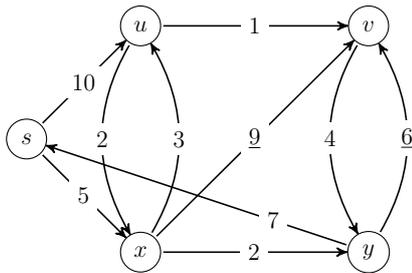


Exercice 1

Soit $G_1 = (X_1, A_1)$ le graphe orienté avec une fonction de pondération $w : A_1 \rightarrow \mathbb{R}$ représenté ci-dessous.



1. En appliquant l'algorithme de *Dijkstra*, déterminer la distance (longueur du plus court chemin) du sommet s à chacun des sommets de G_1 .

Soit les graphes orientés G_2, G_3 et G_4 (avec une fonction de pondération sur les arcs) représentés ci-dessous.

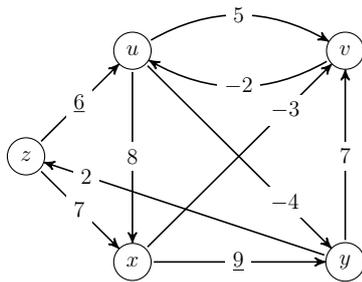


FIGURE 1 – G_2

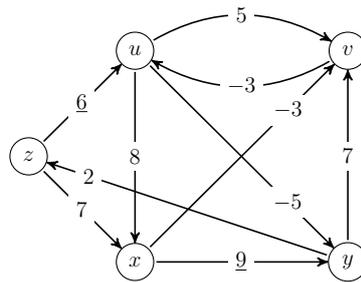


FIGURE 2 – G_3

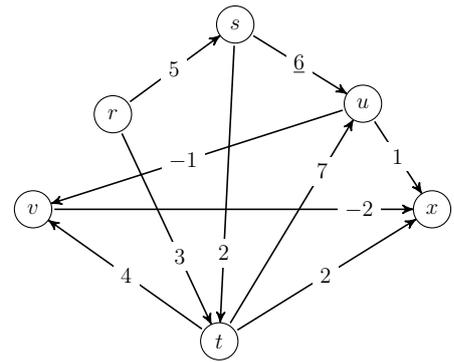


FIGURE 3 – G_4

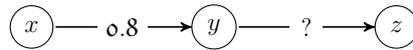
2. Appliquer l'algorithme de *Bellman-Ford* pour résoudre le problème des plus courts chemins ayant pour origine le sommet z dans les graphes G_2 et G_3 .
3. En remarquant que le graphe G_4 est sans circuit, utiliser un algorithme efficace pour calculer les plus courts chemins ayant pour origine le sommet s . Vous explicitez la complexité de votre algorithme.

Exercice 2

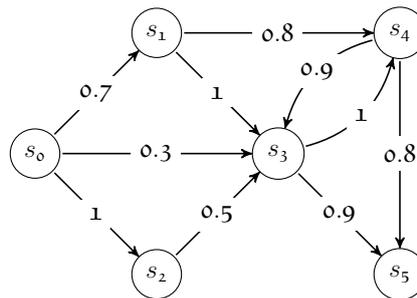
On considère un graphe orienté modélisant un réseau avec une source. Le sommet *source* émet des messages vers les autres sommets. Sur chaque arc il y a une probabilité (comprise entre 0 et 1) de transmission sans erreur d'un message. La probabilité de ne pas avoir d'erreur de transmission le long d'un chemin dépend des probabilités des arcs qui le composent.

1. Quelle doit être la probabilité sur chaque arc d'un chemin si l'on veut être sûr qu'un message soit transmis sans erreur ?

2. Quelle doit être la probabilité de l'arc (y,z) du graphe ci-dessous si l'on souhaite que le chemin de x à z ait une probabilité de transmission sans erreur de 0.4? Quid si l'on souhaite que le chemin de x à z ait probabilité nulle?



3. Comment obtient-on la probabilité de transmission sans erreur le long d'un chemin à partir des probabilités des arcs qui le composent?
4. Modifier l'algorithme de *Dijkstra* afin de calculer les canaux de transmission les plus sûrs entre la source et chacun des autres sommets et l'appliquer au graphe suivant en utilisant s_0 comme source :



Exercice 3

(*) Pour un graphe G donné, on notera $d_s(G)$ les distances des plus courts chemins issu du sommet s vers tous les autres sommets de G . On notera $d_s(G, u)$ la distance d'un plus court chemin issu du sommet s vers le sommet u dans G .

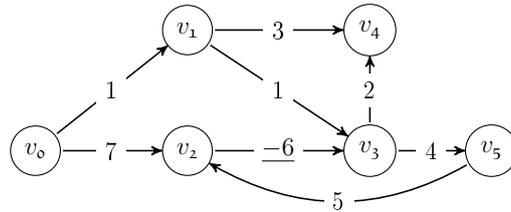
Soit H un graphe orienté et valué ayant un seul arc de poids négatif. On note α l'origine et β l'extrémité de cet arc, *i.e.* $w(\alpha, \beta) < 0$.

- En utilisant l'algorithme de *Dijkstra*, trouver une façon de détecter l'existence d'un circuit dont la somme des poids des arcs est négative (un tel circuit sera appelé un **circuit absorbant**).

Dans la suite on suppose que le graphe H ne possède pas de circuit absorbant, mais qu'il possède un arc (α, β) tel que $w(\alpha, \beta) < 0$. Pour chaque sommet de H , on veut savoir si un plus court chemin depuis un sommet distingué s_* doit emprunter l'arc de poids négatif. Pour ce faire, on procède en deux étapes :

- On applique l'algorithme de *Dijkstra* au graphe H_1 obtenu en supprimant de H l'arc de poids négatif et on calcule la distance $d_{s_*}(H_1)$ entre s_* et tous les sommets de H_1 .
- Soit H_2 le graphe ayant comme sommets ceux qui dans H_1 sont accessibles depuis β et tous les arcs de H les reliant excepté l'arc (α, β) de poids négatif. On calcule, en appliquant l'algorithme de *Dijkstra*, la distance $d_\beta(H_2)$ entre β et tous les sommets de H_2 .

- Comment utiliser les distances ainsi calculées pour savoir si le plus court chemin de s_* à un sommet s quelconque est l'un de ceux qui contiennent l'arc de poids négatif?
- Appliquer votre algorithme au graphe G ci-dessous en utilisant le sommet $s_* = v_0$ comme sommet distingué.



Annexe

A Algorithme de Bellman-Ford

BELLMANN-FORD(G, w, s)

```

1  SOURCE-UNIQUE-INITIALISATION( $G, s$ )
2  pour  $i \leftarrow 1$  à  $|V[G]| - 1$  faire
3      pour chaque arc  $(u, v) \in E[G]$  faire
4          RELÂCHER( $u, v, w$ )
5  pour chaque arc  $(u, v) \in E[G]$  faire
6      si  $d[v] > d[u] + w(u, v)$  alors
7          retourner FAUX
8  retourner VRAI

```

SOURCE-UNIQUE-INITIALISATION(G, s)

```

1  pour chaque sommet  $v \in V[G]$  faire
2       $d[v] \leftarrow \infty$ 
3      père[ $v$ ]  $\leftarrow$  NIL
4   $d[s] \leftarrow 0$ 

```

RELÂCHER(u, v, w)

```

1  si  $d[v] > d[u] + w(u, v)$  alors
2       $d[v] \leftarrow d[u] + w(u, v)$ 
3      père[ $v$ ]  $\leftarrow u$ 

```

Histoire (sources WIKIPÉDIA)

- **RICHARD ERNEST BELLMAN**, né le 29 août 1920 à Brooklyn et mort le 19 mars 1984 à Los Angeles, est un mathématicien américain.
- **EDSGER WYBE DIJKSTRA**, né à Rotterdam le 11 mai 1930 et mort à Nuenen le 6 août 2002, est un mathématicien et informaticien néerlandais du XX^e siècle.
- **LESTER RANDOLPH FORD, Sr.**, né le 29 octobre 1886 et mort le 11 novembre 1967, est un mathématicien américain.

Solutions des exercices optionnels

Correction de l'exercice 3

1. On exécute l'algorithme de *Dijkstra* depuis β dans la graphe $H \setminus \{(\alpha, \beta)\}$ pour calculer $d_\beta(H \setminus \{(\alpha, \beta)\})$. Si $d_\beta(H \setminus \{(\alpha, \beta)\}, \alpha) + w(\alpha, \beta) < 0$ alors il y a un circuit négatif.
2. Pour chaque sommet s , on calcule le minimum entre $d_{s_\star}(H_1, s)$ et $d_{s_\star}(H_1, \alpha) + w(\alpha, \beta) + d_\beta(H_2, s)$. Si le minimum est obtenu par $d_{s_\star}(H_1, \alpha) + w(\alpha, \beta) + d_\beta(H_2, s)$ cela signifie que le plus court chemin de $s_\star \rightsquigarrow s$ emprunte l'arc (α, β) , sinon il ne l'emprunte pas.

Si $\min(d_{s_\star}(H_1, s), d_{s_\star}(H_1, \alpha) + w(\alpha, \beta) + d_\beta(H_2, s)) = d_{s_\star}(H_1, s)$

alors $s_\star \xrightarrow{H_1} s$,

sinon $s_\star \xrightarrow{H_1} \alpha \rightarrow \beta \xrightarrow{H_2} s$.

3. Sur le graphe G , $\alpha = v_2$ et $\beta = v_3$. On teste s'il y a un circuit de poids négatif : non car $d_{v_3}(G \setminus \{(v_2, v_3)\}, v_2) = 9$ et donc $d_{v_3}(G \setminus \{(v_2, v_3)\}, v_2) + w(v_2, v_3) = 9 - 6 > 0$

On définit les graphes G_1 et G_2 en utilisant les mêmes définitions que pour les graphes H_1 et H_2 de l'énoncé.

On calcule $d_{v_0}(G_1)$ dans le graphe sans arc négatif depuis la source v_0 et $d_{v_3}(G_2)$ dans le graphe des sommets accessibles dans G_1 depuis le sommet v_3 (sans l'arc négatif) :

$d_{v_0}(G_1, v_0) = 0$ v_0 est non accessible dans G_1 depuis v_3 , donc $(v_2, v_3) \notin v_0 \rightsquigarrow v_0$.

$d_{v_0}(G_1, v_2) = 7$ $d_{v_3}(G_2, v_2) = 9$

$d_{v_0}(G_1, v_3) = 2$ $d_{v_3}(G_2, v_3) = 0$

$d_{v_0}(G_1, v_1) = 1$ v_1 est non accessible dans G_1 depuis v_3 , donc $(v_2, v_3) \notin v_0 \rightsquigarrow v_1$.

$d_{v_0}(G_1, v_5) = 6$ $d_{v_3}(G_2, v_5) = 4$

$d_{v_0}(G_1, v_4) = 4$ $d_{v_3}(G_2, v_4) = 2$

Calcul du minimum pour les sommets de G_2 : on a $d_{v_0}(G_1, v_2) + w(v_2, v_3) = 7 - 6 = 1$

v_i	$d_{v_0}(G_1, v_i)$	comparaison	$d_{v_0}(G_1, v_2) + w(v_2, v_3) + d_{v_3}(G_2, v_i)$	conclusion
v_2	7	<	$1 + 9 = 10$	$(v_2, v_3) \notin v_0 \rightsquigarrow v_2$
v_3	2	>	$1 + 0 = 1$	$(v_2, v_3) \in v_0 \rightsquigarrow v_3$
v_4	4	>	$1 + 2 = 3$	$(v_2, v_3) \in v_0 \rightsquigarrow v_4$
v_5	6	>	$1 + 4 = 5$	$(v_2, v_3) \in v_0 \rightsquigarrow v_5$

Conclusion : Il faut emprunter l'arc négatif pour obtenir le plus court chemin entre le sommet v_0 et les sommets v_3 , v_4 et v_5 .