Chapitre III - Chemin de longueur minimale ou maximale

07/01/2019 (15)

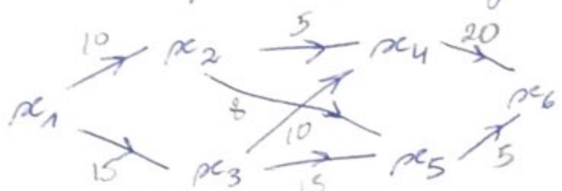
Supposons q dns un graphe orienté, on décide d'attribuer à chaque arc, une longueur + ou non. Il est alors naturel de définir la longueur d'1 chemin quelconque comme la somme ds longueurs des arcs qui le composent. Un problème fondamental qui se pose fréquemment dans les applications est celui de la recherche de chemin de longueur minimale ou maximale.

L'objet de ce chapitre est de présenter les ≠ algos simples permettant d'obtenir ces chemins

Ces algorithmes seront utilisés dans la suite pour la recherche de flots et ordonnancement optimaux.

I - Formulation du problème

Soit $G=(X,U)$ un graphe orienté sans boucles comportant n sommets, à ts arcs $(x_i, x_j) \in U \longleftrightarrow l_{ij}$ (un nbre réel) appelé longueur de l'arc (x_i, x_j). La longueur d'un chemin alpha quelconque noté $l(\alpha)$ est alors définie comme la somme des longueurs des arcs qui le composent.



$\alpha_1 = (x_1, x_2, x_4, x_6) \Rightarrow l(\alpha_1) = 10+5+20 = 35$

$\alpha_2 = (x_1, x_2, x_5, x_6) \Rightarrow l(\alpha_2) = 10+8+5 = 23$

$\alpha_3 = (x_1, x_3, x_4, x_6) \Rightarrow l(\alpha_3) = 15+10+20 = 45$

$\alpha_4 = (x_1, x_3, x_5, x_6) \Rightarrow l(\alpha_4) = 15+15+5 = 35$

Un chemin joignant un sommet x_a à x_b est dit de longueur min ou max s'il minimise ou maximise cette longueur $l(\alpha)$ dans l'ens. de ts les chemins $\alpha(x_a, x_b)$ joignant x_a à x_b.

$$l(\alpha^*_{a,b}) = \underset{\alpha \in \alpha(x_a, x_b)}{\text{Min} \atop \text{Max}} \; l(\alpha)$$

La longueur d'1 tel chemin est appelé distance min ou mx de x_a à x_b selon le cas $\quad l(\alpha^*_{x_1, x_6}) = \underset{\alpha \in \alpha(x_1, x_6)}{\text{Max}}$

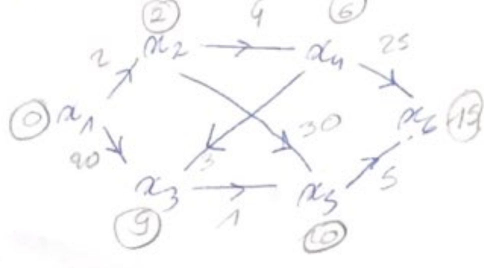
Ex. $l(\alpha^*_{x_1,x_6}) = \text{Max}\left(l(\alpha_1), l(\alpha_2), l(\alpha_3), l(\alpha_4)\right)$

$= \text{Max}(35; 23; 45; 35) = 45 = l(\alpha_3) \Rightarrow \alpha^*_{x_1,x_6} = \alpha_3$

$l(\alpha^{*\prime}_{x_1,x_6}) = \text{Min}(l(\alpha_1), l(\alpha_2), l(\alpha_3), l(\alpha_4)) = \text{Min}(35; 23; 45; 35) = 23 = l(\alpha_2) \Rightarrow \alpha^{*\prime}_{x_1,x_6} = \alpha_2$

Par abus de langage et bien q l'unicité ne soit pas nécessairement réalisée (15)
on parle parfois du chemin le plus court ou du plus long de x_a à x_b. Ns allons présenter 3 algos permettant d'obtenir ds chemins de longueur min ou max. Une nlle numérotation des chemins étant tjrs possible, ns ne considérons q ds chemins de x_1 à x_n. Ls hypothèses faites sur les signes des longueurs l_{ij} varient selon l'algorithme proposé. Pour q le problème étudié admette une solut° de longueur finie, ns supposerons q le graphe G étudié ne comporte aucun circuit de longueur négative dns le cas de la recherche d'1 chemin de longueur min, ni aucun circuit de longueur + dns le cas de la recherche d'un chemin de longueur max.
Il sera commode d'introduire une matrice l dite matrice des longueurs où la composante l_{ij} se définit de façon ≠ selon le problème considéré.

$$L = (l_{ij}) \quad \begin{matrix} i = 1 \text{ — } n \\ j = 1 \text{ — } n \end{matrix}$$



$$l_{ij} = \begin{cases} l_{ij} & \text{si } (x_i, x_j) \in U \\ 0 & \text{si } x_i = x_j \\ -\infty & \text{si } (x_i, x_j) \notin U \text{ et Pb Max} \\ +\infty & \text{si } (x_i, x_j) \notin U \text{ et Pb Min} \end{cases}$$

II - Algorithme de résolution

1) Algorithme de FORD : obtention de chemin de longueur min

Cet algorithme permet l'obtention de chemin de l min ou max ∀ les signes ds longueurs l_{ij}.

principe de l'algo : On marque ts sommets x_j d'une marque λ_j qui correspond à une borne supérieur pr la distance min de x_1 à x_j ($x_j \to x_j$). On diminue progressivement ces marques en suivant la procédure décrite dans l'algo lorsqu'une telle diminution n'est plus possible, la marque λ_j est la dist. min de x_1 à x_j. Un chemin de long min peut alors être construit à partir ds distances λ_j par application de l'algo d'identification.

Etapes	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	
DEBUT	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	initialisat°
x_1	0	2	20	"	"	"	ces successeurs
x_2	"	2	"	6	32	"	
x_3	"	"	20	"	21	"	
x_4	"	"	9	6	"	31	
x_3	"	"	9	"		"	

$x_i : \lambda_j - \lambda_i > l_{ij}$ Pr x_1 : $\lambda_2 - \lambda_1 = +\infty > l_{12}$ $x_2 : \lambda_4 - \lambda_2 = +\infty > l_{24}$

$\Leftrightarrow \lambda_j > \lambda_i + l_{ij}$ $\lambda_2 \leftarrow \lambda_1 + l_{12}$ $\lambda_4 \leftarrow \lambda_2 + l_{24}$

$\lambda_j \leftarrow \lambda_i + l_{ij}$ $\lambda_3 - \lambda_1 = +\infty > l_{13}$ $\lambda_5 - \lambda_2 = +\infty > l_{25}$

$\lambda_3 \leftarrow \lambda_1 + l_{13}$ $\lambda_5 \leftarrow \lambda_2 + l_{25}$

$x_3 \lambda_5 - \lambda_3 = 32 - 20 = 12 > 1$

$\Rightarrow \lambda_5 = \lambda_3 + l_{35} = 20 + 1 = 21$

On revient sur x_3 à cause du changement de valeur.

$\lambda_5 - \lambda_3 = 12 > l_{35}$

$\lambda_5 \leftarrow \lambda_3 + l_{35}$

$x_4 : \lambda_5 - \lambda_3 = 12 > 2$; $\lambda_6 - \lambda_4 = +\infty > 25$

$\lambda_5 \leftarrow \lambda_3 + l_{35}$ $\lambda_6 \leftarrow x_4 + l_{46}$

$\lambda_3 - \lambda_4 = 14 > 3$

$\lambda_3 \leftarrow \lambda_4 + l_{43}$

Pour x_5 : $\lambda_6 - \lambda_5 = 21 > 5$

$\lambda_6 = \lambda_5 + l_{56}$

Suite du tableau

Etape	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
DEBUT						
x_4	"	"	9	6	~~10~~	31
x_5	"	"	9	6	10	15
x_6	0	2	9	6	10	15

On regarde les marques des arcs incidents intérieurement. ex $x_2 x_1$ et $x_1 \to x_1$.

Ex Det les distances max pr les chemins qui partent de x_1.

Etape	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
DEBUT	0	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
x_1	0	2	20	"	"	"
x_2	"	2	"	6	32	"
x_3	"	"	20	"	32	"
x_4	"	"	20	6	"	31
x_5	"	"	"	"	32	37
x_6	0	2	20	6	32	37

$x_i : \lambda_j - \lambda_i < l_{ij}$

$\lambda_j < \lambda_i + l_{ij}$

$\lambda_j \leftarrow \lambda_i + l_{ij}$

2) Algorithme de Bellman - Kalaba : obtention de chemin de longueur minimale

14/01/2019

Cet algorithme permet l'obtention de chemin de longueur minimale ou maximale $\forall$ le signe des longueurs l_{ij}.

Principe : On considère au départ de x_i, les chemins d'un arc (étape 1) puis ceux de deux arcs au plus (étape 2) ... et ceux de k-arcs au plus (étape k)

A l'étape k, on marque tout sommet x_i d'une marque λ_j de k qui représente la longueur du plus court chemin joignant x_i à x_j au k arcs au plus. Lorsque toutes les marques obtenues aux étapes k-1 et k st identiques λ_{jk} est la distance minimale de x_i à x_j, un chemin de longueur minimale peut donc être construit à partir de ces distances par application de l'algorithme d'identification

Version Forward

Initialisation:

Sommet x_1 donc $\lambda_1(k) = 0$; autre = 0

étape k:

$k : \lambda_j(k) = \min(\lambda_i(k-1) + l_{ij}) \quad x_i \in X$

$\lambda_1(0) = 0 \ , \ \lambda_j(0) = +\infty \quad j \neq 1$

Ex $\lambda_1(1) = \text{Min}(\lambda_1(0) + l_{ij})$

$\min(\lambda_1(0) + l_{ij} \ ; \ \lambda_2(0) + l_{ij} \ ; \lambda_3(0) + l_{ij} \ ; \ \lambda_4(0) + l_{ij} \ ; \lambda_5(0) + l_{ij} \ ; \lambda_6(0) + l_{ij} \ ;$

$\lambda_1(1) = 0$

* critère d'arrêt : 2 lignes identiques

$\lambda_2(1) = \min(\lambda_1(0) + l_{i2})$

$= \min(\lambda_1(0) + l_{12} \ ; \ \lambda_2(0) + l_{22} \ ; \ \lambda_3(0) + l_{32} \ ; \ \lambda_4(0) + l_{43} \ ; \ \lambda_5(0) + l_{53} \ ; \ \lambda_6(0) + l_{63}) \quad + \infty$

$\min(2, +\infty)$

Etape k	$\lambda_1(k)$	$\lambda_2(k)$	$\lambda_3(k)$	$\lambda_4(k)$	$\lambda_5(k)$	$\lambda_6(k)$
k = 0	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
k = 1	0	2	20	$+\infty$	$+\infty$	$+\infty$
k = 2	0	2	20	6	21	$+\infty$
k = 3	0	2	9	6	21	26
k = 4	0	2	9	6	10	26
k = 5	0	2	9	6	10	15
k = 6	0	2	9	6	10	15

distances minimales chemin partant de x_i et composé de 3 arcs au plus

distances minimales chemin qui partent de x_1

$$\begin{bmatrix} 0 & 2 & 20 & +\infty & +\infty & +\infty \\ +\infty & 0 & +\infty & 4 & 30 & +\infty \\ +\infty & +\infty & 0 & +\infty & 1 & +\infty \\ +\infty & +\infty & 3 & 0 & +\infty & 25 \\ +\infty & +\infty & +\infty & +\infty & 0 & 5 \\ +\infty & +\infty & +\infty & +\infty & +\infty & 0 \end{bmatrix} \quad \underset{P_r\ k=1}{\begin{bmatrix} 0 \\ +\infty \\ +\infty \\ +\infty \\ +\infty \\ +\infty \end{bmatrix}} \quad \underset{P_r\ k=2}{\begin{bmatrix} 0 \\ 2 \\ 20 \\ +\infty \\ +\infty \\ +\infty \end{bmatrix}} \quad \underset{k=3}{\begin{bmatrix} 0 \\ 2 \\ 20 \\ 6 \\ 21 \\ +\infty \end{bmatrix}}$$

valeur $k=1$ (trouvée

Étape k	$\lambda_1(k)$	$\lambda_2(k)$	$\lambda_3(k)$	$\lambda_4(k)$	$\lambda_5(k)$	$\lambda_6(k)$
$k=0$	0	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
$k=1$	0	2	20	$-\infty$	$-\infty$	$-\infty$
$k=2$	0	2	20	6	32	$-\infty$
$k=3$	0	2	20	6	32	37
$k=4$	0	2	20	6	32	37

$\lambda_1(0)=0$, $\lambda_j(0)=-\infty$; $\lambda_j(k)=\max(\lambda_i(k+1)+l_{ij})$

distance maximale chemin partant de x_1

Version Backward

On veut chercher les distances minimales ou maximales des chemins qui arrivent en x_6.

$k=6$ $\lambda_n(0)=0$ $\quad \lambda_j(0)=\begin{smallmatrix}-\infty\\+\infty\end{smallmatrix}$ $j\neq n$

k: $\lambda_j(k)=\underset{x_i\in X}{\begin{smallmatrix}\max\\\min\end{smallmatrix}}(\lambda_i(k-1)+l_{ji})$

distances maximales

	$\lambda_1(k)$	$\lambda_2(k)$	$\lambda_3(k)$	$\lambda_4(k)$	$\lambda_5(k)$	$\lambda_6(k)$
$k=0$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0
$k=1$	$-\infty$	$-\infty$	$-\infty$	25	5	0
$k=2$	$-\infty$	35	6	25	5	0
$k=3$	37	35	6	25	5	0
$k=4$	37	35	6	25	5	0

distance max chemin arrivant x_6 constituée de 2 arcs au plus

Pour $k=2$ $[-\infty \quad -\infty \quad -\infty \quad 25 \quad 5 \quad 0]$

Pour $k=1$ $[-\infty \quad -\infty \quad -\infty \quad -\infty \quad -\infty \quad 0]$

$$\begin{bmatrix} 0 & 2 & 20 & -\infty & -\infty & -\infty \\ -\infty & 0 & -\infty & 4 & 30 & -\infty \\ -\infty & -\infty & 0 & -\infty & 1 & -\infty \\ -\infty & -\infty & 3 & 0 & -\infty & 25 \\ -\infty & -\infty & -\infty & -\infty & 0 & 5 \\ -\infty & -\infty & -\infty & -\infty & -\infty & 0 \end{bmatrix}$$

algorithme de Bellman - Kalaba dans le cas d'1 partage à niveau de G : obtention simplifié de chemin de longueur minimale (20)

21/01/2019

Forward (distance minimale chemin partant d'1 sommet)

$\lambda_1(0) = 0 \quad ; \lambda_j(0) = +\infty \quad j \neq 1$

$k : \lambda_j(k) = \min(\lambda_i(k-1) + l_{ij}) \quad x_i \in X \quad j = 1 — n$

Backward (arrivant à un sommet)

$\lambda n(0) = 0 \quad \lambda_j(0) = +\infty$

$k : \lambda j(k) = \min(\lambda_i(k-1) + l_{ji}) \quad x_i \in X \quad j = 1 — n$

L'algorithme est une application beaucoup plus directe lorsque le graphe G peu être préalablement partitionner en niveau

Principe de l'algorithme :

Rappelons qu'un graphe G sans circuit peut être partager en niveau. Soit $X(0), X(1), \dots, X(k-1)$ les k Niveaux du Graphe G

Pour faciliter la présentation, rénumérotons les en sens inverse et notons les respectivement $N_K, N_{K-1} \dots N_1$

Supposons que $x_1 \in N_1$ et $x_n \in N_k$, l'algorithme peut alors se simplifier comme suit :

A l'étape $k = 1 — K$, on marque chaque sommet $x_j \in N_k$ d'une marque λ_j qui représente la distance minimale de x_1 à x_j les marques ainsi attribuées sont donc définitives, ce qui réduit fortement le nbre d'opérations à effectuer.

x_j	$\Gamma^+(x_j)$
x_1	$\cancel{x_2}, \cancel{x_3}$
$\cancel{x_2}$	$\cancel{x_4}, \cancel{x_5}$
$\cancel{x_3}$	$\cancel{x_5}$
$\cancel{x_4}$	$\cancel{x_3}, \cancel{x_6}$
$\cancel{x_5}$	$\cancel{x_6}$
$\cancel{x_6}$	

$X(0) = \{x_6\} = N_6$

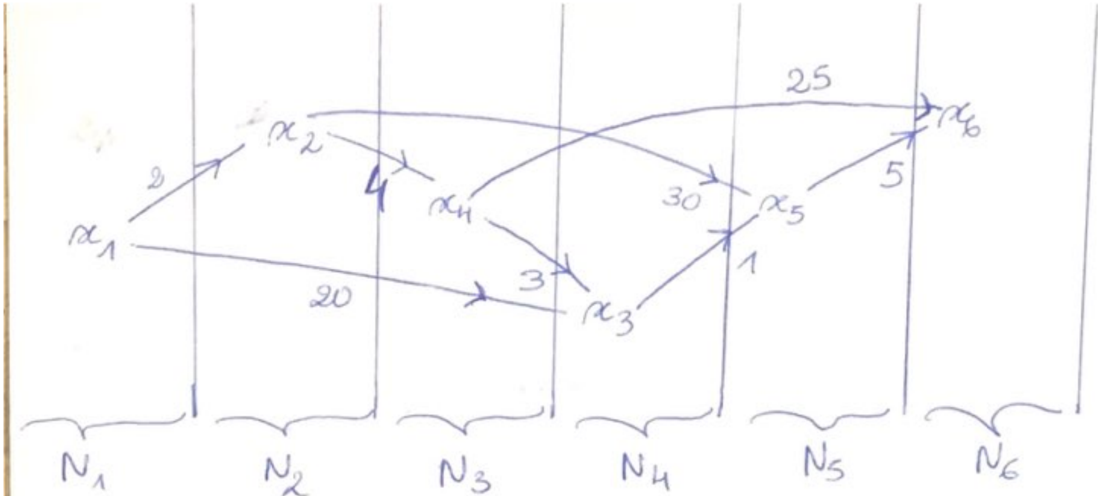
$X(1) = \{x_5\} = N_5$

$X(2) = \{x_3\} = N_4$

$X(3) = \{x_4\} = N_3$

$X(4) = \{x_2\} = N_2$

$X(5) = \{x_1\} = N_1$



Forward

$\lambda_1 = 0$

$N_k : x_j \in N_k$

$$\lambda_j = \underset{x_i \in \Gamma^-(x_i)}{\mathrm{Min}} (\lambda_i + l_{ij})$$

Backward

$\lambda_n = 0$

$N_k : x_i\, x_j \in N_k$

$$\lambda_j = \underset{x_i \in \Gamma^+(x_j)}{\mathrm{Min}} (\lambda_i + l_{ji})$$

Δ! Max

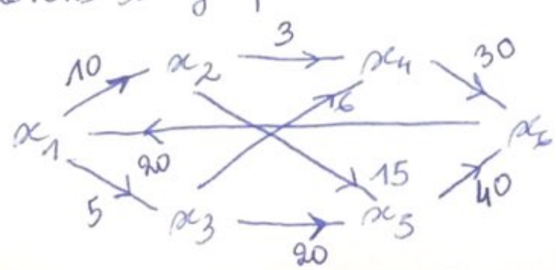
Etp	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
N_1	0					
N_2		2				
N_3				6		
N_4			9 20			
N_5					32 10	
N_6						37 15

Forward

Etp	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
N_6						0
N_5					5	
N_4			6			
N_3				9 25		
N_2		35 13				
N_1	37 15					

Backward

Considérons le graphe suivant :



Dt. les distances minimales pour les chemins :

① Partant de x_4

② Arrivant en x_4

(22)

11/03/2019

①

k	$\lambda_1(k)$	$\lambda_2(k)$	$\lambda_3(k)$	$\lambda_4(k)$	$\lambda_5(k)$	$\lambda_6(k)$
0	$+\infty$	$+\infty$	$+\infty$	0	$+\infty$	$+\infty$
1	$+\infty$	$+\infty$	$+\infty$	0	$+\infty$	30
2	50	$+\infty$	$+\infty$	0	$+\infty$	30
3	50	60	55	0	$+\infty$	30
4	50	60	55	0	75	30
5	50	60	55	0	75	30
6						

$$\begin{bmatrix} 0 & 10 & 5 & +\infty & +\infty & +\infty \\ +\infty & 0 & +\infty & 3 & 15 & +\infty \\ +\infty & +\infty & 0 & 6 & 20 & +\infty \\ +\infty & +\infty & +\infty & 0 & +\infty & 30 \\ +\infty & +\infty & -\infty & +\infty & 0 & 40 \\ 20 & +\infty & +\infty & +\infty & +\infty & 0 \end{bmatrix} \begin{bmatrix} \\ \\ \\ \\ \\ \\ \end{bmatrix}$$

②

	x_1	x_2	x_3	x_4	x_5	x_6
0	$+\infty$	$+\infty$	$+\infty$	0	$+\infty$	$+\infty$
1	$+\infty$	3	6	0	$+\infty$	$+\infty$
2	11	3	6	0	$+\infty$	$+\infty$
3	11	3	6	0	$+\infty$	31
4	11	3	6	0	71	31
5	11	3	6	0	71	31
6						

Application : analyse de chemin critique dans le cadre d'un projet. (3)

Un projet a l'exemple d'une construction d'un immeuble comprend un certain nombre d'activités liés : aménagement du site, réalisat° des fondations ... . En général certaines activités d'un projet peuvent être exécuter en m^e tps. Par contre, il y'a des activités dont la réalisation ne peut se faire qu'à la réalisat° de certaines autres. Dans la GDP, les dates de démarrage au plutôt de chaque activité et les dates de démarrages au plutard mais aussi le délai minimum de réalisat° du projet.

Le tableau svt ns montre comment les liaisons entre les activités d'un projet doivent être présenter.

Activ	Durée (ut)	Act. Antérieures
1	0	–
2	4	1
3	10	1
4	6	2
5	2	2
6	11	2
7	22	4;5
8	3	5
9	17	3;6;8
10	0	7;9

Les activités n° 1 et 10 st fictives de durée nulle représentant respectivement le début et la fin du projet. Pour chaque activité i, le tableau donne sa durée d_i et la liste de ces prédecesseurs ie la liste des activités qu'il faut terminer avant de débuter i. A partir de ces informat°, on peut construire un graphe d'activité pr représenter le projet. Dans ce graphe, les sommets correspondent aux activites du projet. plus précisément chaque sommet s_i représente un evt, le commencement de l'activité i. Les contraintes de temps liées à l'exécution des activités st representées par des arcs du graphe. Si l'activité i ne peut pas être débuter avant que l'activité k ne st terminée ie au moins d_k ut ~~avant~~ après l'act. k ne soit débuté. Cette condition s'exprime par : $x_k \xrightarrow{u_k} x_i$

Ns noterons qu'un graphe d'activité est forcément sans circuit

$N_1 = \{x_1\}$

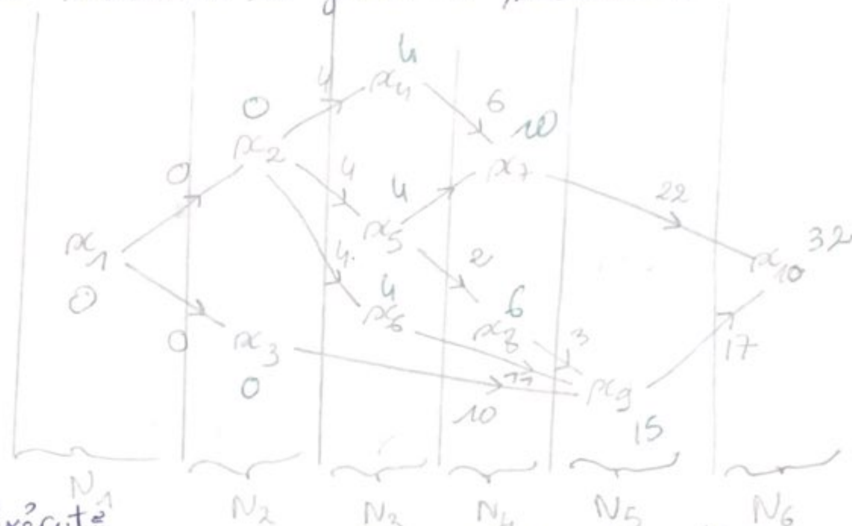
$N_2 = \{x_2, x_3\}$

$N_3 = \{x_4, x_5, x_6\}$

$N_4 = \{x_7, x_8\}$

$N_5 = \{x_9\}$

$N_6 = \{x_{10}\}$



Délais minimum de l'exécut° de l'ensemble du projet = distance maximale allant de x_1 à x_{10}

$= 32\ ut = \theta$

x_j : distance max de x_1 à x_j

λ_j : date de début au plut tôt de l'act j.

λ'_j : dist. max de x_j à x_n

: délai min séparant le dbt d'exécution de l'act j de la fin ~~projet~~ projet.

$\lambda\lambda''_j$: date au dbut au plutard de l'act j.

$\boxed{\lambda\lambda''_j = \theta - \lambda'_j}$

là où il y a coincidence entre la date de début au plustard et au plut tôt = activités critiques. leurs arcs fait partie du chemin

x_j	λ_j	λ'_j	$\lambda''_j = \theta - \lambda'_j$	Δ_j
x_1	0	32	0	0
x_2	0	32	0	0
x_3	0	27	5	5
x_4	4	28	4	0
x_5	4	24	8	4
x_6	4	28	4	0
x_7	10	22	10	0
x_8	6	20	12	6
x_9	15	17	15	0
x_{10}	32	0	32	0

Δ_j = retard qu'on peut accuser dans l'exécut° de x_j sans retarder le projet

$\Delta_j = \lambda''_j - x_j$

Si le retard est inférieur à la marge totale, il n'y aura pas de conséquences concernant la durée du projet

La marge libre est le retard qu'on peut accuser dans l'exécut° d'une activité sans retarder les dates au plutot ds act. svtes :