# Algorithmique et Conteneurs

## Graphe : Algorithme de Dijkstra
## Graphe : Algorithme de Ford Fulkerson

Département Informatique

# Problèmes de plus courts chemins

Utilité

- Temps minimal de transmission dans un réseau (télécom,...)
- Cout minimal d'un trajet
- Meilleur routage dans un réseau

- La réponse algorithmique :

  Algorithme de Dijkstra

# Algorithme de Dijkstra

Principes de l'algorithme

1. Soit un graphe orienté et valué (*) G(S,A), on construit une arborescence T(r , S') à partir d'un sommet r (racine)
2. Cette arborescence couvre un ensemble de sommets {s ∈ S'} ⊂ S et donne le plus court chemin de r à chacun des sommets de S'.
3. On définit 2 fonctions sur S' :
   - п(s') est la longueur du plus court chemin de r à s' `a une itération donnée (on utilise seulement comme sommets intermédiaires, des sommets qui sont déjà dans S')
   - pere(s') désigne à une itération donnée, le prédécesseur de s' sur un tel chemin

(*) *On notera p(x,y) la valuation d'un arc (x,y) de A*

# Assertions de l'algorithme

2 assertions à vérifier :

1. Pour tout sommet $y \notin S'$ et qui est lié par un arc avec au moins un élément de $S'$ on a :

   $$\pi(y) = \min_{x \in S', (x,y) \in A} (\pi(x) + p(x, y))$$

2. pere(pivot) ≡ longueur d'un plus court chemin de r à pivot et pere(pivot) est le prédécesseur de pivot dans ce plus court chemin.

# Description de l'algorithme de Dijkstra

$pivot \leftarrow r$, $S' \leftarrow \emptyset$, $\pi(r) \leftarrow 0$
Pour tout sommet $x \neq r$ faire $\pi(x) \leftarrow \infty$
Pour j variant de 1 `a n − 1, faire // n le nombre de sommets
    Pour tout sommet $y \notin S'$ et successeur de pivot Faire
        Si $(\pi(pivot) + p(pivot, y)) < \pi(y)$ alors
            $\pi(y) \leftarrow \pi(pivot) + p(pivot, y)$
            $pere(y) \leftarrow pivot$
        Fin Si
    Fin pour
    chercher $y_0 \notin S' \,/\, \pi(y_0) = \min_{y \notin S'} \pi(y)$
    $pivot \leftarrow y_0$
    $S' \leftarrow S' \cup (pivot)$
Fin pour

# Flot maximum

## Utilité

- Résolution des problèmes de transports dans un réseau de télécommunications, de canalisations (débits), de transport de produits (capacités des camions, wagons, etc ...)
- Etude de la connectivité des graphes (Résistance aux pannes dans un réseau)

# Flot maximum : Définitions

Soit G(S,A) un graphe orienté

- La capacité C : A → R
- Le flot f : A → R

- ∀a ∈ A : c(a) représente le flux maximum pouvant circuler sur l'arête a
- ∀a ∈ A : f (a) représente le flux circulant sur l'arête a
- 2 sommets privilégiés de S :
  - la source $s_0$
  - le puits p
- Soit $S_0 \subset S$ / $s_0 \in S_0$ et $p \notin S_0$, on définit $(S_0, S_0^c)$ l'ensemble des arcs (x,y) de A où $x \in S_0$ et $y \notin S_0$
- On dit que l'ensemble $(S_0, S_0^c)$ est une coupe qui sépare la source du puits.

# Flot maximum : Définitions

- Un chemin reliant $s_0$ à p est saturé s'il existe au moins une arête *a* saturée : $c(a) = f(a)$
- Un flot f du réseau doit vérifier :
  - $\forall a \in A, 0 \leq f(a) \leq c(a)$
  - Conservation du flot (loi de Kirshoff) : $\forall s \in S$ :

$$\sum_{a \text{ arc d'orig. } s} f(a) = \sum_{a \text{ arc d'extr. } s} f(a)$$

- Capacité d'une coupe $(S_0, S_0^c)$

$$c(S_0, S_0^c) = \sum_{a \in (S_0, S_0^c)} c(a)$$

# Algorithme de Ford Fulkerson : Graphe résiduel

- Pour un flot f du graphe G, on définit le graphe valué $G_{résiduel}$ par :
  - $G_{résiduel}$ a les mêmes sommets que G,
  - Toutes les arêtes a non saturées du flot sont des arêtes de $G_{résiduel}$ avec la valuation c(a) – f(a),
  - Pour toutes les arêtes a = (x,y) ayant un flot strictement positif, on crée dans $G_{résiduel}$ l'arête a' = (y,x) avec la valuation f(a).

- Tout chemin de $G_{résiduel}$ reliant la source $s_0$ au puits p est appelé chemin augmentant.

- Ce graphe résiduel servira à améliorer le flot f, en faisant (si c'est possible) passer par ce graphe un flot additionnel au flot f.
- Cette amélioration utilisera un chemin augmentant.

# Algorithme de Ford Fulkerson : Chemin augmentant

- Dans ce qui suit, G est un graphe valué, f un flot de G, $G_{\text{résiduel}}$ le graphe résiduel de f et cp un chemin augmentant dans le graphe résiduel du flot f.
- Saturer le chemin cp consiste à valuer les arêtes de cp avec le minimum des capacités de ces arêtes dans $G_{\text{résiduel}}$.
- Soient f un flot sur G et cp un chemin valué dans $G_{\text{résiduel}}$ . On définit l'opération **f ⊕ cp** renvoie un flot g tel que
  - On sature cp et on note $\alpha$ le minimum des capacités
  - Pour toute arête a=(x,y) de cp :
    - Si a est une arête de G alors $g(a) = f(a) + \alpha$
    - Si a' = (y,x) est une arête de G alors $g(a) = f(a) - \alpha$

# Algorithme de Ford Fulkerson

- En entrée un graphe G valué positiviment, $s_0$ une source et p un puits
- En sortie f est le flot maximum entre $s_0$ et p.

Soit f un flot quelconque de G // On peut prendre le flot nul

$G_{résiduel} \leftarrow$ le graphe résiduel de f

Tant qu'il existe un chemin cp augmentant de $G_{résiduel}$

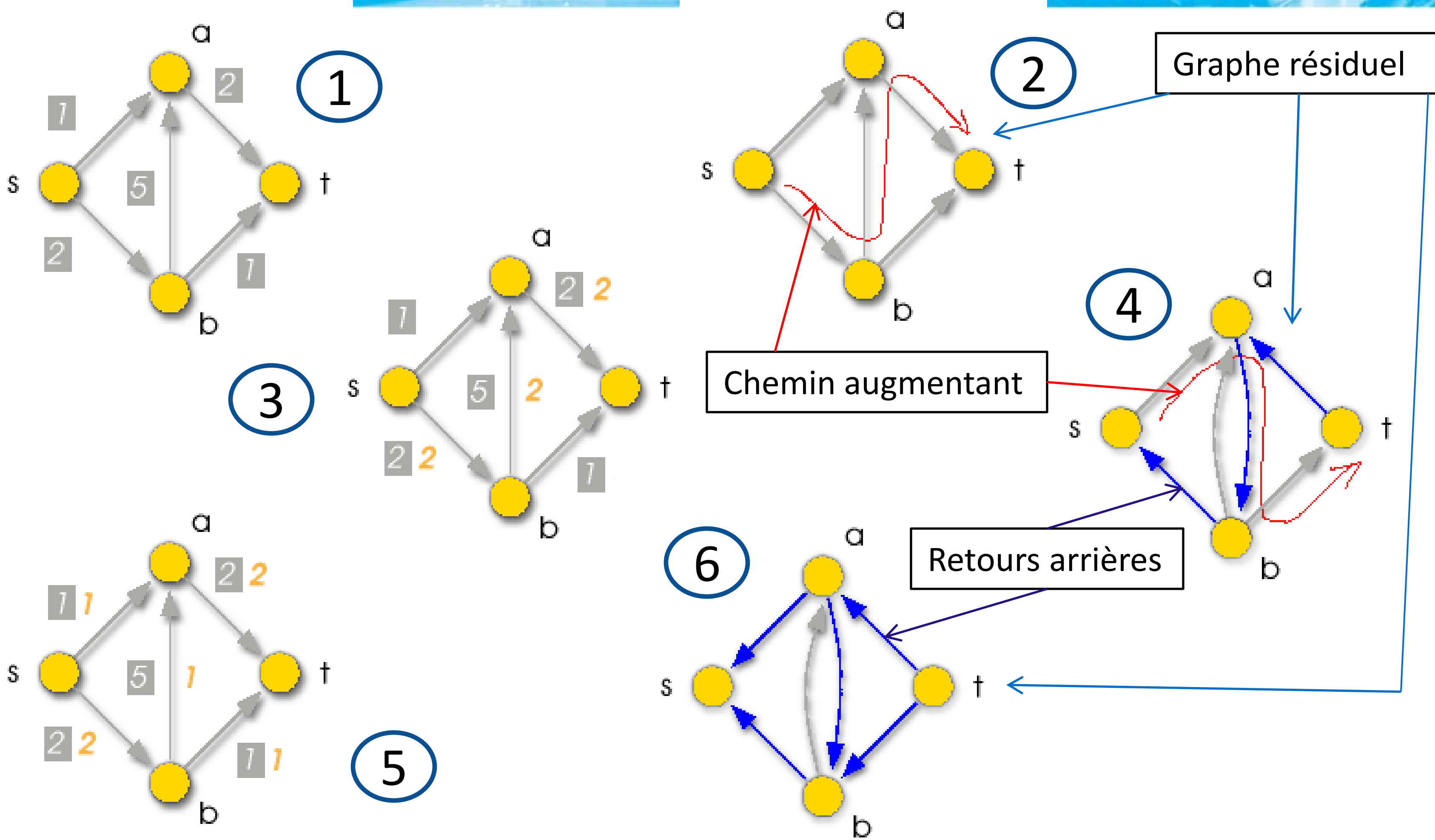
  saturer cp dans $G_{résiduel}$ // (*)

  $f \leftarrow f \oplus cp$ // (**)

  $G_{résiduel} \leftarrow$ le graphe résiduel de f

Fin Tantque

# Algorithme de Ford Fulkerson : Exemple




http://www.g-scop.inpg.fr/~rapinec/Graphe/Flot/fordFulkerson.html