

Algorithme de Ford

F.M.

2024/2025

Rappel : l'algorithme de base

Soit $G = (S, A, p)$ un graphe pondéré sans circuit absorbant et soit r un sommet de G . L'algorithme de base détermine une r -arborescence de chemins de poids minimum.

```
pour tout  $s \in S$  faire  
  |  $\Pi[s] \leftarrow \infty; \mathcal{A}[s] \leftarrow s$   
fin pour  
 $\Pi[r] \leftarrow 0; \mathcal{A}[r] \leftarrow \emptyset$   
tant que  $\exists (s, x) \in A$  tel que  $\Pi[s] + p(s, x) < \Pi[x]$  faire  
  |  $\Pi[x] \leftarrow \Pi[s] + p(s, x); \mathcal{A}[x] \leftarrow s$   
fin tant que
```

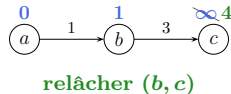
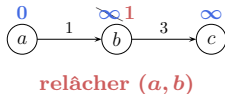
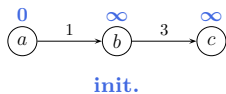
L'algorithme se termine lorsqu'il n'existe plus d'arc **améliorant** et dans ce cas : $\forall x \in S, \Pi[x] = \text{DIST}(r, x)$. Notons que le résultat final ne dépend pas du choix effectué à chaque itération.

La question est : comment reformuler la boucle **tant que il existe** ?

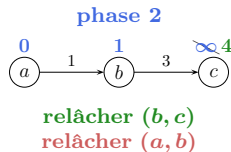
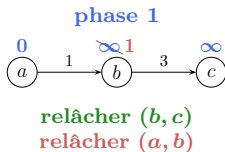
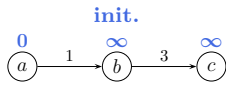
Phase de relâchement

On appelle **phase de relâchement** le fait de relâcher, l'un après l'autre, tous les arcs du graphe.

Soit $\mathcal{C} = (u_1, \dots, u_k)$ un chemin de k arcs. Le poids de \mathcal{C} peut être calculé en une seule phase si les arcs de \mathcal{C} sont relâchés de u_1 à u_k .



Mais si les arcs sont relâchés de u_k à u_1 , il faudra alors effectuer k phases pour calculer le poids de \mathcal{C} .



Algorithme de Ford : le principe

Considérons **un graphe G d'ordre n et ne contenant aucun circuit absorbant**. Rappelons que dans ce cas :

- ☞ Tout plus court chemin est élémentaire.
- ☞ Tout chemin élémentaire est de longueur **au plus $n - 1$ arcs**.

Si le plus long chemin élémentaire de G est de longueur $k \leq n - 1$, alors k phases au plus suffiront pour calculer son poids.

Ces k phases vont aussi permettre de calculer les poids de tous les chemins élémentaires de G tout en ne conservant que les plus courts d'entre eux (grâce à la procédure de relâchement).

C'est le principe de l'algorithme de Ford.

Algorithme de Ford : le principe

Soit G un graphe d'ordre n .

L'algorithme de Ford consiste en la **répétition d'au plus n phases de relâchement**. Le résultat est soit une r -arborescence de cpm soit un message signalant la présence de circuits absorbants.

Les trois situations permettant de conclure sont les suivantes :

- ☞ **Une phase n'améliore aucun potentiel** : le graphe ne contient plus d'arcs améliorants et une r -arborescence de cpm a été trouvée.
- ☞ **Le potentiel de r devient négatif lors d'une phase** : le sommet r appartient à un circuit absorbant.
- ☞ **La n^{e} phase améliore des potentiels** : le graphe contient toujours des arcs améliorants et donc des circuits absorbants.

Remarque : la n^{e} phase ne se produit que si (1) il existe un cpm qui est hamiltonien ou (2) le graphe contient des circuits absorbants.

Algorithme de Ford détaillé

pour tout $s \in S$ **faire**

$\Pi[s] \leftarrow \infty$; $\mathcal{A}[s] \leftarrow s$

fin pour

$\Pi[r] \leftarrow 0$; $\mathcal{A}[r] \leftarrow \emptyset$

$k \leftarrow 0$  k est le compteur de phases

répéter

$ameliorant \leftarrow \text{faux}$  mis à vrai s'il existe un arc améliorant

$k \leftarrow k + 1$

 k^e phase de relâchement

pour tout $(s, x) \in A$ **faire**  relâcher tous les arcs du graphe

si $\Pi[s] + p(s, x) < \Pi[x]$ **alors**

$\Pi[x] \leftarrow \Pi[s] + p(s, x)$; $\mathcal{A}[x] \leftarrow s$

$ameliorant \leftarrow \text{vrai}$

fin si

fin pour

jusqu'à non $ameliorant$ **ou** $k = n$ **ou** $\Pi[r] < 0$

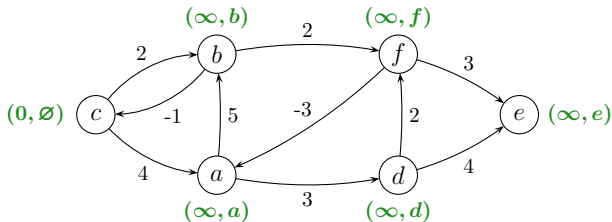
si $ameliorant$ **alors** écrire("présence de circuits absorbants")

sinon écrire(Π, \mathcal{A})

Trace de l'algorithme de Ford

initialisation

racine = c



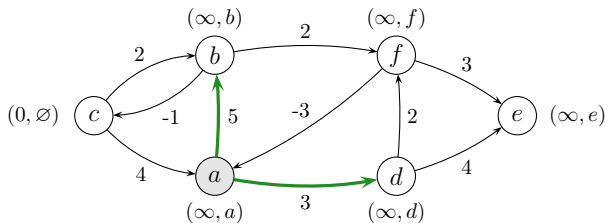
sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f

- Déroulement d'une phase : on modifie les annotations des sommets sur le diagramme et on ajoute une ligne au tableau des résultats.
- On convient de relâcher les arcs dans l'ordre suivant : on relâche tous les arcs d'origine a , puis tous ceux d'origine b , etc.

Trace de l'algorithme de Ford

phase 1

relâchement des
arcs d'origine a

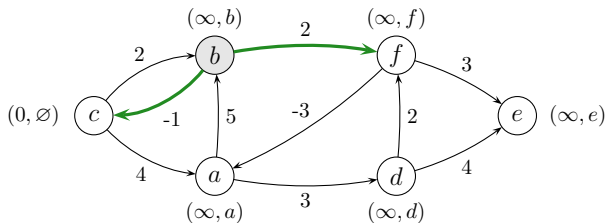


sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1						

Trace de l'algorithme de Ford

phase 1

relâchement des
arcs d'origine b

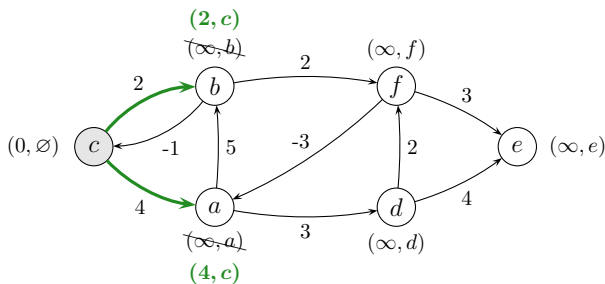


sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1						

Trace de l'algorithme de Ford

phase 1

relâchement des
arcs d'origine c

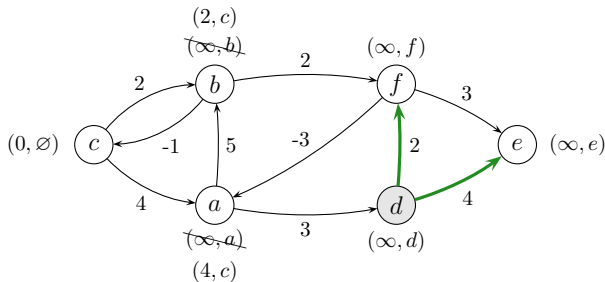


sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$				

Trace de l'algorithme de Ford

phase 1

relâchement des
arcs d'origine d

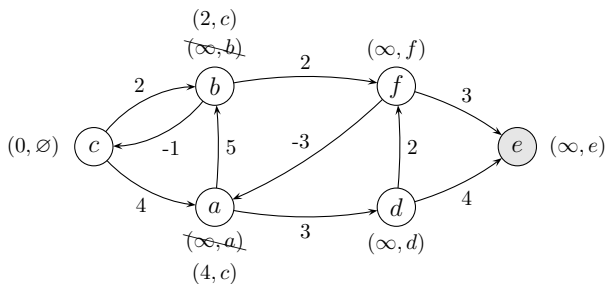


sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$				

Trace de l'algorithme de Ford

phase 1

relâchement des
arcs d'origine e

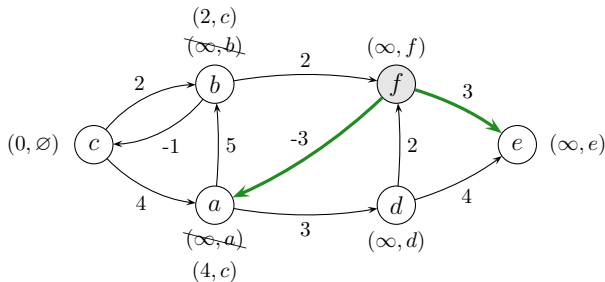


sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$				

Trace de l'algorithme de Ford

phase 1

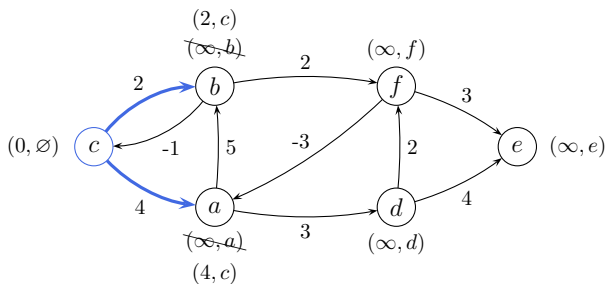
relâchement des
arcs d'origine f



sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$				

Trace de l'algorithme de Ford

fin phase 1

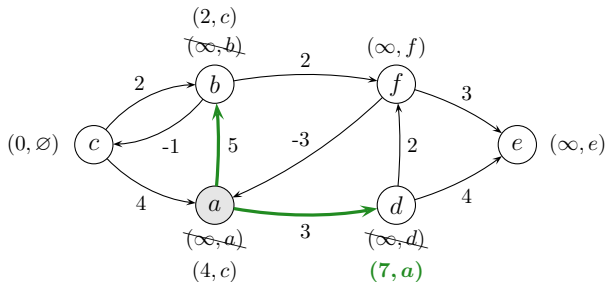


sommet	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$	$0, \emptyset$	∞, d	∞, e	∞, f

Trace de l'algorithme de Ford

phase 2

relâchement des
arcs d'origine a

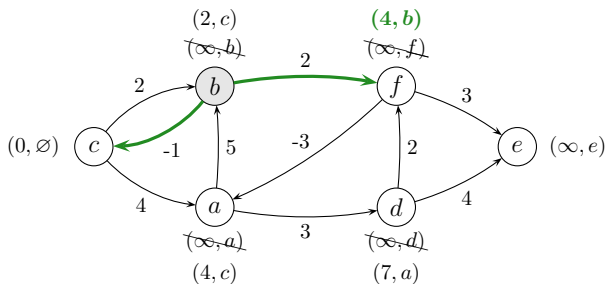


sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$	$0, \emptyset$	∞, d	∞, e	∞, f
phase 2				$7, a$		

Trace de l'algorithme de Ford

phase 2

relâchement des
arcs d'origine b

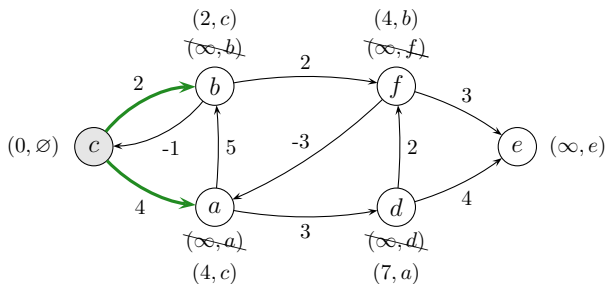


sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$	$0, \emptyset$	∞, d	∞, e	∞, f
phase 2				$7, a$		$4, b$

Trace de l'algorithme de Ford

phase 2

relâchement des
arcs d'origine c

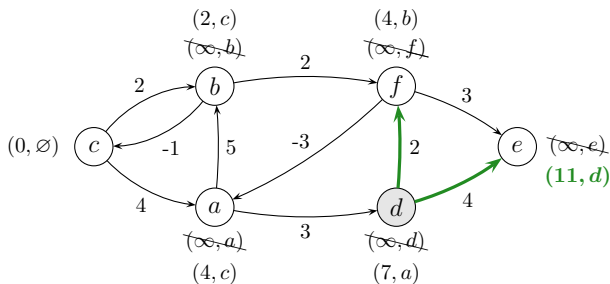


sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$	$0, \emptyset$	∞, d	∞, e	∞, f
phase 2				$7, a$		$4, b$

Trace de l'algorithme de Ford

phase 2

relâchement des
arcs d'origine d

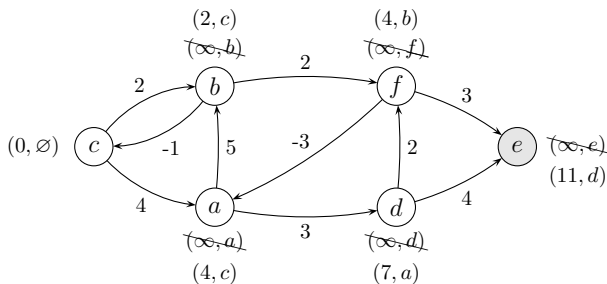


sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$	$0, \emptyset$	∞, d	∞, e	∞, f
phase 2				$7, a$	$11, d$	$4, b$

Trace de l'algorithme de Ford

phase 2

relâchement des
arcs d'origine e

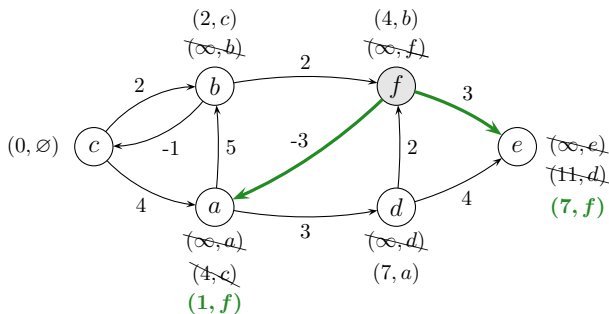


sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$	$0, \emptyset$	∞, d	∞, e	∞, f
phase 2				$7, a$	$11, d$	$4, b$

Trace de l'algorithme de Ford

phase 2

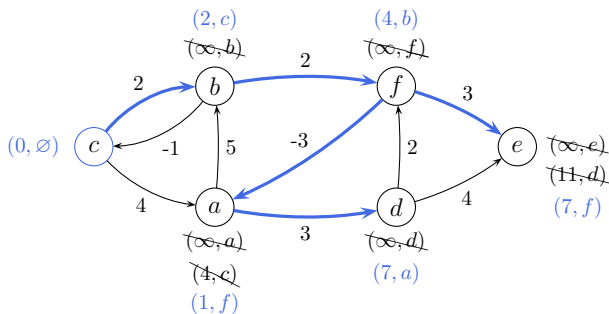
relâchement des
arcs d'origine f



sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$	$0, \emptyset$	∞, d	∞, e	∞, f
phase 2	$1, f$			$7, a$	$11, d$ $7, f$	$4, b$

Trace de l'algorithme de Ford

fin phase 2

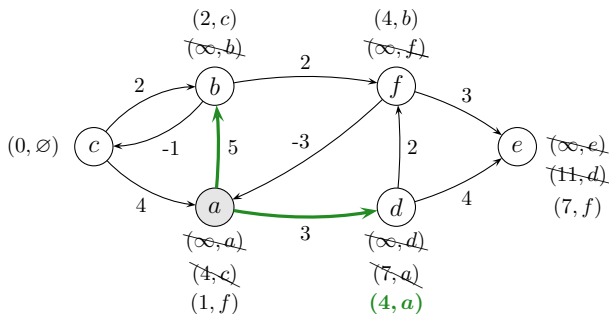


sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$	$0, \emptyset$	∞, d	∞, e	∞, f
phase 2	$1, f$	$2, c$	$0, \emptyset$	$7, a$	$7, f$	$4, b$

Trace de l'algorithme de Ford

phase 3

relâchement des
arcs d'origine a

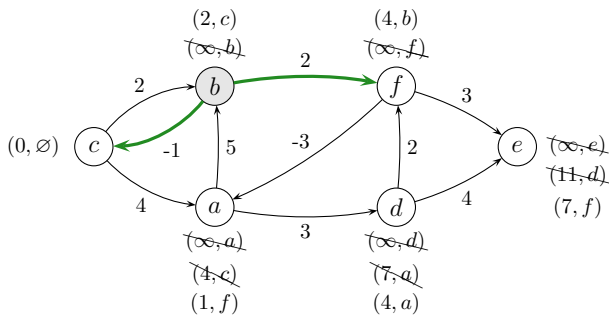


sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$	$0, \emptyset$	∞, d	∞, e	∞, f
phase 2	$1, f$	$2, c$	$0, \emptyset$	$7, a$	$7, f$	$4, b$
phase 3				$4, a$		

Trace de l'algorithme de Ford

phase 3

relâchement des
arcs d'origine b

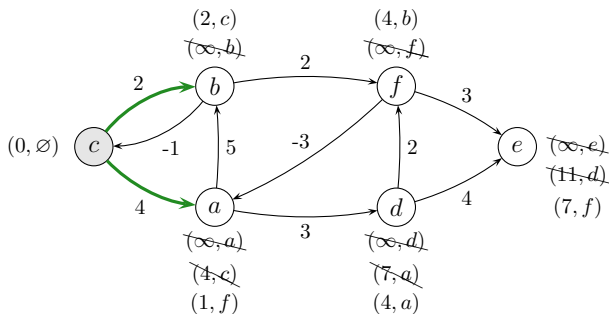


sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$	$0, \emptyset$	∞, d	∞, e	∞, f
phase 2	$1, f$	$2, c$	$0, \emptyset$	$7, a$	$7, f$	$4, b$
phase 3				$4, a$		

Trace de l'algorithme de Ford

phase 3

relâchement des
arcs d'origine c

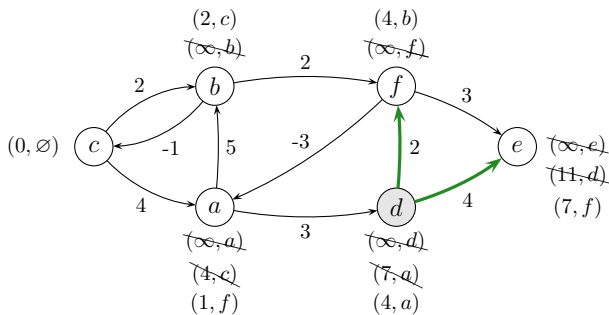


sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$	$0, \emptyset$	∞, d	∞, e	∞, f
phase 2	$1, f$	$2, c$	$0, \emptyset$	$7, a$	$7, f$	$4, b$
phase 3				$4, a$		

Trace de l'algorithme de Ford

phase 3

relâchement des
arcs d'origine d

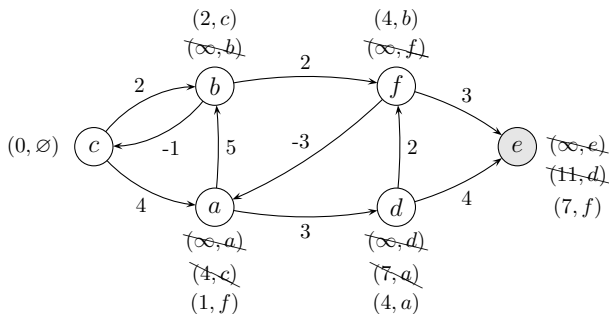


sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$	$0, \emptyset$	∞, d	∞, e	∞, f
phase 2	$1, f$	$2, c$	$0, \emptyset$	$7, a$	$7, f$	$4, b$
phase 3				$4, a$		

Trace de l'algorithme de Ford

phase 3

relâchement des
arcs d'origine e

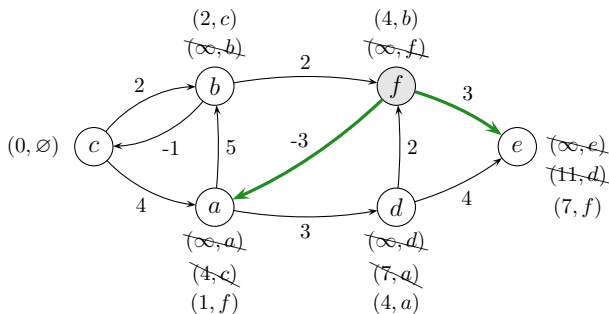


sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$	$0, \emptyset$	∞, d	∞, e	∞, f
phase 2	$1, f$	$2, c$	$0, \emptyset$	$7, a$	$7, f$	$4, b$
phase 3				$4, a$		

Trace de l'algorithme de Ford

phase 3

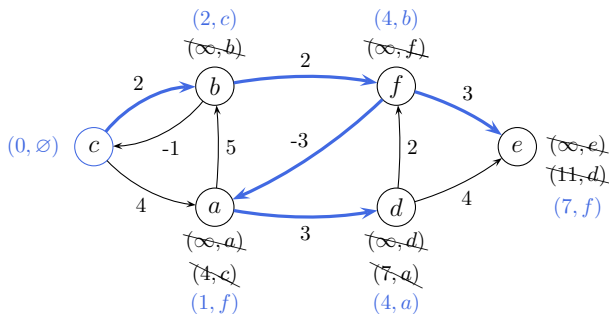
relâchement des
arcs d'origine f



sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$	$0, \emptyset$	∞, d	∞, e	∞, f
phase 2	$1, f$	$2, c$	$0, \emptyset$	$7, a$	$7, f$	$4, b$
phase 3				$4, a$		

Trace de l'algorithme de Ford

fin phase 3



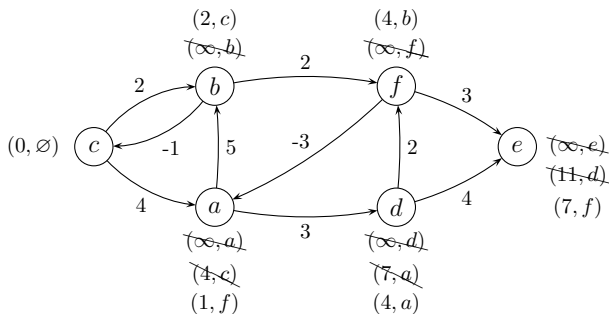
sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$	$0, \emptyset$	∞, d	∞, e	∞, f
phase 2	$1, f$	$2, c$	$0, \emptyset$	$7, a$	$7, f$	$4, b$
phase 3	$1, f$	$2, c$	$0, \emptyset$	$4, a$	$7, f$	$4, b$

Trace de l'algorithme de Ford

phase 4

aucune
amélioration

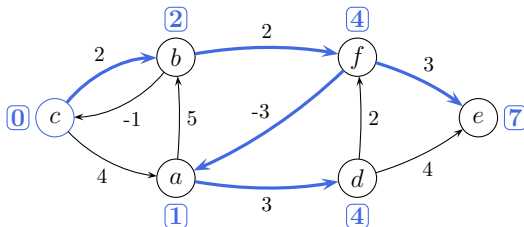
⇒ fin de l'algo



sommet	a	b	c	d	e	f
init.	∞, a	∞, b	$0, \emptyset$	∞, d	∞, e	∞, f
phase 1	$4, c$	$2, c$	$0, \emptyset$	∞, d	∞, e	∞, f
phase 2	$1, f$	$2, c$	$0, \emptyset$	$7, a$	$7, f$	$4, b$
phase 3	$1, f$	$2, c$	$0, \emptyset$	$4, a$	$7, f$	$4, b$
phase 4	$1, f$	$2, c$	$0, \emptyset$	$4, a$	$7, f$	$4, b$

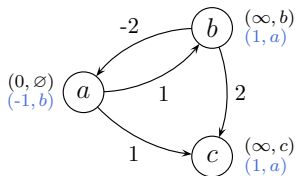
Trace de l'algorithme de Ford

résultat final



s	a	b	c	d	e	f
$\Pi[s]$	1	2	0	4	7	4
$\mathcal{A}[s]$	f	c	\emptyset	a	f	b

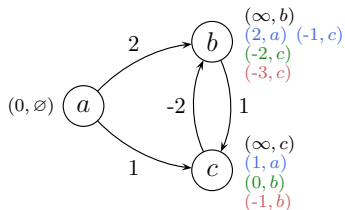
Algorithme de Ford et circuits absorbants



sommet	<i>a</i>	<i>b</i>	<i>c</i>
init.	0, ∅	∞, <i>b</i>	∞, <i>c</i>
phase 1	-1, <i>b</i>	1, <i>a</i>	1, <i>a</i>

$$\Pi[a] < 0$$

⇒ présence de circuits absorbants



sommet	<i>a</i>	<i>b</i>	<i>c</i>
init.	0, ∅	∞, <i>b</i>	∞, <i>c</i>
phase 1	0, ∅	-1, <i>c</i>	1, <i>a</i>
phase 2	0, ∅	-2, <i>c</i>	0, <i>b</i>
phase 3	0, ∅	-3, <i>c</i>	-1, <i>b</i>

des améliorations en phase 3

⇒ présence de circuits absorbants

Accélération de l'algorithme de Ford

L'algorithme de Ford relâche chaque arc au minimum 1 fois et au maximum n fois. Parmi les arcs relâchés au cours d'une phase, certains ne sont pas améliorants et ils sont inutilement relâchés.

Si le potentiel d'un sommet s n'a pas été amélioré au cours d'une phase, alors les arcs d'origine s ne seront pas améliorants lors de la phase suivante.

Au lieu de répéter un maximum de n phases de relâchement, l'idée est d'utiliser une **file (FIFO)** pour mémoriser les sommets dont le potentiel a diminué. À chaque itération, on extrait le sommet en tête de file, on relâche les arcs sortant de ce sommet et on ajoute en queue de file chaque successeur de ce sommet dont le potentiel a diminué.

Si la file devient vide, c'est que l'algorithme a trouvé une solution. Si un sommet passe plus de n fois dans la file, c'est qu'il appartient à un circuit absorbant.

Cas des graphes sans circuit absorbant

Savoir qu'un graphe ne contient pas de circuits absorbants simplifie considérablement la recherche de cpm. Partant d'un sommet, on avance dans le graphe en identifiant et en mémorisant des cpm progressivement (il est impossible de savoir si un chemin est un cpm en présence de circuits absorbants).

Si un graphe est sans circuit absorbant, nous verrons qu'il est possible de déterminer des cpm en effectuant une unique phase de relâchement (à condition de relâcher les arcs dans un certain ordre).

C'est le cas pour les **graphes à pondération positive**. L'algorithme qui tient compte de ce fait est **l'algorithme de Dijkstra**.

C'est le cas pour les **graphes sans circuits**. L'algorithme qui tient compte de ce fait est **l'algorithme de Bellman**.

Ces deux algorithmes sont présentés dans les chapitres suivants.