Arch des Réseaux / M1 RID: Belman-Ford & Floyd-Warshall

Ali Benzerbadj

Ain Témouchent University Belhadj Bouchaïb (ATU-2B)

20 avril 2021

Plan

Algorithme de Bellman-Ford

Remarque

Ce cours a été inspiré de la vidéo d'Antoine GOURHAND et al. de l'ECAM/Rennes.

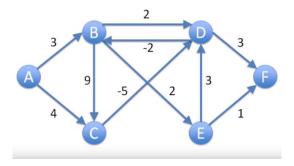
- Dijkstra : Shortest path from one node to all nodes
- Bellman-Ford : Shortest path from one node to all nodes, negative edges allowed
- Floyd-Warshall: Shortest path between all pairs of vertices, negative edges allowed

```
Graphe = (S;A)
Cij appartient R (\leq 0 ou \geq 0)
Cxy = ∞ s'il n'y a pas d'arc entre x et y
                                                        Déclaration des variables
Bellman Ford (G, C, s)
     G = Graphe C = Cii s = point initial
d(s) \leftarrow 0
Pour chaque v E S sauf (s)
          Faire: d(v) \leftarrow \infty
Pouri ← 1 à S-1
Faire, pour chaque arc (u,v) ε A,
          Faire: si d(v) > d(u) + C(u,v)
          d(v) \leftarrow d(u) + C(u,v)
          Fin
Fin
Pour chaque arc (u,v) ε A
          Faire: si d(v) > d(u) + C(u,v)
                     Alors existence d'une boucle négative
Sinon retour à d(v)
```

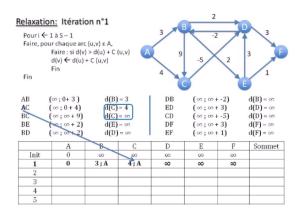
```
Graphe = (S ; A)
Cij appartient R (≤ 0 ou ≥ 0)
Cxy = ∞ s'il n'y a pas d'arc entre x et y
Bellman Ford (G, C, s)
     G = Graphe C = Cij s = point initial
d(s) \leftarrow 0
                                                     Initialisation
Pour chaque v E S sauf (s)
          Faire: d(v) ← ∞
Pouri ← 1 à S-1
Faire, pour chaque arc (u,v) & A.
          Faire: si d(v) > d(u) + C(u,v)
          d(v) \leftarrow d(u) + C(u,v)
          Fin
Fin
Pour chaque arc (u,v) ε A
          Faire: si d(v) > d(u) + C(u,v)
                    Alors existence d'une boucle négative
Sinon retour à d(v)
```

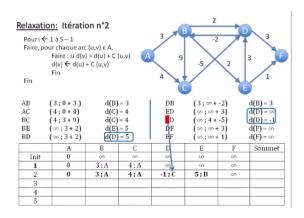
```
Graphe = (S : A)
Cij appartient R (\leq 0 ou \geq 0)
Cxy = ∞ s'il n'y a pas d'arc entre x et y
Bellman Ford (G, C, s)
     G = Graphe C = Cij s = point initial
d(s) \leftarrow 0
Pour chaque v E S sauf (s)
          Faire: d(v) \leftarrow \infty
Pouri ← 1 à S - 1
Faire, pour chaque arc (u,v) ε A,
                                                       Relaxation
          Faire: si d(v) > d(u) + C(u,v)
          d(v) \leftarrow d(u) + C(u,v)
          Fin
Fin
Pour chaque arc (u,v) ε A
          Faire: si d(v) > d(u) + C(u,v)
                     Alors existence d'une boucle négative
Sinon retour à d(v)
```

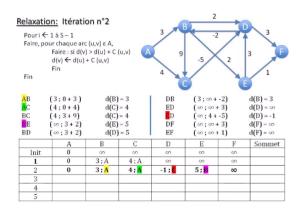
```
Graphe = (S;A)
Cij appartient R (\leq 0 ou \geq 0)
Cxy = ∞ s'il n'y a pas d'arc entre x et y
Bellman Ford (G, C, s)
     G = Graphe C = Cij s = point initial
d(s) \leftarrow 0
Pour chaque v E S sauf (s)
          Faire: d(v) \leftarrow \infty
Pouri ← 1 à S - 1
Faire, pour chaque arc (u,v) & A,
          Faire: sid(v) > d(u) + C(u,v)
          d(v) \leftarrow d(u) + C(u,v)
          Fin
Fin
Pour chaque arc (u,v) ε A
                                                                     Contrôle de la présence
          Faire: si d(v) > d(u) + C(u,v)
                                                                     d'une boucle négative.
                     Alors existence d'une boucle négative
Sinon retour à d(v)
```

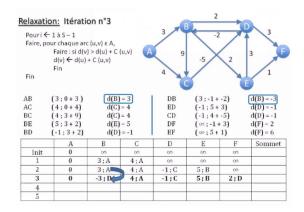


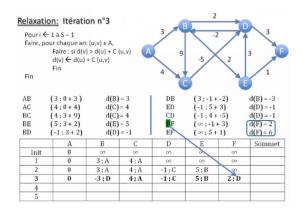
Relaxation: Pour $i \leftarrow 1$ à S - 1Faire, pour chaque arc $(u,v) \in A$, Faire : si d(v) > d(u) + C(u,v) $d(v) \leftarrow d(u) + C(u,v)$ Fin Fin E B 2 D 3 B -2 3 4 9 -5 2 1

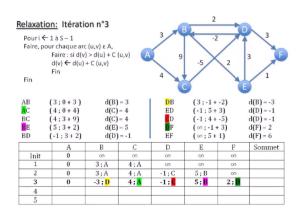


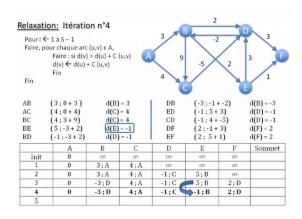


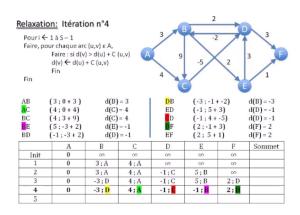


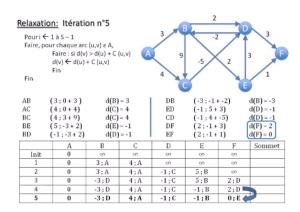


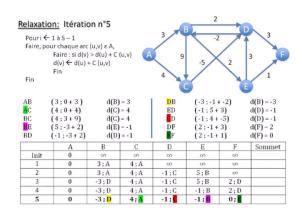










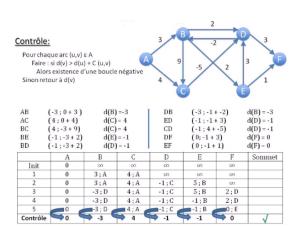


```
Graphe = (S;A)
Cij appartient R (\leq 0 ou \geq 0)
Cxy = ∞ s'il n'y a pas d'arc entre x et y
Bellman Ford (G, C, s)
     G = Graphe C = Cii s = point initial
d(s) \leftarrow 0
Pour chaque v E S sauf (s)
          Faire: d(v) \leftarrow \infty
Pouri ← 1 à S-1
Faire, pour chaque arc (u,v) & A,
          Faire: si d(v) > d(u) + C(u,v)
          d(v) \leftarrow d(u) + C(u,v)
          Fin
Fin
Pour chaque arc (u,v) ε A
                                                                      Contrôle si présence
          Faire: si d(v) > d(u) + C(u,v)
                                                                     d'une boucle négative.
                     Alors existence d'une boucle négative
Sinon retour à d(v)
```

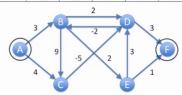
Contrôle:

Pour chaque arc (u,v) ε A
Faire : si d(v) > d(u) + C (u,v)
Alors existence d'une boucle négative
Sinon retour à d(v)

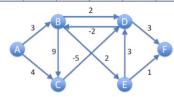
- Consiste à contrôler la **non existence** d'une boucle négative.
- Permet de vérifier que la dernière itération est bien la bonne.



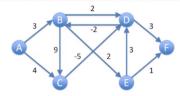
	A	В	C	D	E	F	Sommet
Init	0	00	00	00	00	00	
1	0	3;A	4;A	00	00	00	
2	0	3;A	4;A	-1;C	5;B	00	
3	0	-3; D	4;A	-1;C	5;B	2;D	
4	0	-3; D	4;A	-1;C	-1;B	2;D	
5	0	-3; D	4;A	-1;C	-1;B	0;E	



	A	В	C	D	E	F	Sommet
Init	0	00	00	00	00	00	
1	0	3;A	4;A	00	00	00	
2	0	3;A	4;A	-1;C	5;B	00	
3	0	-3;D	4;A	-1;C	5;B	2;D	
4	0	-3;D	4;A	-1;C	-1;B	2; D	
5	0	-3:D	4:A	-1:C	-1:B	0:E	



	A	В	C	D	Е	F	Sommet
Init	0	00	00	00	00	00	
1	0	3;A	4;A	00	00	00	
2	0	3;A	4;A	-1;C	5;B	00	
3	0	-3;D	4;A	-1;C	5;B	2;D	
4	0	-3;D	4;A	-1;C	-1;B	2;D	
5	0	-3;D	4;A	-1;C	-1;B	0;E	Е



	A	В	C	D	E	F	Sommet
Init	0	00	00	00	00	00	
1	0	3;A	4;A	00	00	00	A
2	0	3;A	4;A	-1;C	5;B	00	C
3	0	-3;D	4;A	-1;C	5;B	2;D	D
4	0	-3;D	4;A	-1;C	-1;B	2;D	В
5	0	-3;D	4;A	-1;C	-1;B	0;E	Е
							F -

