Plus court chemin

Christophe Viroulaud
Terminale - NSI
Archi 15

Plus court chemin

Christophe Viroulaud

Terminale - NSI

Archi 15

ation des

pondéré

Plus court chemin

hme de an-Ford

_{lexité} F : algorithme

ijkstra pe

oe n application exité 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 ?

se en application mplexité

PF: algorithme Dijkstra .

pe on application

e en application nplexité



Plus court chemin

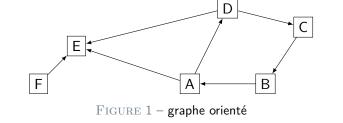
-Représentation des réseaux

-Graphe orienté

Représentation des réseaux - Graphe orienté

Représentation des réseaux - Graphe orienté

Représentation des réseaux - Graphe orienté

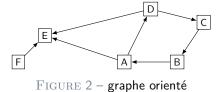


Plus court chemin

Graphe orienté

Plus court chemin Représentation des réseaux Graphe orienté





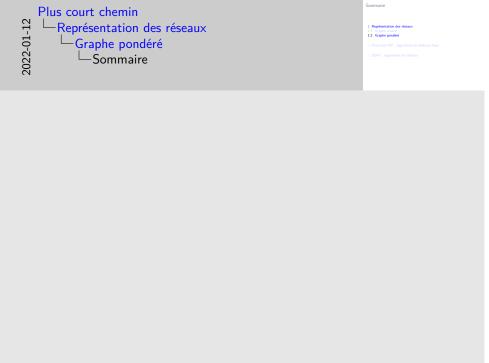
À retenir

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*.

Plus court chemin

Graphe orienté



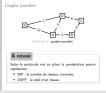
Sommaire

- 1. Représentation des réseaux
- 1.2 Graphe pondéré
- -
- 3. OSPF : algorithme de Dijkstra

Plus court chemin

Graphe pondéré

2022-01-12



pondération peut être négative

Graphe pondéré

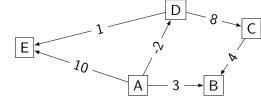


FIGURE 3 – graphe pondéré

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

C I III

Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

Mise en application

OSPF : algorithme

Principe



Sommaire

Représentation des réseaux

- 2. Protocole RIP : algorithme de Bellman-Ford
- 0.1 D.1.........
- 2.1 Fillicipe
- 2.2 Mise en application
- 2.2 Mise en application
 2.3 Complexité
- 3. OSPF : algorithme de Diikstr

Plus court chemin

Protocole RIP algorithme de Bellman-Ford

redécouvert par Moore en 59 dans autre contexte

Algorithme de Bellman-Ford : Principe

Algorithme de Bellman-Ford : Principe

1956 - 1958

► 1956 - 1958

Plus court chemin

Principe

redécouvert par Moore en 59 dans autre contexte

2022-01-12

Algorithme de Bellman-Ford : Principe

► 1956 - 1958

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

Plus court chemin

2022-01-12

► Richard Bellman (père programmation dynamique)

Lester Ford (problème de flot maximum)

► 1956 - 1958

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

Plus court chemin

Représentation des réseaux

Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

Principe

se en application

PF : algorithme

ncipe

rincipe lise en application omplexité Algorithme de Bellman-Ford : Principe

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

Algorithme de Bellman-Ford : Principe

redécouvert par Moore en 59 dans autre contexte

- **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 oriente Graphe pondéré

Protocole RIP :

Principe

e en application

DSPF : algorithmo le Dijkstra

Dijkstra

ncipe

redécouvert par Moore en 59 dans autre contexte

Algorithme de Bellman-Ford : Principe

▶ 1956 - 1958

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

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

Représentation des réseaux

Graphe pondéré

Protocole RIP : algorithme de

Principe

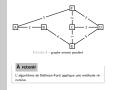
ise en application implexité

SPF: algorithme Dijkstra

Principe

Principe Mise en application

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



1. Bellman « père de l'approche dynamique »

Remarque

2. Dans le graphe figure 4 les pondérations représentent un nombre de routeurs traversés pour atteindre le nœud (routeur) suivant.

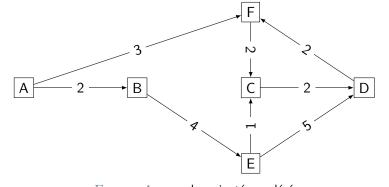


FIGURE 4 – graphe orienté pondéré

À retenir

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

Plus court chemin

Représentation des réseaux

Graphe orienté

Graphe pondéré

Protocole RIP

Principe

Complexité

OSPF : algorithme de Dijkstra

rincipe

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
- 2.1 Princip
- 2.2 Mise en application
- 2.2 IVIISE en application
- 3. OSPF : algorithme de Diiksti

Plus court chemin

Mise en application

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

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

Observations

No effectus autant d'itérations qu'il y a de routeurs.

On regarde chaque arc à chaque 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

Graphe pondéré

Protocole RIP : algorithme de

Principe Mise en application

omplexité

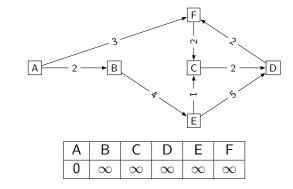
SPF : algorithme Dijkstra

Principe

rincipe lise en application 2022-01-12

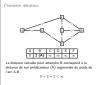


Initialisation

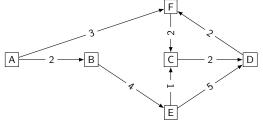


Plus court chemin

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



Première itération



Α	В	C	D	Ε	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 :

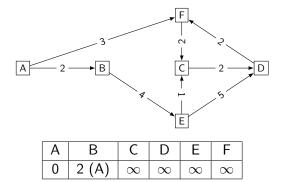
$$0 + 2 = 2 < \infty$$

Plus court chemin



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

Première itération



Premier prédécesseur de C : E

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

Plus court chemin

Représentation des réseaux

Graphe prienté

Protocole RIP

eliman-Fo

Mise en application

SPF: algorithme

de Dijkstra

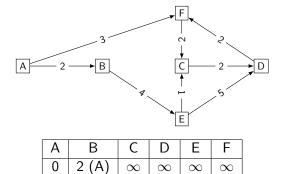
Principe Mise en application

Mise en application Complexité

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



Première itération



Second prédécesseur de C : F

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

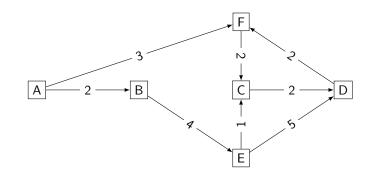
Plus court chemin

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 des réseaux

Graphe orienté

Protocole RIP : algorithme de

Principe

Mise en application Complexité

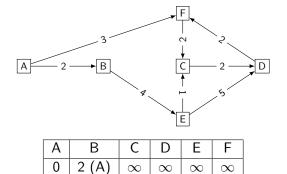
OSPF : algorithme de Dijkstra

Principe

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



Première itération



Même constat pour D

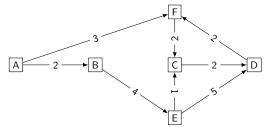
Plus court chemin

Mise en application

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



Première itération



4	В	С	D	E	F
0	2 (A)	∞	∞	6 (B)	∞

Prédécesseur de E : B

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

Plus court chemin

eprésentation des

Graphe orienté

Protocole RIP

Principe

Mise en application Complexité

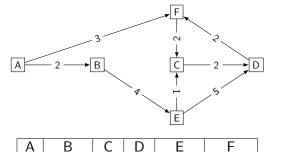
OSPF : algorithme

Principe

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



Première itération



 ∞

6 (B)

3 (A)

2 (A) Premier prédécesseur de F : A

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

 ∞

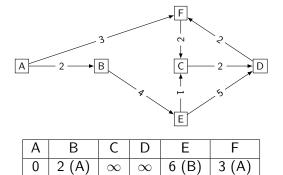
Plus court chemin

Mise en application

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



Première itération



Second prédécesseur de F : D

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

Plus court chemin

Représentation des réseaux

Graphe orienté

Protocole RIP algorithme de

Principe

Mise en application

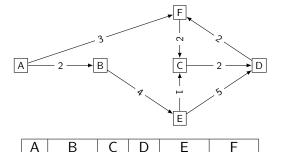
OSPF : algorithme de Dijkstra

Principe

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



Première itération



Fin de la première itération

 ∞

 ∞

6 (B)

2 (A)

3 (A)

Plus court chemin

Représentation des réseaux

Graphe orienté
Graphe pondéré

Protocole RIP algorithme de

Principe

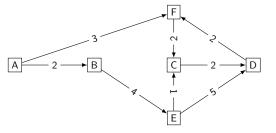
Mise en application

DSPF : algorithme de Dijkstra

Principe



Deuxième itération



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

Pas de modification pour A et B

Plus court chemin

Représentation des réseaux

Graphe orienté Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

rincipe

Mise en application

DSPF : algorithme le Diikstra

Principe

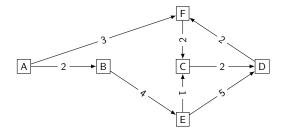
Mise en application

omplexité

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 : E

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

Plus court chemin

Représentation des réseaux

Graphe orienté
Graphe pondéré

Protocole RIP algorithme de Bellman-Ford

Principe Mise en application

omplexité

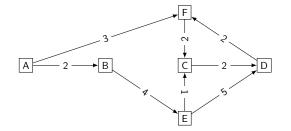
OSPF : algorithmo le Dijkstra

Principe

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)	5 (F)	∞	6 (B)	3 (A)

Second prédécesseur de C : F

$$3+2=5<7$$

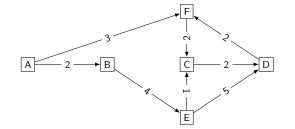
Plus court chemin

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)	7 (C)	6 (B)	3 (A)

Premier prédécesseur de D : C

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

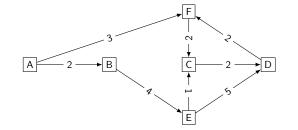
Plus court chemin

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)	7 (C)	6 (B)	3 (A)	

Second prédécesseur de D : E

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

Plus court chemin

Représentation des réseaux

Graphe orienté
Graphe pondéré

Protocole RIP algorithme de

Principe Mise en application

omplexité

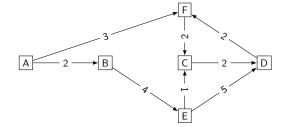
DSPF : algorithmo le Dijkstra

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)	7 (C)	6 (B)	3 (A)

Pas de changement pour E et F

Plus court chemin

Représentation des réseaux

Graphe orienté

Protocole RIP : algorithme de

Dringing

Mise en application

Complexite

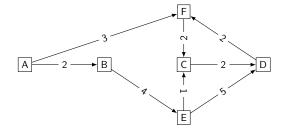
DSPF : algorithme le Dijkstra

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)	7 (C)	6 (B)	3 (A)

Fin de la deuxième itération

Plus court chemin

Mise en application

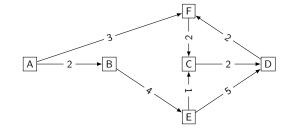
Principe

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

- Trousieme Itération

 | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme Itération | Trousieme 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)	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 onence Graphe pondéré

Protocole RIP algorithme de

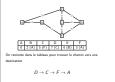
Principe

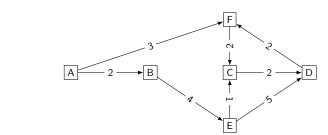
Mise en application

OSPF: 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

Représentation des réseaux

Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

Principe
Mise en application

OSPF : algorithme de Diikstra

de Dijkstra Principe



Sommaire

- 2. Protocole RIP : algorithme de Bellman-Ford

- 2.2 Mise en application 2.3 Complexité

Plus court chemin

Complexité

2022-01-12

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

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)

Plus court chemin

Complexité

37 / 59

2022-01-12

Complexité Tant que (le nombre d'itérations) < (nombre de routeurs) regarde tous les arcs du graphe Pour chaque arc du 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

Complexité

Complexité de l'algorithme de Bellman Ford

O(S.A)

Plus court chemin

resentation des aux he orienté

tocole RIP : prithme de Iman-Ford

Mise en application Complexité

PF : algorithme Diikstra

Principe Mise en application Complexité



39 / 59

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

Principe

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

Keprésentation des éseaux

raphe pondéré

otocole RIP : gorithme de ellman-Ford

e en application

OSPF: algorithme

ncine

Principe Mise en application

Plus court chemin OSPF : algorithme de Dijkstra Principe



Remarque

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

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

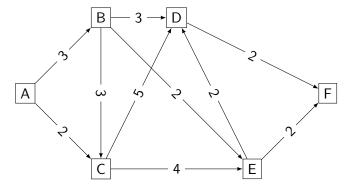


FIGURE 5 – graphe orienté et pondéré

Plus court chemin

Représentation des réseaux

Graphe oriente

Protocole RIP :

Principe

Complexité

OSPF : algorithme de Dijkstra

Principe



Sommaire

- Représentation des réseaux
- 2 Protocole RIP : algorithme de Bellman-Ford
- 3. OSPF : algorithme de Dijkstra
 - Drincina
 - Princip
- 3.2 Mise en application

43 / 59

Plus court chemin

Principe Mise en application

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

Mise en application Créer un tableau des distances entre A et les routeurs.

Mise en application

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

Initialisation

- 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

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
- ► Tant mi'il reste des routeurs non sélectionné Parmi les nucteurs non-silectionnés choisir le route
- ⇒ 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

- (noté S) ayant la plus petite distance.

 Pour chaque routeur adjacent à S (noté V) et non déli

Initialisation:

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

Déroulement :

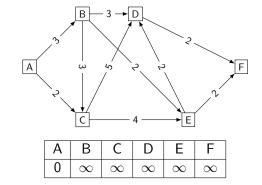
- ► Tant qu'il reste des routeurs non sélectionnés
 - Parmi les routeurs non-sélectionnés, choisir le routeur (noté S) ayant la plus petite distance.
 - Pour chaque routeur adjacent à S (noté V) et non déjà sélectionné :
 - ► Si (la distance de V) > (la distance de S + poids S-V) ⇒ Mettre à jour la distance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

Plus court chemin



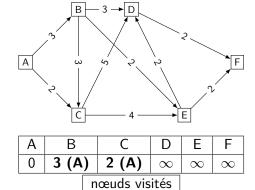
Initialisation

Initialisation



Plus court chemin

Sélection de A



Plus court chemin

Principe

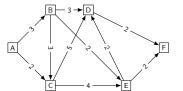
2022-01-12

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 oriente

Protocole RIP algorithme de

rincipe

Complexité

OSPF : algorithme de Dijkstra

Principe

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 seaux

Graphe pondéré

Protocole RIP :

rincipe lise en application

SPE · algorithme

Dijkstra

Principe

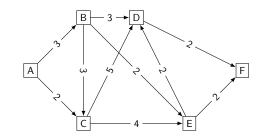
Plus court chemin

OSPF: algorithme de Dijkstra

Mise en application



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



Plus court chemin

Représentation des réseaux

Graphe prienté

Protocole RIP: algorithme de

Principe

Complexité

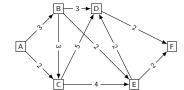
JSPF : algorithme le Dijkstra

Principe



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

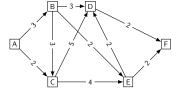
Mise en application

51/59

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	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 Plus court chemin

Représentation des réseaux

Graphe orienté
Graphe pondéré

Protocole RIP algorithme de

rincipe

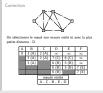
Complexité

OSPF : algorithme de Dijkstra

Principe

Mise en application

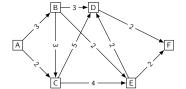
Complexité



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	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 Plus court chemin

Représentation des réseaux

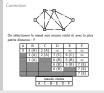
Graphe oriente Graphe pondéré

Protocole RIP :

se en application

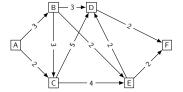
OSPF : algorithm

Principe



pas de modification

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 oriente Graphe pondéré

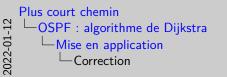
Protocole RIP :

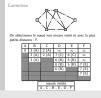
rincipe

Complexité

OSPF : algorithm de Dijkstra

Principe

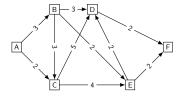




tous les nœuds ont été visités = fin de l'algorithme. On peut reconstruire le chemin.

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

Correction

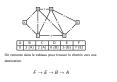


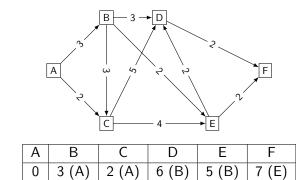
On sélectionne le nœud non encore visité et avec la plus petite distance : F.

Α	В	С	D	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 Plus court chemin

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





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

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

Plus court chemin

Principe Mise en application



Sommaire

- 1. Représentation des réseaux
- 2. Protocole RIP : algorithme de Bellman-Ford
- 3. OSPF : algorithme de Dijkstra
 - Principo
 - rincipe
 - ise en applicatio
- 3.3 Complexité

Plus court chemin

Principe Mise en app 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

2022-01-12

2. utilisation de tas par exemple

Plus court chemin

Principe

Complexité

Complexité

nombre d'arcs A.

distance minimale.

Parmi les routeurs non-sélectionnés, choisir le

routeur (noté S) ayant la plus petite distance.

Plus court chemin

Complexité

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.

Complexité de l'algorithme de Dijkstra

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

Plus court chemin

seaux aphe orienté

tocole RIP :

en application plexité

de Dijkstra Principe

Principe
Mise en application
Complexité