

Recherche de chemin de longueur optimal

Bellman-Ford

Algorithme de Bellman-Ford (2)

- **Algorithme de Bellman-Ford**
- L'algorithme de Bellman-Ford résout le problème des plus courts chemins avec origine unique où les poids des arcs peuvent avoir des valeurs négatives.
- Étant donné un graphe orienté $G = (V, E)$, de fonction de poids w , et une origine s , l'algorithme retourne une valeur booléenne indiquant s'il existe un circuit de poids négatif accessible depuis s . S'il n'existe pas, l'algorithme donne les plus courts chemins ainsi que leurs poids.

Algorithme de Bellman-Ford (3)

- Les notations sont les suivantes : $\pi [v]$ contient le prédécesseur de v sur le chemin (NIL s'il n'y en a pas), $\delta (u, v)$ est le poids du plus court chemin de u vers v (∞ s'il n'existe pas), $d [v]$ est une variable qui est une borne supérieure du poids du plus court chemin de s vers v . On définit le poids d'un chemin

$$p = \langle \hat{v}_0, v_1, \dots, \hat{v}_k \rangle, \text{ est } w(p) = \sum_{i=1}^k w(v_{i-1}, v_i).$$

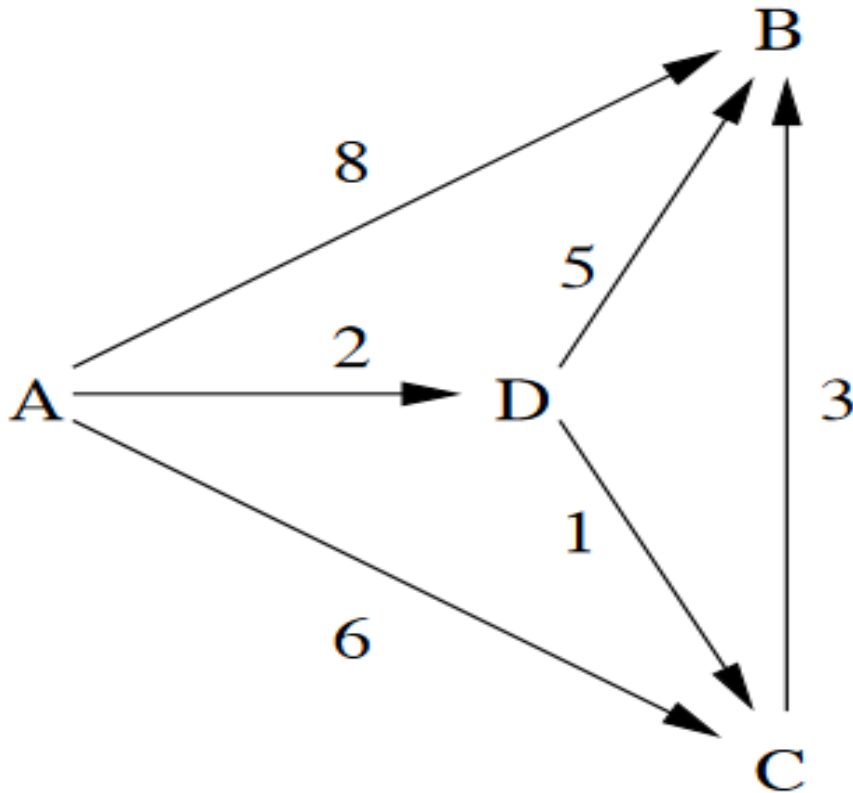
Algorithme de Bellman-Ford (4)

Algorithme 1 Bellman-Ford(G, w, s)

- 1: **pour tout** sommet $v \in V$ **faire** // Initialisation
 - 2: $d[v] \leftarrow \infty, \pi[v] \leftarrow \text{NIL}$
 - 3: $d[s] \leftarrow 0$
 - 4: **pour** i de 1 à $|V| - 1$ **faire**
 - 5: **pour tout** arc $(u, v) \in E$ **faire** // relâchement de l'arc (u, v)
 - 6: **si** $d[v] > d[u] + w(u, v)$ **alors**
 - 7: $d[v] \leftarrow d[u] + w(u, v), \pi[v] \leftarrow u$
 - 8: **pour tout** arc $(u, v) \in E$ **faire** // détection des circuits négatifs
 - 9: **si** $d[v] > d[u] + w(u, v)$ **alors**
 - 10: **retourner Faux**
 - 11: **retourner Vrai**
-

Algorithme de Bellman-Ford (5)

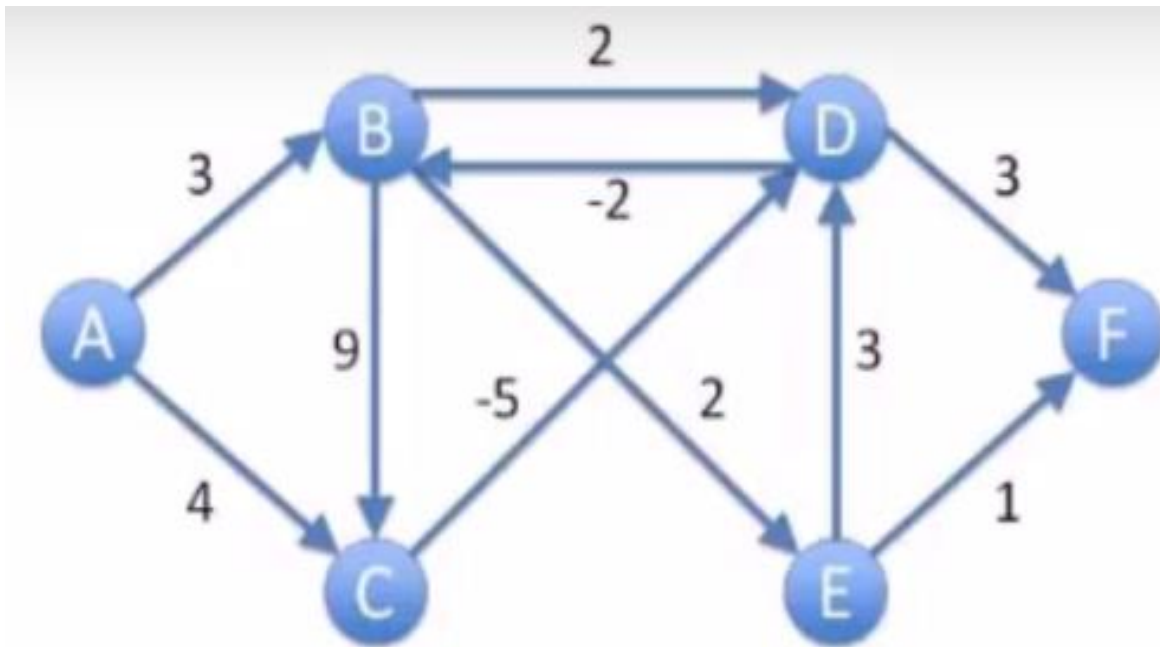
Exemple : trouver les plus courts chemins de A à B et vérifier s'il n'existe pas de boucle de poids négatif à l'aide de l'algorithme de Ford sur le graphe de la figure qui suit : relâchement



--	--	--	--	--

Algorithme de Bellman-Ford (6)

Exercice 1 : trouver les plus courts chemins de A à F et vérifier s'il n'existe pas de boucle de poids négatif à l'aide de l'algorithme de Ford sur le graphe de la figure qui suit :



RIP (1)

Fonctionnement du protocole RIP

RIP utilise le concept de “ vecteur de distance ”, qui s’appuie sur un algorithme de calcul du chemin le plus court dans un graphe **Bellman-Ford**. Le graphe est celui des routeurs, la longueur du chemin est établie en nombre de **sauts** (hop) entre la source et la destination. Cette distance est exprimée comme un nombre entier variant entre 1 et 15 ; la valeur 16 est considérée comme l’infini et indique une mise à l’écart de la route.

RIP (2)

Fonctionnement du protocole RIP

- Chaque routeur **émit** dans un datagramme portant une adresse IP de broadcast (255.255.255.255), à fréquence fixe (environ 30 secondes) le contenu de **sa table de routage** et **écoute** celle des autres routeurs pour compléter sa propre table.
- Si une route n'est **pas annoncée** au moins une fois en **trois** minutes, la distance devient “ infinie ”, et la route sera **retirée** un peu plus tard de la table

RIP (3)

Mise à jour de la table de routage

Lorsqu'un routeur reçoit une route, il compare la route avec les informations de sa propre table de routage :

1. Si la destination n'existe pas, et que la métrique n'est pas infinie alors la route est ajoutée.
2. Si la destination reçue existe, et que la métrique obtenue est meilleure alors la table de routage est mise à jour.
3. Sinon on ne change rien.

RIP V2 (1)

Fonctionnement du protocole RIP

Il existe deux versions de RIP, la deuxième étant une amélioration de la première.

RIPv2 apporte les modifications suivantes à RIPv1 :

1. RIP V1 ne supporte pas les **sous réseaux** ce qui n'est pas le cas pour RIP V2.
2. Le support de **V.L.S.M.** (Variable Length Subnet Masking) ;
3. utilisation d'une adresse de **multicast** (224.0.0.9) pour diffuser les vecteurs de distance au lieu de l'adresse de broadcast ; ce qui réduit l'encombrement sur le réseau ;

RIP V2 (2)

Fonctionnement du protocole RIP

4. **L'authentification** de la source de la mise à jour de routage par un texte en clair (actif par défaut), ou un texte crypté suivant l'algorithme MD5 (Message-Digest 5).
5. **interopérabilité** entre protocoles de routage en diffusant des routes produites par d'autres protocoles.

IGRP(1)

Interior Gateway Routing Protocol

Le protocoles de routage IGRP a été développé par Cisco, il a été spécifiquement développé pour résoudre les problèmes associés au routage dans de grands réseaux, qui dépassaient la portée des protocoles tels que RIP. Tout comme RIP, le protocole IGRP est un protocole à vecteur de distance. Toutefois, pour déterminer le meilleur chemin, il tient compte d'autres facteurs tels que la bande passante, la charge et le délai.

IGRP(2)

Interior Gateway Routing Protocol

Metric= (puiss(10,7)/bande passante +
somme(délais))

	RIP v1	RIP v2	IGRP	OSPF	IS-IS
<i>Interior/Exterior?</i>	Interior	Interior	Interior	Interior	Interior
<i>Type</i>	Distance Vector	Distance Vector	Distance Vector	Link-state	Link-state
<i>Default Metric</i>	Hopcount	Hopcount	Bandwidth/Delay	Cost	Cost
<i>Administrative Distance</i>	120	120	100	110	115
<i>Hopcount Limit</i>	15	15	255 (100 default)	None	None
<i>Convergence</i>	Slow	Slow	Slow	Fast	Fast
<i>Update timers</i>	30 seconds	30 seconds	90 seconds	Only when changes occur; (LSA table is refreshed every 30 minutes, however)	Only when changes occur
<i>Updates</i>	Full table	Full table	Full table	Only Changes	Only changes
<i>Classless</i>	No	Yes	No	Yes	Yes
<i>Supports VLSM</i>	No	Yes	No	Yes	Yes
<i>Algorithm</i>	Bellman-Ford	Bellman-Ford	Bellman-Ford	Dijkstra	Dijkstra