Notes de cours
Semaine 21

Cours Turing

Table des matières

1 Les graphes dirigés

Un graphe dirigé est un graphe où chaque arête a un sens.

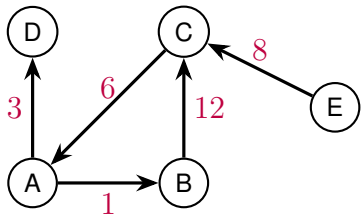


Figure 1: Un graphe dirigé

Une chaîne dans un graphe pondéré n'est valide que si le sens de l'arête est respecté.
Dans la Figure 1, il n'existe pas de chemin entre D et E mais il existe un chemin entre E et D.

1.1 Arête entrante

Pour un sommet, une arête entrante est une arête permettant d'arriver à ce sommet. Dans la Figure 1, C possède 2 arrêtes entrantes.

1.2 Arête sortante

Pour un sommet, une arête sortante est une arête permettant de partir de ce sommet. Dans la Figure 1, A possède 2 arrêtes sortantes.

2 PageRank

L'algorithme PageRank est un algorithme utilisé par Google pour classer les pages web.
L'algorithme a été inventé par Larry Page et Sergey Brin, les fondateurs de Google.
Il fonctionne en attribuant à chaque page un score qui dépend de la quantité de sites qui pointent vers cette page mais également du score de ces sites.

Pour un ensemble de pages web, l'algorithme fonctionne de la manière suivante :

1. Il commence par attribuer à chaque page un score égal
2. Pour chaque page, l'algorithme calcule le nouveau score qui dépend des scores actuels des pages qui pointent vers cette page
3. Les scores des pages sont mis à jour
4. Si la somme des scores ne s'est pas stabilisée, l'algorithme recommence à l'étape 2

! La valeur d'initialisation, la manière dont le score est calculé et la définition de la stabilisation dépendent de l'implémentation de l'algorithme.
Cela signifie que les résultats peuvent varier d'une implémentation à l'autre.

2.1 Exemple

Soit :

- $Rank(I)$ la valeur de PageRank de la page I
- $Sortant(I)$ le nombre de liens sortants de la page I
- $Pointant(I)$ la liste des pages pointant vers la page I
- α un taux d'apprentissage

De manière générale, l'algorithme PageRank calcule le nouveau score de PageRank pour chaque page de la manière suivante :
$Rank_{nouveau}(I) = \alpha \cdot Rank(I) + (1 - \alpha) \cdot \sum_{J \in Pointant(I)}^{n} \frac{Rank(J)}{Sortant(J)}$

Pour alléger la formule, nous allons considérer $\alpha = 0$, c'est-à-dire que nous n'allons pas prendre en compte le score actuel de la page. Ce qui donnera la formule suivante :

$$Rank_{nouveau}(I) = \sum_{J \in Pointant(I)}^{n} \frac{Rank(J)}{Sortant(J)}$$

Soit la situation initiale présentée dans le graphe de la figure 2. On suppose que chaque page a un score initial de PageRank égal à 1.

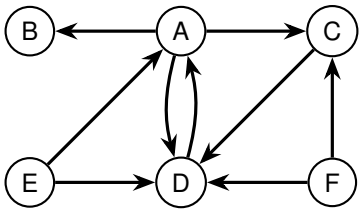


Figure 2: Représentation d'un réseau de pages web

Voici les calculs pour la première itération de l'algorithme :

- $Rank_{nouveau}(A) = \frac{Rank(E)}{Sortant(E)} + \frac{Rank(D)}{Sortant(D)} = \frac{1}{2} + 1 = 1.5$
- $Rank_{nouveau}(B) = \frac{Rank(A)}{Sortant(A)} = \frac{1}{3} = 0.33$
- $Rank_{nouveau}(C) = \frac{Rank(A)}{Sortant(A)} + \frac{Rank(F)}{Sortant(F)} = \frac{1}{3} + \frac{1}{2} = 0.83$
- $Rank_{nouveau}(D) = \frac{Rank(A)}{Sortant(A)} + \frac{Rank(C)}{Sortant(C)} + \frac{Rank(E)}{Sortant(E)} + \frac{Rank(F)}{Sortant(F)} = \frac{1}{3} + 1 + \frac{1}{2} + \frac{1}{2} = 2.33$
- $Rank_{nouveau}(E) = 0$
- $Rank_{nouveau}(F) = 0$

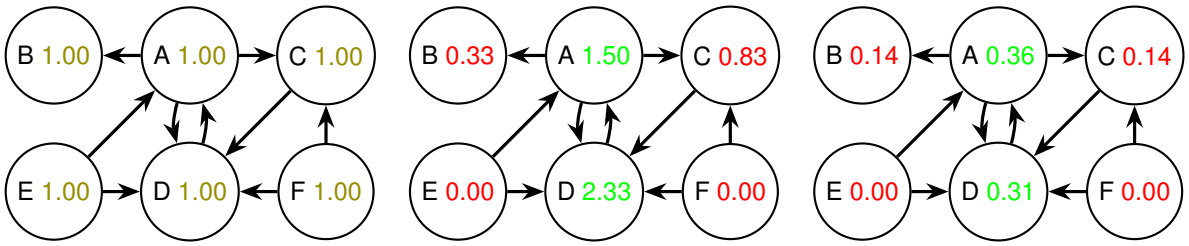


Figure 3: A gauche, situation initiale. Au milieu, situation après une itération. A droite, situation après convergence

2.2 Implémentation en Python

Une implémentation en Python de l'algorithme PageRank est présentée ci-dessous.

```
import numpy as np

sommets = ['A', 'B', 'C', 'D', 'E', 'F']
links = np.array([
    [0, 1/3, 1/3, 1/3, 0, 0],
    [0, 0, 0, 0, 0, 0],
    [0, 0, 0, 1, 0, 0],
    [1, 0, 0, 0, 0, 0],
    [1/2, 0, 0, 1/2, 0, 0],
    [0, 0, 1/2, 1/2, 0, 0]]
)

ranks = np.array([1, 1, 1, 1, 1, 1])
while np.sum(ranks) > 1:
    ranks = np.matmul(ranks, links)

for sommet, rank in zip(sommets, ranks):
    print(sommet +': ' + str(round(rank,2)))
```

3 Réseau de flot

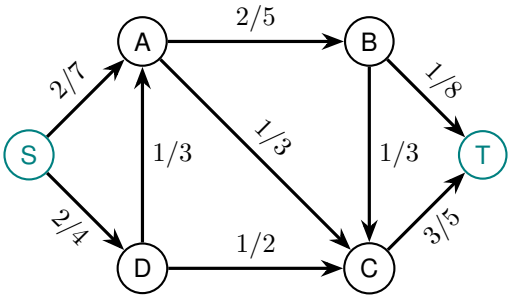


Figure 4: Un réseau de flot

Un réseau de flot est un graphe dirigé où chaque arête contient deux valeurs:

1. Le flot actuel assigné à cette arête
2. Le flot maximal assignable à cette arête

Un réseau de flot contient également 2 sommets spéciaux (généralement nommés S et T):

- Un sommet source qui n'a que des arêtes sortantes
- Un sommet puits qui n'a que des arêtes entrantes

Dans la Figure 4, l'arête C-T a un flot de 3 et une capacité maximale de 5.

Figure 5: Le fleuve Dniepr

Les réseaux de flot sont utilisés pour modéliser des problèmes de flot. Par exemple, dans la Figure 5, le fleuve Dniepr pourrait être modélisé par un réseau de flot. Quel est l'intérêt de modéliser un fleuve par un réseau de flot ? Et bien s'il était possible de savoir le montant d'eau qui peut s'écouler dans ce fleuve, il serait possible de prédire les inondations. Les réseaux de flot sont également utilisés pour modéliser des problèmes de transport, de communication, etc.

3.1 Flot valide

Pour qu'un flot soit valide il faut:

1. Que pour chaque arrête, le flot ne dépasse pas la capacité maximale de l'arête
2. Que pour chaque sommet, sauf la source et le puits, la somme des flots des arêtes entrantes soit égal à la somme des flots des arêtes sortantes.

Valeur de flot

La valeur du flot d'un réseau de flot est définissable de deux manières.

- La somme des flots des arrêtes qui sortent de la source
- La somme des flots des arrêtes qui arrivent au puits.

Dans la Figure 4, la valeur du flot est de 4.

3.2 Flot maximal d'un réseau de flot

Le flot maximal est la valeur maximale que peut prendre le flot d'un réseau de flot.

3.3 Graphes résiduels

Un graphe résiduel est un graphe qui représente les possibilités de modifications du flot, qu'elles soient positives ou négatives.
Dans un graphe résiduel, chaque arête du réseau de flot original est transformée en deux arêtes.

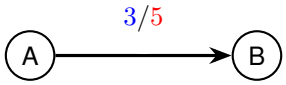


Figure 6: Une arête dans un réseau de flot

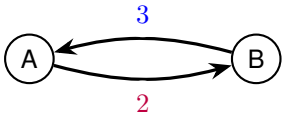


Figure 7: Les deux arrêtes correspondantes dans le graphe résiduel

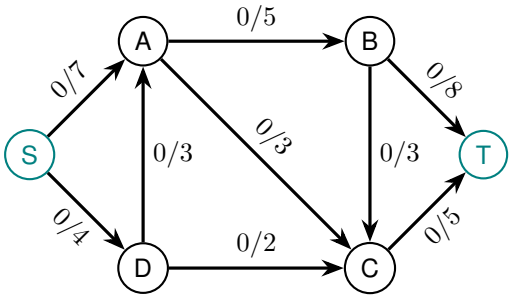
Chaque arête du réseau de flot original sera transformé en deux arêtes dans le graphe résiduel. Dans Figure 6, nous voyons une arête **partant de A vers B** avec:

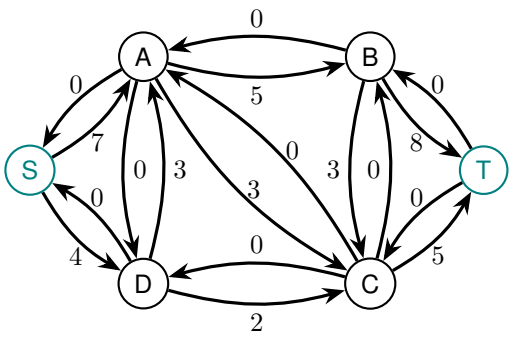
- Un flot actuel de 3
- une capacité maximale de 5

Dans la Figure 7, les deux arrêtes résiduelles représentent :

- La quantité de flot qu'il est encore possible d'ajouter **depuis A vers B** : 5 - 3 = 2
- La quantité de flot qu'il est possible d'enlever **depuis B vers A** : 3

Exemple





Il est possible d'omettre les arrêtes avec un poids de 0 dans le graphe résiduel.

4 Algorithme de Ford-Fulkerson

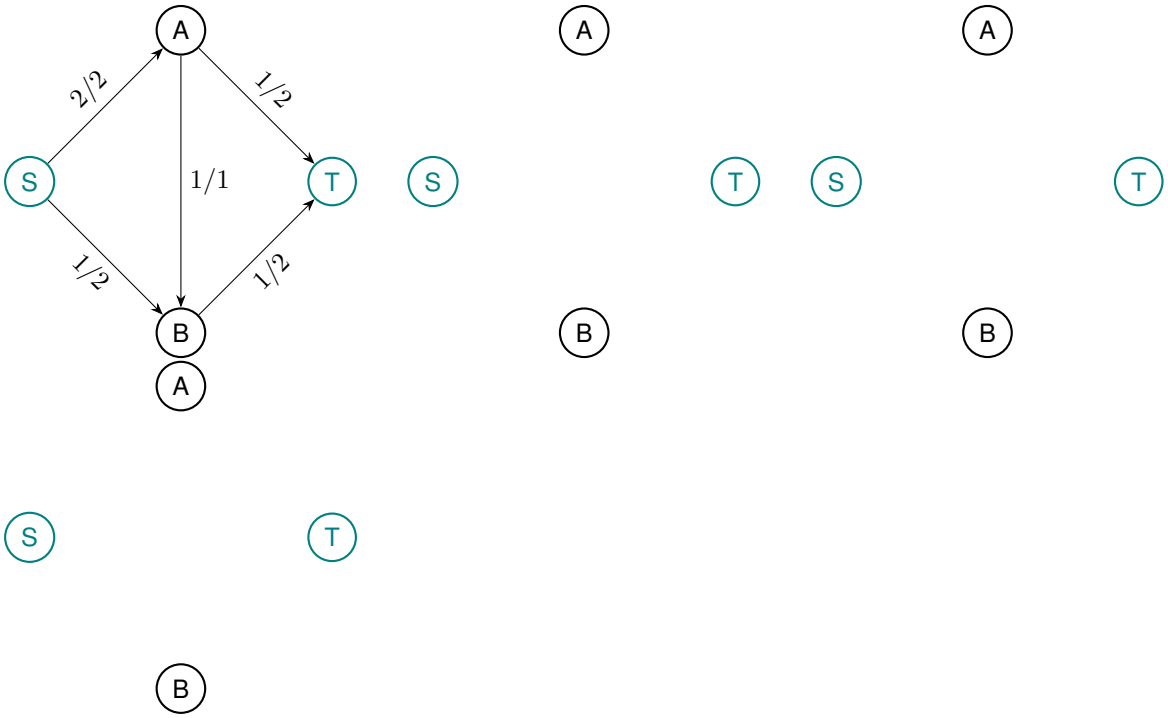
Le but de cet algorithme est de trouver le flot maximal réalisable dans un réseau de flot. L'entrée de l'algorithme sera un réseau de flot F.

L'algorithme de Ford-Fulkerson est un algorithme itératif qui fonctionne de la manière suivante:

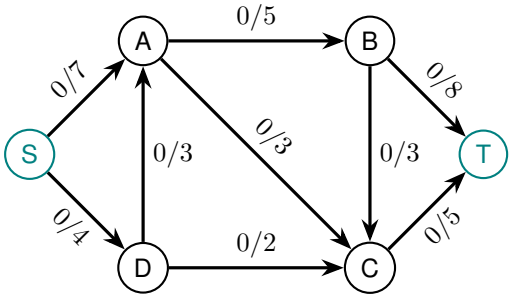
1. Calculer le graphe résiduel du réseau de flot actuel F_{actuel}
2. Trouver un chemin entre la source et le puits dans le graphe résiduel. Sera défini comme valeur V de ce chemin, la valeur minimale des arêtes sur ce chemin. S'il n'existe pas de chemin dans le graphe résiduel, l'algorithme se termine.
3. Mettre à jour F_{actuel} : Pour les arrêtes associées à ce chemin de valeur V dans le graphe résiduel :
   (a) Ajouter V aux flots si l'orientation de l'arête du graphe résiduel est la **même** que dans F_{actuel}.
   (b) Soustraire V aux flots si l'orientation de l'arête du graphe résiduel est **différente** de F_{actuel}.
4. Retourner en 1

! Sans l'étape 3b, il n'est pas possbile de garantir que l'algorithme retournera un flot maximal.

4.1 Exemple 1

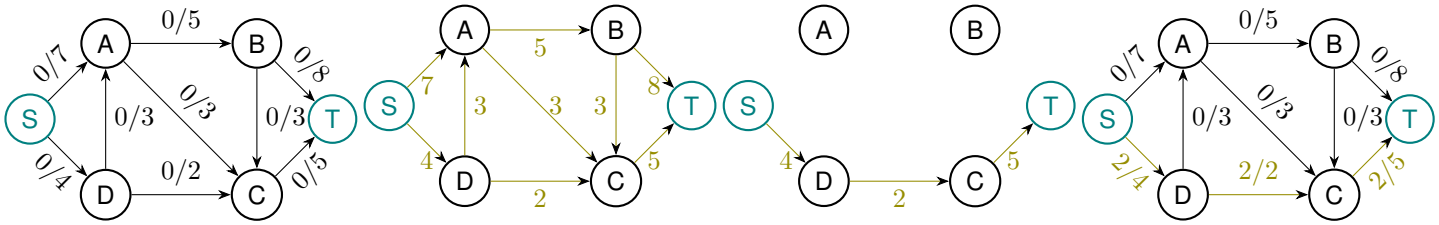


4.2 Exercice

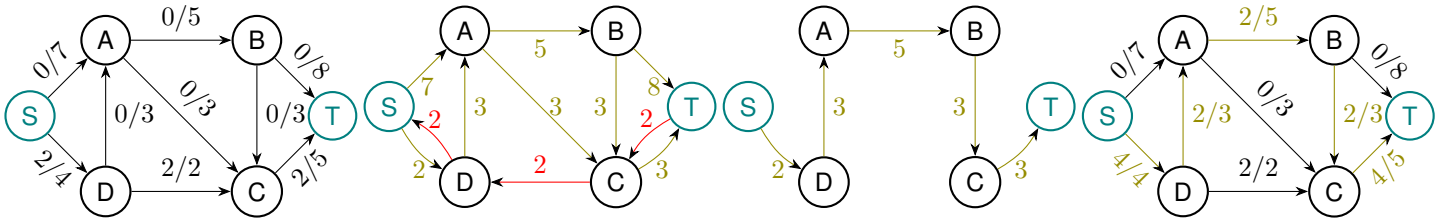


4.3 Corrigé

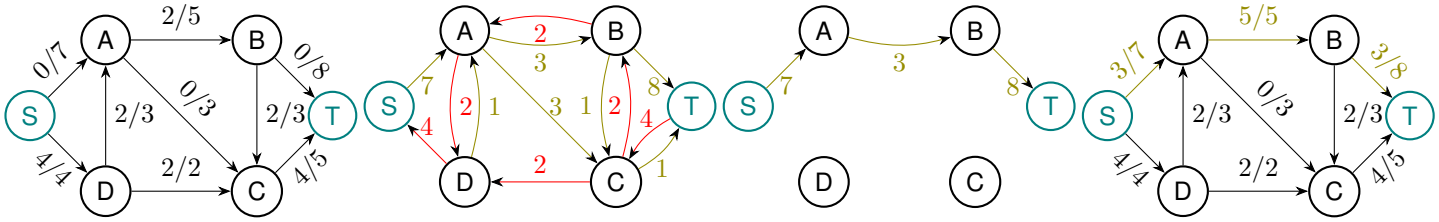
Etape 1



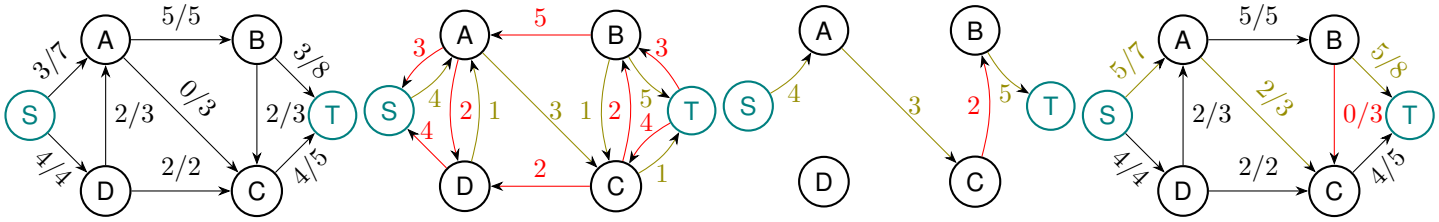
Etape 2



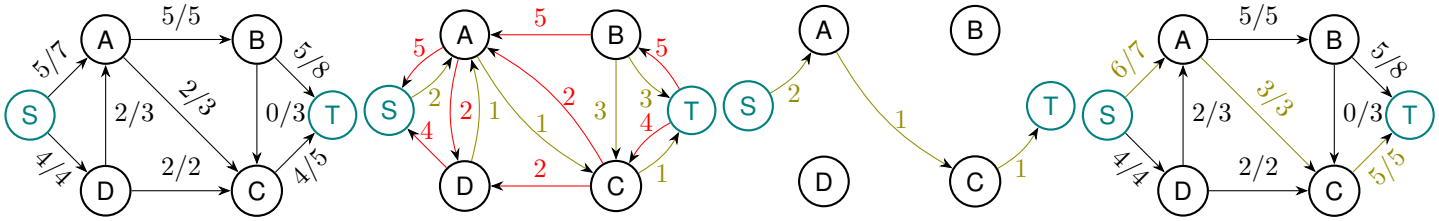
Etape 3



Etape 4



Etape 5



Etape 6