

# Algorithmique des graphes - Cours 7

Olivier Baudon

Université de Bordeaux

11 octobre 2021

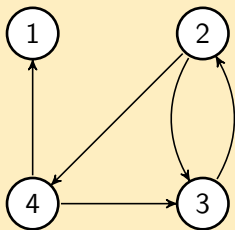
## Algorithme de Warshall - Principe

L'algorithme de Floyd peut être adapté pour calculer la matrice d'adjacence de la fermeture transitive d'un graphe  $G$  orienté, c'est à dire du graphe  $G'$  tel que  $ij \in E(G') \Leftrightarrow$  il existe un chemin de  $i$  à  $j$  dans  $G$ .

## Algorithme de Warshall - Énoncé

Voir document "Algorithmes des graphes" page 10.

## Algorithme de Warshall - Exemple



*Graphe G*

$$\begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

*Matrice W*

## Algorithme de Warshall - Exemple

Valeurs de  $W$  initiale, puis après  $k = 1, 2$  et  $3$ .

$$\begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \quad \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

$$\begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} \quad \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

## Algorithme de Warshall - Exemple

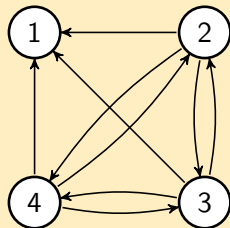
Valeurs de  $W$  pour  $k = 3$  et  $k = 4$

$$\begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

## Algorithme de Warshall - Exemple

Valeur de  $W$  pour  $k = 4$  et graphe associé

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$



## Réseau

Un **réseau** est un quadruplet  $(G, s, t, c)$  où :

- ▶  $G$  est un graphe orienté ;
- ▶  $s$  et  $t$  sont deux sommets de  $G$ , appelés respectivement **source** et **puits** ;
- ▶  $c$  est une fonction de  $E(G) \rightarrow \mathbb{R}^+$ , qui donne la **capacité** de chaque arc.

## Flot

Un flot  $f$  dans un réseau  $(G, s, t, c)$  est une fonction de  $E(G) \rightarrow \mathbb{R}^+$  telle que :

$$\forall e \in E(G), c(e) \geq f(e) \geq 0$$

$$\forall v \in V(G) \setminus \{s, t\} \quad \sum_{e=uv \in E(G)} f(e) = \sum_{e=vw \in E(G)} f(e)$$

Remarque : la deuxième contrainte est appelée **loi de Kirchhoff**, en référence à la conservation d'énergie dans les circuits électriques, et s'exprime généralement par "tout ce qui rentre est égal à tout ce qui sort".

La valeur du flot, notée  $val(f)$  sera égale à  $\sum_{e=sv \in E(G)} f(e)$ .

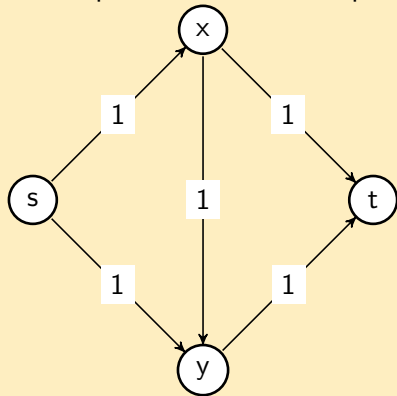
Remarque : on a également  $val(f) = \sum_{e=vt \in E(G)} f(e)$ .

L'objectif est de trouver pour un réseau  $(G, s, t, c)$  son flot de valeur maximum.



## Exemple de réseau

Les étiquettes sur les arcs représentent la capacité de chaque arc.

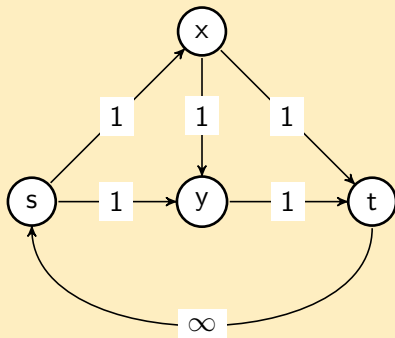


## Arc retour

Si  $f$  est un flot, alors

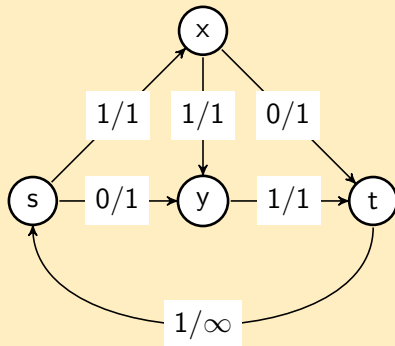
$$\sum_{e=sv \in E(G)} f(e) = \sum_{e=vt \in E(G)} f(e) = \text{val}(f).$$

Il est habituel de rajouter un arc, dit "de retour", de  $t$  vers  $s$ , de capacité infinie, et dont la valeur du flot est  $\text{val}(f)$ . Ainsi, la loi de Kirchhoff sera respectée par tous les sommets.



## Exemple de flot réalisable

Sur chaque arc  $e$ , on note deux valeurs " $f(e)/c(e)$ " représentant respectivement la valeur du flot et la capacité de l'arc  $e$ .  
Le flot ci-dessous est de valeur 1.



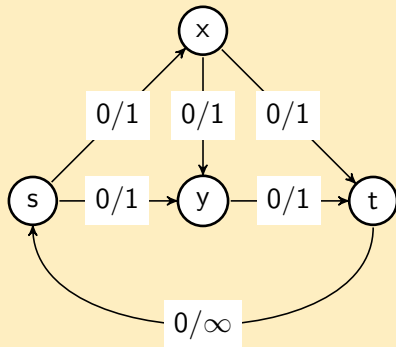
## Formulation sous forme d'un programme linéaire

Le problème de flots peut s'exprimer à l'aide d'un programme linéaire :  $\max\{f(ts) \mid 0 \leq f \leq c, Bf = 0\}$  où  $B$  est la matrice d'incidence de  $G$  augmenté de l'arc retour,  $f$  et  $c$  des vecteurs indicés par  $E(G) \cup \{ts\}$  contenant respectivement la valeur du flot sur chaque arc et les capacités des arcs.

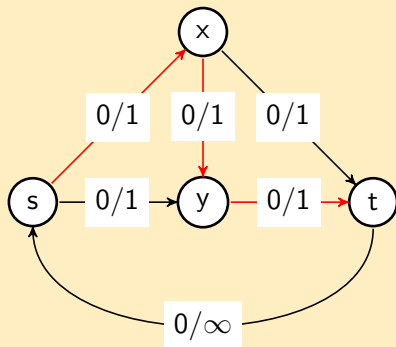
Rappel :  $B_{ij} = -1$  si l'arc  $j$  est un arc sortant de  $i$ ,  $1$  si  $j$  est un arc entrant de  $i$ ,  $0$  sinon.

On peut donc résoudre le problème de flots à l'aide d'un algorithme de résolution d'un programme linéaire, tel que l'algorithme du simplexe.

## Recherche d'une chaîne augmentante par marquage direct

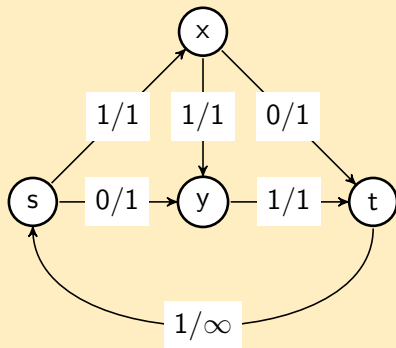


## Recherche d'une chaîne augmentante par marquage direct

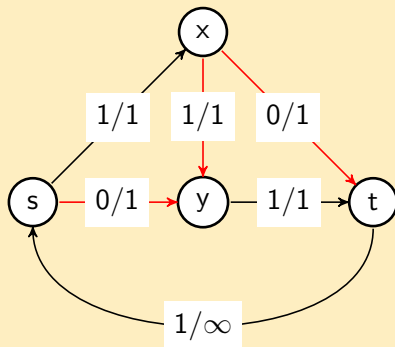


Chaîne augmentante :  $s, x, y, t$ . Augmentation :  $+1$ .

## Recherche d'une chaîne augmentante par marquage indirect



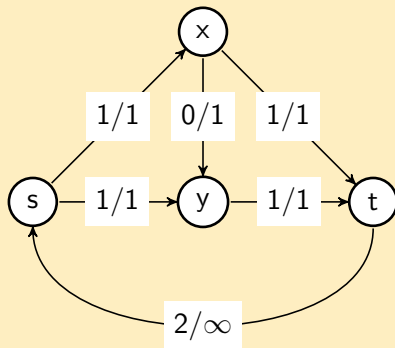
## Recherche d'une chaîne augmentante par marquage indirect



Chaîne augmentante :  $s, y, x, t$ . Augmentation :  $+1$ .



## Recherche d'une chaîne augmentante par marquage indirect

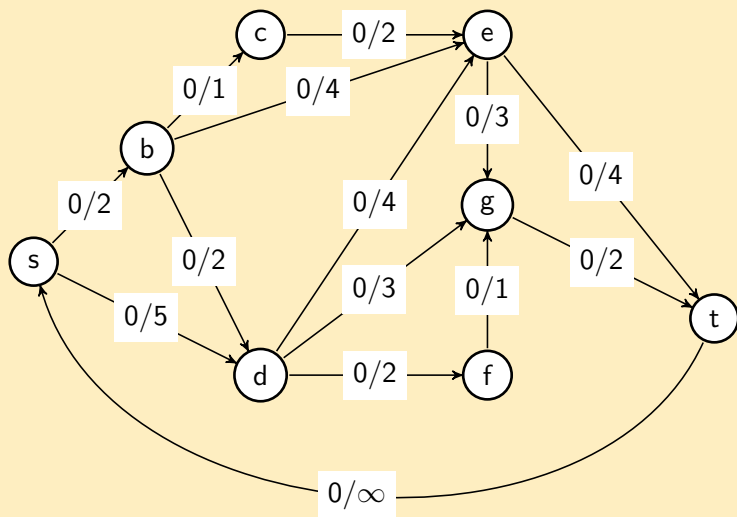


## Algorithme de Ford-Fulkerson

Voir document "Algorithmes des graphes" pages 11 et 12.

# Flots - résolution

## Exemple (Ford-Fulkerson)

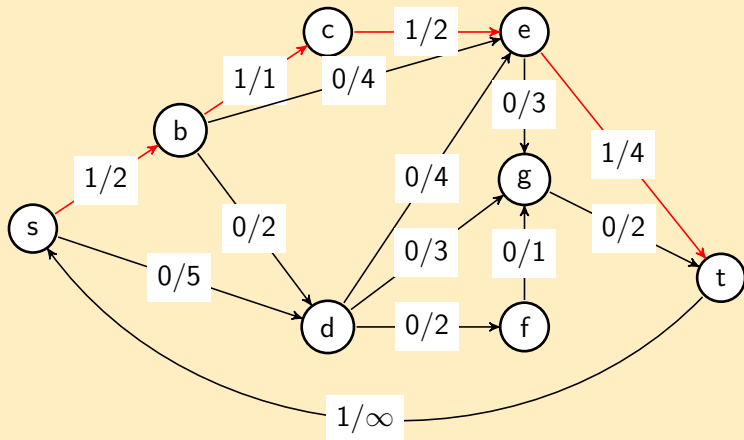


# Flots - résolution

## Exemple (Ford-Fulkerson)

Chaîne augmentante :

s	b	c	e	t
	+2	+1	+1	+1

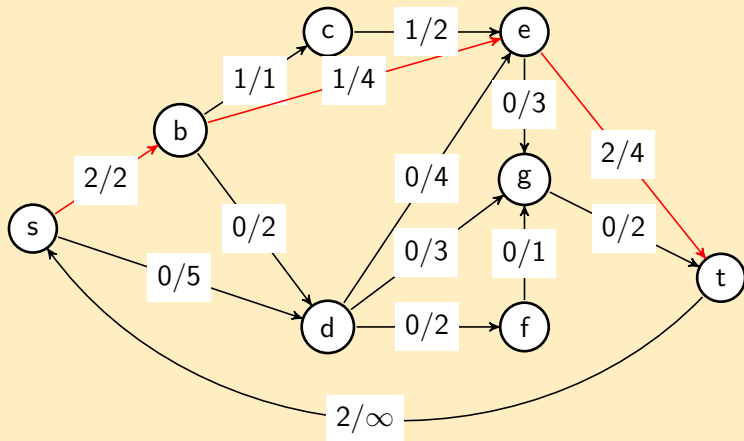


# Flots - résolution

## Exemple (Ford-Fulkerson)

Chaîne augmentante :

s	b	e	t
	+1	+1	+1

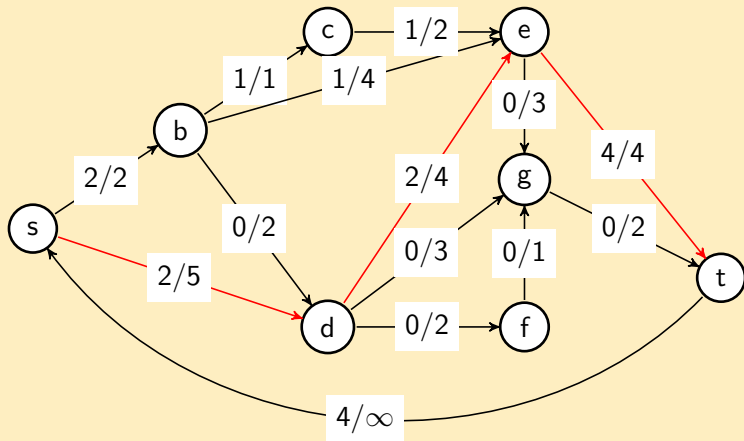


# Flots - résolution

## Exemple (Ford-Fulkerson)

Chaîne augmentante :

<i>s</i>	<i>d</i>	<i>e</i>	<i>t</i>
	+5	+4	+2

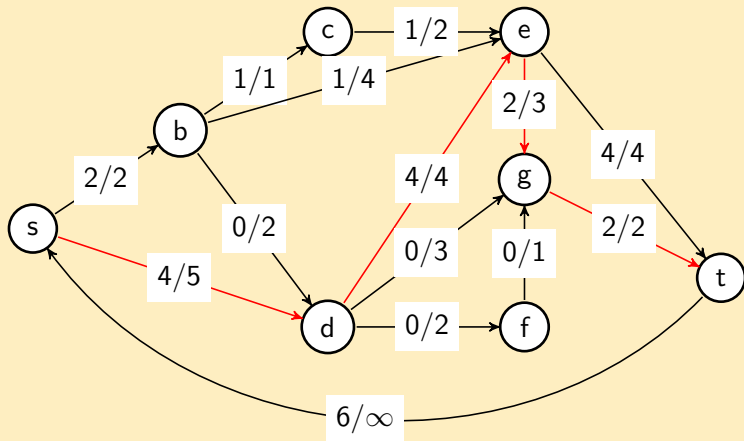


# Flots - résolution

## Exemple (Ford-Fulkerson)

Chaîne augmentante :

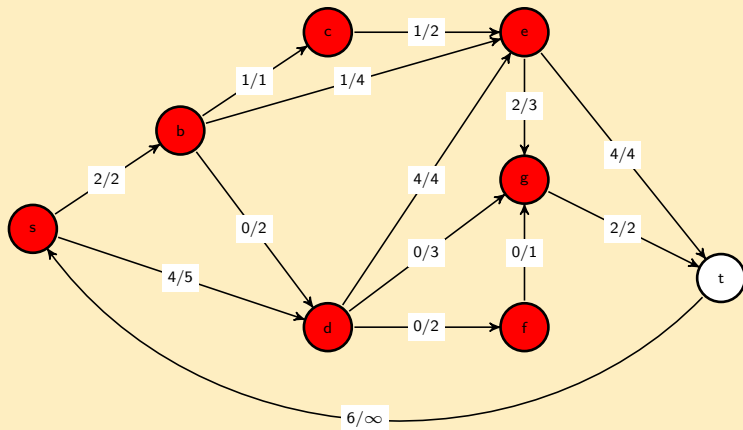
<i>s</i>	<i>d</i>	<i>e</i>	<i>g</i>	<i>t</i>
	+3	+2	+2	+2



## Exemple (Ford-Fulkerson)

Chaîne augmentante :

<i>s</i>	<i>d</i>	<i>g</i>	<i>e</i>	<i>c</i>	<i>b</i>
	+1	+1	+1	+1	+1
<i>s</i>	<i>d</i>	<i>f</i>			
	+1	+1			





## Notion de coupe

Une coupe est un sous-ensemble  $C$  de  $V(G)$  contenant  $s$  et ne contenant pas  $t$ .

La valeur d'une coupe  $C$ ,  $val(C)$ , est définie par :

$$val(C) = \sum_{e=uv, u \in C, v \notin C} c(e)$$

.

## Théorème

Soit  $f$  un flot réalisable sur un réseau  $R = (G, s, t, c)$  et  $C$  une coupe de  $R$ . Alors  $val(f) \leq val(C)$

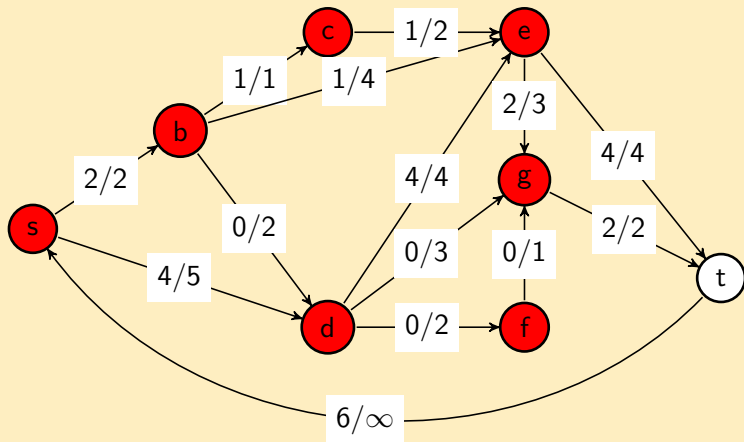
## Théorème "Flot max = coupe min"

Soit  $f$  un flot maximum sur un réseau  $R = (G, s, t, c)$  et  $C$  une coupe minimum de  $R$ . Alors  $val(f) = val(C)$

# Flots - résolution

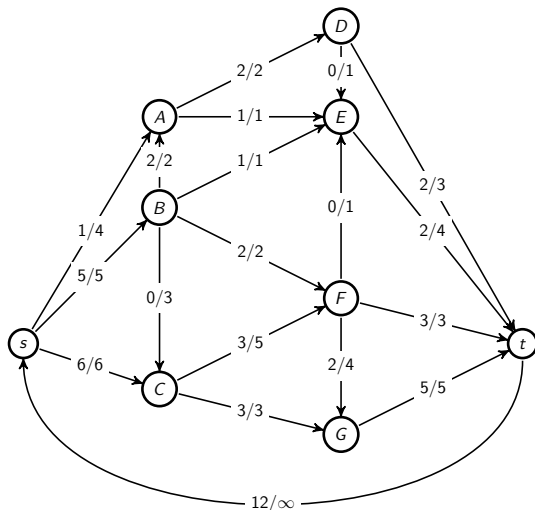
## Exemple (Ford-Fulkerson)

$Y = \{s, b, c, d, e, f, g\}$ .  $val(Y) = c(et) + c(gt) = 6 = val(f)$



# Flots - exercice (examen 2016)

Résoudre le problème du flot maximum sur le graphe ci-dessous et donner la coupe minimum correspondante.

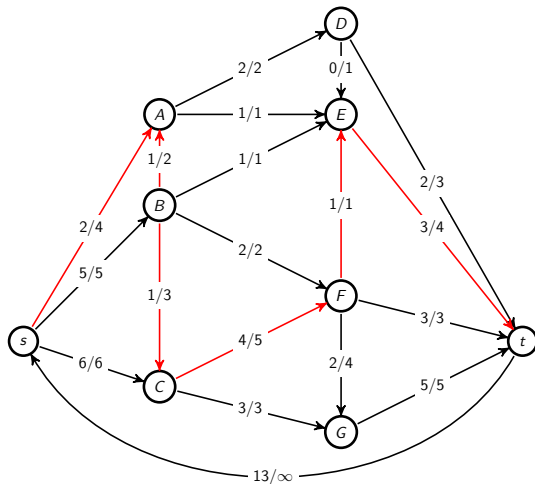


# Flots - exercice (examen 2016)

Chaîne augmentante (arcs en rouge avec la nouvelle valeur de  $f$ ) :

$v$	$s$	$A$	$B$	$C$	$F$	$E$	$t$
$\delta$	$\infty$	+3	+2	+2	+2	+1	+1

$C^+ = \{ts, sA, BC, CF, FE, Et\}$   $C^- = \{AB\}$



# Flots - exercice (examen 2016)

Recherche d'un chaîne augmentante (sommets de  $Y$  en rouge) :

$v$	$s$	$A$	$B$	$C$	$F$	$G$	
$\delta$	$\infty$	+2	+1	+1	+1	+1	

On n'a pas réussi à atteindre  $t$ , donc  $Y = \{s, A, B, C, F, G\}$  doit être une coupe minimum.

$$\text{val}(Y) = c(AD) + c(AE) + c(BE) + c(FE) + c(Ft) + c(Gt) = 2 + 1 + 1 + 1 + 3 + 5 = 13 = \text{val}(f)$$

