

Plus court chemin

Christophe Viroulaud

Terminale - NSI

Archi 15

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Pour calculer rapidement les distances dans le réseau, les routeurs appliquent des algorithmes conçus au début de l'ère de l'informatique.

Comment fonctionnent les algorithmes de plus court chemin ?

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

1. Représentation des réseaux

1.1 Graphe orienté

1.2 Graphe pondéré

2. Protocole RIP : algorithme de Bellman-Ford

3. OSPF : algorithme de Dijkstra

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme de Dijkstra

Principe

Mise en application

Complexité

Représentation des réseaux - Graphe orienté

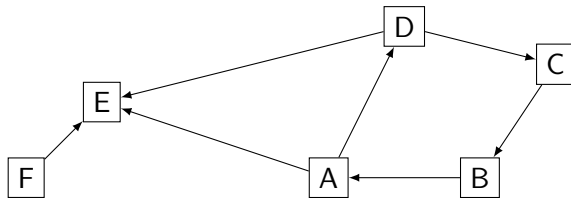


FIGURE 1 – graphe orienté

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

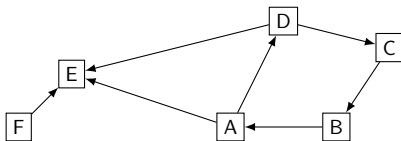


FIGURE 2 – graphe orienté

À retenir

Pour chaque nœud on peut définir ses **prédécesseurs** et ses **successeurs**.

- ▶ Le nœud F ne possède pas de *prédécesseur*.
- ▶ Le nœud E ne possède pas de *successeur*.

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

1. Représentation des réseaux

1.1 Graphe orienté

1.2 Graphe pondéré

2. Protocole RIP : algorithme de Bellman-Ford

3. OSPF : algorithme de Dijkstra

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

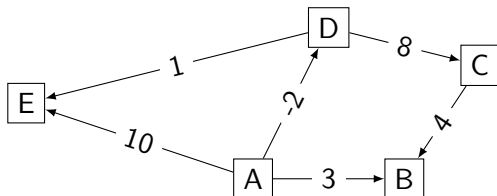


FIGURE 3 – graphe pondéré

À retenir

Selon le protocole mis en place la pondération pourra représenter :

- ▶ RIP : le nombre de réseaux traversés,
- ▶ OSPF : le coût d'un réseau.

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

1. Représentation des réseaux

2. Protocole RIP : algorithme de Bellman-Ford

2.1 Principe

2.2 Mise en application

2.3 Complexité

3. OSPF : algorithme de Dijkstra

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme de Dijkstra

Principe

Mise en application

Complexité

Algorithme de Bellman-Ford : Principe

Plus court chemin

► 1956 - 1958

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Algorithme de Bellman-Ford : Principe

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

- ▶ 1956 - 1958
- ▶ Richard Bellman (père programmation dynamique)

Algorithme de Bellman-Ford : Principe

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

- ▶ 1956 - 1958
- ▶ Richard Bellman (père programmation dynamique)
- ▶ Lester Ford (problème de flot maximum)

Algorithme de Bellman-Ford : Principe

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

- ▶ 1956 - 1958
- ▶ Richard Bellman (père programmation dynamique)
- ▶ Lester Ford (problème de flot maximum)
- ▶ redécouvert par Edward Moore en 1959

Algorithme de Bellman-Ford : Principe

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

- ▶ 1956 - 1958
- ▶ Richard Bellman (père programmation dynamique)
- ▶ Lester Ford (problème de flot maximum)
- ▶ redécouvert par Edward Moore en 1959
- ▶ Le protocole RIP applique cet algorithme.

À retenir

La distance pour atteindre chaque nœud correspond à la distance pour atteindre son prédécesseur à laquelle on ajoute le poids de l'arête les séparant.

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

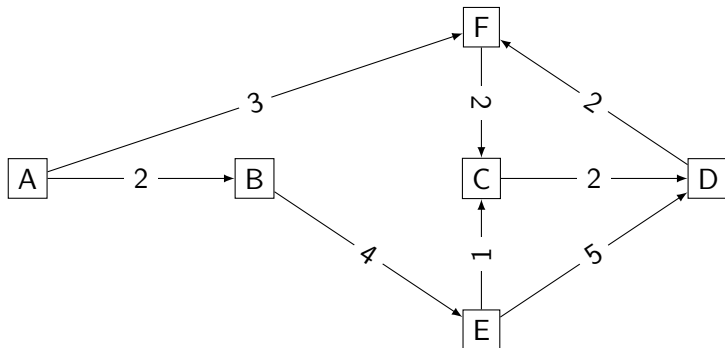


FIGURE 4 – graphe orienté pondéré

À retenir

L'algorithme de Bellman-Ford applique une méthode récursive.

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Pour chaque routeur, on obtient un tableau contenant la distance minimale entre le routeur de départ et chaque autre routeur.

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

1. Représentation des réseaux

2. Protocole RIP : algorithme de Bellman-Ford

2.1 Principe

2.2 Mise en application

2.3 Complexité

3. OSPF : algorithme de Dijkstra

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Initialisation :

- ▶ Créer un tableau des distances entre A et les routeurs, initialisées à l'infini.
- ▶ Modifier la distance vers A à 0.

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Initialisation :

- ▶ Créer un tableau des distances entre A et les routeurs, initialisées à l'infini.
- ▶ Modifier la distance vers A à 0.

Déroulement :

- ▶ Tant que (nombre d'itérations) < (nombre de routeurs)
 - ▶ Pour chaque arc du graphe
 - ▶ Si (distance du routeur) > (distance de son prédécesseur + poids de l'arc entre les deux routeurs)
⇒ Mettre à jour distance du routeur

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

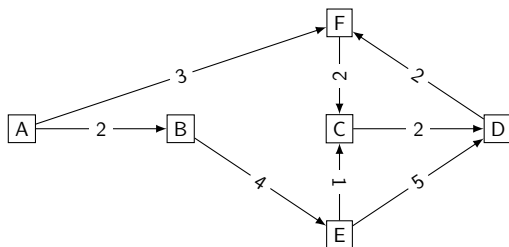
Mise en application

Complexité

Observations

- ▶ On effectue autant d'itérations qu'il y a de routeurs.
- ▶ On regarde chaque arc à chaque tour.

Initialisation



A	B	C	D	E	F
0	∞	∞	∞	∞	∞

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

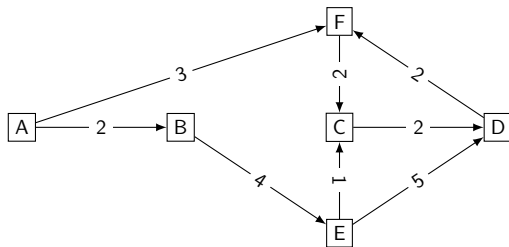
Principe

Mise en application

Complexité

Première itération

Plus court chemin



A	B	C	D	E	F
0	2 (A)	∞	∞	∞	∞

La distance calculée pour atteindre B correspond à la distance de son prédécesseur (A) augmentée du poids de l'arc A-B :

$$0 + 2 = 2 < \infty$$

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

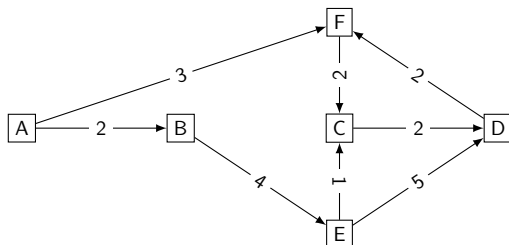
OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Première itération



A	B	C	D	E	F
0	2 (A)	∞	∞	∞	∞

Premier prédécesseur de C : E

$$\infty + 1 = \infty \not< \infty$$

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

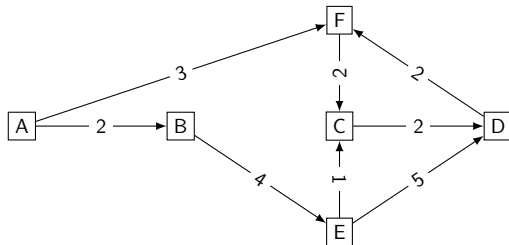
OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Première itération



A	B	C	D	E	F
0	2 (A)	∞	∞	∞	∞

Second prédécesseur de C : F

$$\infty + 2 = \infty \neq \infty$$

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

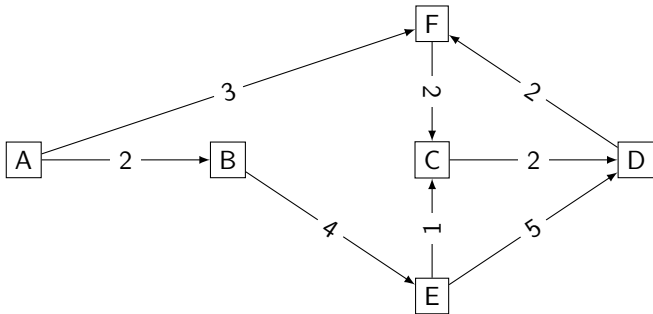
OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Activité 1 : Continuer de dérouler la première itération de l'algorithme sur le graphe.



Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

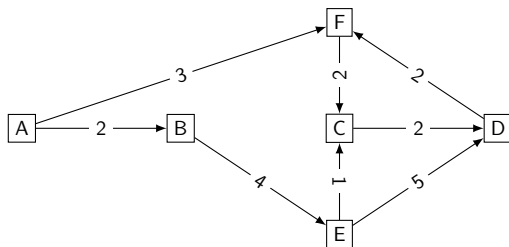
OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Première itération



A	B	C	D	E	F
0	2 (A)	∞	∞	∞	∞

Même constat pour D

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

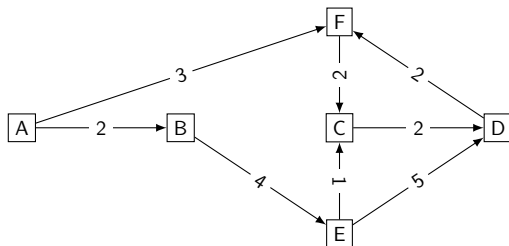
OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Première itération



A	B	C	D	E	F
0	2 (A)	∞	∞	6 (B)	∞

Prédécesseur de E : B

$$2 + 4 = 6 < \infty$$

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

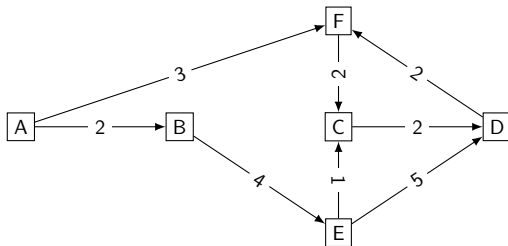
OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Première itération



A	B	C	D	E	F
0	2 (A)	∞	∞	6 (B)	3 (A)

Premier prédécesseur de F : A

$$0 + 3 = 3 < \infty$$

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

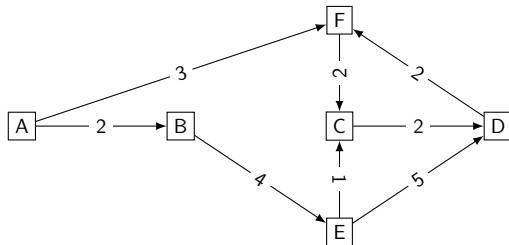
OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Première itération



A	B	C	D	E	F
0	2 (A)	∞	∞	6 (B)	3 (A)

Second prédécesseur de F : D

$$\infty + 2 = \infty \not< 3$$

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

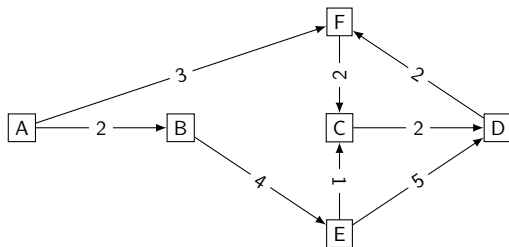
OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Première itération



A	B	C	D	E	F
0	2 (A)	∞	∞	6 (B)	3 (A)

Fin de la première itération

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

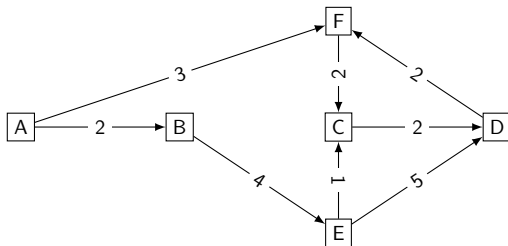
OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

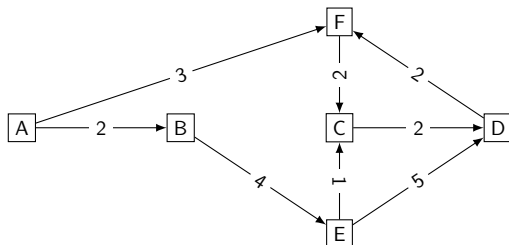
Deuxième itération



A	B	C	D	E	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	∞	∞	6 (B)	3 (A)

Pas de modification pour A et B

Deuxième itération

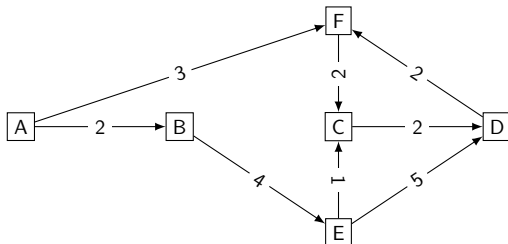


A	B	C	D	E	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	7 (E)	∞	6 (B)	3 (A)

Premier prédécesseur de C : E

$$6 + 1 = 7 < \infty$$

Deuxième itération



A	B	C	D	E	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	5 (F)	∞	6 (B)	3 (A)

Second prédécesseur de C : F

$$3 + 2 = 5 < 7$$

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

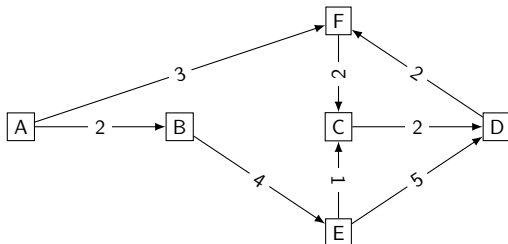
Principe

Mise en application

Complexité

Deuxième itération

Plus court chemin



A	B	C	D	E	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	5 (F)	7 (C)	6 (B)	3 (A)

Premier prédécesseur de D : C

$$5 + 2 = 7 < \infty$$

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

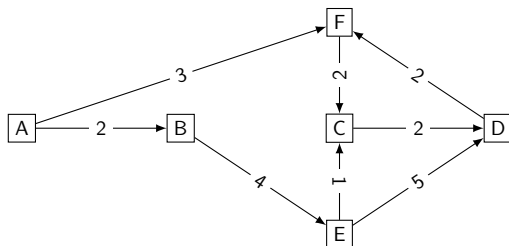
Principe

Mise en application

Complexité

Deuxième itération

Plus court chemin



A	B	C	D	E	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	5 (F)	7 (C)	6 (B)	3 (A)

Second prédécesseur de D : E

$$6 + 5 = 11 \not< 7$$

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

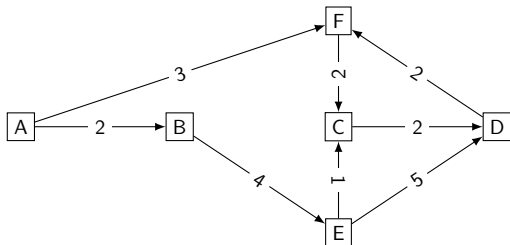
OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

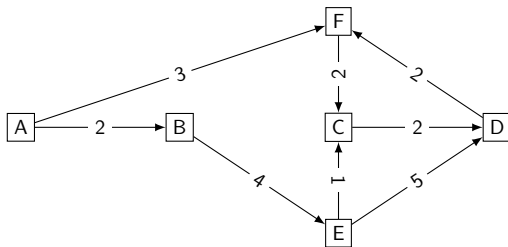
Deuxième itération



A	B	C	D	E	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	5 (F)	7 (C)	6 (B)	3 (A)

Pas de changement pour E et F

Deuxième itération



A	B	C	D	E	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	5 (F)	7 (C)	6 (B)	3 (A)

Fin de la deuxième itération

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

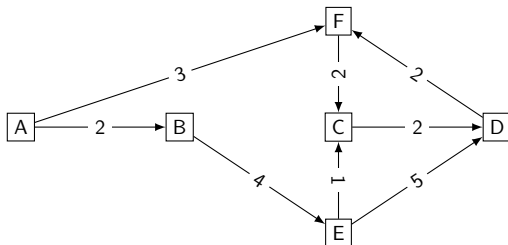
OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Troisième itération



A	B	C	D	E	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	5 (F)	7 (C)	6 (B)	3 (A)
0	2 (A)	5 (F)	7 (C)	6 (B)	3 (A)

Pas de changement lors de la troisième itération

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

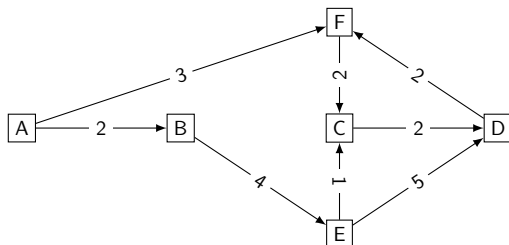
Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité



A	B	C	D	E	F
0	2 (A)	5 (F)	7 (C)	6 (B)	3 (A)

On remonte dans le tableau pour trouver le chemin vers une destination.

$$D \rightarrow C \rightarrow F \rightarrow A$$

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

1. Représentation des réseaux

2. Protocole RIP : algorithme de Bellman-Ford

2.1 Principe

2.2 Mise en application

2.3 Complexité

3. OSPF : algorithme de Dijkstra

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

La complexité dépend de :

- ▶ du nombre de sommets (notée S) : on visite chaque sommet

1 Tant que (le nombre d'itérations) $<$ (nombre de routeurs)

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme de Dijkstra

Principe

Mise en application

Complexité

La complexité dépend de :

- ▶ du nombre de sommets (notée S) : on visite chaque sommet

1 Tant que (le nombre d'itérations) $<$ (nombre de routeurs)

- ▶ du nombre d'arcs (notée A) : pour chaque sommet on regarde tous les arcs du graphe

1 Pour chaque arc du graphe

Complexité de l'algorithme de Bellman Ford

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

$$O(S.A)$$

1. Représentation des réseaux

2. Protocole RIP : algorithme de Bellman-Ford

3. OSPF : algorithme de Dijkstra

3.1 Principe

3.2 Mise en application

3.3 Complexité

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme de Dijkstra

Principe

Mise en application

Complexité

OSPF : algorithme de Dijkstra - principe

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

- ▶ Edsger Dijkstra : mathématicien néerlandais
- ▶ Algorithme utilisé dans GPS

À retenir

principe : construire un sous-graphe en ajoutant à chaque itération un sommet de distance minimale.

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

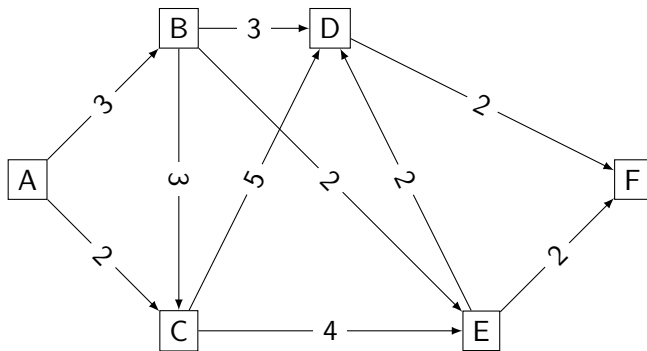


FIGURE 5 – graphe orienté et pondéré

1. Représentation des réseaux

2. Protocole RIP : algorithme de Bellman-Ford

3. OSPF : algorithme de Dijkstra

3.1 Principe

3.2 Mise en application

3.3 Complexité

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme de Dijkstra

Principe

Mise en application

Complexité

Initialisation :

- ▶ Créer un tableau des distances entre A et les routeurs, initialisées à l'infini.
- ▶ Modifier la distance vers A à 0.

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Initialisation :

- ▶ Créer un tableau des distances entre A et les routeurs, initialisées à l'infini.
- ▶ Modifier la distance vers A à 0.

Déroulement :

- ▶ Tant qu'il reste des routeurs non sélectionnés
 - ▶ Parmi les routeurs non-sélectionnés, choisir le routeur (noté S) ayant la plus petite distance.
 - ▶ Pour chaque routeur adjacent à S (noté V) et non déjà sélectionné :
 - ▶ Si $(\text{la distance de V}) > (\text{la distance de S} + \text{poids S-V})$
⇒ Mettre à jour la distance de V.

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Initialisation

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

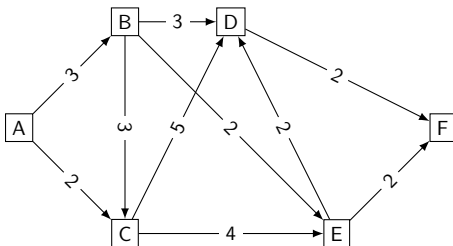
Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

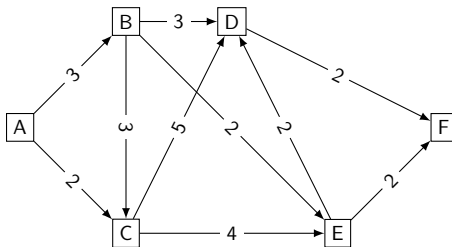
Complexité



A	B	C	D	E	F
0	∞	∞	∞	∞	∞

Sélection de A

Plus court chemin



A	B	C	D	E	F
0	3 (A)	2 (A)	∞	∞	∞
nœuds visités					
A					

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

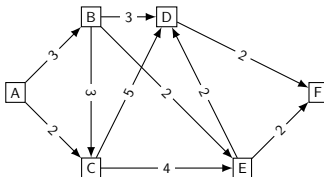
Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité



On sélectionne le nœud non encore visité et avec la plus petite distance : C.

A	B	C	D	E	F
0	3 (A)	2 (A)	∞	∞	∞
	3 (A)	2 (A)	7 (C)	6 (C)	∞

nœuds visités

A - C

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Observation

La route la plus courte a déjà été déterminée pour les nœuds déjà visités. Ils ne seront plus modifiés (cellule grise).

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

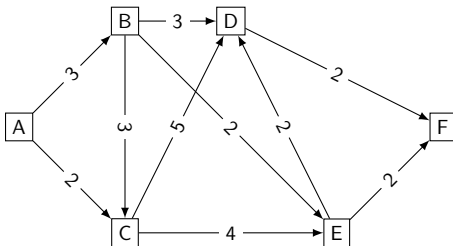
OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Activité 2 : Continuer de dérouler l'algorithme sur le graphe.



Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

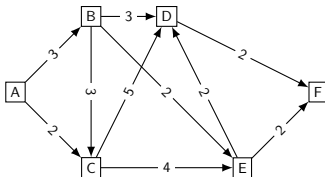
Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité



On sélectionne le nœud non encore visité et avec la plus petite distance : B.

A	B	C	D	E	F
0	3 (A)	2 (A)	∞	∞	∞
	3 (A)	2 (A)	7 (C)	6 (C)	∞
	3 (A)		6 (B)	5 (B)	∞

nœuds visités

A - C - B

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

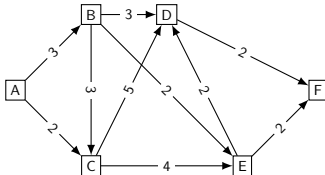
Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité



On sélectionne le nœud non encore visité et avec la plus petite distance : E.

A	B	C	D	E	F
0	3 (A)	2 (A)	∞	∞	∞
	3 (A)	2 (A)	7 (C)	6 (C)	∞
	3 (A)		6 (B)	5 (B)	∞
			6 (B)	5 (B)	7 (E)

nœuds visités

A - C - B - E

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

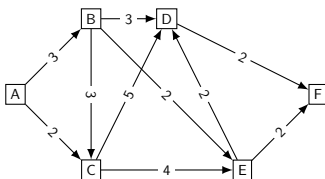
Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité



On sélectionne le nœud non encore visité et avec la plus petite distance : D.

A	B	C	D	E	F
0	3 (A)	2 (A)	∞	∞	∞
	3 (A)	2 (A)	7 (C)	6 (C)	∞
	3 (A)		6 (B)	5 (B)	∞
			6 (B)	5 (B)	7 (E)
			6 (B)		7 (E)

nœuds visités

A - C - B - E - D

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

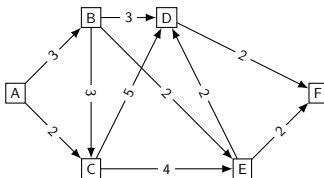
OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Correction



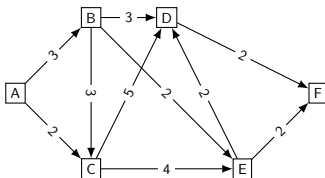
On sélectionne le nœud non encore visité et avec la plus petite distance : F.

A	B	C	D	E	F
0	3 (A)	2 (A)	∞	∞	∞
	3 (A)	2 (A)	7 (C)	6 (C)	∞
	3 (A)		6 (B)	5 (B)	∞
			6 (B)	5 (B)	7 (E)
			6 (B)		7 (E)
					7 (E)

nœuds visités

A - C - B - E - D - F

Correction

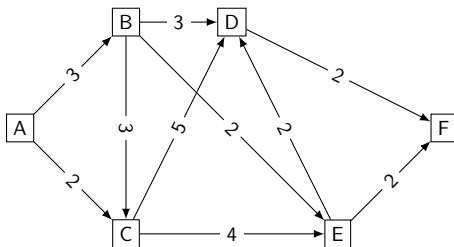


On sélectionne le nœud non encore visité et avec la plus petite distance : F.

A	B	C	D	E	F
0	3 (A)	2 (A)	∞	∞	∞
	3 (A)	2 (A)	7 (C)	6 (C)	∞
	3 (A)		6 (B)	5 (B)	∞
			6 (B)	5 (B)	7 (E)
			6 (B)		7 (E)
					7 (E)

nœuds visités

A - C - B - E - D - F



A	B	C	D	E	F
0	3 (A)	2 (A)	6 (B)	5 (B)	7 (E)

On remonte dans le tableau pour trouver le chemin vers une destination.

$$F \rightarrow E \rightarrow B \rightarrow A$$

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

1. Représentation des réseaux
2. Protocole RIP : algorithme de Bellman-Ford
3. OSPF : algorithme de Dijkstra
 - 3.1 Principe
 - 3.2 Mise en application
 - 3.3 Complexité

Représentation des réseaux

Graphes orientés

Graphes pondérés

Protocole RIP : algorithme de Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme de Dijkstra

Principe

Mise en application

Complexité

- La complexité dépend du nombre de sommets S et du nombre d'arcs A .

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

- La complexité dépend du nombre de sommets S et du nombre d'arcs A .
- Le point clé de l'algorithme tient dans la recherche de la distance minimale.

- 1 Parmi les routeurs non-sélectionnés, choisir le routeur (noté S) ayant la plus petite distance.

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

Complexité de l'algorithme de Dijkstra

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP :
algorithme de
Bellman-Ford

Principe

Mise en application

Complexité

OSPF : algorithme
de Dijkstra

Principe

Mise en application

Complexité

$$O((A + S) \times \log S)$$