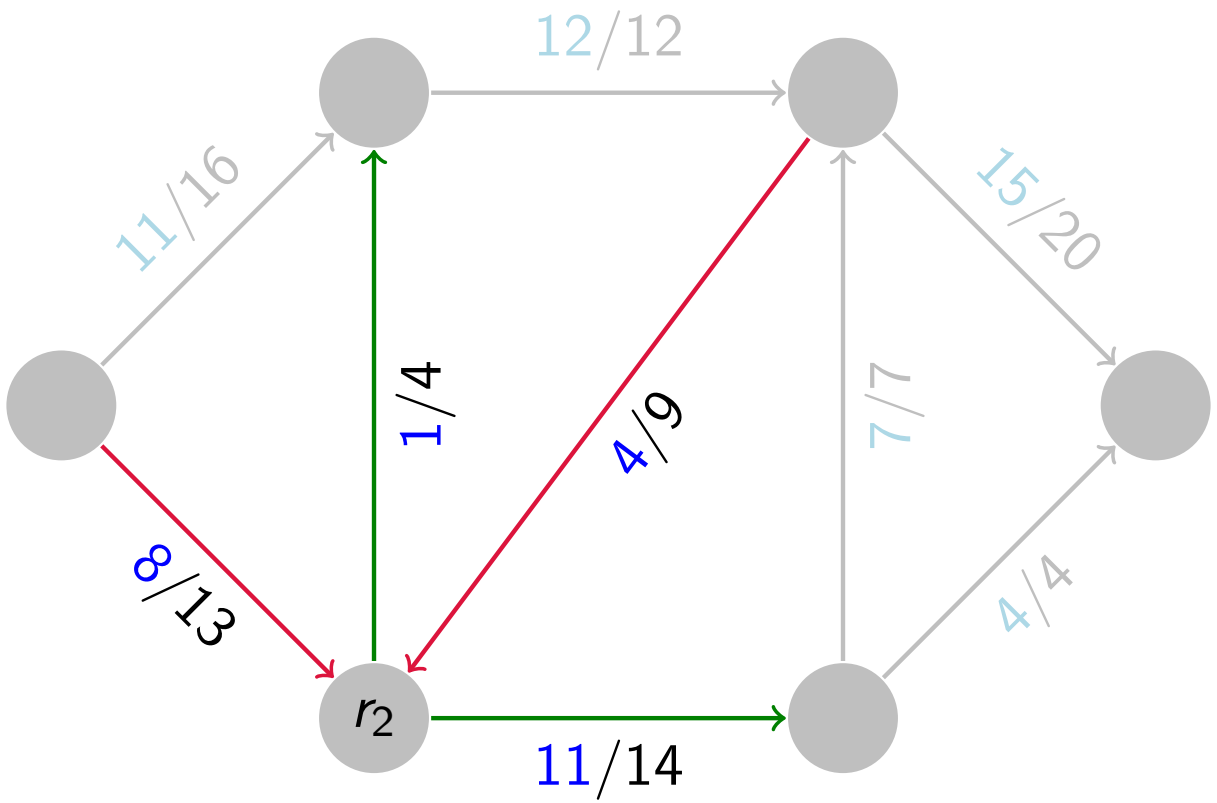


Exemple graphe à flot : Règle Flot–Capacité



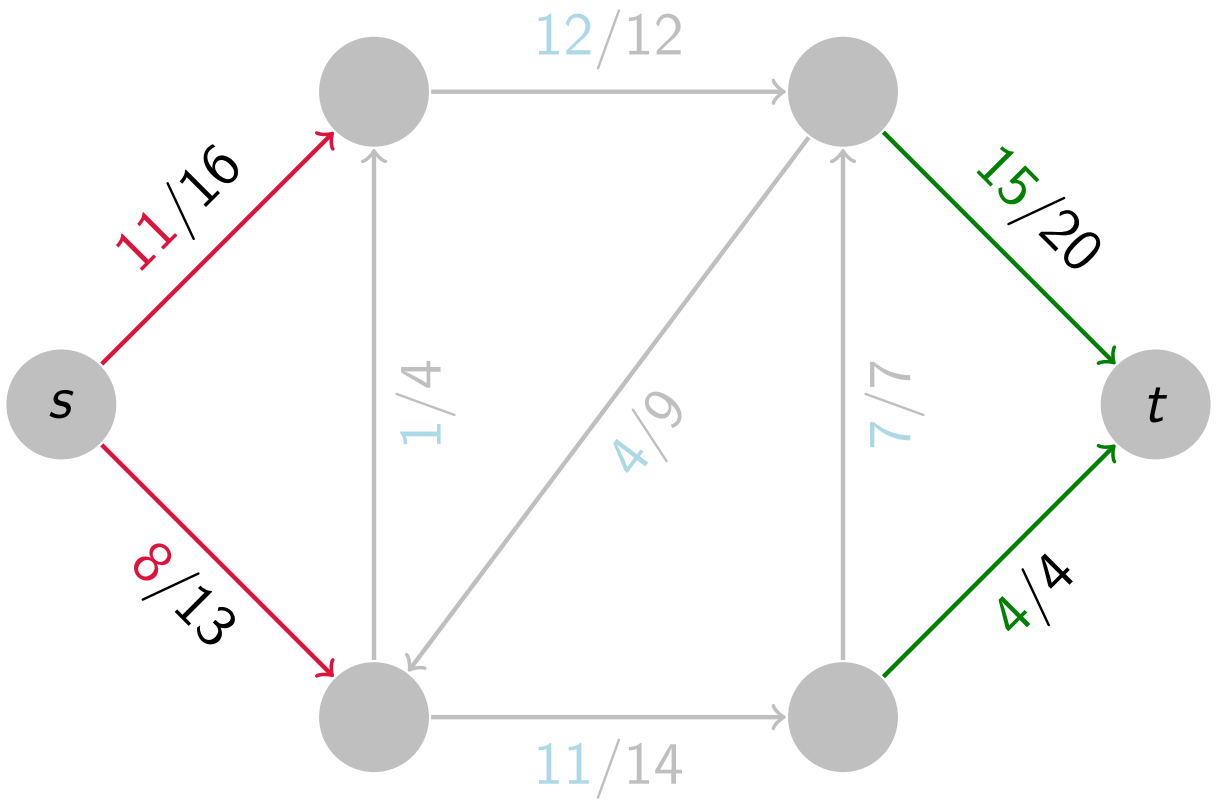
Capacité : $\forall u, v \in V \times V \quad 0 \leq f(u, v) \leq c(u, v)$

Exemple graphe à flot : Conservation du flot



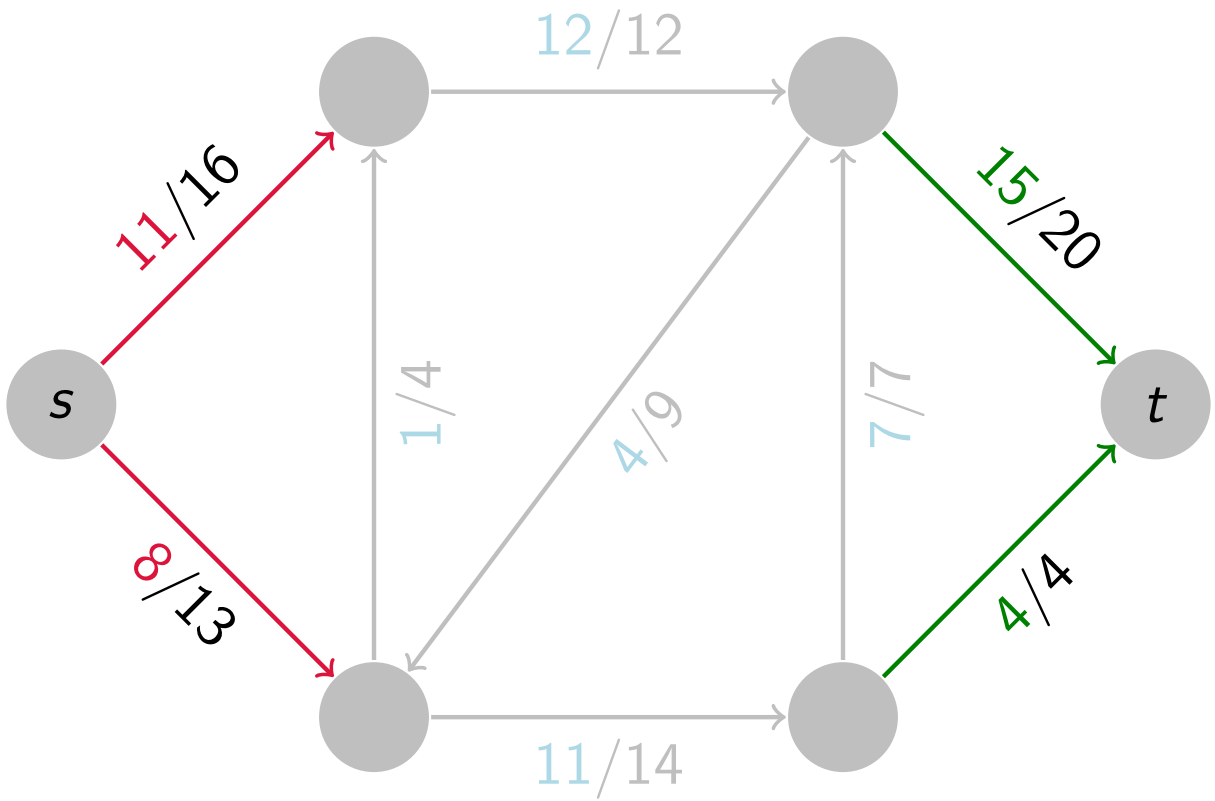
Loi de conservation : $\forall u \in V \setminus \{s, t\}$ $\sum_{v \in V} f(u, v) = \sum_{v \in V} f(v, u)$

Exemple graphe à flot : Flot courant



le flot courant est : $\varphi = \sum_{u \in V} f(s,u) = \sum_{u \in V} f(u,t)$

Exemple graphe à flot : Flot courant



le flot courant est : $\varphi = \sum_{u \in V} f(s,u) = \sum_{u \in V} f(u,t)$

Quel est le flot maximal ?

Définition du problème

Graphe à flot

- Un graphe orienté $G = (V, E)$
- deux sommets $s \in V$: source et $t \in V$: terminal
- une fonction de capacité $c : E \longrightarrow \mathbb{R}^+$

Définition du problème

Graphe à flot

- Un graphe orienté $G = (V, E)$
- deux sommets $s \in V$: source et $t \in V$: terminal
- une fonction de capacité $c : E \longrightarrow \mathbb{R}^+$

Définitions

On appelle flot une fonction $f : V \times V \longrightarrow \mathbb{R}$ telle que :

Définition du problème

Graphe à flot

- Un graphe orienté $G = (V, E)$
- deux sommets $s \in V$: source et $t \in V$: terminal
- une fonction de capacité $c : E \longrightarrow \mathbb{R}^+$

Définitions

On appelle flot une fonction $f : V \times V \longrightarrow \mathbb{R}$ telle que :

- contrainte de capacité :

$$\forall (u, v) \in E \quad 0 \leq f(u, v) \leq c(u, v)$$

Définition du problème

Graphe à flot

- Un graphe orienté $G = (V, E)$
- deux sommets $s \in V$: source et $t \in V$: terminal
- une fonction de capacité $c : E \longrightarrow \mathbb{R}^+$

Définitions

On appelle flot une fonction $f : V \times V \longrightarrow \mathbb{R}$ telle que :

- contrainte de capacité :

$$\forall (u, v) \in E \quad 0 \leq f(u, v) \leq c(u, v)$$

- contrainte de conservation du flot :

$$\forall u \in V \setminus \{s, t\} \quad \sum_{v \in V} f(u, v) = \sum_{v \in V} f(v, u)$$

On pose $f(u, v) = 0$ si $(u, v) \notin E$

Définition du problème II

Valeur du flot

On appelle valeur du flot f et on note

$$\varphi = \sum_{v \in V} f(s, v) = \sum_{v \in V} f(v, t)$$

la quantité absolue qui sort de la source. C'est aussi la quantité qui entre dans le terminal.

Définition du problème II

Valeur du flot

On appelle valeur du flot f et on note

$$\varphi = \sum_{v \in V} f(s, v) = \sum_{v \in V} f(v, t)$$

la quantité absolue qui sort de la source. C'est aussi la quantité qui entre dans le terminal.

Le problème du flot maximal consiste à trouver un flot f ayant la valeur du flot maximale φ_{max}.

Hypothèses simplificatrices

Pour simplifier, nous supposerons :

1. Tout sommet est sur un chemin entre s et t

$$\forall v \in V \quad s \rightsquigarrow v \rightsquigarrow t$$

Hypothèses simplificatrices

Pour simplifier, nous supposerons :

1. Tout sommet est sur un chemin entre s et t

$$\forall v \in V \quad s \rightsquigarrow v \rightsquigarrow t$$

2. Les capacités sont strictement positives

$$(u, v) \in E \iff c(u, v) \neq 0$$

Hypothèses simplificatrices

Pour simplifier, nous supposerons :

1. Tout sommet est sur un chemin entre s et t

$$\forall v \in V \quad s \rightsquigarrow v \rightsquigarrow t$$

2. Les capacités sont strictement positives

$$(u, v) \in E \iff c(u, v) \neq 0$$

3. Pas de boucle d'un sommet sur lui-même

$$(u, u) \notin E$$

Hypothèses simplificatrices

Pour simplifier, nous supposerons :

1. Tout sommet est sur un chemin entre s et t

$$\forall v \in V \quad s \rightsquigarrow v \rightsquigarrow t$$

2. Les capacités sont strictement positives

$$(u, v) \in E \iff c(u, v) \neq 0$$

3. Pas de boucle d'un sommet sur lui-même

$$(u, u) \notin E$$

4. Nous interdisons $(u, v) \in E$ et $(v, u) \in E$ simultanément

Hypothèses simplificatrices

Pour simplifier, nous supposerons :

1. Tout sommet est sur un chemin entre s et t

$$\forall v \in V \quad s \rightsquigarrow v \rightsquigarrow t$$

2. Les capacités sont strictement positives

$$(u, v) \in E \iff c(u, v) \neq 0$$

3. Pas de boucle d'un sommet sur lui-même

$$(u, u) \notin E$$

4. Nous interdisons $(u, v) \in E$ et $(v, u) \in E$ simultanément

5. Pour simplifier : pas d'arc entrant vers s ou sortants de t

$$\forall u \in V \quad (u, s) \notin E \quad \text{et} \quad (t, u) \notin E$$

Plan

1 Problème pratique

2 Modélisation du problème

3 Ford-Fulkerson
- Principe
- Graphe résiduel
- Exemple
- Augmentation du flot
- Flot et coupe
- Max-Flow-Min-Cut
- Implémentation

4 Graphe “à flot” comme modèle pour d’autres problèmes

Méthode de Ford-Fulkerson (1962)

Principe général

Algorithme itératif

➜ Augmenter progressivement le flot jusqu'à saturation

Méthode de Ford-Fulkerson (1962)

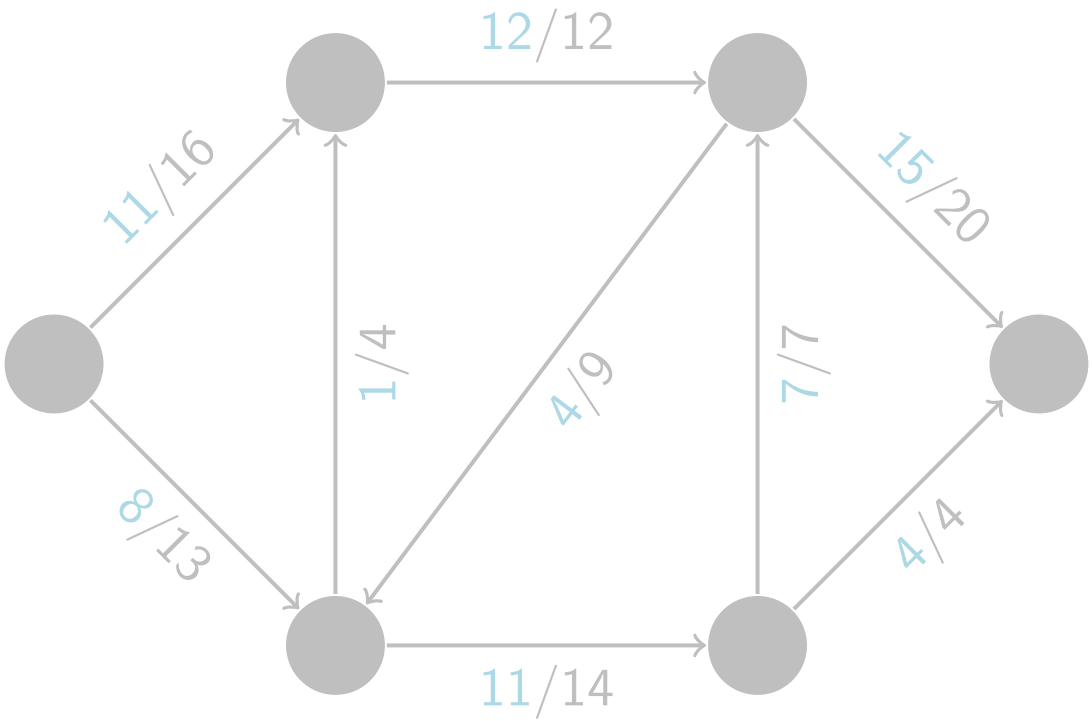
Principe général

Algorithme itératif

➜ Augmenter progressivement le flot jusqu'à saturation

Repose sur le calcul de capacité résiduelle et de chaînes augmentantes

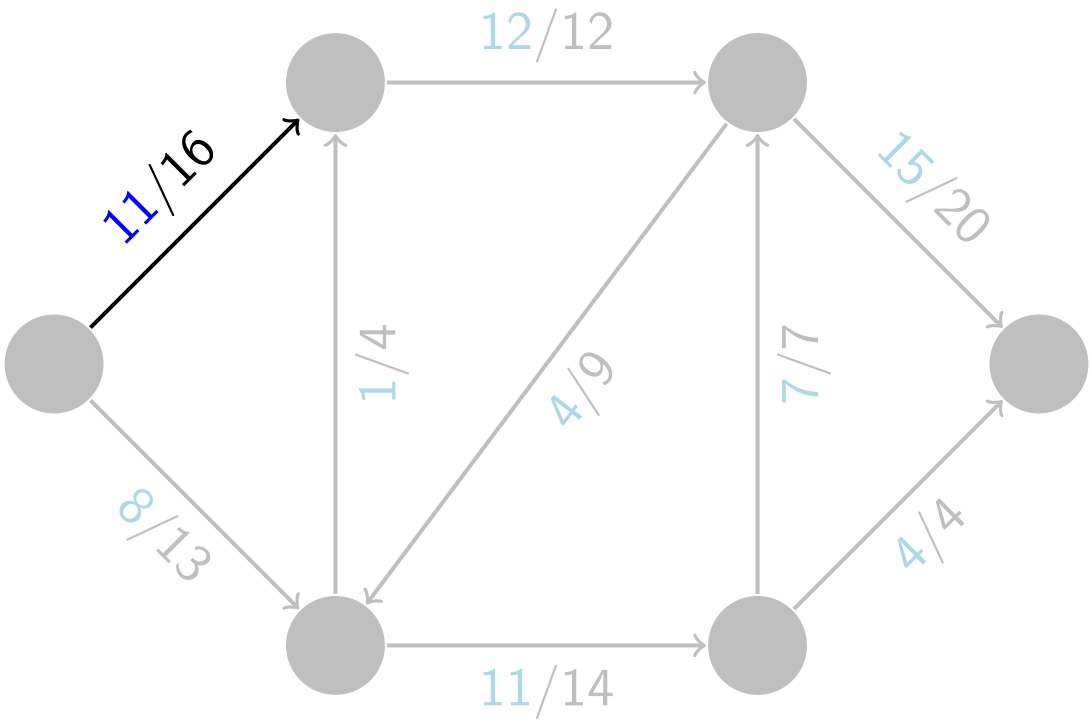
Capacité résiduelle



Capacité résiduelle

- Ce que l'on peut encore envoyer le long d'un arc

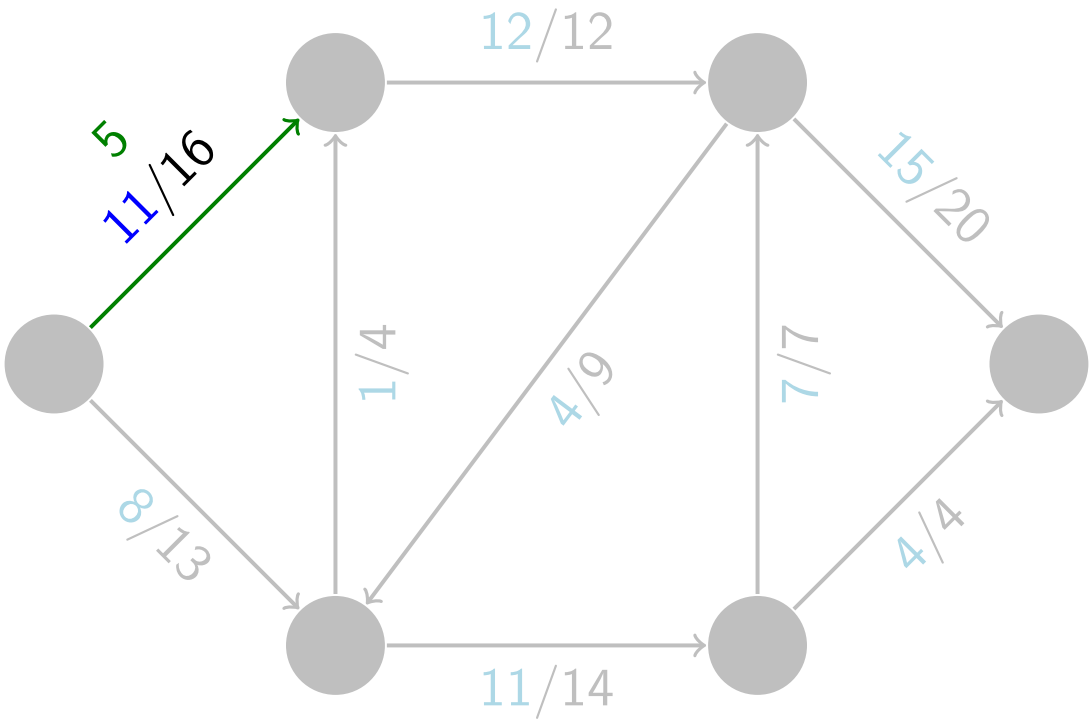
Capacité résiduelle



Capacité résiduelle

- Ce que l'on peut encore envoyer le long d'un arc

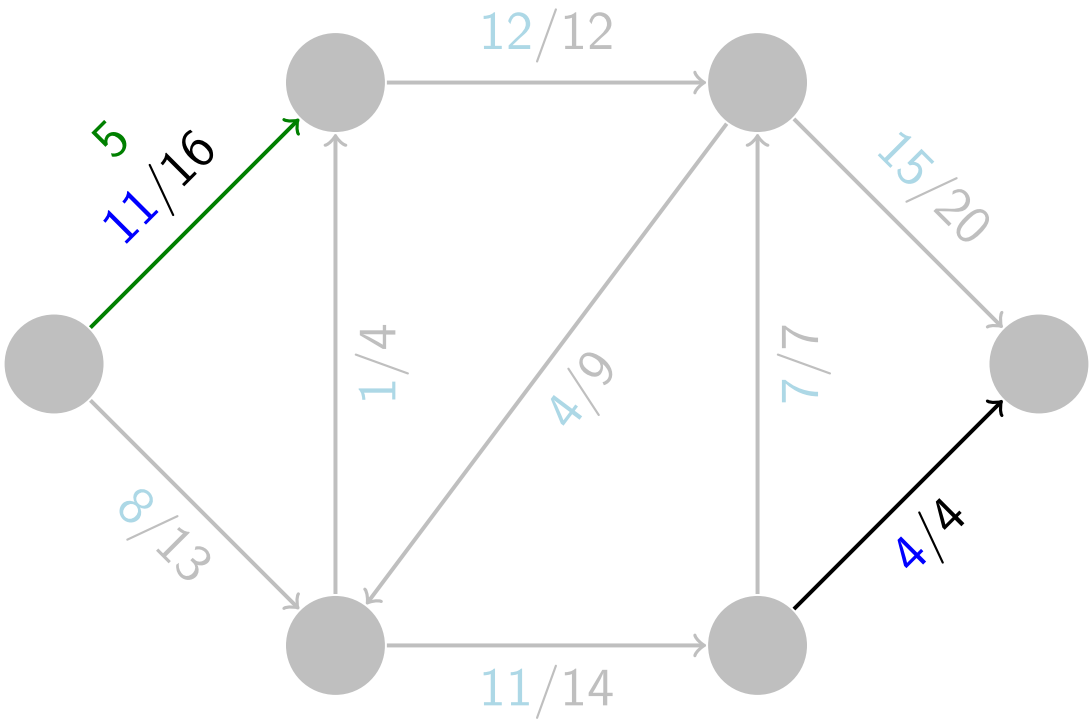
Capacité résiduelle



Capacité résiduelle

- Ce que l'on peut encore envoyer le long d'un arc

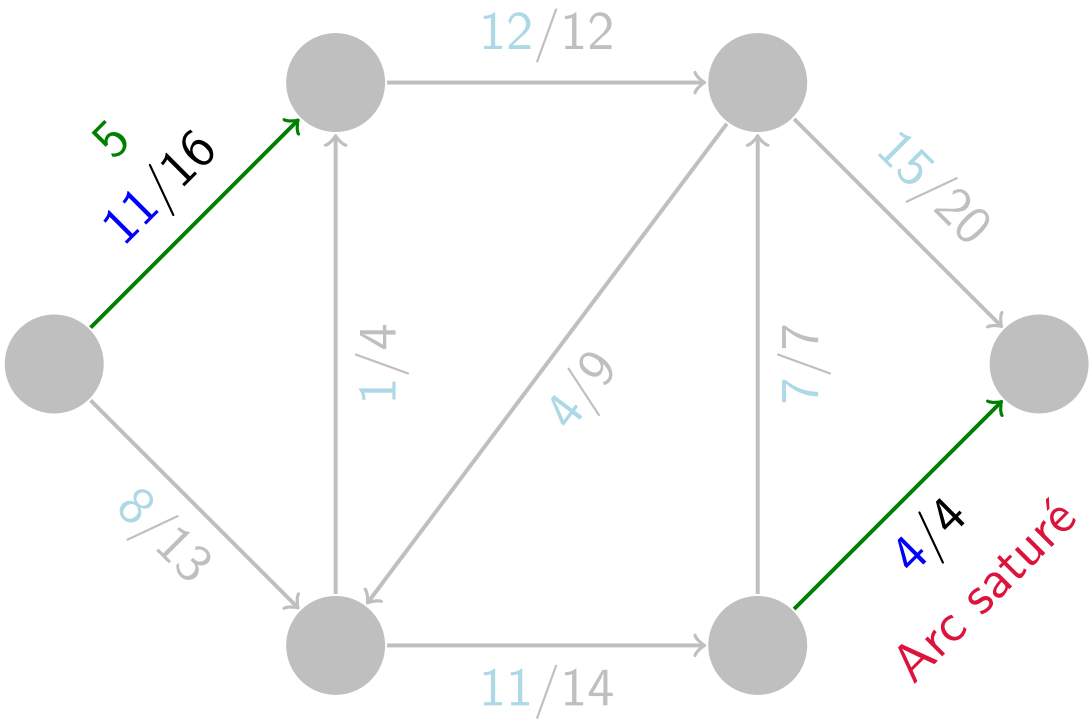
Capacité résiduelle



Capacité résiduelle

- Ce que l'on peut encore envoyer le long d'un arc

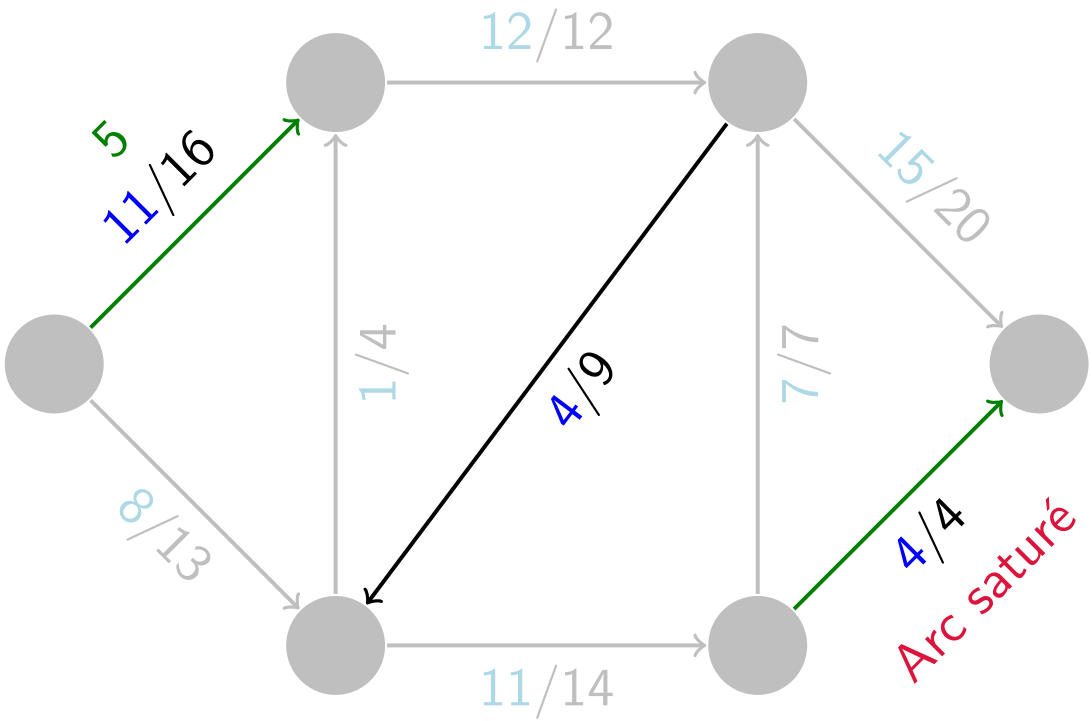
Capacité résiduelle



Capacité résiduelle

- Ce que l'on peut encore envoyer le long d'un arc

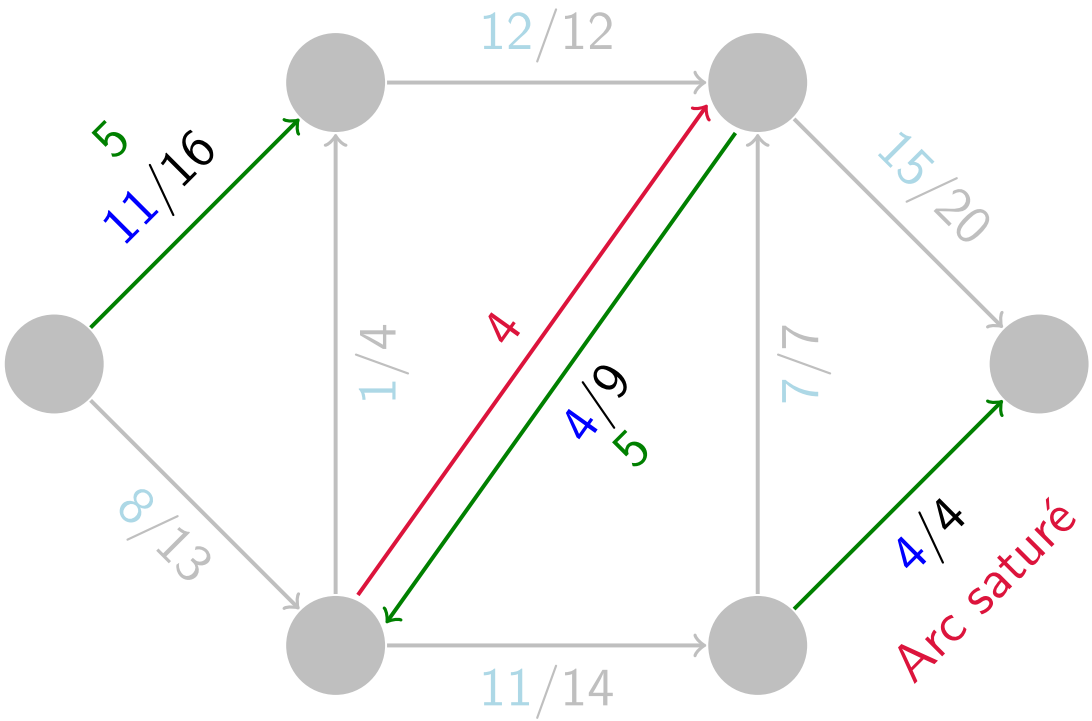
Capacité résiduelle



Capacité résiduelle

- Ce que l'on peut encore envoyer le long d'un arc

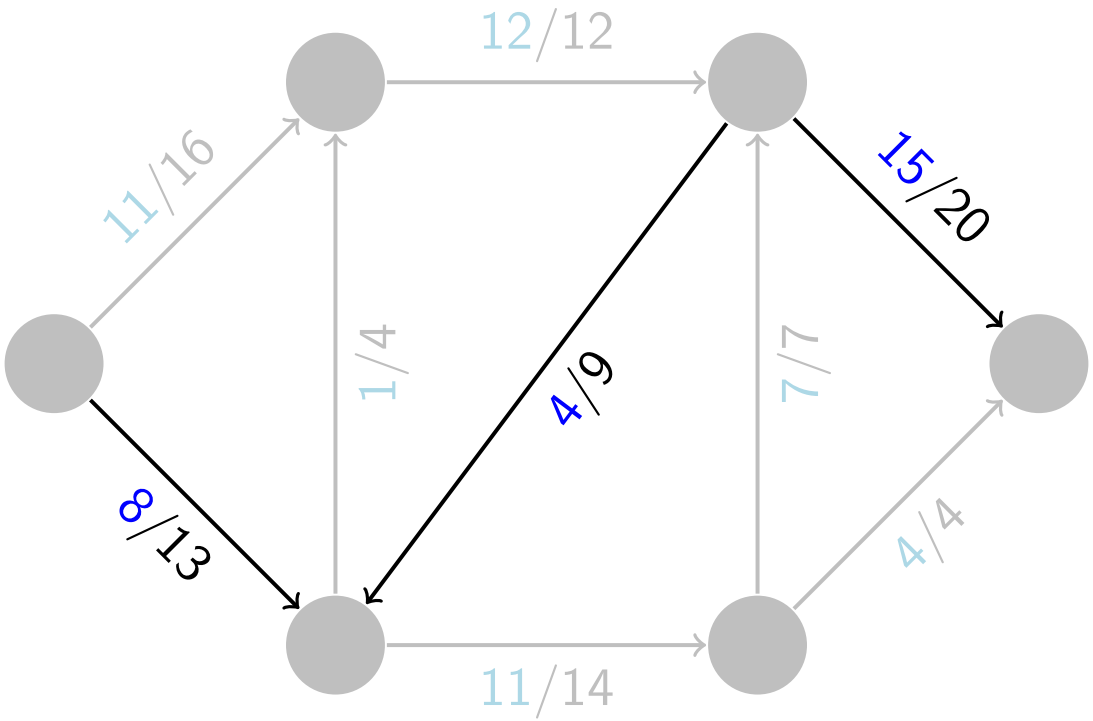
Capacité résiduelle



Capacité résiduelle

- Ce que l'on peut encore envoyer le long d'un arc
- Ce que l'on peut annuler en sens inverse
  - pour simplifier : pas d'annulation vers s ou depuis t.

Chaîne augmentante (principe)

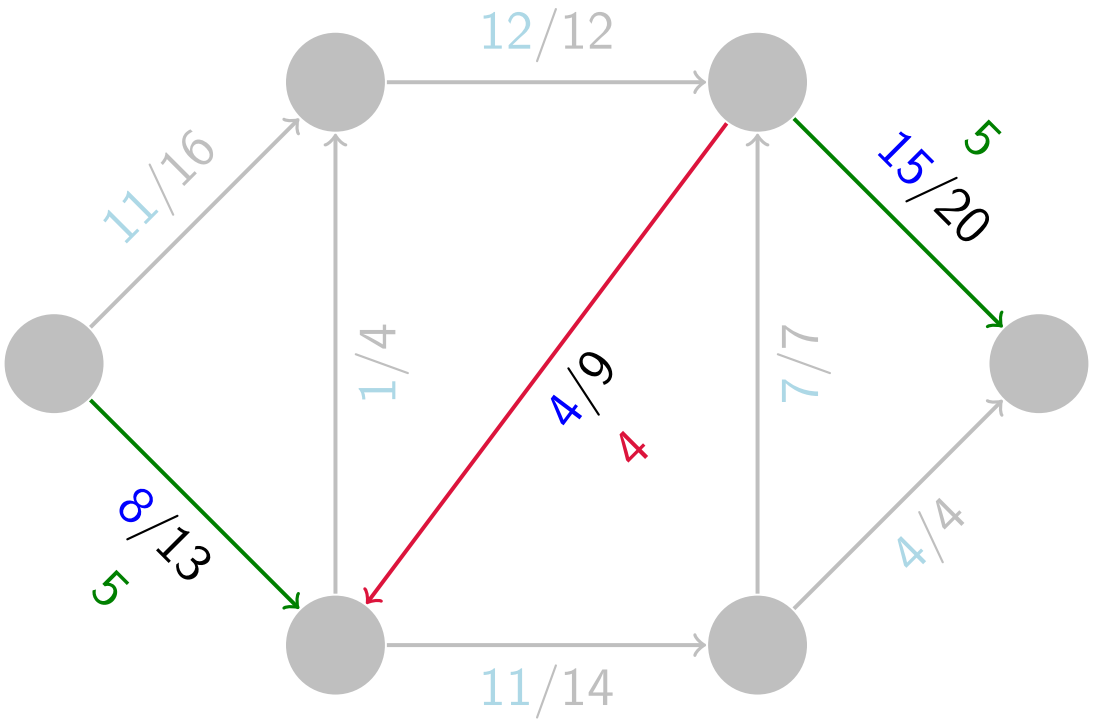


Chaîne augmentante

➜ « chemin » de s à t avec des capacités résiduelles

dans un sens ou dans l'autre...

Chaîne augmentante (principe)



Chaîne augmentante

➜ « chemin » de s à t avec des capacités résiduelles

dans un sens ou dans l'autre...

Méthode de Ford-Fulkerson (1962)

Principe général

Algorithme itératif

➜ Augmenter progressivement le flot jusqu'à saturation

Idée de l'algorithme

1. Trouver une chaîne qui relie s et t et qui possède une capacité résiduelle

   par une méthode qui reste à déterminer...
2. Augmenter au maximum le flot dans cette chaîne
3. Recommencer jusqu'à ce qu'on ne puisse plus augmenter le flot

Méthode de Ford-Fulkerson (1962)

Principe général

Algorithme itératif

➜ Augmenter progressivement le flot jusqu'à saturation

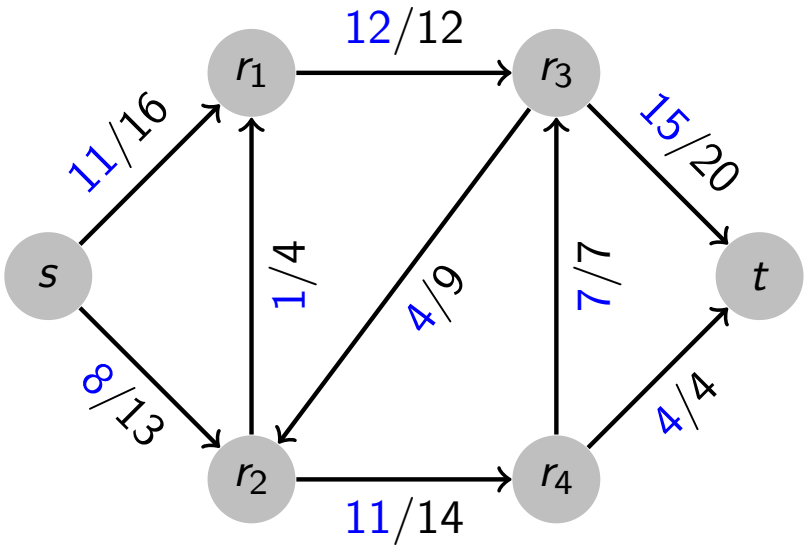
Idée de l'algorithme

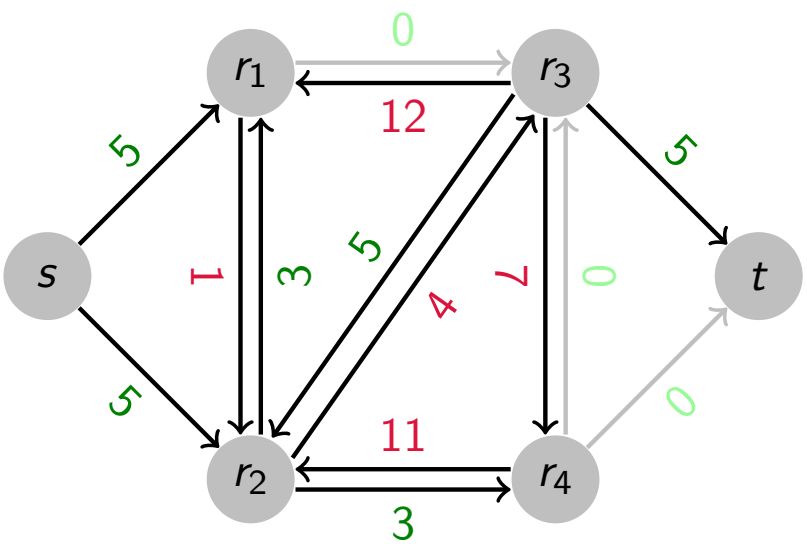
1. Trouver une chaîne qui relie s et t et qui possède une capacité résiduelle

   par une méthode qui reste à déterminer...
2. Augmenter au maximum le flot dans cette chaîne
3. Recommencer jusqu'à ce qu'on ne puisse plus augmenter le flot

➜ Comment implémenter cet algorithme ?

Définition : Graphe résiduel

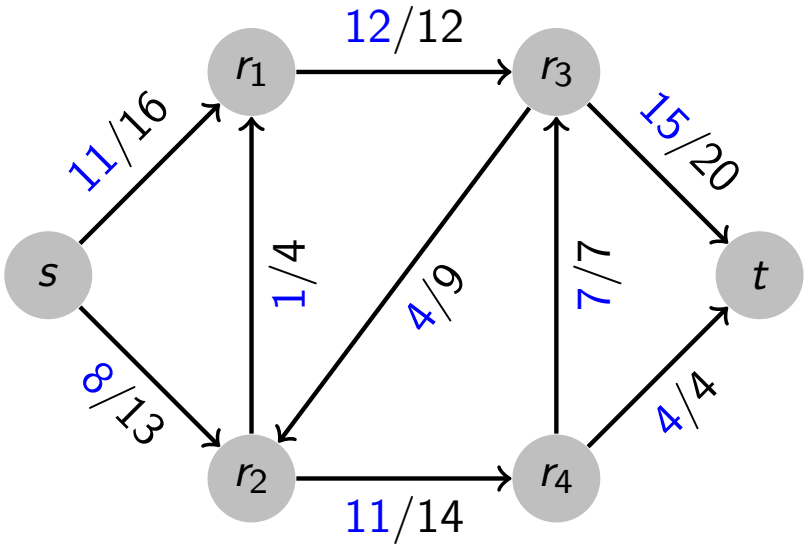


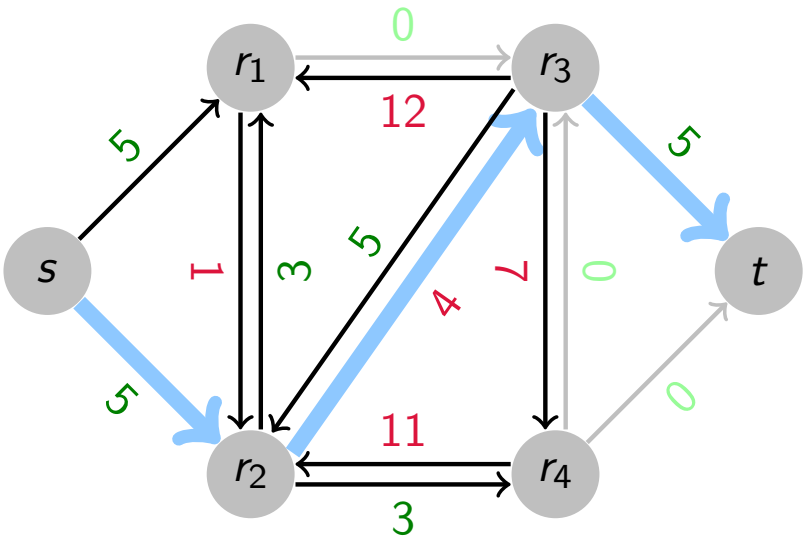


Graphe résiduel

- Le graphe induit par les capacités résiduelles

Exemple graphe de flot : Chaîne augmentante





Chaîne augmentante

Une chaîne augmentante est un chemin entre s et t dans le graphe résiduel

➜ On peut augmenter le flot par la valeur minimale des capacités résiduelles sur une chaîne augmentante

Définitions

Capacité résiduelle

La capacité résiduelle de (u, v) est la valeur de flot supplémentaire que l'on peut envoyer ou délester :

$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v) & \text{si } (u,v) \in E \\ f(v,u) & \text{si } (v,u) \in E \\ 0 & \text{sinon.} \end{cases}$$

Graphe résiduel

Le graphe résiduel de G induit par f est le graphe $G_f = (V, E_f)$ où :

$$E_f = \{(u,v) \in V \times V \mid c_f(u,v) > 0\}.$$

Définitions

Chaîne augmentante

Une chaîne augmentante dans G est un chemin entre s et t dans le graphe résiduel de G induit par f.

Capacité d'une chaîne

La capacité résiduelle d'une chaîne augmentante p est la valeur maximale par laquelle on peut augmenter le flot le long de p :

$$c_f(p) = \min\{c_f(u, v) \mid (u, v) \in p\}$$

Définitions

Chaîne augmentante

Une chaîne augmentante dans G est un chemin entre s et t dans le graphe résiduel de G induit par f.

Capacité d'une chaîne

La capacité résiduelle d'une chaîne augmentante p est la valeur maximale par laquelle on peut augmenter le flot le long de p :

$$c_f(p) = \min\{c_f(u, v) \mid (u, v) \in p\}$$

Lors de l'augmentation

- Uniquement la chaîne en question est concernée.
- La loi de conservation du flot reste respectée.

  La sommes des flots entrants est égale à la somme des flots sortants.

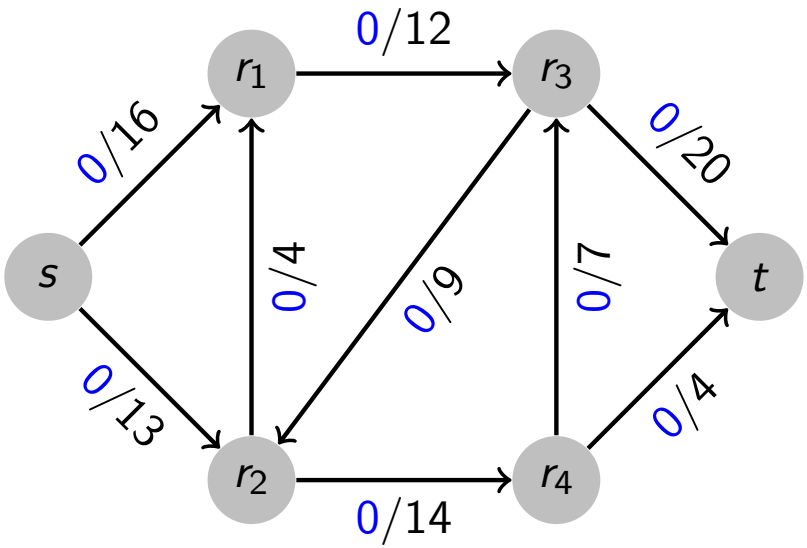
Méthode de Ford-Fulkerson (1962)

Idée générale

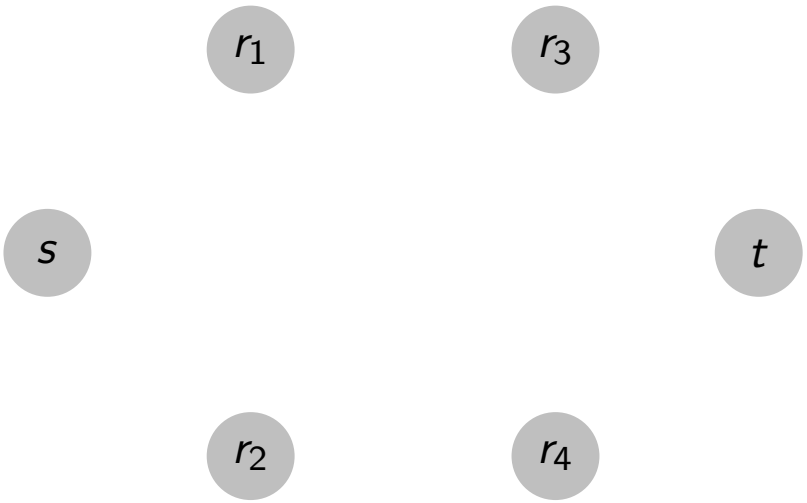
```
def FordFulkerson(G, s, t):
    # initialiser Gr par G et f par 0
    Gr, f = ...

    while True:
        # trouver une chaine augmentante
        aug_path = search_aug_path(Gr, s, t)
        if not aug_path :
            break

        # calculer la capacite residuelle de la chaine
        aug_flow = cf_path(Gr, aug_path)

        # mise a jour du flot et du graphe residuel
        Gr, f = update_flow_graph(Gr, f, aug_path, aug_flow)

    return f
```

Ford-Fulkerson : animation

Graphe à flot

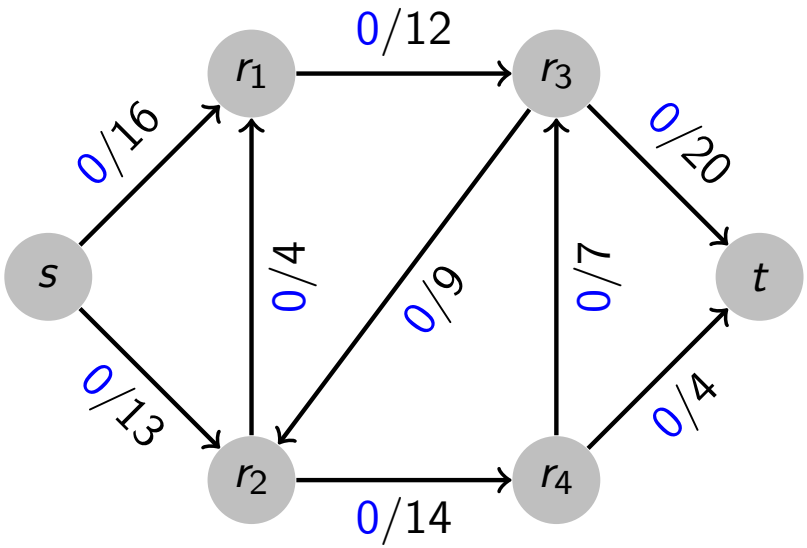


Graphe résiduel

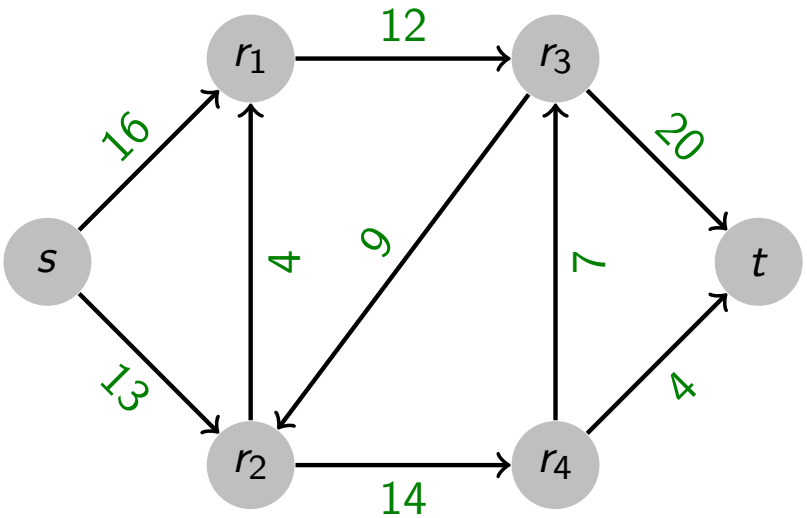


Ford-Fulkerson : animation

Graphe à flot

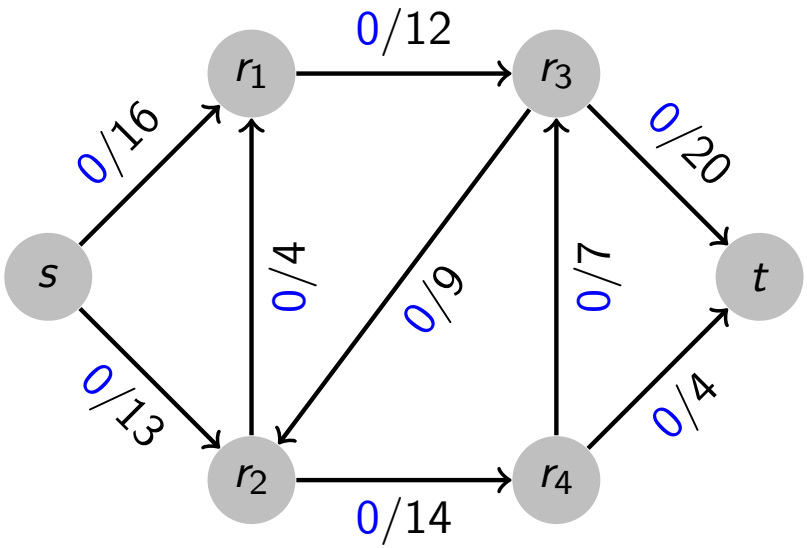


Graphe résiduel

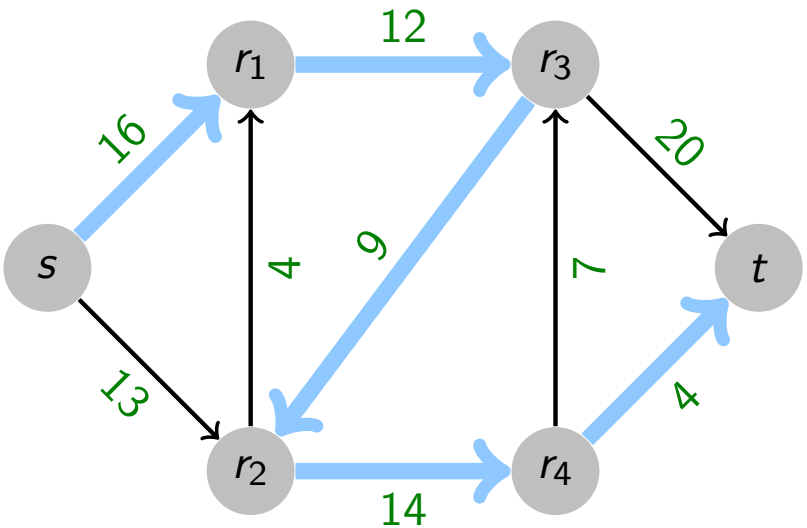


Ford-Fulkerson : animation

Graphe à flot

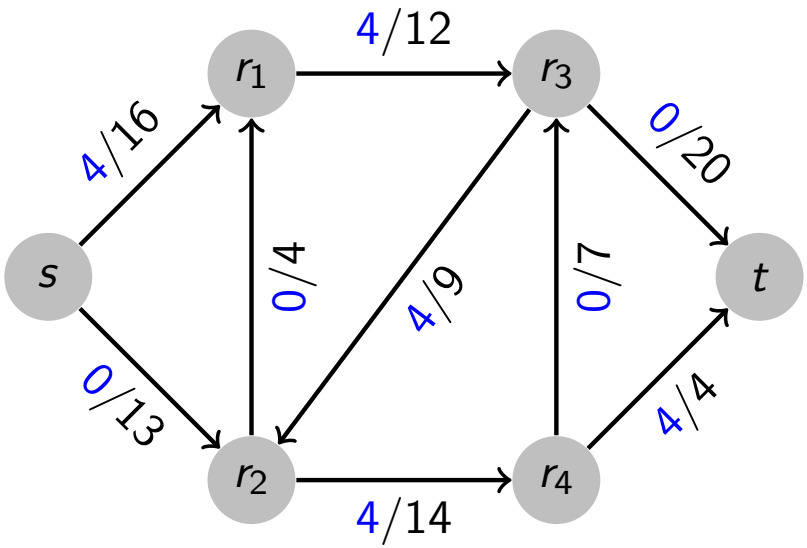


Graphe résiduel

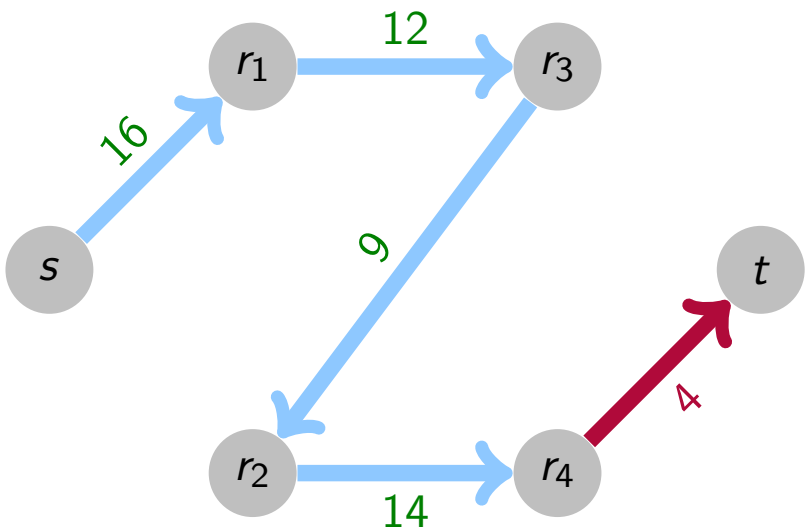


Ford-Fulkerson : animation

Graphe à flot

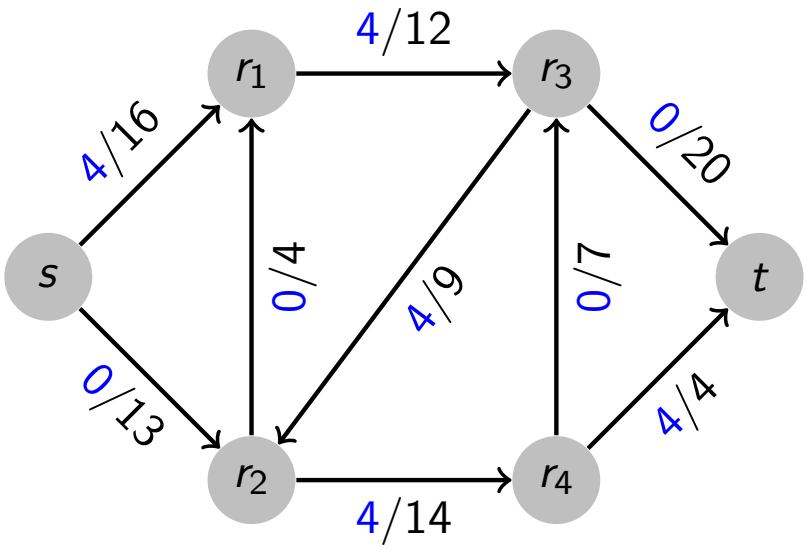


Graphe résiduel

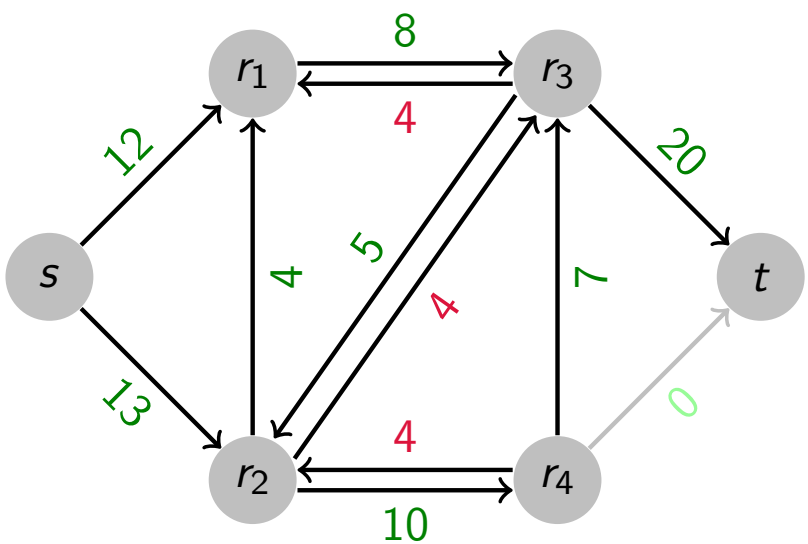


Ford-Fulkerson : animation

Graphe à flot

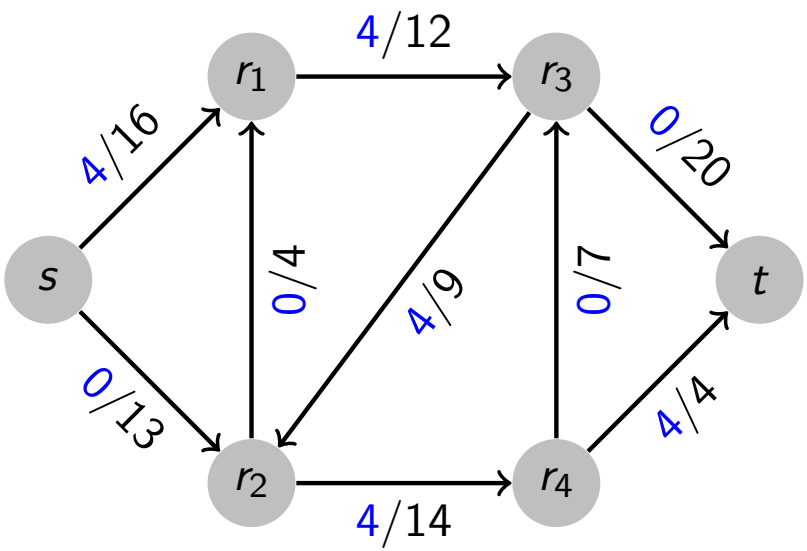


Graphe résiduel

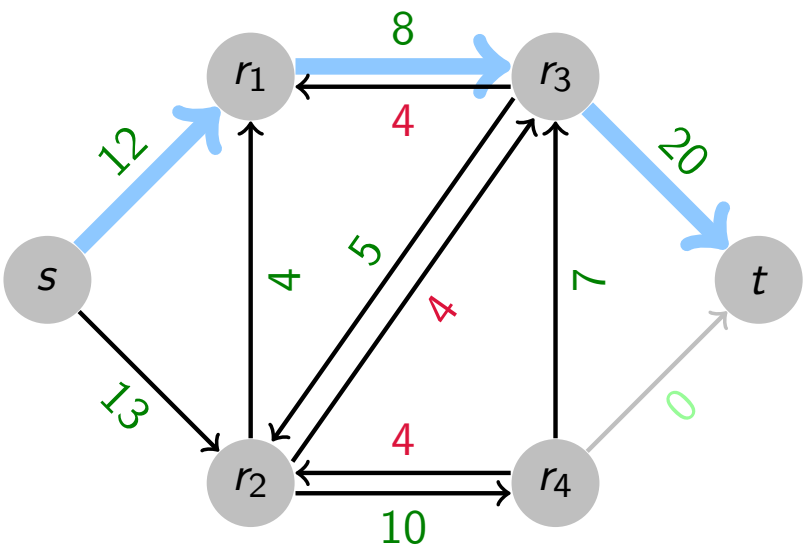


Ford-Fulkerson : animation

Graphe à flot

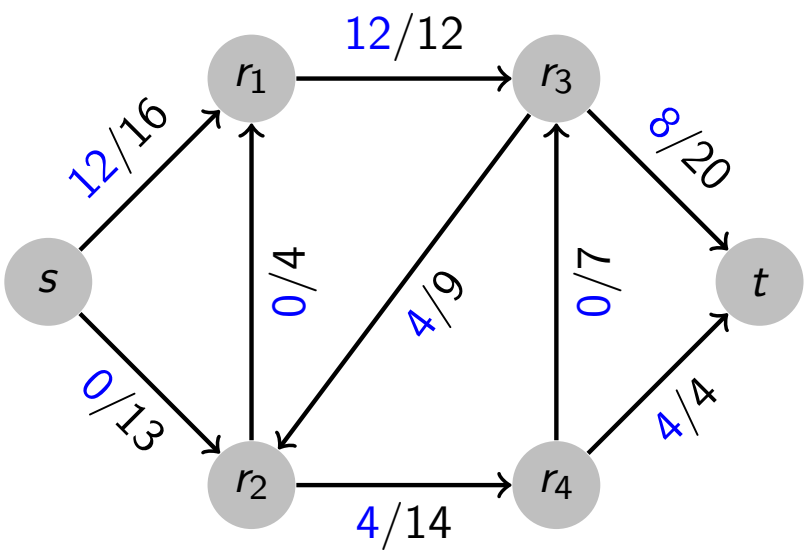


Graphe résiduel

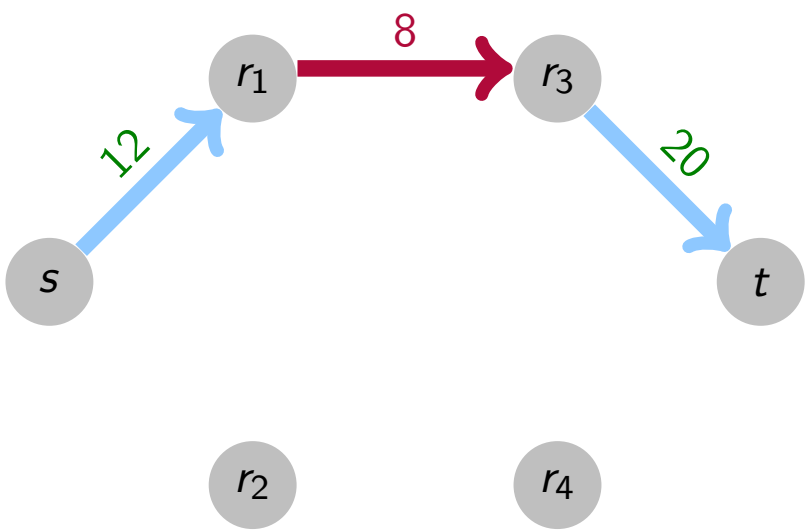


Ford-Fulkerson : animation

Graphe à flot

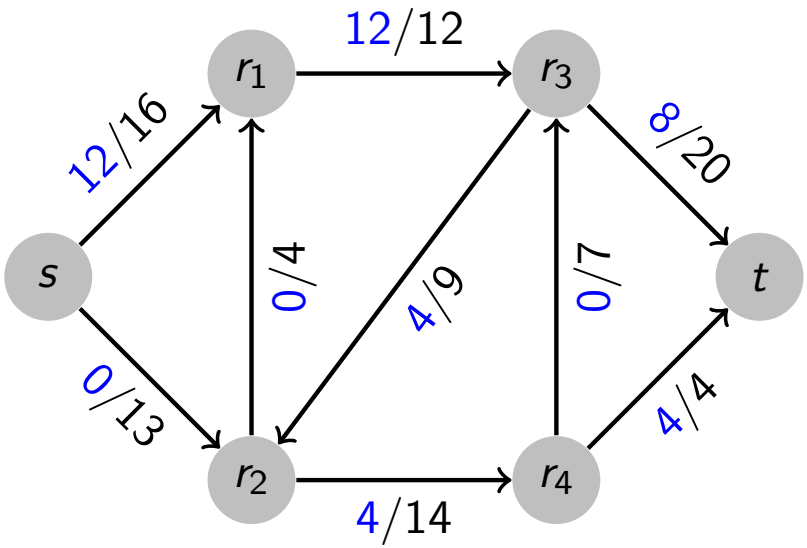


Graphe résiduel

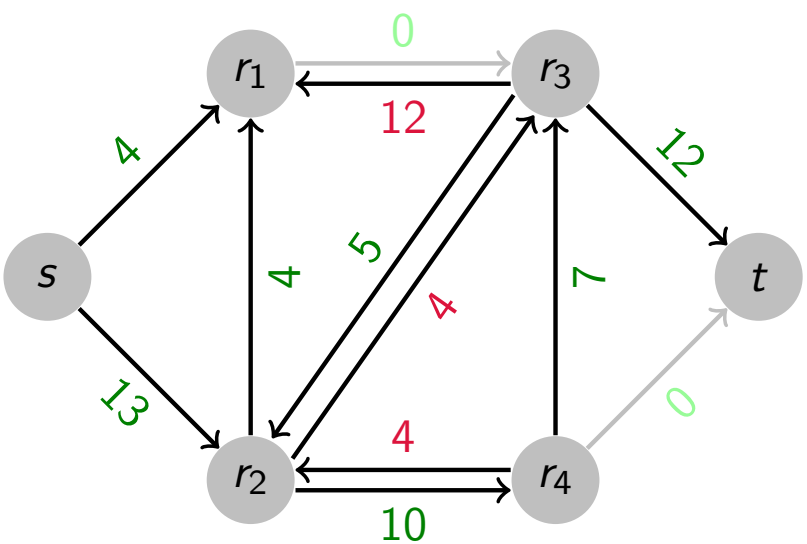


Ford-Fulkerson : animation

Graphe à flot

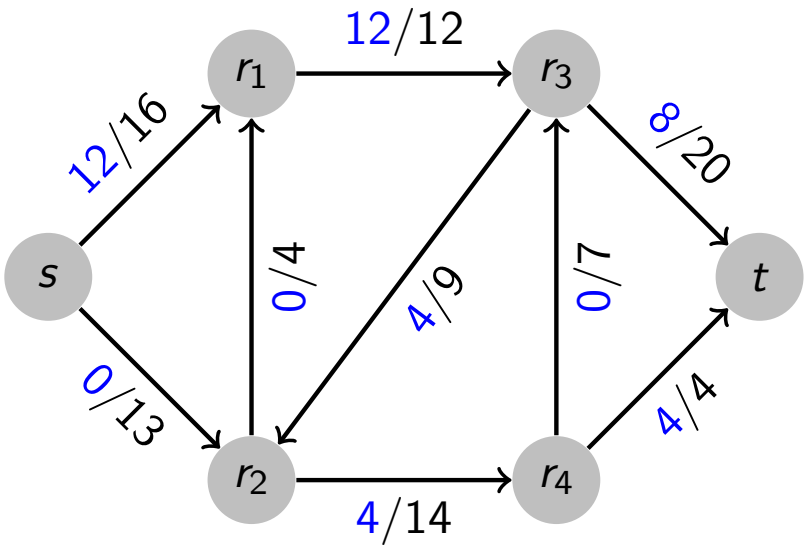


Graphe résiduel

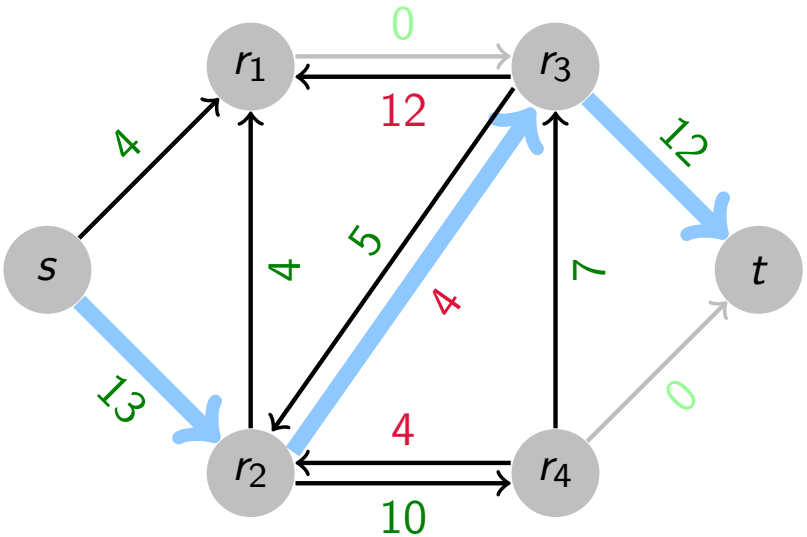


Ford-Fulkerson : animation

Graphe à flot

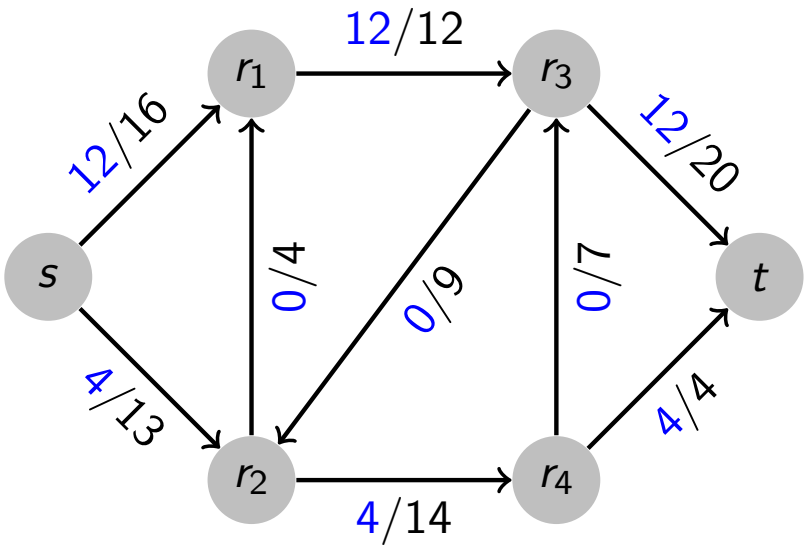


Graphe résiduel

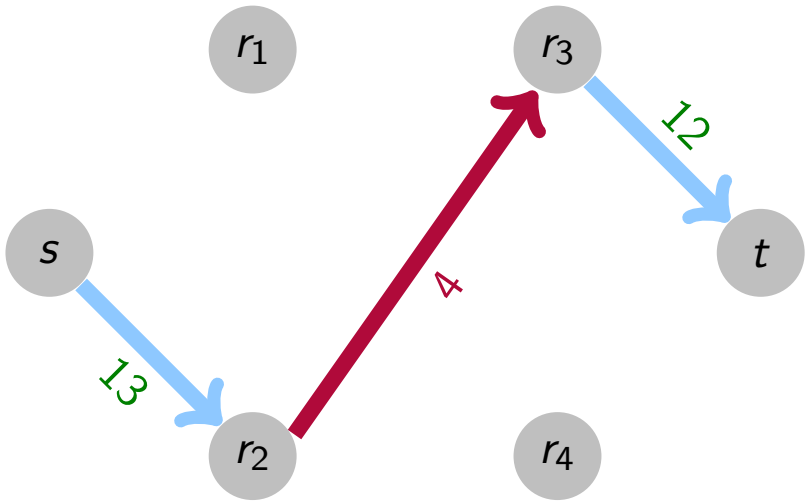


Ford-Fulkerson : animation

Graphe à flot

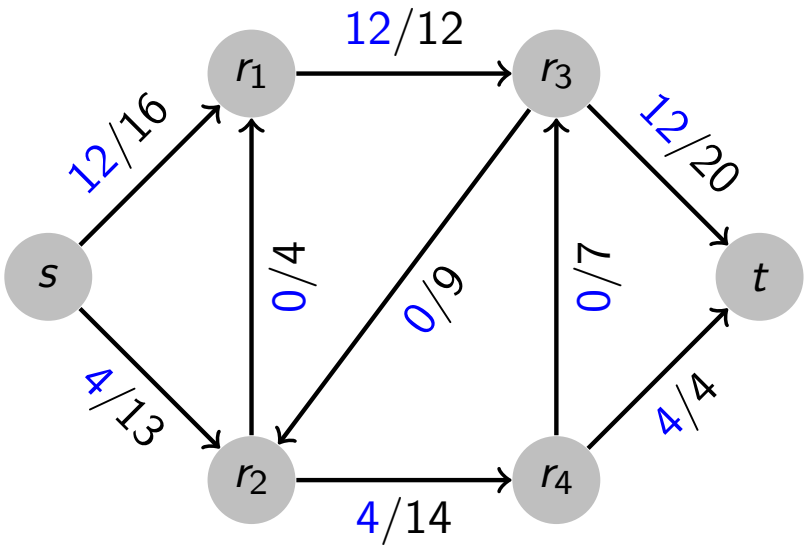


Graphe résiduel

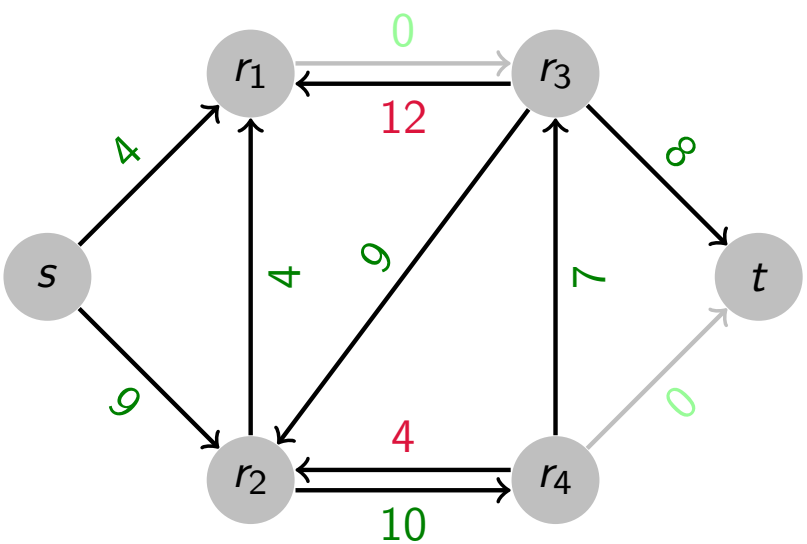


Ford-Fulkerson : animation

Graphe à flot

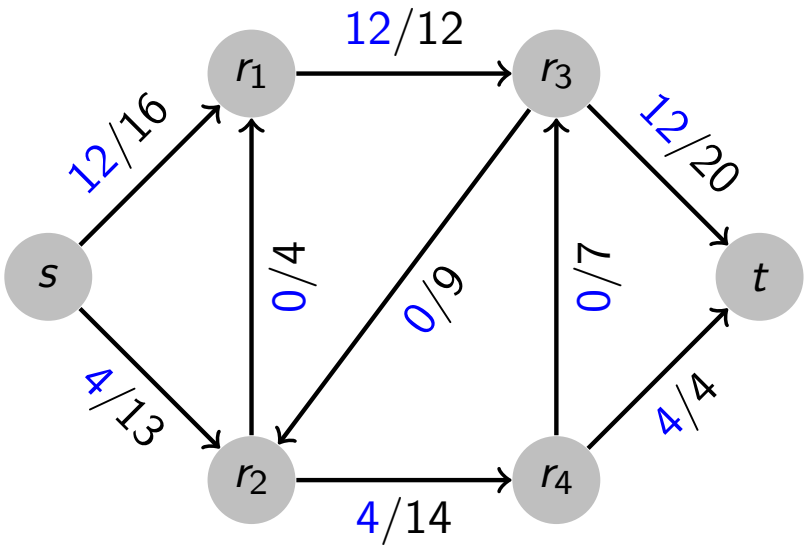


Graphe résiduel

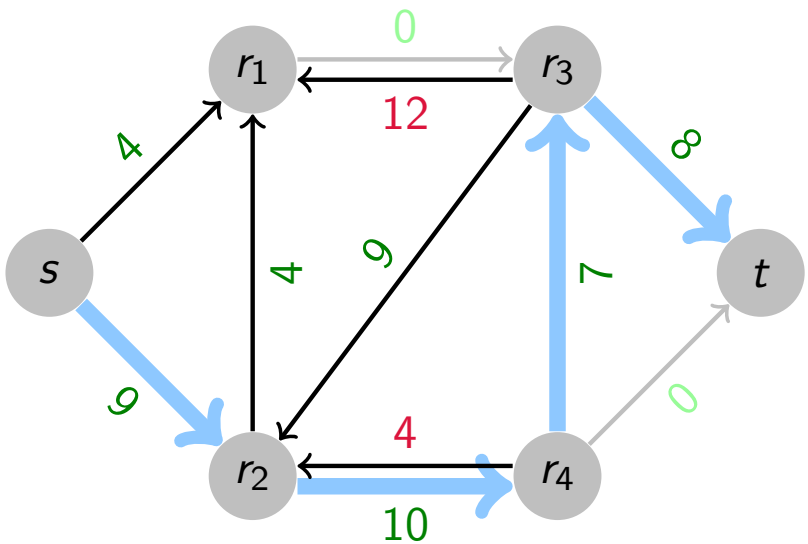


Ford-Fulkerson : animation

Graphe à flot

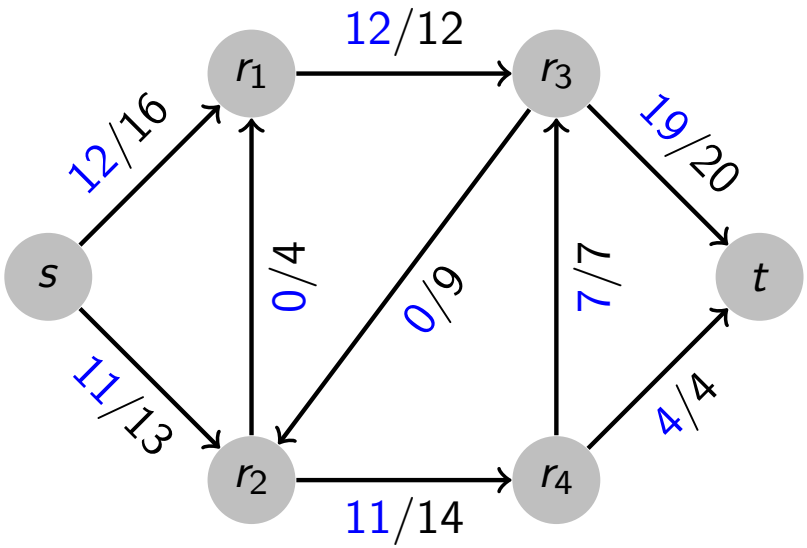


Graphe résiduel

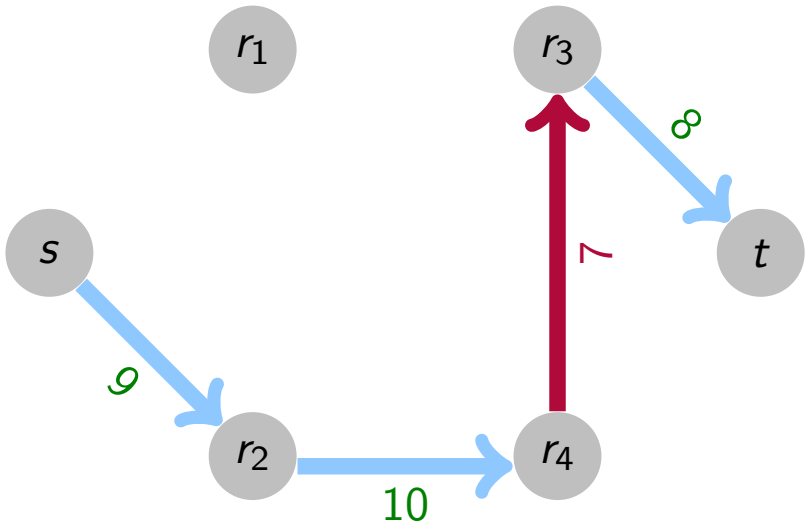


Ford-Fulkerson : animation

Graphe à flot

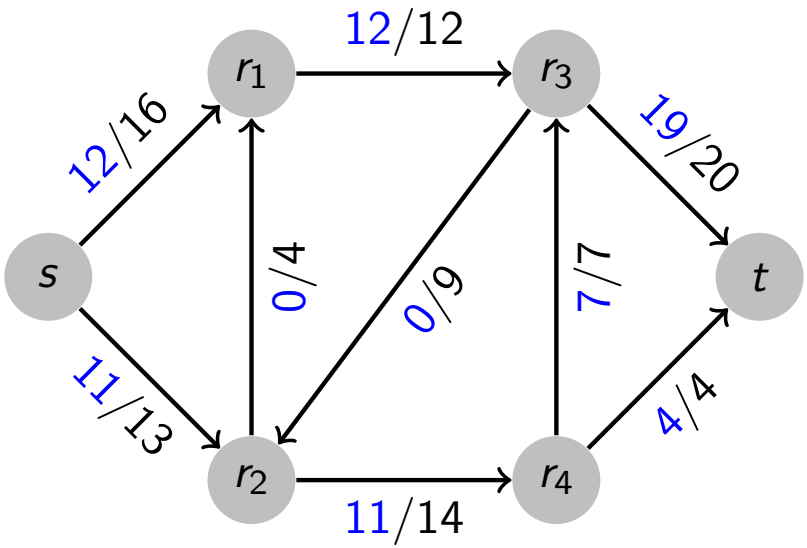


Graphe résiduel

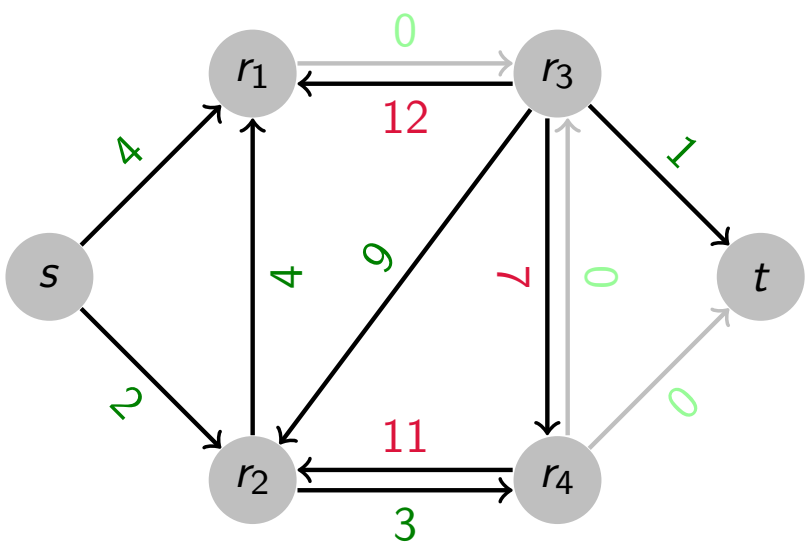


Ford-Fulkerson : animation

Graphe à flot

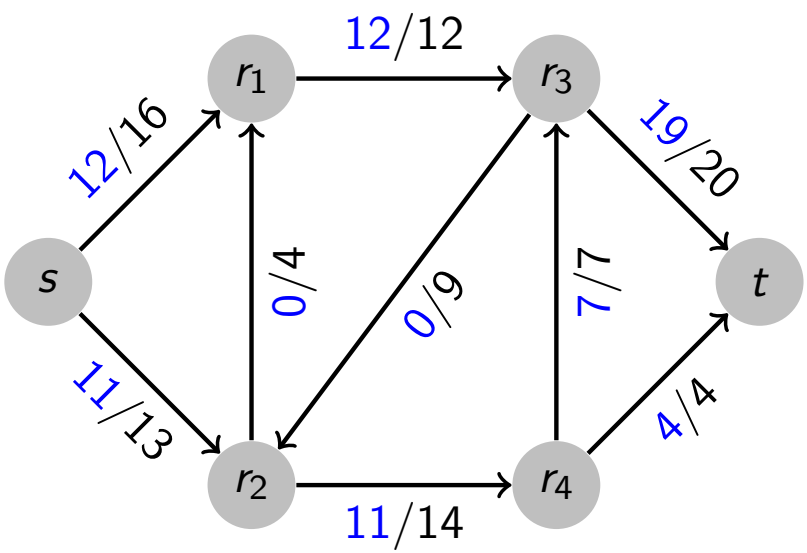


Graphe résiduel

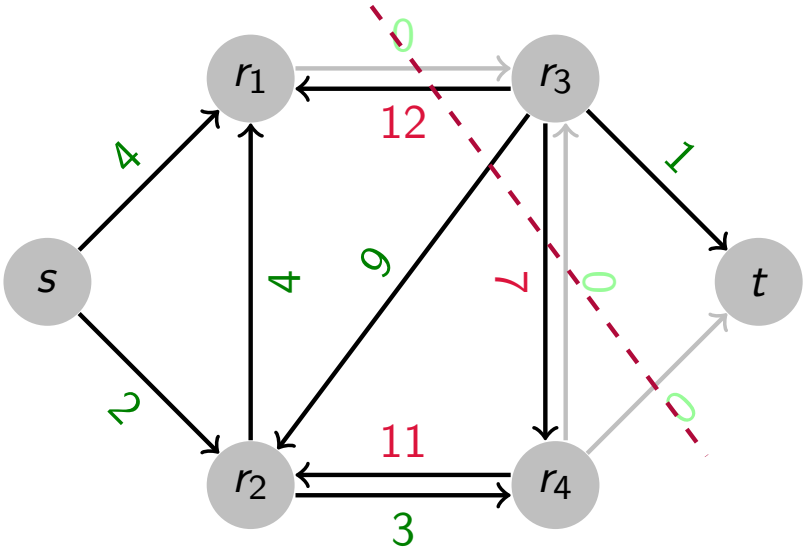


Ford-Fulkerson : animation

Graphe à flot



Graphe résiduel



Terminaison

L'algorithme s'arrête lorsqu'il n'y a plus de chemin entre s et t dans le graphe résiduel

Ford-Fulkerson et augmentation du flot

Théorème

- Soit f un flot dans un graphe de flot G.
- Soit $c_f(p)$ la capacité résiduelle d'une chaîne augmentante p dans le graphe résiduel G_f induit par f sur G.

➜ Le flot $f' = f + c_f(p)$ déterminé par l'ajout de $c_f(p)$ le long de p dans f est un flot sur G et $\varphi' > \varphi$

- Ford-Fulkerson augmente bien le flot de façon répétitive,

Ford-Fulkerson et augmentation du flot

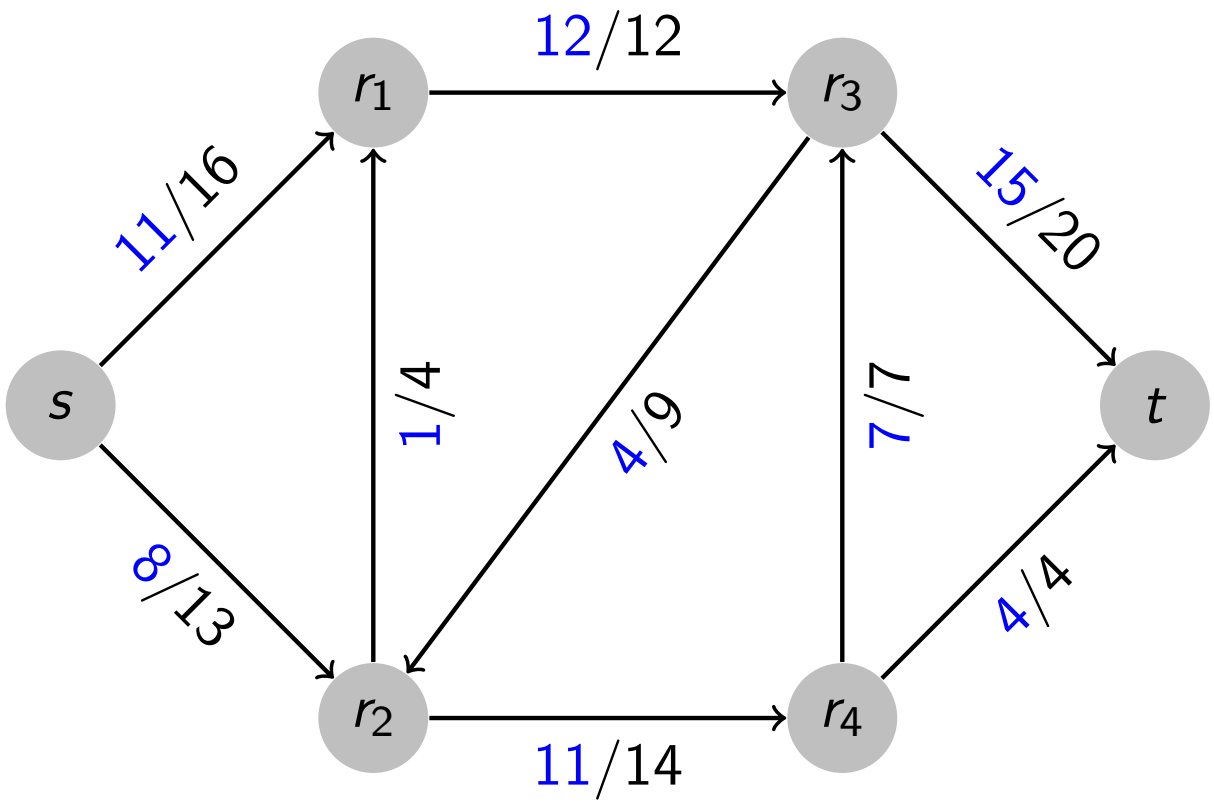
Théorème

- Soit f un flot dans un graphe de flot G.
- Soit $c_f(p)$ la capacité résiduelle d'une chaîne augmentante p dans le graphe résiduel G_f induit par f sur G.

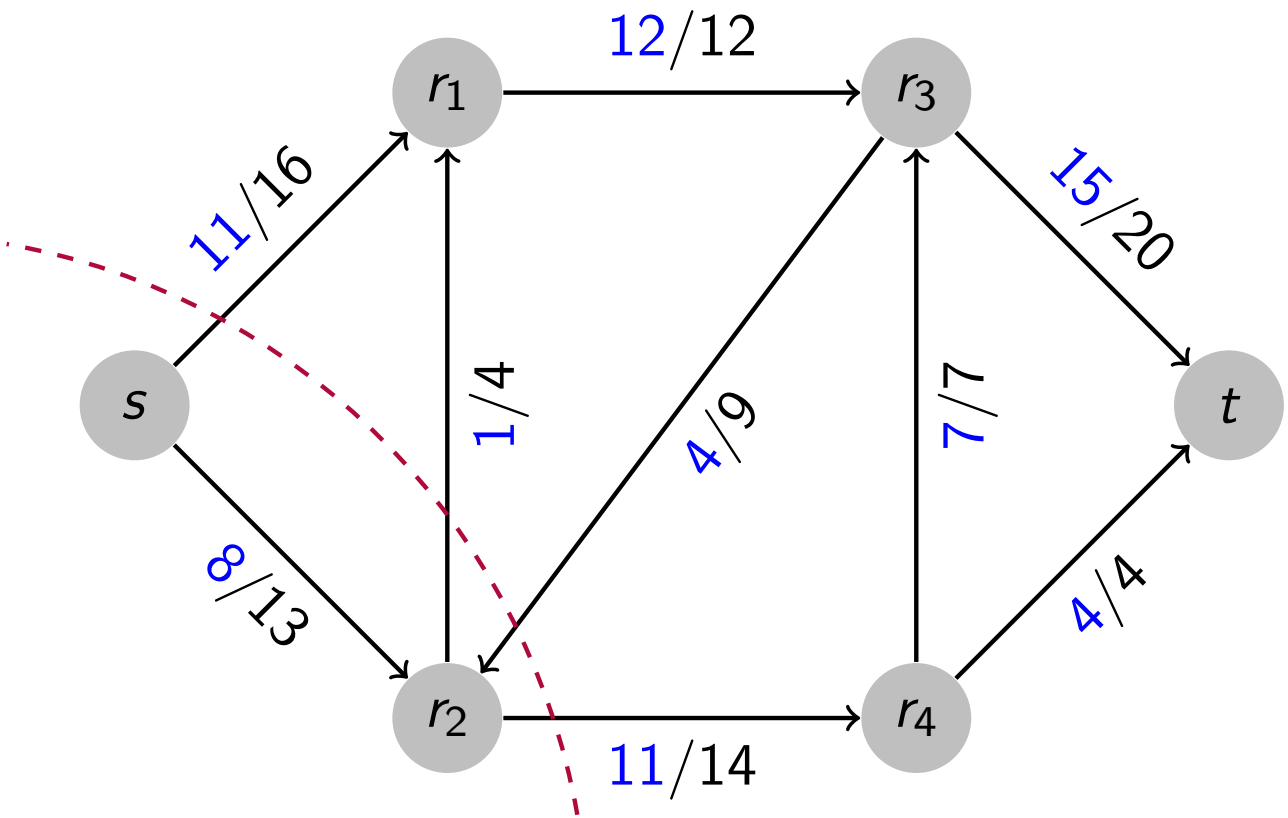
➜ Le flot $f' = f + c_f(p)$ déterminé par l'ajout de $c_f(p)$ le long de p dans f est un flot sur G et $\varphi' > \varphi$

- Ford-Fulkerson augmente bien le flot de façon répétitive,
- Pourquoi Ford-Fulkerson converge vers le flot maximum ?

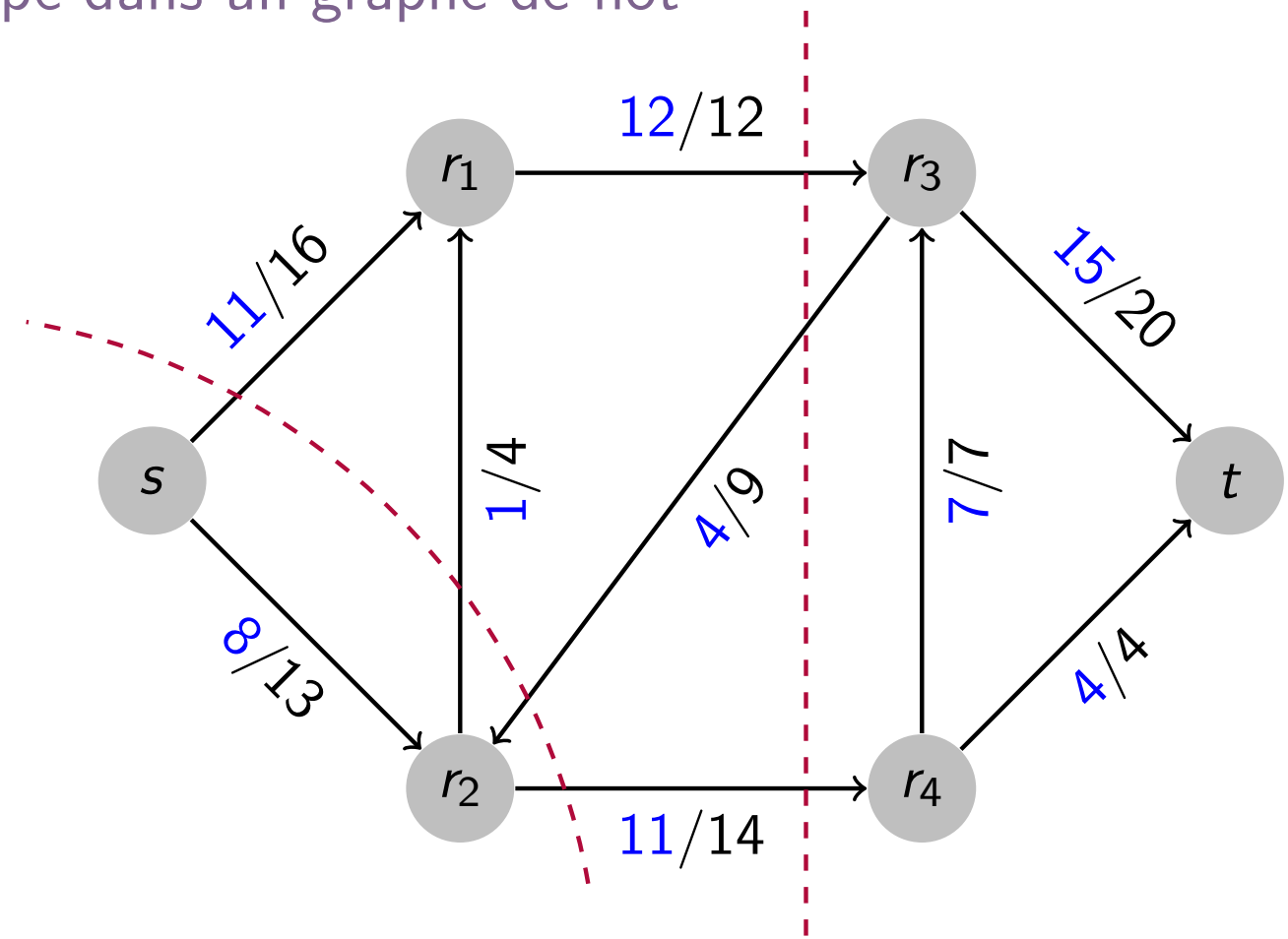
Coupe dans un graphe de flot



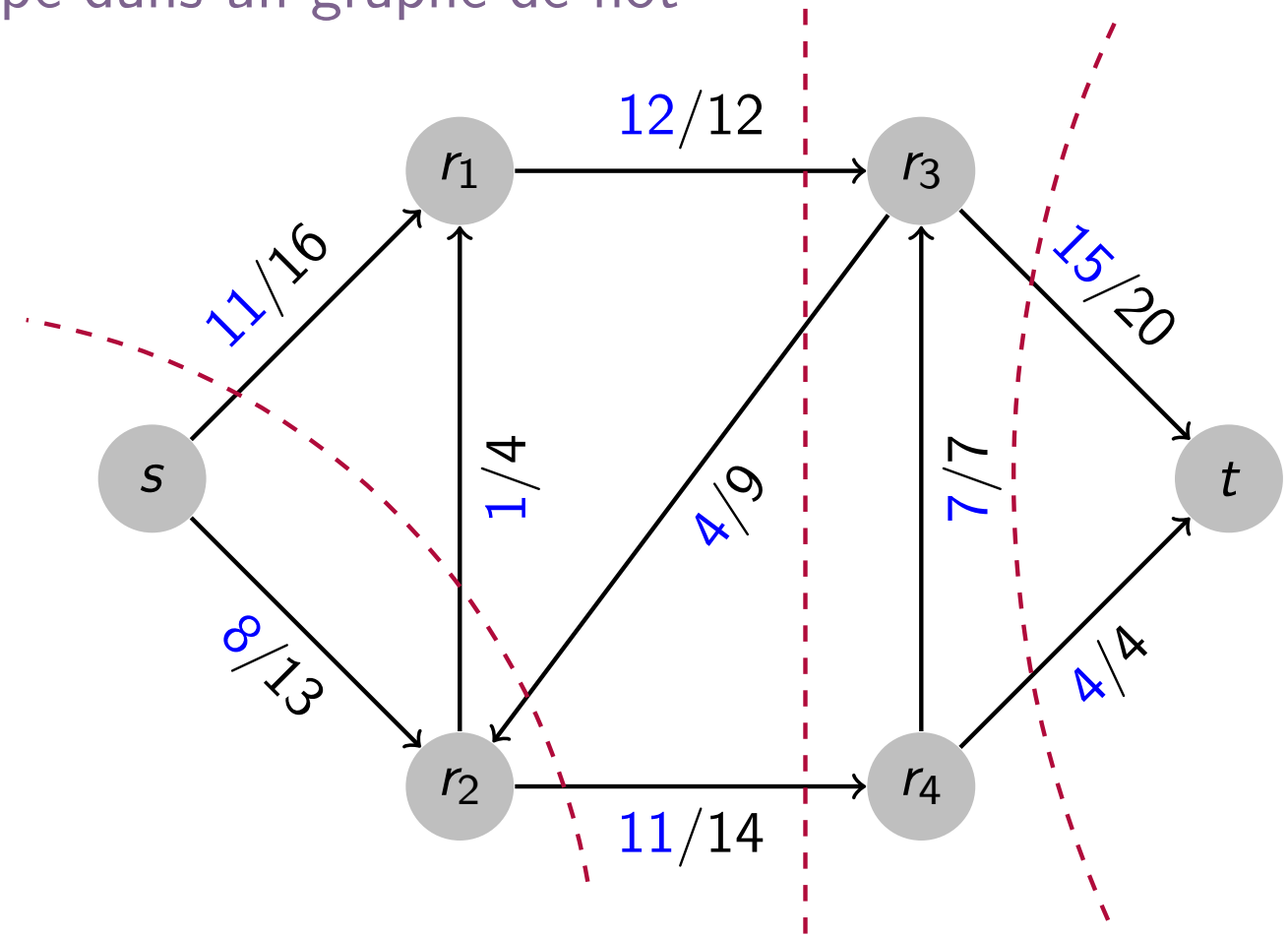
Coupe dans un graphe de flot



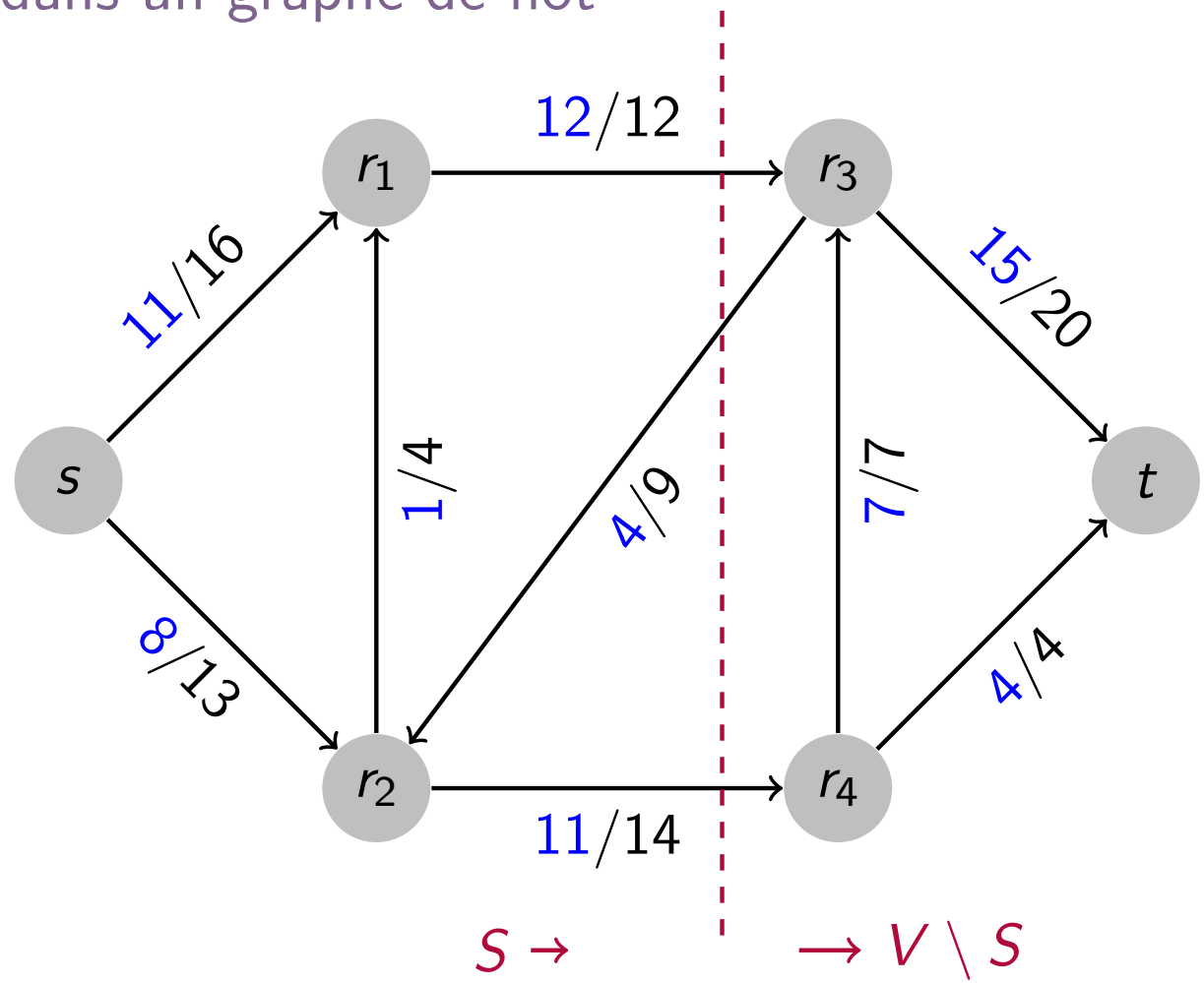
Coupe dans un graphe de flot



Coupe dans un graphe de flot



Coupe dans un graphe de flot



Coupe dans un graphe de flot

Précisons la définition d'une coupe pour un graphe "à flot"

Definition

Une coupe s–t est une partition de V en S et $T = V \setminus S$ telle que $s \in S$ et $t \in T = V \setminus S$.

Coupe dans un graphe de flot

Précisons la définition d'une coupe pour un graphe "à flot"

Definition

Une coupe s–t est une partition de V en S et $T = V \setminus S$ telle que $s \in S$ et $t \in T = V \setminus S$.
Sa capacité

$$c(S, T) = \sum_{(u,v) \in S \times T} c(u, v).$$

Coupe dans un graphe de flot

Précisons la définition d'une coupe pour un graphe "à flot"

Definition

Une coupe s–t est une partition de V en S et $T = V \setminus S$ telle que $s \in S$ et $t \in T = V \setminus S$.
Sa capacité

$$c(S, T) = \sum_{(u,v) \in S \times T} c(u, v).$$

La valeur du flot transporté par cette coupe

$$f(S, T) = \sum_{(u,v) \in S \times T} f(u, v) - \sum_{(u,v) \in T \times S} f(u, v).$$

Comment déterminer la valeur du flot acheminé par G ?

Définition et théorème

La valeur de flot φ dans G est égale à $f(S, T)$ pour n'importe quelle coupe s–t.

Comment déterminer la valeur du flot acheminé par G ?

Définition et théorème

La valeur de flot φ dans G est égale à $f(S, T)$ pour n'importe quelle coupe s–t.

$\rightarrow$ On peut montrer que c'est toujours le même !

Intuition : si on fait passer un seul nœud de S à T, cela ne change rien pour le flot. . .

Comment déterminer la valeur du flot acheminé par G ?

Définition et théorème

La valeur de flot φ dans G est égale à $f(S, T)$ pour n'importe quelle coupe s–t.

$\rightarrow$ On peut montrer que c'est toujours le même !

Intuition : si on fait passer un seul nœud de S à T, cela ne change rien pour le flot. . .

Deux coupes s–t quelconques, néanmoins "spéciales"

- $S = \{s\}$ (valeur à la source)
- $S = V \setminus \{t\}$ (valeur au sommet terminal)

Définitions

La valeur du flot est bornée par la capacité d'une coupe :

$$\varphi \leq c(S, T)$$

Définitions

La valeur du flot est bornée par la capacité d'une coupe :

$$\varphi \leq c(S, T)$$

Definition

Un arc $(u, v) \in E$ est dit saturé si $f(u, v) = c(u, v)$.

Par extension, une coupe s–t est saturée si $\varphi = c(S, T)$.

(Le flot est égal à la capacité de la coupe)

Définitions

La valeur du flot est bornée par la capacité d'une coupe :

$$\varphi \leq c(S, T)$$

Definition

Un arc $(u, v) \in E$ est dit saturé si $f(u, v) = c(u, v)$.

Par extension, une coupe s–t est saturée si $\varphi = c(S, T)$.

(Le flot est égal à la capacité de la coupe)

Definition

On parle de coupe minimale pour désigner une coupe de capacité minimale.

Définitions

La valeur du flot est bornée par la capacité d'une coupe :

$$\varphi \leq c(S, T)$$

Definition

Un arc $(u, v) \in E$ est dit saturé si $f(u, v) = c(u, v)$.

Par extension, une coupe s–t est saturée si $\varphi = c(S, T)$.

(Le flot est égal à la capacité de la coupe)

Definition

On parle de coupe minimale pour désigner une coupe de capacité minimale.

Corollaire : Une coupe saturée est une coupe minimale.

Théorème principal : flot max $\Leftrightarrow$ coupe min $\Leftrightarrow$ aucune chaîne augmentante

Théorème

Les trois propositions ci-dessous sont équivalentes :

1. Le flot φ entre s et t est maximal.
2. Il n'y a aucune chaîne augmentante.
3. Il existe une coupe s–t dont la capacité est égale à φ.

Théorème principal : flot max $\Leftrightarrow$ coupe min $\Leftrightarrow$ aucune chaîne augmentante

Théorème

Les trois propositions ci-dessous sont équivalentes :

1. Le flot φ entre s et t est maximal.
2. Il n'y a aucune chaîne augmentante.
3. Il existe une coupe s–t dont la capacité est égale à φ.

Démonstration

1. 3 $\Rightarrow$ 1 : coupe=flot $\Rightarrow$ flot max
2. 1 $\Rightarrow$ 2 : flot max $\Rightarrow$ pas de chaîne augmentante
3. 2 $\Rightarrow$ 3 : pas de chaîne augmentante $\Rightarrow$ coupe=flot

⏩ skip proof

3 ⇒ 1 : coupe min et flot max

- Par définition du flot ($\forall e, f(e) \leq c(e)$) et de la capacité d'une coupe (somme des capacités), la valeur du flot est bornée par la capacité d'une coupe :

$$\varphi \leq c(S, T)$$

$\rightarrow$ C'est vrai quelque soit la coupe...

3 ⇒ 1 : coupe min et flot max

- Par définition du flot ($\forall e, f(e) \leq c(e)$) et de la capacité d'une coupe (somme des capacités), la valeur du flot est bornée par la capacité d'une coupe :

$$\varphi \leq c(S, T)$$

$\rightarrow$ C'est vrai quelque soit la coupe...

- Si $c(S, T) = \varphi$ (3) alors nécessairement :
  - φ est max (1)
  - $c(S, T)$ est min

$3 \Rightarrow 1$: coupe min et flot max

- Par définition du flot ($\forall e, f(e) \leq c(e)$) et de la capacité d'une coupe (somme des capacités), la valeur du flot est bornée par la capacité d'une coupe :

$$\varphi \leq c(S, T)$$

$\rightarrow$ C'est vrai quelque soit la coupe...

- Si $c(S, T) = \varphi$ (3) alors nécessairement :
  - φ est max (1)
  - $c(S, T)$ est min

On a montré

$3 \Rightarrow 1$

mais aussi : $3 \Rightarrow$ la coupe est minimale

$1 \Rightarrow 2$: flot max et chaîne augmentante

Preuve par contraposition

S'il existe une chaîne augmentante...

... alors on peut augmenter le flot (donc il n'était pas maximum)

1 ⇒ 2 : flot max et chaîne augmentante

Preuve par contraposition

S'il existe une chaîne augmentante. . .

. . . alors on peut augmenter le flot (donc il n'était pas maximum)

Par contraposition, flot max (1) ⇒ pas de chaîne augmentante (2)

2 ⇒ 3 : chaîne augmentante et coupe min

Preuve (1956)

Preuve donnée à la fois par Ford et Fulkerson et par Elias, Feinstein et Shannon

Soit φ la valeur d'un flot sans chaîne augmentante.

Soit S l'ensemble des nœuds atteignables depuis s par des chaînes augmentantes

- $s \in S$ (par définition) et $t \notin S$
  (car il n'y a pas de chaîne augmentante atteignant t)

2 ⇒ 3 : chaîne augmentante et coupe min

Preuve (1956)

Preuve donnée à la fois par Ford et Fulkerson et par Elias, Feinstein et Shannon

Soit φ la valeur d'un flot sans chaîne augmentante.

Soit S l'ensemble des nœuds atteignables depuis s par des chaînes augmentantes

- $s \in S$ (par définition) et $t \notin S$
  (car il n'y a pas de chaîne augmentante atteignant t)
- Nous avons donc une coupe s–t de flot :

$$f(S,T) = \sum_{u \in S, v \in T} f(u,v) - \sum_{u \in T, v \in S} f(u,v)$$

(def. du flot d'une coupe)

2 ⇒ 3 : chaîne augmentante et coupe min

Preuve (1956)

Preuve donnée à la fois par Ford et Fulkerson et par Elias, Feinstein et Shannon

Soit φ la valeur d'un flot sans chaîne augmentante.

Soit S l'ensemble des nœuds atteignables depuis s par des chaînes augmentantes

- $s \in S$ (par définition) et $t \notin S$
  (car il n'y a pas de chaîne augmentante atteignant t)
- Nous avons donc une coupe s–t de flot :

$$f(S,T) = \sum_{u \in S, v \in T} c(u,v) - \sum_{u \in T, v \in S} f(u,v)$$

(les arcs sortants sont saturés, sinon on pourrait atteindre T)

2 ⇒ 3 : chaîne augmentante et coupe min

Preuve (1956)

Preuve donnée à la fois par Ford et Fulkerson et par Elias, Feinstein et Shannon

Soit φ la valeur d'un flot sans chaîne augmentante.

Soit S l'ensemble des nœuds atteignables depuis s par des chaînes augmentantes

- $s \in S$ (par définition) et $t \notin S$
  (car il n'y a pas de chaîne augmentante atteignant t)
- Nous avons donc une coupe s–t de flot :

$$f(S,T) = \sum_{u \in S, v \in T} c(u,v) - \sum_{u \in T, v \in S} 0$$

(le flot venant de T est nul, sinon on pourrait le délester)

2 ⇒ 3 : chaîne augmentante et coupe min

Preuve (1956)

Preuve donnée à la fois par Ford et Fulkerson et par Elias, Feinstein et Shannon

Soit φ la valeur d'un flot sans chaîne augmentante.

Soit S l'ensemble des nœuds atteignables depuis s par des chaînes augmentantes

- $s \in S$ (par définition) et $t \notin S$
  (car il n'y a pas de chaîne augmentante atteignant t)
- Nous avons donc une coupe s–t de flot :

$$f(S,T) = \sum_{u \in S, v \in T} c(u,v)$$

$\Rightarrow$ flot = capacité de la coupe ■

Algorithme de Ford-Fulkerson en Python

```
def FordFulkerson(G, s, t):
    # initialiser Gr et f
    ...

    while True:
        # trouver une chaine augmentante
        aug_path = search_aug_path(Gr, s, t)
        if aug_path == None:
            break
        # calculer la capacite residuelle de la chaine
        aug_flow = cf_path(Gr, aug_path)

        for k in range(len(aug_path)-1):
            u, v = aug_path[k], aug_path[k+1]
            # mise a jour du flot et des capacites residuelles
            # le long de la chaine augmentante
            if v in neighbours(G, u) :
                f[u][v] += aug_flow
                cf[u][v], cf[v][u] = c[u][v] - f[u][v], f[u][v]
            else :
                f[v][u] -= aug_flow
                cf[v][u], cf[u][v] = c[v][u] - f[v][u], f[v][u]
```

Recherche de la chaîne augmentante en Python

```
def search_aug_path(Gr,s,t):

    lnext = [s]
    parent = {s:None} # plus besoin de visited

    while len(lnext)>0:
      n = pop_end(lnext) # DFS ou pop_begin pour BFS

      if n==t:
          return path(parent, t) # retourne la chaine augmentante

      for v in neighbours(Gr, n):
        if not v in parent:
          add_end(v,lnext)
          parent[v] = n

    return None # aucune chaine augmentante
```

N.B. F-F classique utilise DFS, mais le choix de la méthode pour établir une chaîne est libre, par exemple, Edmonds et Karp suggèrent BFS.

Complexité et convergence de l'algorithme de Ford-Fulkerson

Supposons que $c : E \longrightarrow \mathbb{N}$

- DFS est en $\mathcal{O}(|V| + |E|)$.

Complexité et convergence de l'algorithme de Ford-Fulkerson

Supposons que $c : E \longrightarrow \mathbb{N}$

- DFS est en $\mathcal{O}(|V| + |E|)$.
- complexité de l'algorithme F-F ?

Complexité et convergence de l'algorithme de Ford-Fulkerson

Supposons que $c : E \longrightarrow \mathbb{N}$

- DFS est en $\mathcal{O}(|V| + |E|)$.
- complexité de l'algorithme F-F : $\mathcal{O}((|V| + |E|) \times \varphi_{max})$
  - dépend de la valeur de la solution $\varphi_{max} \in \mathbb{N}$
  - et si $\varphi_{max} >> |V|$, par exemple : $\varphi_{max} \approx 2^{|V|}$!!

Complexité et convergence de l'algorithme de Ford-Fulkerson

Supposons que $c : E \longrightarrow \mathbb{N}$

- DFS est en $\mathcal{O}(|V| + |E|)$.
- complexité de l'algorithme F-F : $\mathcal{O}((|V| + |E|) \times \varphi_{max})$
  - dépend de la valeur de la solution $\varphi_{max} \in \mathbb{N}$
  - et si $\varphi_{max} >> |V|$, par exemple : $\varphi_{max} \approx 2^{|V|}$!!

Supposons que $c : E \longrightarrow \mathbb{Q}$?

Complexité et convergence de l'algorithme de Ford-Fulkerson

Supposons que $c : E \longrightarrow \mathbb{N}$

- DFS est en $\mathcal{O}(|V| + |E|)$.
- complexité de l'algorithme F-F : $\mathcal{O}((|V| + |E|) \times \varphi_{max})$
  - dépend de la valeur de la solution $\varphi_{max} \in \mathbb{N}$
  - et si $\varphi_{max} >> |V|$, par exemple : $\varphi_{max} \approx 2^{|V|}$!!

Supposons que $c : E \longrightarrow \mathbb{Q}$

- complexité de l'algorithme F-F est $\mathcal{O}((|V| + |E|) \times \varphi_{max} \times d)$! où d est le dénominateur commun

Complexité et convergence de l'algorithme de Ford-Fulkerson

Supposons que $c : E \longrightarrow \mathbb{N}$

- DFS est en $\mathcal{O}(|V| + |E|)$.
- complexité de l'algorithme F-F : $\mathcal{O}((|V| + |E|) \times \varphi_{max})$
  - dépend de la valeur de la solution $\varphi_{max} \in \mathbb{N}$
  - et si $\varphi_{max} >> |V|$, par exemple : $\varphi_{max} \approx 2^{|V|}$!!

Supposons que $c : E \longrightarrow \mathbb{Q}$

- complexité de l'algorithme F-F est $\mathcal{O}((|V| + |E|) \times \varphi_{max} \times d)$! où d est le dénominateur commun

Supposons que $c : E \longrightarrow \mathbb{R}$?

Complexité et convergence de l'algorithme de Ford-Fulkerson

Supposons que $c : E \longrightarrow \mathbb{N}$

- DFS est en $\mathcal{O}(|V| + |E|)$.
- complexité de l'algorithme F-F : $\mathcal{O}((|V| + |E|) \times \varphi_{max})$
  - dépend de la valeur de la solution $\varphi_{max} \in \mathbb{N}$
  - et si $\varphi_{max} >> |V|$, par exemple : $\varphi_{max} \approx 2^{|V|}$!!

Supposons que $c : E \longrightarrow \mathbb{Q}$

- complexité de l'algorithme F-F est $\mathcal{O}((|V| + |E|) \times \varphi_{max} \times d)$! où d est le dénominateur commun

Supposons que $c : E \longrightarrow \mathbb{R}$

- il arrive que F-F ne termine jamais !
- des chaînes augmentantes avec des flots de plus en plus petits
- sans risque pour un ordinateur ! (voir exercice implémentation TD3)

Alternatives

Algorithmes dont la complexité ne dépend pas de $\varphi_{\max}$

- Edmonds-Karp (F-F basé sur un BFS), 1970 :
  - en $\mathcal{O}(|E|^2 \times |V|)$

Alternatives

Algorithmes dont la complexité ne dépend pas de $\varphi_{\max}$

- Edmonds-Karp (F-F basé sur un BFS), 1970 :
  - en $\mathcal{O}(|E|^2 \times |V|)$
  - converge avec des capacités parmi $\mathbb{N}$, $\mathbb{Q}$ ou $\mathbb{R}$
  - en moins de $|V| \times |E|$ chaînes augmentantes (itérations)

Alternatives

Algorithmes dont la complexité ne dépend pas de $\varphi_{\max}$

- Edmonds-Karp (F-F basé sur un BFS), 1970 :
  - en $\mathcal{O}(|E|^2 \times |V|)$
  - converge avec des capacités parmi $\mathbb{N}$, $\mathbb{Q}$ ou $\mathbb{R}$
  - en moins de $|V| \times |E|$ chaînes augmentantes (itérations)
- Dinic (Dinitz), 1970, en $\mathcal{O}(|E| \times |V|^2)$
- Orlin, 2013, en $\mathcal{O}(|E| \times |V|)$ et même en $\mathcal{O}(\frac{|V|^2}{\log(|V|)})$ quand $|E|$ est en $\mathcal{O}(|V|)$

Plan

1. Problème pratique
2. Modélisation du problème
3. Ford-Fulkerson
4. Graphe “à flot” comme modèle pour d’autres problèmes
   - Résidence étudiantes
   - Semaine du BDE
5. Conclusion

Gestions des résidences estudiantines

Contexte

Une université dispose de M résidences, le nombre d'étudiants que chaque résidence peut héberger est connu, m_i avec $i \in 1 \dots M$. Cette université accueille N étudiants. Un étudiant communique une liste des résidences dans lesquelles il souhaite être logé.

Gestions des résidences estudiantines

Contexte

Une université dispose de M résidences, le nombre d'étudiants que chaque résidence peut héberger est connu, m_i avec $i \in 1 \dots M$. Cette université accueille N étudiants. Un étudiant communique une liste des résidences dans lesquelles il souhaite être logé.

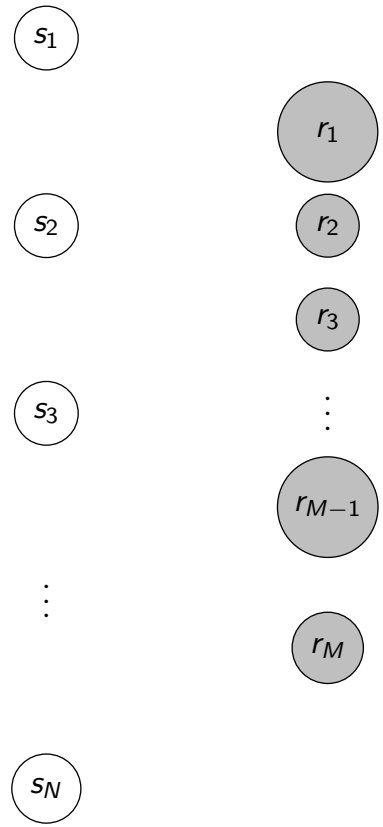
Objectif

Proposer une attribution des places dans les résidences en maximisant le nombre d'étudiants acceptés.

Construction du modèle : Graphe à flot

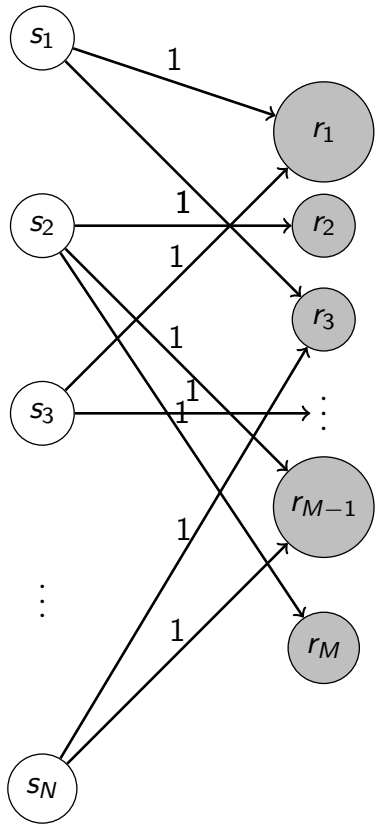
Construction du modèle : Graphe à flot

1. un graphe biparti $(V_N \cup V_M, E = V_N \times V_M)$, V_N les sommets étudiants et V_M les sommets résidences



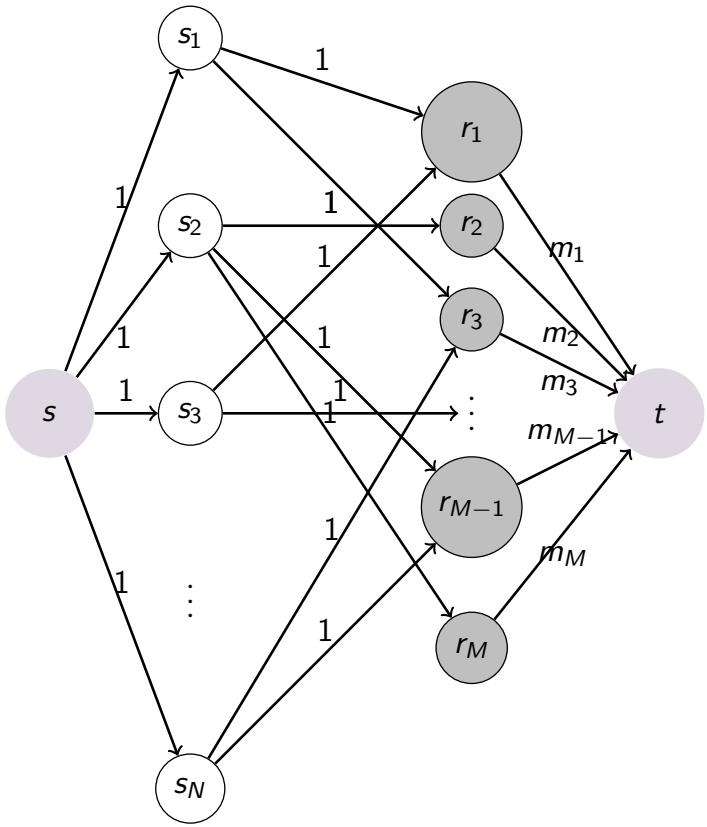
Construction du modèle : Graphe à flot

1. un graphe biparti $(V_N \cup V_M, E = V_N \times V_M)$, V_N les sommets étudiants et V_M les sommets résidences
2. un arc $(s_i, r_j) \in V_N \times V_M$ pour chaque souhait d'un étudiant s_i portant sur la résidence m_j avec $c((s_i, m_j)) = 1$



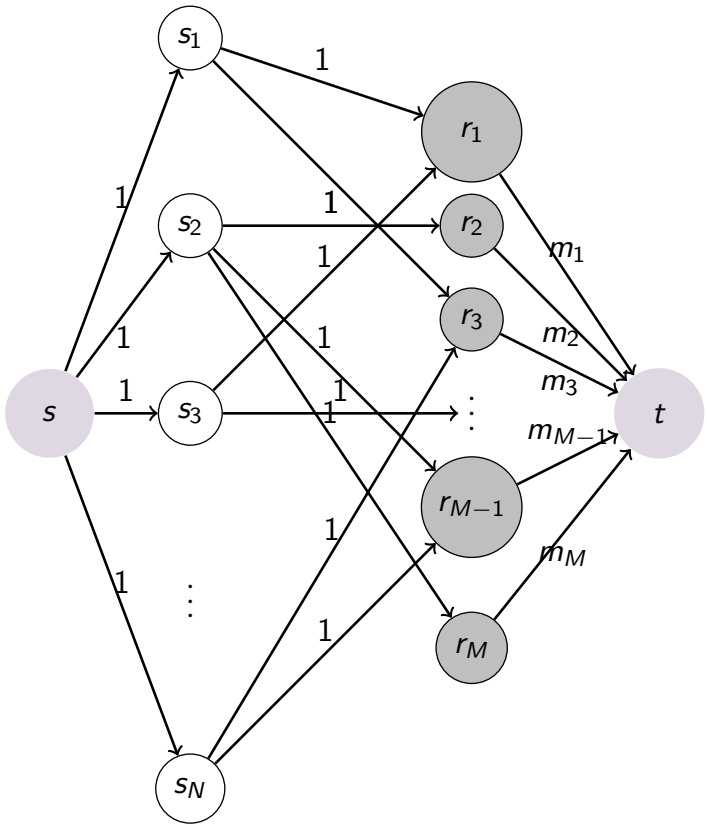
Construction du modèle : Graphe à flot

1. un graphe biparti $(V_N \cup V_M, E = V_N \times V_M)$, V_N les sommets étudiants et V_M les sommets résidences
2. un arc $(s_i, r_j) \in V_N \times V_M$ pour chaque souhait d'un étudiant s_i portant sur la résidence m_j avec $c((s_i, m_j)) = 1$
3. on ajoute ensuite deux sommets fictifs, s et t et les arcs suivants :
   - $e = (s, s_i)$ pour chaque $s_i \in V_N$ avec $c(e) = 1$
   - $e = (r_i, t)$ pour chaque $r_i \in V_M$ avec $c(e) = m_i$



Construction du modèle : Graphe à flot

1. un graphe biparti $(V_N \cup V_M, E = V_N \times V_M)$, V_N les sommets étudiants et V_M les sommets résidences
2. un arc $(s_i, r_j) \in V_N \times V_M$ pour chaque souhait d'un étudiant s_i portant sur la résidence m_j avec $c((s_i, m_j)) = 1$
3. on ajoute ensuite deux sommets fictifs, s et t et les arcs suivants :
   - $e = (s, s_i)$ pour chaque $s_i \in V_N$ avec $c(e) = 1$
   - $e = (r_i, t)$ pour chaque $r_i \in V_M$ avec $c(e) = m_i$



Conclusion

La recherche d'une attribution des logements revient à la recherche d'un flot de valeur maximale dans le graphe "à flot" construit.

Semaine du BDE

Contexte

Au cours de la journée, les étudiants de CentraleSupélec vont de la résidence étudiante (le matin) vers la cantine (le midi) en passant par différentes salles de cours et de TD.

Problème

L'équipe du BDE veut poster des volontaires sur le trajet des élèves de manière à ce qu'aucun ne puisse échapper à la distribution de tracts. Selon la taille des zones de passage, il peut être nécessaire de mettre plusieurs volontaires pour couvrir un large passage.

Semaine du BDE

Contexte

Au cours de la journée, les étudiants de CentraleSupélec vont de la résidence étudiante (le matin) vers la cantine (le midi) en passant par différentes salles de cours et de TD.

Problème

L'équipe du BDE veut poster des volontaires sur le trajet des élèves de manière à ce qu'aucun ne puisse échapper à la distribution de tracts. Selon la taille des zones de passage, il peut être nécessaire de mettre plusieurs volontaires pour couvrir un large passage.

Objectif

Proposer une affectation d'un minimum de volontaires pour ne manquer aucun étudiant.

Construction du modèle :

1. Un graphe dont les sommets représentent la résidence, les salles de cours et la cantine. Les arcs représentent les différents chemins permettant d'aller directement d'un point à l'autre du campus.

Construction du modèle :

1. Un graphe dont les sommets représentent la résidence, les salles de cours et la cantine. Les arcs représentent les différents chemins permettant d'aller directement d'un point à l'autre du campus.
2. On définit la capacité d'un arc comme le nombre de volontaires qu'il faut positionner pour intercepter tous les élèves qui passent par ce chemin.

Construction du modèle :

1. Un graphe dont les sommets représentent la résidence, les salles de cours et la cantine. Les arcs représentent les différents chemins permettant d'aller directement d'un point à l'autre du campus.
2. On définit la capacité d'un arc comme le nombre de volontaires qu'il faut positionner pour intercepter tous les élèves qui passent par ce chemin.
3. Deux sommets spéciaux : la résidence (s) et la cantine (t).

Problème de coupe minimum

Quelle est la coupe minimum dans ce graphe séparant s et t ?

Construction du modèle :

1. Un graphe dont les sommets représentent la résidence, les salles de cours et la cantine. Les arcs représentent les différents chemins permettant d'aller directement d'un point à l'autre du campus.
2. On définit la capacité d'un arc comme le nombre de volontaires qu'il faut positionner pour intercepter tous les élèves qui passent par ce chemin.
3. Deux sommets spéciaux : la résidence (s) et la cantine (t).

Problème de coupe minimum

Quelle est la coupe minimum dans ce graphe séparant s et t ?

➜ La capacité de cette coupe donne le nombre de volontaires requis.

Construction du modèle :

1. Un graphe dont les sommets représentent la résidence, les salles de cours et la cantine. Les arcs représentent les différents chemins permettant d'aller directement d'un point à l'autre du campus.
2. On définit la capacité d'un arc comme le nombre de volontaires qu'il faut positionner pour intercepter tous les élèves qui passent par ce chemin.
3. Deux sommets spéciaux : la résidence (s) et la cantine (t).

Problème de coupe minimum

Quelle est la coupe minimum dans ce graphe séparant s et t ?

➜ La capacité de cette coupe donne le nombre de volontaires requis.

Exercice : il s'agit d'un graphe non-orienté, comment le transformer en graphe de flot ?

Plan

1 Problème pratique

2 Modélisation du problème

3 Ford-Fulkerson

4 Graphe “à flot” comme modèle pour d’autres problèmes

5 Conclusion

Ce qu'il faut retenir

- Graphe orienté : $G = (V, E)$ avec capacités $c : E \to \mathbb{R}^+$
- Théorème principal :

flot max $\iff$ coupe min $\iff$ pas de chaînes augmentantes

- Algorithme de Ford-Fulkerson :
  - Recherche de chaînes augmentantes (parcours libre) ;
  - Complexité en $\mathcal{O}((|V| + |E|) \times \varphi_{max})$;
  - Quand F-F se termine, on obtient le flot maximal (théorème) ;

  ➜ Variante Edmonds-Karp à base d'un BFS en $\mathcal{O}(|E|^2 \times |V|)$
- Plusieurs applications pratiques :
  - réseaux de toute sorte...