Plus court chemin

Archi 15

Christophe Viroulaud

Plus court chemin

Christophe Viroulaud

Terminale - NSI

Archi 15

Plus court chemin

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

Comment fonctionnent les algorithmes de plus court

RIP déjà utilisé avec ARPANET!

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?

Plus court chemin

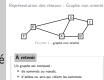


Plus court chemin

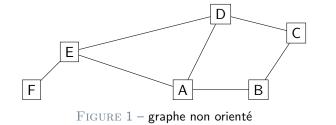
Représentation des réseaux

—Graphe non orienté

Représentation des réseaux - Graphe non orienté



Représentation des réseaux - Graphe non orienté



À retenir

Un graphe est composé :

- de sommets ou nœuds,
- d'arêtes ou arcs qui relient les sommets.

Plus court chemin

Représentation des réseaux

Graphe non orienté

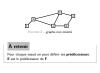
Graphe pondéré

rotocole RIP : Igorithme de sellman-Ford

Complexité

de Dijkstra Principe

rincipe Mise en application Complexité Plus court chemin
Représentation des réseaux
Graphe non orienté



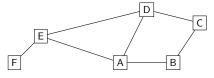


FIGURE 2 – graphe non orienté

À retenir

Pour chaque nœud on peut définir ses **prédécesseurs**. **E** est le prédécesseur de **F**.

Plus court chemin

Représentation des réseaux

Graphe non orienté

Graphe pondéré

rotocole RIP : gorithme de

cipe

Complexité

de Dijkstra

Principe



Sommaire

- 1. Représentation des réseaux
- 1.2 Graphe pondéré

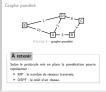
Plus court chemin

Graphe pondéré

Principe

6/67

2022-01-12



pondération peut être négative

Graphe pondéré

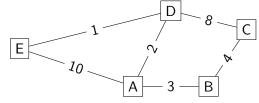


FIGURE 3 – graphe pondéré

À retenir

Selon le protocole mis en place la pondération pourra représenter :

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

Plus court chemin

Représentation des réseaux

Graphe pondéré

Protocole RIP

Protocole RIP : algorithme de Bellman-Ford Principe

OSPF : algorithme de Dijkstra

Principe Mise en application

7 / 67



Sommaire

1 Représentation des réseaux

- 2. Protocole RIP : algorithme de Bellman-Ford
- 2.1 Dringing
- 2.1 Fillicipe
- 2.2 Mise en application
- 2.3 Complexité
- 3. OSPF : algorithme de Diikstr

Plus court chemin

Protocole RIP algorithme de Bellman-Ford

Algorithme de Bellman-Ford : Principe

1956 - 1958

Principe

Plus court chemin

redécouvert par Moore en 59 dans autre contexte

2022-01-12

Algorithme de Bellman-Ford : Principe

► Richard Bellman (père programmation dynamique)

► 1956 - 1958

Algorithme de Bellman-Ford : Principe

- **1956 1958**
- ► Richard Bellman (père programmation dynamique)

Plus court chemin

2022-01-12

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

Algorithme de Bellman-Ford : Principe

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

Algorithme de Bellman-Ford : Principe

redécouvert par Moore en 59 dans autre contexte

Plus court chemin

redécouvert par Moore en 59 dans autre contexte

► 1956 - 1958

Algorithme de Bellman-Ford : Principe

Algorithme de Bellman-Ford : Principe

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

Plus court chemin

Représentation des réseaux

Graphe non orient

Protocole RIP : algorithme de Bellman-Ford

Principe

omplexité

SPF: algorithme Dijkstra

cipe

lise en application omplexité redécouvert par Moore en 59 dans autre contexte

► Le protocole RIP :

► 1956 - 1958

Algorithme de Bellman-Ford : Principe

Algorithme de Bellman-Ford : Principe

► 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.

Plus court chemin

Représentation des réseaux

Graphe pondéré

Protocole RIP : algorithme de

Principe

omplexité

SPF : algorithme e Dijkstra

cipe

Plus court chemin
Protocole RIP : algorithme de Bellman-Ford
Principe



Remarque

L'algorithme de Bellman-Ford est normalement appliqué dans un graphe orienté. Cependant, si les pondérations sont toutes positives, il est possible d'utiliser un graphe non orienté.

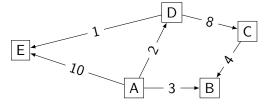


FIGURE 4 – graphe orienté

Plus court chemin

Représentation des réseaux

Graphe non oriente

Protocole RIP : algorithme de

Principe

Vlise en application Complexité

OSPF : algorithm de Dijkstra

Principe

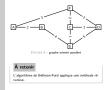
À retenir

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.

Plus court chemin

Principe

Plus court chemin Protocole RIP : algorithme de Bellman-Ford Principe



1. Bellman « père de l'approche dynamique »

Remarque

2. Dans le graphe figure 5 les pondérations représentent un nombre de routeurs traversés pour atteindre le nœud (routeur) suivant.

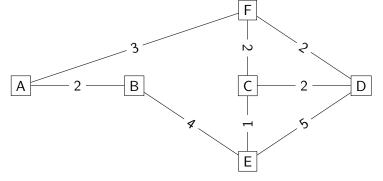


FIGURE 5 – graphe orienté pondéré

À retenir

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

Plus court chemin

Représentation des réseaux

Graphe pondéré

Bellman-For

Principe

Complexité

DSPF : algorithm le Dijkstra

rincipe

distance minimale entre le routeur de départ et chaque autre

peut servir pour retrouver chemin, distance mini...

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

Plus court chemin

Principe



Sommaire

- 1. Représentation des réseaux
- 2. Protocole RIP : algorithme de Bellman-Ford
 - 1 Principa
- 2.2 Mise en application
- 2.2 IVIISE en application
- 3. OSPF : algorithme de Diikstr

Plus court chemin

Mise en application

Initialisation:

Mise en application

initialisées à l'infini. Modifier la distance vers A à 0.

► Créer un tableau des distances entre A et les routeurs.

Initialisation :

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

Plus court chemin

Mise en application

Plus court chemin Protocole RIP : algorithme de Bellman-Ford ─Mise en application

► Créer un tableau des distances entre A et les routeurs, initialisées à l'infini

► Modifier la distance vers A à fl

► Tant que (nombre d'itérations) < (nombre de routeurs prédécesseur + poids de l'arc entre les deux routeurs ⇒ Mettre à jour distance du routeur

- 1. ligne 2 de l'algo : en effet distance de A à A = 0
- 2. ligne 4 : on fait autant d'itérations qu'il y a de routeurs = pire des $cas \rightarrow on pourra améliorer$

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

Plus court chemin
Protocole RIP : algorithme de Bellman-Ford
Mise en application

Observations

• On effectue autant d'itérations qu'il y a de routeurs.

• On regarde chaqué arc à chique tour.

Observations

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

Plus court chemin

Représentation des réseaux

Protocole RIP :

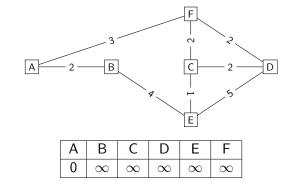
Principe Mise en application

SPF: algorithme

de Dijkstra Principe

rincipe Nise en application

Initialisation



Plus court chemin

Représentation des réseaux

Graphe non orienté

Protocole RIP :

rincipe

Mise en application

SPF : algorithme

Dijkstra ncipe

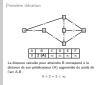
incipe ise en application

se en application mplexité Plus court chemin

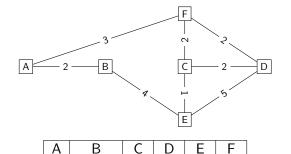
Protocole RIP : algorithme de Bellman-Ford

Mise en application

Première itération



Première itération



	0	2 (A)	∞	∞	∞	∞	
La distance c	ماحييا	ée nour :	ttain	dra B	corr	osnor	, ,

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

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

Plus court chemin

Représentation des réseaux

Graphe non orient Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

Mise en application

Complexité DSPF : algorithme

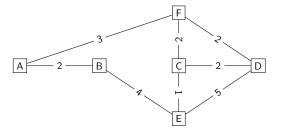
OSPF : algorithme de Dijkstra Principe

Principe Mise en application

Plus court chemin -Protocole RIP : algorithme de Bellman-Ford ☐ Mise en application -Première itération



Première itération



4	В	C	D	Ε	F
0	2 (A)	∞	∞	∞	∞

Second prédécesseur de B : E

$$\infty + 4 = \infty < \infty$$

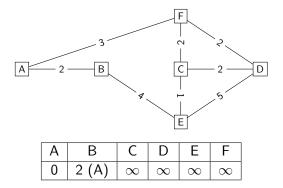
Plus court chemin

Mise en application



On est toujours dans la première itération; on vérifie chaque arc.

Première itération



Premier prédécesseur de C : D

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

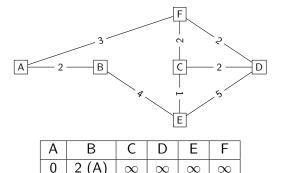
Plus court chemin

Mise en application

Plus court chemin -Protocole RIP: algorithme de Bellman-Ford ☐Mise en application -Première itération



Première itération



Deuxième prédécesseur de C : E

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

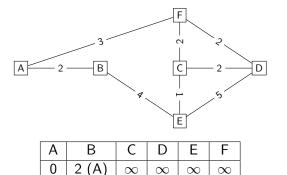
Plus court chemin

Mise en application

Plus court chemin -Protocole RIP: algorithme de Bellman-Ford ☐Mise en application -Première itération



Première itération



Troisième prédécesseur de C : F

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

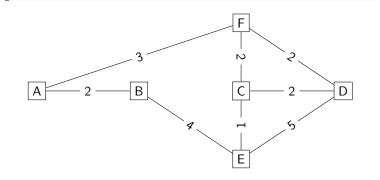
Plus court chemin

Mise en application

Plus court chemin
— Protocole RIP : algorithme de Bellman-Ford
— Mise en application



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



Plus court chemin

Représentation de réseaux

Graphe non orienté

Protocole RIP : algorithme de Bellman-Ford

Principe

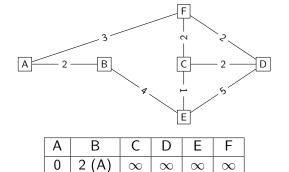
Mise en application Complexité

OSPF : algorithme de Dijkstra

Principe



Première itération



Même constat pour D

Plus court chemin

Représentation des réseaux

Graphe non orienté Graphe nondéré

Protocole RIP : algorithme de Bellman-Ford

Principe

Mise en application Complexité

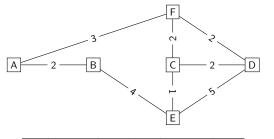
> SPF : algorithme e Dijkstra

Principe
Mise en application

Plus court chemin -Protocole RIP : algorithme de Bellman-Ford ☐ Mise en application -Première itération



Première itération



Α	В	C	D	Е	F
0	2 (A)	∞	8	6 (B)	∞

Premier prédécesseur de E : B

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

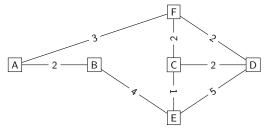
Plus court chemin

Mise en application

Plus court chemin -Protocole RIP : algorithme de Bellman-Ford ☐ Mise en application -Première itération



Première itération



Α	В	С	D	Е	F
0	2 (A)	∞	∞	6 (B)	∞

Deuxième prédécesseur de E : C

$$\infty + 1 = \infty > 6$$

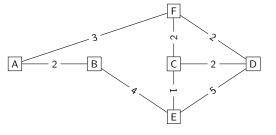
Plus court chemin

Mise en application

Plus court chemin Protocole RIP : algorithme de Bellman-Ford Mise en application Première itération



Première itération



Α	В	С	D	Е	F
0	2 (A)	∞	∞	6 (B)	∞

Troisième prédécesseur de E : D

$$\infty + 5 = \infty > 6$$

Plus court chemin

Représentation des réseaux

Graphe non orient

Protocole RIP algorithme de

rincipe

Mise en application Complexité

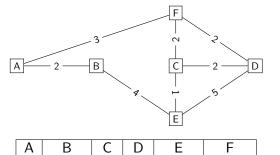
DSPF : algorithme le Dijkstra

Principe Mise en application

Plus court chemin -Protocole RIP: algorithme de Bellman-Ford ☐Mise en application -Première itération



Première itération



 ∞

6 (B)

3 (A)

Premier prédécesseur de F : A

2 (A)

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

 ∞

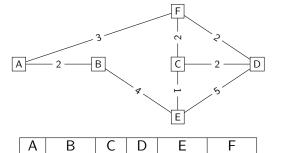
Plus court chemin

Mise en application

Plus court chemin -Protocole RIP: algorithme de Bellman-Ford ☐Mise en application -Première itération



Première itération



 ∞

6 (B)

3 (A)

Deuxième prédécesseur de F : C

2 (A)

$$\infty + 2 = \infty \nless 3$$

 ∞

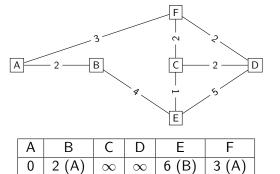
Plus court chemin

Mise en application

Plus court chemin Protocole RIP : algorithme de Bellman-Ford Mise en application Première itération



Première itération



Troisième prédécesseur de F : D

$$\infty + 2 = \infty \nless 3$$

Plus court chemin

Représentation de réseaux

Graphe non orient Graphe pondéré

Protocole RIP : algorithme de

rincipe

Mise en application Complexité

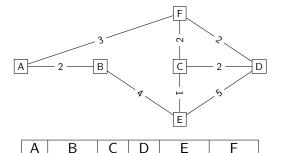
OSPF : algorithme de Dijkstra

Principe

Plus court chemin -Protocole RIP : algorithme de Bellman-Ford ☐ Mise en application -Première itération



Première itération



┌:		I _	: `	: + 4 + :
ГIП	ae	ıa	bremiere	itération

 ∞

 ∞

6 (B)

3 (A)

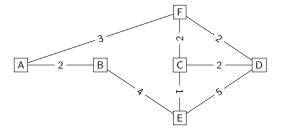
В

2 (A)

Plus court chemin

Mise en application

Deuxième itération



 ∞

 ∞

 ∞

6 (B)

6 (B)

3 (A)

3 (A)

Pas de modification pour A et B

B 2 (A)

2 (A)

Plus court chemin

Représentation des réseaux

Graphe non orient Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

rincipe

Mise en application

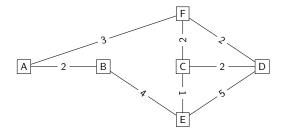
DSPF : algorithme de Dijkstra

Principe Mise en application

Plus court chemin -Protocole RIP : algorithme de Bellman-Ford ☐ Mise en application -Deuxième itération



Deuxième itération



Α	В	С	D	E	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	7 (E)	∞	6 (B)	3 (A)

Premier prédécesseur de C : D

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

Plus court chemin

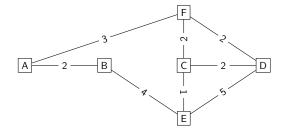
Mise en application

Principe

Plus court chemin -Protocole RIP : algorithme de Bellman-Ford ☐ Mise en application -Deuxième itération



Deuxième itération



Α	В	С	D	Е	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	7 (E)	∞	6 (B)	3 (A)

Deuxième prédécesseur de C : E

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

Plus court chemin

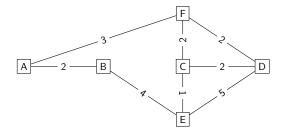
Mise en application

Principe

Plus court chemin —Protocole RIP : algorithme de Bellman-Ford —Mise en application —Deuxième itération



Deuxième itération



Α	В	С	D	Е	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	5 (F)	∞	6 (B)	3 (A)

Troisième prédécesseur de C : F

$$3 + 2 = 5 < 7$$

Plus court chemin

Représentation des réseaux

Graphe non oriente Graphe pondéré

Protocole RIP algorithme de Bellman-Ford

Mise en application

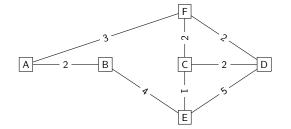
OSPF : algorithme de Diikstra

Principe

Plus court chemin -Protocole RIP : algorithme de Bellman-Ford ☐ Mise en application -Deuxième itération



Deuxième itération



Α	В	С	D	Е	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	5 (F)	7 (C)	6 (B)	3 (A)

Premier prédécesseur de D : C

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

Plus court chemin

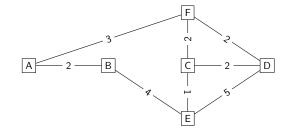
Mise en application

Principe

Plus court chemin —Protocole RIP : algorithme de Bellman-Ford —Mise en application —Deuxième itération



Deuxième itération



Α	В	С	D	Е	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	5 (F)	7 (C)	6 (B)	3 (A)

Second prédécesseur de D : E

$$6 + 5 = 11 \nless 7$$

Plus court chemin

Représentation des réseaux

Graphe non orienté Graphe pondéré

Protocole RIP algorithme de Bellman-Ford

Principe Mise en application

omplexité

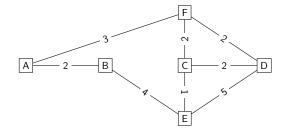
DSPF : algorithm le Dijkstra

Principe

Plus court chemin Protocole RIP : algorithme de Bellman-Ford Mise en application Deuxième itération



Deuxième itération



Α	В	С	D	Е	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	5 (F)	5 (F)	6 (B)	3 (A)

Troisième prédécesseur de D : F

$$3 + 2 = 5 < 7$$

Plus court chemin

Représentation des réseaux

Graphe non orienté Graphe pondéré

Protocole RIP algorithme de Bellman-Ford

Mise en application Complexité

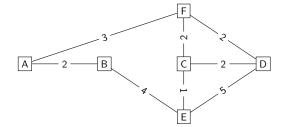
DSPF : algorithmo le Dijkstra

Principe Mise en application

Plus court chemin —Protocole RIP : algorithme de Bellman-Ford —Mise en application —Deuxième itération



Deuxième itération



Α	В	С	D	Е	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	5 (F)	5 (F)	6 (B)	3 (A)

Pas de changement pour E et F

Plus court chemin

Représentation des réseaux

Graphe non orienti Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

Principe

Mise en application

OSPF : algorithme de Dijkstra

Principe

Plus court chemin

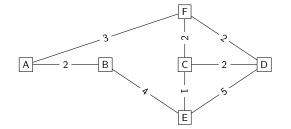
—Protocole RIP : algorithme de Bellman-Ford

—Mise en application

—Deuxième itération



Deuxième itération



Α	В	С	D	Е	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	5 (F)	5 (F)	6 (B)	3 (A)

Fin de la deuxième itération

Plus court chemin

Représentation des réseaux

Graphe non orient Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

Principe

Mise en application

OSPF : algorithme

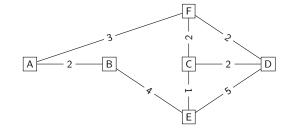
Principe

Plus court chemin —Protocole RIP : algorithme de Bellman-Ford —Mise en application

Troisième itération

- 1. amélioration de l'algorithme : on peut s'arrêter là
- 2. On peut retracer le chemin en partant de la fin : D \rightarrow C \rightarrow F \rightarrow A

Troisième itération



Α	В	С	D	Е	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	5 (F)	5 (F)	6 (B)	3 (A)
0	2 (A)	5 (F)	5 (F)	6 (B)	3 (A)

Pas de changement lors de la troisième itération

Plus court chemin

Représentation de réseaux

Graphe pondéré

Protocole RIP algorithme de Bellman-Ford

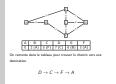
rincipe

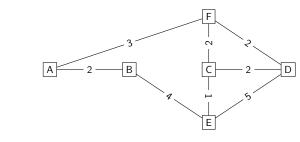
Mise en application Complexité

DSPF : algorithme

Dijkstra

Plus court chemin -Protocole RIP : algorithme de Bellman-Ford ☐Mise en application





Α	В	С	D	Е	F
0	2 (A)	5 (F)	7 (C)	6 (B)	3 (A)

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

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

Plus court chemin

Mise en application

Principe



Sommaire

- . Representation des reseaux
- 2. Protocole RIP : algorithme de Bellman-Ford
- 2.1 Princip
- 2.2 Mise en application
- 2.3 Complexité
- 3. OSPF : algorithme de Diikstr

Plus court chemin

Complexité

44 / 67

2022-01-12

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

Thet que (le nombre d'idérations) < (nombre de souteurs)

Complexité

Complexité

ligne 4 = on peut faire une amélioration : si pas de modification pendant l'itération, on peut s'arrêter

La complexité dépend de :

- du nombre de sommets (notée S) : on visite chaque sommet
- Tant que (le nombre d'itérations) < (nombre de routeurs)

Plus court chemin

Représentation de réseaux

Graphe non orienté

Protocole RIP : algorithme de Bellman-Ford

> cipe e en application

Complexité

PF : algorithme Dijkstra

cipe en application

nexite

2022-01-12

Complexité

La complexité dépend de :

* de nombre de semmets (notée 5) : on vitile chaque
commet

1. Tent que (ils montre d'abstratus) : (remine de notions)

* de nombre d'aris (notée à () ; pour chaque semmet
on apple to tou la vie art de graphe

1. (Pour chaque art de graphe

ligne 4 = on peut faire une amélioration : si pas de modification pendant l'itération, on peut s'arrêter

Complexité

La complexité dépend de :

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

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

Pour chaque arc du graphe

Plus court chemin

Représentation de réseaux

Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

Mise en application Complexité

DSPF : algorithmo le Dijkstra

Dijkstra

Principe Mise en application non orienté

précisément chaque arc est parcouru deux fois car on est dans un graphe

Complexité de l'algorithme de Bellman Ford

O(S.A)

O(S.A)

ne non orienté
ne pondéré
ocole RIP :

Plus court chemin

Principe
Mise en application
Complexité

OSPF : algorit de Dijkstra Principe

n application exité



Plus court chemin

OSPF : algorithme de Dijkstra

└─Principe

□OSPF : algorithme de Dijkstra - principe

OSPF : algorithme de Dijkstra - principe

1. peut se contenter de renvoyer la distance départ/arrivée (et le chemin)

OSPF: algorithme de Diikstra - principe

► Edsger Dijkstra : mathématicien néerlandais

- 2. graphe non orienté possible
- 3. parfois appelé Moore-Dijkstra

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

Plus court chemin

Représentation de réseaux

Graphe pondéré

gorithme de ellman-Ford incipe

omplexité

SPF: algorithme Dijkstra

Principe

se en application mplexité 2022-01-12

À retenir

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

Plus court chemin

réseaux Graphe non orienté

rotocole RIP :

ise en application omplexité

OSPF : algorithme de Dijkstra

Principe

Plus court chemin OSPF : algorithme de Dijkstra Principe



Remarque

Les poids de chaque arc représentent le coût de chaque connexion.

 $\begin{array}{l} \text{coût 5} \rightarrow 20 \text{Mbit/s} \\ \text{coût 2} \rightarrow 50 \text{Mbit/s} \end{array}$

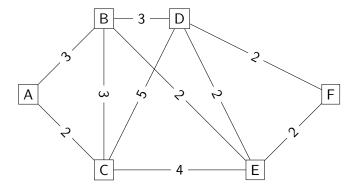


FIGURE 6 – graphe non orienté et pondéré

Plus court chemin

Représentation des réseaux

Graphe nondéré

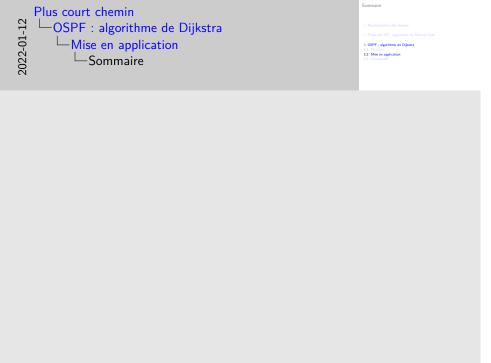
Protocole RIP : algorithme de Bellman-Ford

ise en application

OSPF : algorithme

Dijkstra

Principe
Mise en application



Sommaire

- 3. OSPF : algorithme de Dijkstra
- 3.2 Mise en application

Plus court chemin

Principe Mise en application

Plus court chemin OSPF: algorithme de Dijkstra Mise en application Mise en application

Initialisation :

• Créer un tableau des distances entre A et les routeurs, inivialides à l'infei

Mise en application

Modifier la distance vers A à 0.

IVIIS

Mise en application

- 1. S = suivant; V = voisin
- 2. ligne 7 : on compare encore la distance déjà enregistrée à distance du voisin + arc

Initialisation:

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

Plus court chemin

Représentation des réseaux

Graphe pondéré

Protocole RIP :

Bellman-Ford
Principe
Mise en application

SPF : algorithme e Dijkstra

Principe

Plus court chemin OSPF: algorithme de Dijkstra ─Mise en application

- ► Créer un tableau des distances entre A et les routeurs,
- ► Modifier la distance vers A à 0

- Parmi les nouteurs popuellectionnée choisir relui (noté
- S) ayant la plus petite distance. ▶ Pour chaque routeur adjacent à S (noté V) et non déjà ⇒ Mettre à lour la distance de V.

- 1. S = suivant : V = voisin
- 2. ligne 7 : on compare encore la distance déjà enregistrée à distance du voisin + arc

Initialisation:

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

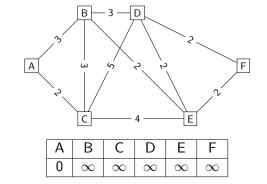
Déroulement :

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

Plus court chemin

Initialisation

Initialisation



Plus court chemin

Représentation des réseaux

Graphe non orienté Graphe nondéré

Protocole RIP : algorithme de Bellman-Ford

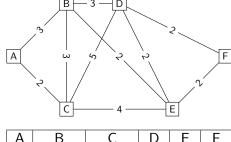
lise en application

SPF : algorithme Dijkstra

rincipe



Sélection de A



Α	В		С	D	E	F
0	3 (A)	2 (A)	∞	∞	∞
	nœuds visités					
			Α			

Plus court chemin

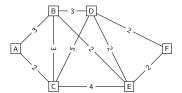
2022-01-12

Plus court chemin

OSPF: algorithme de Dijkstra

Mise en application





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

Α	В	С	D	Е	F
0	3 (A)	2 (A)	∞	∞	∞
	3 (A)	2 (A)	7 (C)	6 (C)	∞

nœuds visités A - C Plus court chemin

Représentation des réseaux

Graphe non orient

Protocole RIP algorithme de

Principe Mise en application

OSDE v al marith man

JSPF : algorithme le Dijkstra

Principe

Plus court chemin

OSPF: algorithme de Dijkstra

Mise en application

Observation

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

Observation

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

eprésentation des éseaux

Graphe non oriente

Protocole RIP : algorithme de

rincipe Mise en application

SPF : algorithme

Dijkstra

Principe

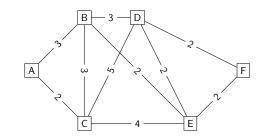
Plus court chemin

OSPF: algorithme de Dijkstra

Mise en application



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



Plus court chemin

deprésentation des éseaux

Graphe non orienté Graphe pondéré

Protocole RIP :

rincipe

omplexité

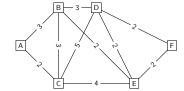
de Dijkstra

Miss sa saulissaiss



C n'est pas regardé : on a déjà trouvé la + petite distance pour lui = il a déjà été visité

Correction



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

Α	В	С	D	Е	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

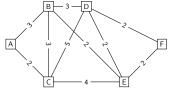
Plus court chemin

Principe

Plus court chemin OSPF: algorithme de Dijkstra ☐Mise en application -Correction



Correction

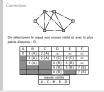


On sélectionne le nœud non encore visité et avec la plus petite distance : E.

Α	В	С	D	Е	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 Plus court chemin

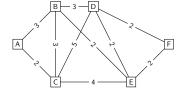
Principe



pas de modification

2022-01-12

Correction



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

Α	В	С	D	Е	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

Plus court chemin

Représentation des réseaux

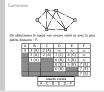
Graphe non orient Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

lise en application

OSPF : algorithm de Dijkstra

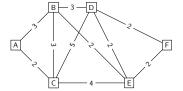
Principe



pas de modification

2022-01-12

Correction



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

Α	В	С	D	Е	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 Plus court chemin

Représentation des réseaux

Graphe non orient Graphe pondéré

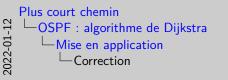
Protocole RIP :

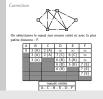
rincipe

Complexité

OSPF : algorithme de Dijkstra

Principe

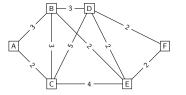




tous les nœuds ont été visités = fin de l'algorithme. On peut reconstruire le chemin.

pour $F: F \rightarrow E \rightarrow B \rightarrow A$

Correction

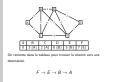


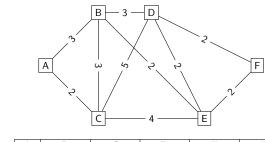
On sélectionne le nœud non encore visité et avec la plus petite distance: F.

Α	В	С	D	Е	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 Plus court chemin

Plus court chemin -OSPF : algorithme de Dijkstra ☐Mise en application





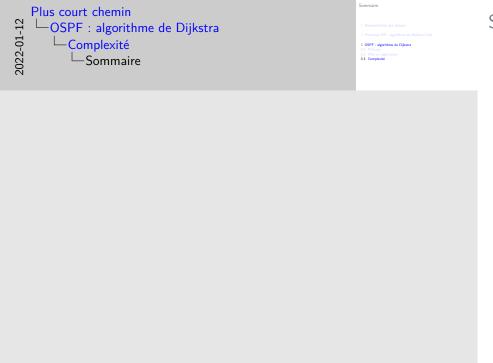
Α	В	С	D	Е	F
0	3 (A)	2 (A)	6 (B)	5 (B)	7 (E)

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

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

Plus court chemin

Principe



Sommaire

- 3. OSPF : algorithme de Dijkstra
- 3.3 Complexité

Complexité

Principe

Plus court chemin

65 / 67

Complexité

Complexité

► La complexité dépend du nombre de sommets S et du

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

- 1. hors programme
- 2. utilisation de tas par exemple

66 / 67

Plus court chemin

Principe Complexité

Complexité

Parmi les routeurs non-sélectionnés, choisir le

routeur (noté S) ayant la plus petite distance.

1. hors programme

2022-01-12

2. utilisation de tas par exemple

- 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.

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

Plus court chemin

Représentation des réseaux

Graphe non oriente Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

lise en application

SPF : algorithm e Dijkstra

Principe

Mise en application Complexité

Complexité de l'algorithme de Dijkstra

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

Principe

Plus court chemin

Complexité