Chapitre 4

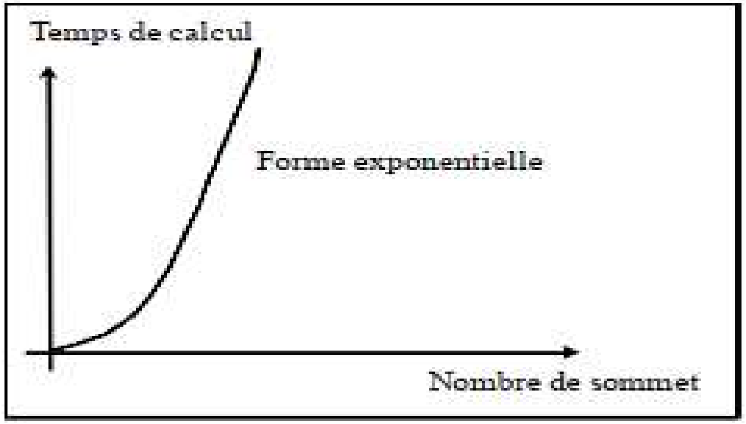
Algorithmes de recherche des plus court/long chemins

Chapitre 4: Algorithmes de recherche des plus court/long chemins

I. Introduction

1) Problématique

Le problème traité dans ce chapitre consiste à étudier des algorithmes astucieux (efficaces) pour résoudre le problème du plus court/long chemin. La complexité de tel algorithme est donnée par la courbe suivante :



Une solution triviale consiste à parcourir tout le graphe, à chercher touts les chemins et à trouver le chemin le plus cours. Cette solution n'est réalisable que pour un petit nombre de sommet.

Chapitre 4: Algorithmes de recherche des plus court/long chemins

2) Définitions

- Soit $G = (X, U)$ un graphe orienté. On définie $d : U \rightarrow R$

  $u \rightarrow d(u)$: fonction coût, longueur, temps,...

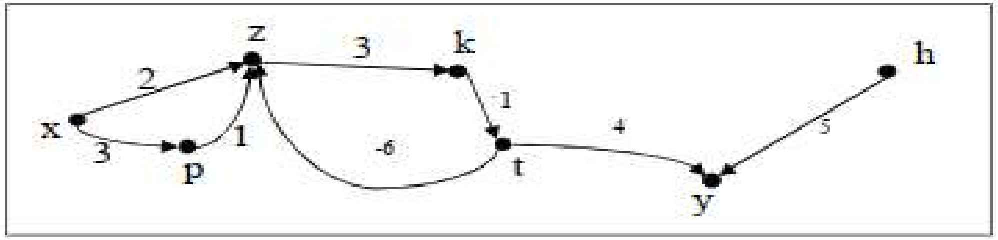
- On appelle réseau : $R = (X, U, d)$.
- Soit un sous ensemble d'arc : $u_1 \subset U$ tel que $d(u_1) = \sum_{u \in U_1}^{n} d(u)$
- On appelle la longueur d'un chemin $L(c) = \sum_{u \in C} d(u)$

Attention : la longueur n'est plus le nombre d'arc qui constitue le chemin

Chapitre 4: Algorithmes de recherche des plus court/long chemins

Exemple :

Soit le réseau R suivant :



Réseau R

- Déterminer le plus court chemin entre x et y ? (il n'y en à pas)
- Soit les chemin C_1, C_2 et C_3 du sommet x vers le sommet y. Calculer la longueur de ces trois chemins ?

C_1 = x z k t y → L(C1) =2 + 3 + 1 + 4 =10.

C_2 = x z k t z k t y → L(C2) = 2 + 3 + 1 -6 + 3 +1 + 4 = 8.

C_3 = x z k t z k t z k t y → L(C3) = 2 + 3 + 1 -6 +3 + 1 -6 + 3 + 1 + 4 = 6.

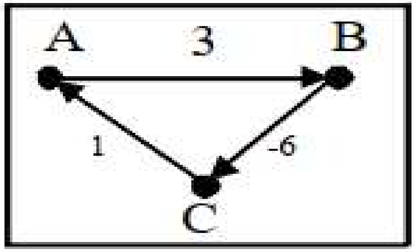
➔ Existence d'un circuit absorbant ! → Il n'existe pas de plus court chemin (Cf. ϕI.4).

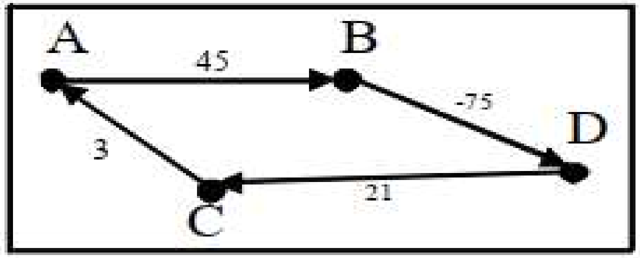
Chapitre 4: Algorithmes de recherche des plus court/long chemins

3) Circuit absorbant

Un circuit C est appelé circuit absorbant si et seulement si : $\sum_{u \in C} d(u) < 0$.

Exemple :





Théorème :

Une condition nécessaire et suffisante (CNS) pour que le problème du plus court chemin entre un sommet source (s) et un sommet destination (p) ait une solution est que :

- L'ensemble des sommets y qui sont à la fois des descendants de (s) et des ascendants de (p) soit différent de l'ensemble vide ; c.à.d. il $\exists$ au moins un chemin de (s) à (p).

Chapitre 4: Algorithmes de recherche des plus court/long chemins

- Il n'existe pas de circuits absorbants dans le sous réseau construit sur y.
- Si les deux conditions précédentes sont satisfaites alors le problème du plus court chemin entre (s) et (p) est équivalent au problème du plus court chemin élémentaire (passe par le sommet une seule fois)

Attention : L'existence d'un circuit absorbant, qui n'intervient dans aucun chemin entre (s) et (p), ne ni pas la préséance d'un court chemin entre (s) et (p).

Chapitre 4: Algorithmes de recherche des plus court/long chemins

4) Situations possibles

Plusieurs situations sont possibles:

a) Il n'existe pas de chemin allant d'un sommet (s) à un sommet (p). Exemple : entre x et h

b) Il existe un chemin entre un sommet (s) et un sommet (p), mais il n'existe pas de chemin de longueur minimum (l'ensemble des chemins joignant (s) à (p) n'est pas bornée inférieurement : existence d'un circuit absorbant).

c) Il existe un chemin allant d'un sommet (s) à un sommet (p) tel que dans l'ensemble des chemins de (s) à (p) il en a un de longueur minimum.

Remarque :

Le problème peut être posé d'une autre manière :

- Trouver un chemin de longueur minimum joignant un sommet (s) à un sommet destination (p).
- A tout $x \in X$ associer un chemin de longueur minimum joignant un sommet (s) à tout $x \in X$.

Chapitre 4: Algorithmes de recherche des plus court/long chemins

5) Racine, Potentiels et Tensions

a) Racine

Dans un graphe orienté G, un sommet $x_0 \in X$ est une racine s'il existe dans G un chemin de x_0 a x pour tout $x \in X$.

b) Potentiel

Soit R = (X, U, d) un réseau sans circuit absorbant et admettant (s) comme racine.

Pour tout $x \in X$, on note $\pi(x)$, le potentiel de x qui est définit par la longueur du plus court chemin de (s) à x.

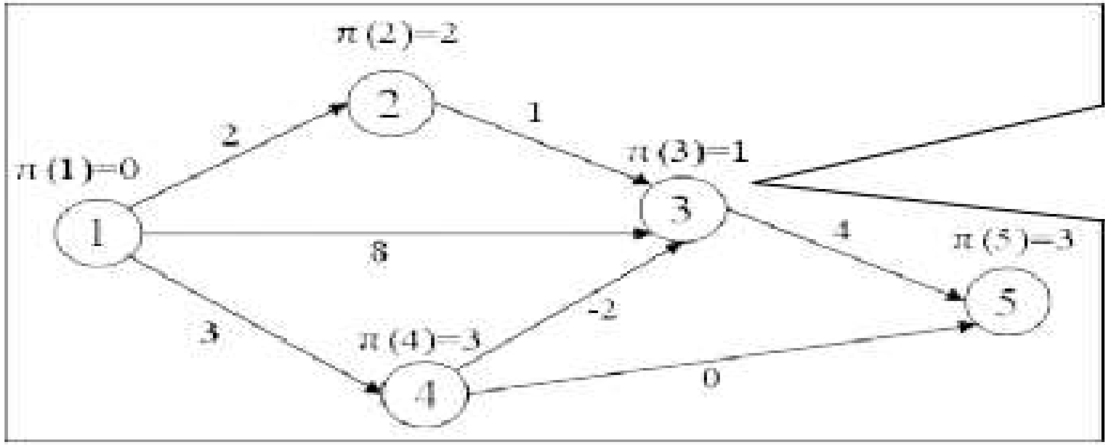
Chapitre 4: Algorithmes de recherche des plus court/long chemins

Remarque : Le sommet 1 est le seul sommet qui peut être une racine !

Exemple :

Soit le réseau R1 suivant avec le sommet 1 comme racine.

Déterminer les potentiels des sommets du réseau.



Détermination du potentiel du sommet (3) : voir tous les arcs incidents puis faire la somme de la valeur de l'arc + le potentiel du sommet précédant puis prendre la valeur minimale pour le potentiel du sommet (3).

$\pi(2)+1 = 3$

$\pi(1)+8 = 8$

$\pi(4)-2 = \underline{\mathbf{1}}$

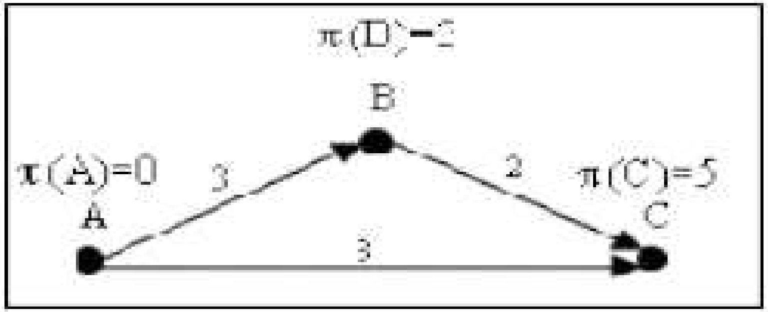
Chapitre 4: Algorithmes de recherche des plus court/long chemins

<u>Remarque :</u>

→ On à toujours :

- $\pi\,(s) = 0$.
- $\pi\,(T(u)) - \pi\,(I(u)) \leq d(u)$

<u>Exemple</u> :



$\pi\,(T(u)) - \pi\,(I(u)) \leq d(u)$ avec $u = A \rightarrow C$

$\pi\,(C) - \pi\,(A) \leq d(A \rightarrow C)$

$5 - 0 \leq 8$

<u>Remarque</u> : d(u), la valeur de l'arc, est plus grande que le plus court chemin déterminé avec les potentiels

Chapitre 4: Algorithmes de recherche des plus court/long chemins

c) Tension

On appelle tension $t : U \rightarrow R$

$$u \rightarrow t(u) = \pi\ (T(u)) - \pi\ (I(u)) \leq d(u)$$

<u>Exemple</u> :

Soit le réseau R1 (exemple du potentiel page 4). Le potentiel du sommet 3 est égale : $\pi\ (3)=1$.

Les chemins entre le sommet 1 et 3 sont :

$$\left.\begin{array}{l} (C1) : 1, 2, 3 \rightarrow L(C1) = 3 \\ (C2) : 1, 3 \rightarrow L(C2) = 8 \\ (C3) : 1, 4, 3 \rightarrow L(C3) = 1 \end{array}\right\} \leq \pi\ (3) = 1$$

Chapitre 4: Algorithmes de recherche des plus court/long chemins

II. Algorithme de recherche du plus court chemin

Soit un réseau R (X, U, d) / |X| = n et |U| = m.

Objectif :

- Chercher un chemin de valeur minimale de $x_0 \in X$ vers un sommet destination.
- Chercher les chemins de valeurs minimales de $x_0 \in X$ (x_0 sommet source) vers les autres sommets du réseau.

Plusieurs algorithmes permettent d'atteindre ces objectifs :

- Algorithme de Ford
- Algorithme de Bellman
- Algorithme de Dijkstra

Chapitre 4: Algorithmes de recherche des plus court/long chemins

1) Algorithme de Ford (recherche ***d'un* chemin** de valeur minimale) (réseau sans circuit négatif)

Etape 1 :

Numéroter les sommets du réseau dans un ordre quelconque à l'exception du sommet d'origine x_0 et du sommet destination x_{n-1}

Le sommet destination est X_{n-1} car il y a n sommet : $|X| = n$ et on commence à partir de x_0

$x_0\ x_1\ x_2\ \ldots\ldots\ldots\ldots\ldots\ldots\ x_{n-2}\ x_{n-1}$

origine ... destination

Etape 2 : Initialisation :

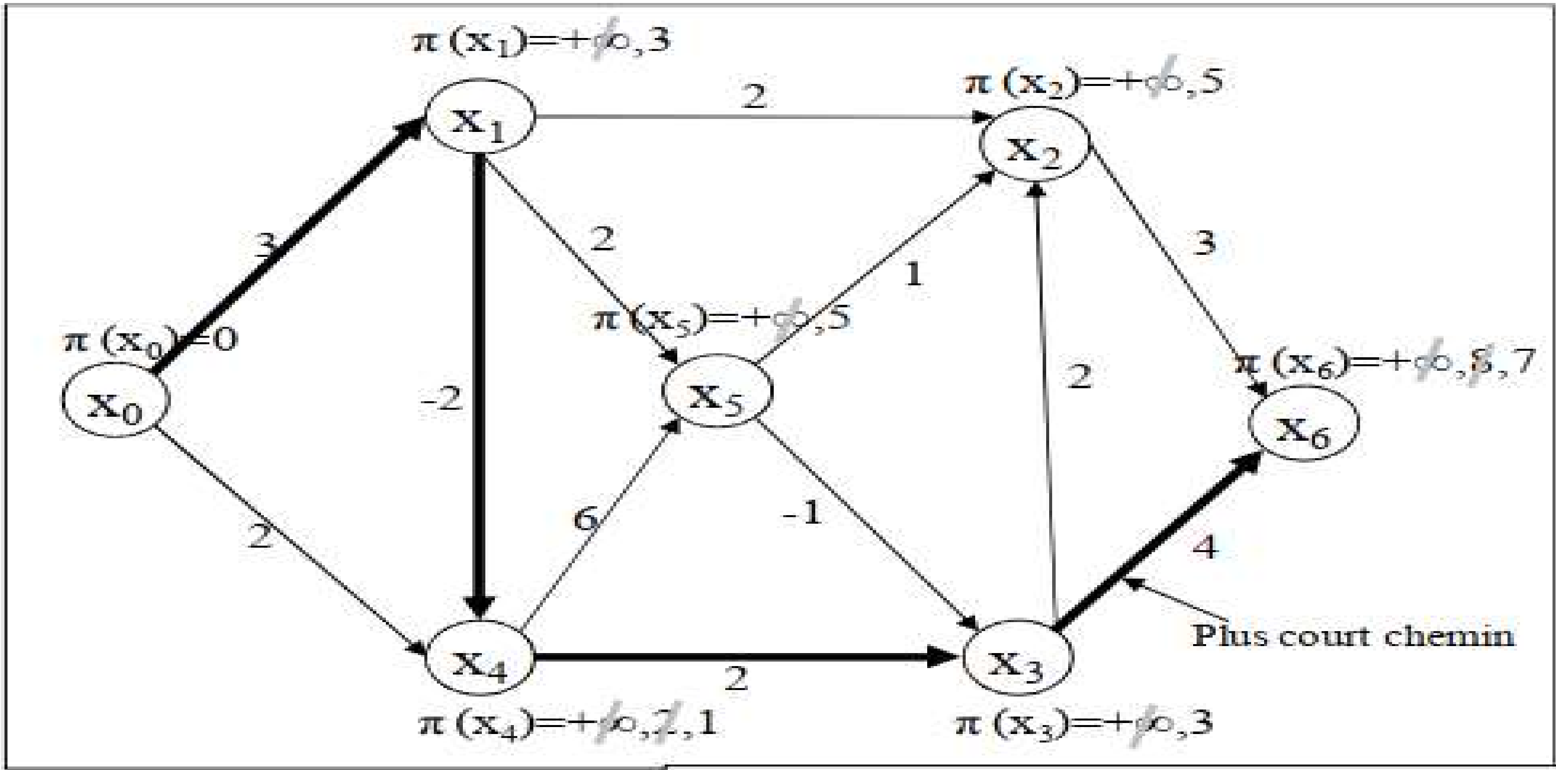
- Mettre $\pi\ (x_0) = 0$; mettre $\pi\ (x_j)$ à $+\infty$ / $j = 1, \ldots, n-1$.
- $i=0$.

Chapitre 4: Algorithmes de recherche des plus court/long chemins

Etape 3 :

```
Tant que i < n-1
{
        Pour tout u / I(u) = xi faire
        {
           Si π (xj) > π (xi) + d(u) alors
               {
                 π (xj)= π (xi) +d(u)
                 Si j<i alors
                    {
                      i = j ;
                      aller à l'étape 3
                    }
               }
        }
   i = i + 1 ;
}
Fin
```

Chapitre 4: Algorithmes de recherche des plus court/long chemins



Etape 1 : Numérotation des sommets ;

Attention : La numérotation des sommets d'une façon différente peut changer le nombre d'itération !

Etape 2 : Mettre $\pi(x_0) = 0$ et $\pi(x_j) = +\infty$ / j= 1,.., n-1 (même pour le sommet destination)

Chapitre 4: Algorithmes de recherche des plus court/long chemins

Etape 3 :

i

j

$I(x_0, x_1) = x_0$

$i = 0 \rightarrow x_0$

$x_i \,/\, i = 0$

$$\begin{cases} \left.\begin{array}{l} (x_0, x_1) : \pi(x_j) > \pi(x_i) + d(u) \text{ c.à.d. } \pi(x_1) > \pi(x_0) + d(x_0, x_1) \\ \qquad\qquad\qquad\qquad\qquad\qquad +\infty \quad > \quad 0 \quad + \quad 3 \end{array}\right\} \rightarrow \pi(x_1) = 3 \quad 1 \not< 0 \\ \left.\begin{array}{l} (x_0, x_4) : \pi(x4) > \pi(x_0) + d(x_0, x_4) \\ \qquad\quad +\infty \quad > \quad 0 \quad + \quad 2 \end{array}\right\} \rightarrow \pi(x_4) = \boxed{2} \quad 4 \not< 0 \end{cases}$$

Inferieur non vérifié

<u>Remarque</u> : ∀ u : ne pas oublier le deuxième arc : (x_0,x_4)

Chapitre 4: Algorithmes de recherche des plus court/long chemins

i $\quad$ j $\quad$ $I(x_1, x_2) = x_1$

$\mathbf{i = 1} \rightarrow x_1$ $\quad$ $x_i \,/\, i = 1$

$$
\left\{
\begin{array}{lll}
(x_1, x_2) : \pi(x_2) > \pi(x_1) + d(x_1, x_2) & \rightarrow & \pi(x_2) = 5 \\
\qquad\qquad +\infty > 3 + 2 & & 2 \leftarrow 1 \\
(x_1, x_5) : +\infty > 3 + 2 & \rightarrow & \pi(x_5) = 5 \\
 & & 5 \leftarrow 1 \\
(x_1, x_4) : \pi(x_4) > \pi(x_1) + d(x_1, x_4) & \rightarrow & \pi(x_4) = 1 \\
\qquad\qquad 2 > 3 + -2 & & 4 \leftarrow 1
\end{array}
\right.
$$

$\mathbf{i = 2} \rightarrow x_2$

$$
\left\{
\begin{array}{lll}
(x_2, x_6) : \pi(x_6) > \pi(x_2) + d(x_2, x_6) & \rightarrow & \pi(x_6) = 8 \\
\qquad\qquad +\infty > 5 + 3 & & 6 \leftarrow 2
\end{array}
\right.
$$

Chapitre 4: Algorithmes de recherche des plus court/long chemins

i = 3 $\rightarrow x_3$
- (x_3, x_6) : $8 > +\infty \; +4$ $\rightarrow$ RAS
- (x_3, x_2) : $5 > +\infty \; +2$ $\rightarrow$ RAS (on ne vérifie même pas si $j<i$ (cf Algo))

i = 4 $\rightarrow x_4$
- (x_4, x_3) : $+\infty > 1 + 2$ $\rightarrow \pi(x_3) = 3$; $j<i \rightarrow 3<4$ $\boxed{\textbf{oui}}$ $\rightarrow i = 3$
- (x_4, x_5) : $\boxed{\text{(Pas la peine de le faire, on y reviens après pour i=4)}}$

i = 3 $\rightarrow x_3$
- (x_3, x_6) : $8 > 3 + 4$ $\rightarrow \pi(x_6) = 7$ $\rightarrow 6 \not< 3$
- (x_3, x_2) : $5 > 3 + 2$ $\rightarrow$ RAS

i = 4 $\rightarrow x_4$
- (x_4, x_3) : $3 > 1 + 2$ $\rightarrow$ RAS
- (x_4, x_5) : $5 > 1 + 5$ $\rightarrow$ RAS

i = 5 $\rightarrow x_5$
- (x_5, x_2) : $5 > 5 + 1$ $\rightarrow$ RAS
- (x_5, x_3) : $3 > 5 - 1$ $\rightarrow$ RAS

i = 6 → or $6 < n-1$ → $6 < 6$ faux → fin de l'algorithme.

On obtient le plus cours chemin en faisant un retour arrière à partir du sommet x_6 C.A.D

$\pi(x_6) = 7$: $\boxed{7-\underbrace{3}_{d(x_2,x_6)} = \underbrace{5}_{\pi(x_2)}}$ ou $\boxed{7-\underbrace{4}_{d(x_3,x_6)} = \underbrace{3}_{\pi(x_3)}}$? → on choisie $7-4 = 3$ → l'arc (x_3,x_6) fais partie du plus cour chemin. On répète la procédure jusqu'à arriver au sommet x_0 (la source).

→ **Le plus court chemin de x_0 à x_6 est x_0, x_1, x_4, x_3, x_6**

Chapitre 4: Algorithmes de recherche des plus court/long chemins

2) Algorithme de Bellman (réseau sans circuit)

Avec l'algorithme de Bellman, on ne calcule la plus courte distance de (s) à x que si on à déjà calculer les plus courtes distances de (s) à tous les prédécesseurs de x.

On désigne par "S" l'ensemble des sommets pour les quels on à déjà calculer leurs plus courte distance.

<u>Algorithme</u>

<u>Etape 0 :</u>

Poser S= {s} ; $\pi(s) = 0$;

<u>Etape 1 :</u>

Tan que ((il $\exists$ x $\notin$ S) ET (tous les prédécesseurs de x $\in$ S)) **faire**

{

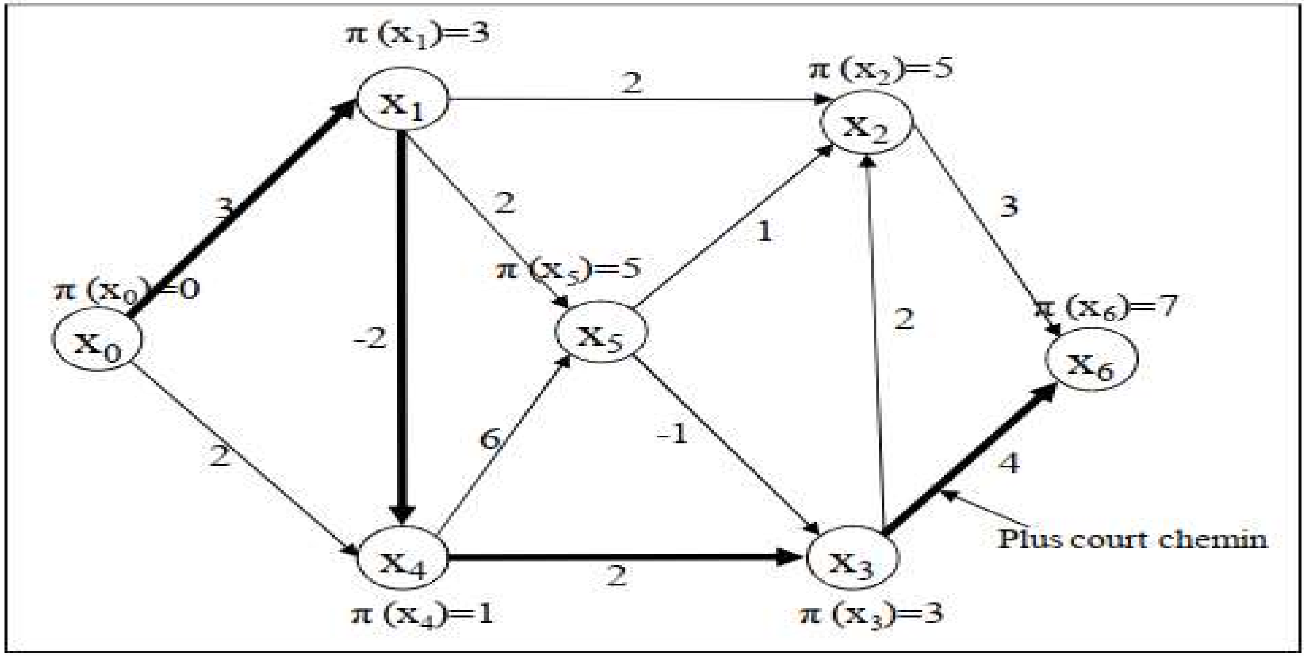
$\pi(x)$= Min[π(I(u) + d(u)] avec T(u) = x

S =S $\cup$ {x}

}

Chapitre 4: Algorithmes de recherche des plus court/long chemins

Exemple 1 :



Etape 0 :

$S_1=\{x_0\}$; $\pi(x_0)=0$.

Chapitre 4: Algorithmes de recherche des plus court/long chemins

Etape 1 :

- Il ∃ le sommet $x_1 \notin S$ tel que tous les prédécesseurs de x_1 $\{x_0\}$ sont dans S.

  $\pi(x_1) = \text{Min}\,[\pi(x_0) + 3] = 3$

  $S_2 = \{x_0, x_1\}$

- Il ∃ le sommet $x_4 \notin S$ tel que tous les prédécesseurs de x_4 $\{x_0\}$ sont dans S.

  $$\pi(x_4) = \text{Min}\begin{cases}[\pi(x_1) + (-2)] = 1 \\ [\pi(x_0) + 2] \quad = 2\end{cases} \rightarrow \text{Min}(1,2) = 1 \rightarrow \pi(x_4) = 1.$$

  $S_3 = \{x_0, x_1, x_4\}$

- Il ∃ le sommet $x_5 \notin S$ tel que tous les prédécesseurs de x_5 $\{x_1, x_4\}$ sont dans S.

  $$\pi(x_5) = \text{Min}\begin{cases}[\pi(x_1) + 2] = 3+2 = 5 \\ [\pi(x_4) + 6] \ = 1+6=7\end{cases} \rightarrow \text{Min}(5,7) = 5 \rightarrow \pi(x_5) = 5.$$

  $S_4 = \{x_0, x_1, x_4, x_5\}$

- Il ∃ le sommet $x_3 \notin S$ tel que tous les prédécesseurs de x_3 $\{x_4, x_5\}$ sont dans S.

  $$\pi(x_3) = \text{Min}\begin{cases}[\pi(x_5) + (-1)] = 5-1 = 4 \\ [\pi(x_4) + 2] \ = 1+2=3\end{cases} \rightarrow \text{Min}(3,4) = 3 \rightarrow \pi(x_3) = 3.$$

  $S_5 = \{x_0, x_1, x_4, x_5, x_3\}$

- Il ∃ le sommet $x_2 \notin S$ tel que tous les prédécesseurs de x_2 $\{x_1, x_3, x_5\}$ sont dans S.

$$\pi(x_2) = \text{Min}\begin{cases}[\pi(x_1) + 2] = 3+2=5 \\ [\pi(x_3) + 2] = 3+2=5 \\ [\pi(x_5) + 1] = 5+1=6\end{cases} \rightarrow \text{Min}(5,5,6) = 5 \rightarrow \pi(x_2) = 5$$

$S_5 = \{x_0, x_1, x_4, x_5, x_3, x_2\}$

- Il ∃ le sommet $x_6 \notin S$ tel que tous les prédécesseurs de x_6 $\{x_2, x_3\}$ sont dans S.

$$\pi(x_6) = \text{Min}\begin{cases}[\pi(x_2) + 3] = 5+3=8 \\ [\pi(x_3) + 4] = 3+4=7\end{cases} \rightarrow \text{Min}(7,8) = 7 \rightarrow \pi(x_6) = 7$$

$S_7 = \{x_0, x_1, x_4, x_5, x_3, x_2, x_6\}$

- Il ∄ de sommet $x_i \notin S$ tel que tous les prédécesseurs de x_i sont dans S.
- FIN

Chapitre 4: Algorithmes de recherche des plus court/long chemins

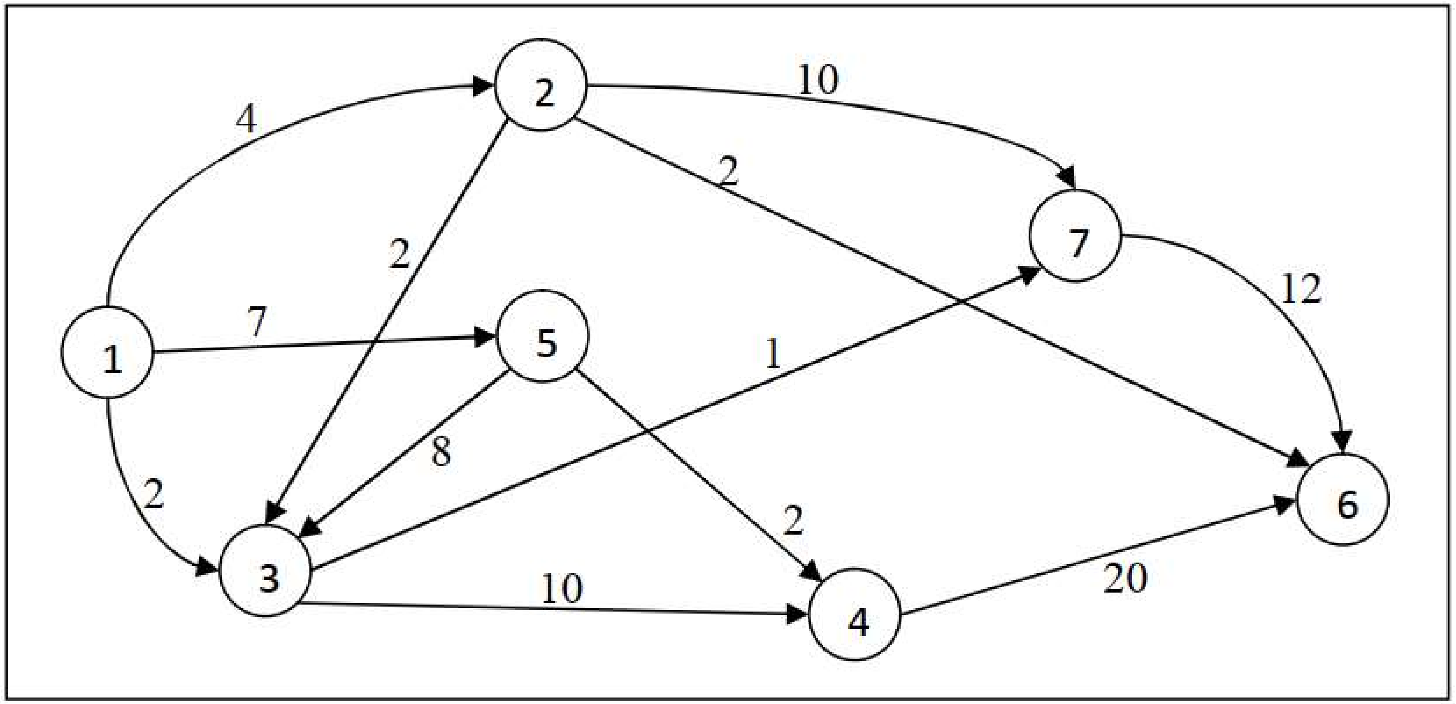
Remarque:

- Si le nombre de sommet $\notin S > 1$, on choisit un sommet au hasard

Exemple 2:

Soit le graphe G suivant. En prenant le sommet (1) comme racine, déterminer avec l'algorithme de Bellman les plus courtes distances vers les autres sommets.

Chapitre 4: Algorithmes de recherche des plus court/long chemins



Question : Est ce que le graphe contient un circuit ?

Avec quel sommet on commence ?

Chapitre 4: Algorithmes de recherche des plus court/long chemins

→ Avant d'exécuter l'algorithme de Bellman il convient **d'ordonnancer le graphe en niveau** afin de **faciliter** l'exécution de l'algorithme.

<u>Détermination des niveaux des sommets d'un graphe (sans circuit)</u>

Déterminons le tableau des prédécesseurs.

x	P(x) : prédécesseurs
1	-
2	1
3	1, 5, 2
4	5, 3
5	1
6	2, 7, 4
7	2, 3

Chapitre 4: Algorithmes de recherche des plus court/long chemins

Algorithme

- Soit N_0 l'ensemble des sommets de niveau 0 (sans prédécesseurs)
- On prend i =0.
- On commence par barrer les sommets de niveau i partout où ils figurent dans la colonne p(x).
- Si une ligne à tous ses sommets barrés, le sommet correspondant est de niveau i+1. On répète la procédure en incrémentant i de 1 jusqu'à ce que touts les éléments soit barrés.
- S'il reste des sommets non barrés, alors le graphe contient un circuit.

Chapitre 4: Algorithmes de recherche des plus court/long chemins

Exemple

x	P(x) : prédécesseurs
~~1~~	-
~~2~~	~~1~~
~~3~~	~~1~~, ~~5~~, ~~2~~
~~4~~	~~5~~, ~~3~~
~~5~~	~~1~~
~~6~~	~~2~~, ~~7~~, ~~4~~
~~7~~	~~2~~, ~~3~~

$N_0 = \{1\}$

$N_1 = \{2, 5\}$

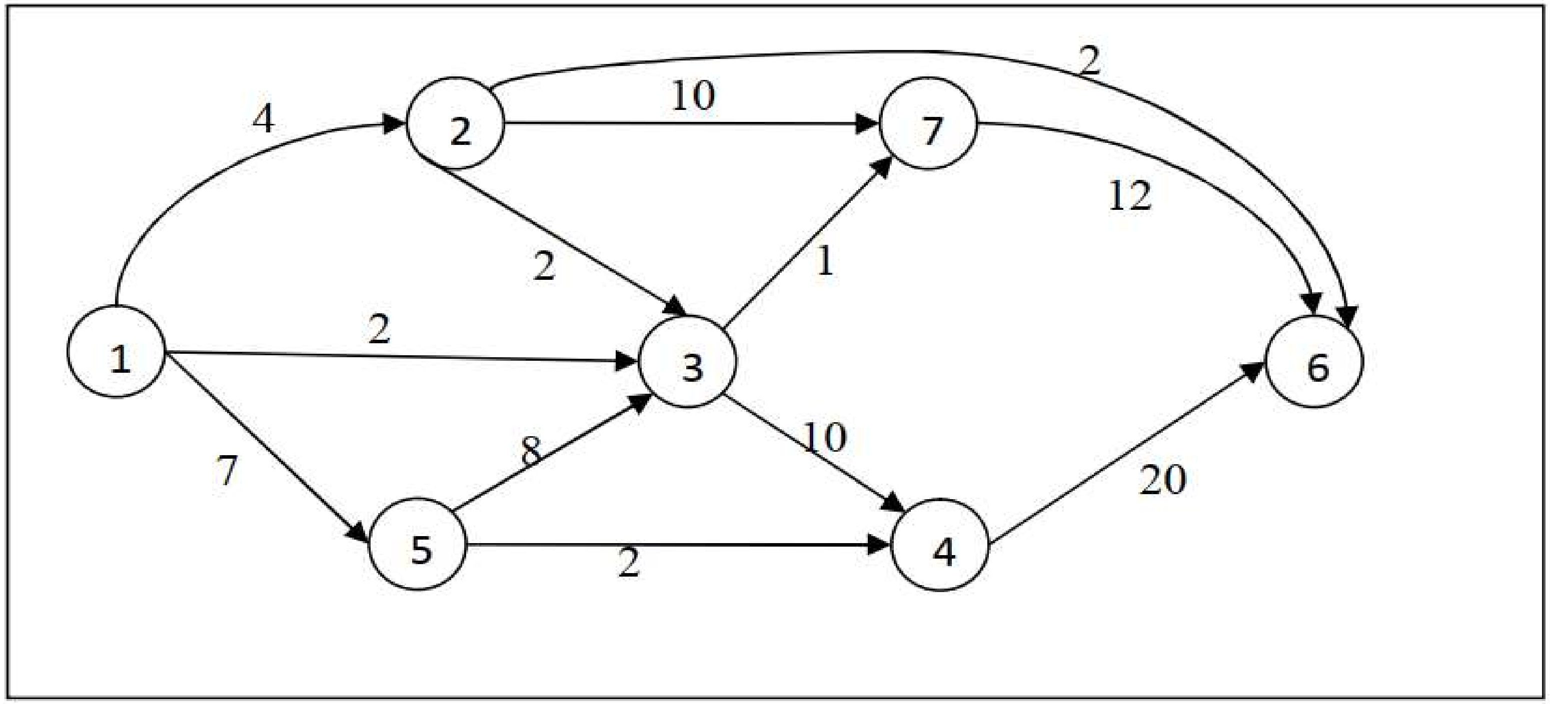
$N_2 = \{3\}$

$N_3 = \{7, 4\}$

$N_4 = \{6\}$

Chapitre 4: Algorithmes de recherche des plus court/long chemins

Nous obtenons alors le graphe suivant ordonnancé en niveau



Chapitre 4: Algorithmes de recherche des plus court/long chemins

Remarque :

- Il ne reste pas des sommets non barrés, alors le graphe ne contient pas de circuit.

➔ Si on rajoute l'arc x_4-x_2, le graphe n'est plus ordonnancable en niveau → le graphe contient le circuit x_2-x_3-x_4-x_2

3) Algorithme de Dijkstra

Cet algorithme ne peut être appliqué que si tous les arcs ont des valeurs positives. c.à.d:

$$\forall\ u \in U : d(u) \geq 0$$

Chapitre 4: Algorithmes de recherche des plus court/long chemins

a) objectif

L'application de l'algorithme de Dijkstra permet de chercher les plus courtes distances de proche en proche. On désigne par S l'ensemble des sommets pour les quels on à déjà trouvés leurs plus courts chemins. Cet ensemble augmente d'une unité à chaque itération.

→Les sommets s'introduisent dans S dans l'ordre de leurs plus courtes distances.

<u>Exemple:</u>

Si $S = \{s, x_1, x_2, x_3, x_4, \ldots\ldots, x_{n-1}\}$ alors $\pi(x_s) \leq \pi(x_1) \leq \pi(x_2) \leq \pi(x_3) \leq \ldots\ldots\ldots \leq \pi(x_{n-1})$

Chapitre 4: Algorithmes de recherche des plus court/long chemins

b) Algorithme

<u>Étape 0 :</u>

$S = \{s\}$

$\pi(x_s) = 0$

$x_p = s$ (sommet pivot)

$\pi(x) = +\infty \; \forall \, x \neq s$

<u>Etape 1 :</u>

<u>Tant que</u> $S \neq X$ et $\pi(x_p) < +\infty$

$\forall \, u \in U \, / \, I(u) = x_p$ faire

$x = T(u)$

si $(\pi(x) > \pi(x_p) + d(u))$ alors

$\pi(x) = \pi(x_p) + d(u)$

choisir $x \notin S \, / \, \pi(x) = \text{Min } \pi(y)$ avec $y \notin S$

$x_p = x$

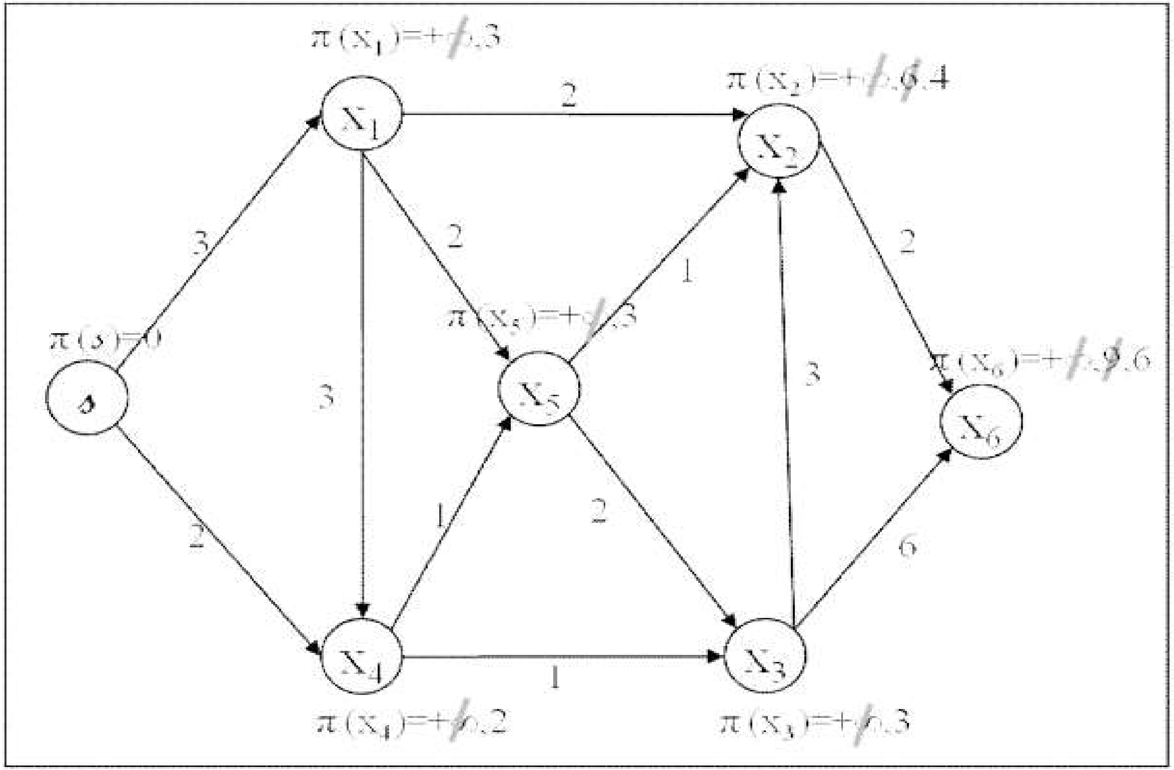
$S = S \cup \{x\}$

Aller à l'étape 1

Fin TQ

Chapitre 4: Algorithmes de recherche des plus court/long chemins

c) Exemple



Chapitre 4: Algorithmes de recherche des plus court/long chemins

Étape 0 :

$S=\{s\}$; $\pi(s)=0$; $\pi(x_i)=+\infty\ \forall\, i\ (1\rightarrow 6)$; $x_p = s$

Étape 1 :

Il existe un chemin entre s et x_1 de longueur 3

- $x_p = s$ $\begin{cases} -\ (s, x_1) : \pi(x_1) > \pi(s) + 3 \rightarrow +\infty > 0 + 3 \rightarrow \pi(x_1) = 3 \\ -\ (s, x_4) : \pi(x_4) > \pi(s) + 2 \rightarrow +\infty > 0 + 2 \rightarrow \pi(x_4) = 2 \end{cases}$

  $x_p = x_4$

  Le chois du **pivot** est tel que : c'est le sommet qui à **le plus petit potentiel** et qui $\notin$ S. on choisie entre x_1 et x_4 car les autres sommets ont un potentiel $= +\infty$

  $S = \{s, x_4\}$

- $x_p = x_4$ $\begin{cases} (x_4, x_3) : \pi(x_3) > \pi(x_4) + 1 \rightarrow +\infty > 2 + 1 \rightarrow \pi(x_3) = 3 \\ -\ (x_4, x_5) : \pi(x_5) > \pi(x_4) + 1 \rightarrow +\infty > 2 + 1 \rightarrow \pi(x_5) = 3 \end{cases}$

  $x_p = x_3$ ou x_1 ou x_5 : on choisie un

  $S = \{s, x_4, x_3\}$

Chapitre 4: Algorithmes de recherche des plus court/long chemins

- $x_p = x_3 \begin{cases} (x_3, x_6) : \pi(x_6) > \pi(x_3) + 6 \rightarrow +\infty > 3 + 6 \rightarrow \pi(x_6) = 9 \\ -\ (x_3, x_2) : \pi(x_2) > \pi(x_3) + 3 \rightarrow +\infty > 3 + 3 \rightarrow \pi(x_2) = 6 \end{cases}$

  $x_p = x_5$ ou x_1

  $S = \{s, x_4, x_3, x_5\}$

- $x_p = x_5 \begin{cases} (x_5, x_2) : \pi(x_2) > \pi(x_5) + 1 \rightarrow +\infty > 3 + 1 \rightarrow \pi(x_2) = 4 \\ -\ (x_5, x_3) : \pi(x_3) > \pi(x_5) + 2 \rightarrow 3 \not> 3 + 2 \rightarrow \text{RAS} \qquad \rightarrow \pi(x_3) = 3 \end{cases}$

  $x_p = x_1$

  $S = \{s, x_4, x_3, x_5, x_1\}$

- $x_p = x_1 \begin{cases} (x_1, x_2) : \pi(x_2) > \pi(x_1) + 2 \rightarrow 4 > 3 + 2\ \text{RAS} \qquad \rightarrow \pi(x_2) = 4 \\ -\ (x_1, x_5) : \pi(x_5) > \pi(x_1) + 2 \rightarrow 3 \not> 3 + 2 \rightarrow \text{RAS} \end{cases}$

  $x_p = x_2$

  $S = \{s, x_4, x_3, x_5, x_1, x_2\}$

Chapitre 4: Algorithmes de recherche des plus court/long chemins

- $x_p = x_2 \{$- $(x_2, x_6) : \pi (x_6) > \pi (x_2) + 2 \rightarrow 9 > 4 + 2 \rightarrow \pi (x_6) = 6$

  $x_p = x_6$

  $S = \{s, x_4, x_3, x_5, x_1, x_2, x_6\}$
- Fin

Remarque:

Nous avons: $\pi (s) \leq \pi (x_4) \leq \pi (x_3) \leq \pi (x_5) \leq \pi (x_1) \leq \pi (x_2) \leq \pi (x_6)$