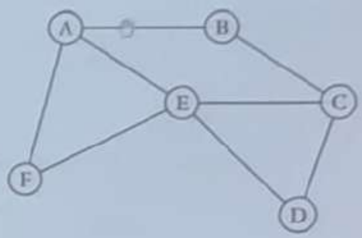
Exercice 1

1. *Cette situation peut être modélisée par le graphe (G) suivant :*
   - *Chaque sommet représente un atelier*
   - *Les canalisations entre les ateliers seront représentées par des arêtes*



2. (a) *Oui, vu que le graphe (G) est connexe. (Un graphe est connexe si deux sommets quelconques peuvent être reliés par une chaîne).*

   (b) *Non, puisque le graphe (G) n'est pas complet. (Un graphe non orienté est complet si et seulement si tous ses sommets sont reliés par une arête).*

   (c) *Oui, vu que le graphe (G) est hamiltonien. On prends le cycle* **A-B-C-D-E-F-A**.

   (d) *Oui, vu que le graphe (G) abrite une chaîne eulérienne. En effet, on a les degrés des sommets sont donnés par le tableau ci-après :*

Sommet	A	B	C	D	E	F
Degré	3	2	3	2	4	2

Le graphe (G) possède deux sommets de degré impair : A et C. Il est donc possible de parcourir l'usine en empruntant chacune des 8 canalisations une fois et une seule, par exemple en suivant le trajet suivant : **A-B-C-D-E-F-A-E-C**.

en empruntant chacune des 8 canalisations une fois et une seule, par exemple en suivant le trajet suivant : **A-B-C-D-E-F-A-E-C**.

3. (a) *La matrice d'adjacence* $\mathcal{M}$ *et la matrice* $\mathcal{M}^3$ *sont comme suit :*

$$\mathcal{M} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \Longrightarrow \mathcal{M}^3 = \begin{pmatrix} 2 & 5 & 2 & 3 & 7 & 4 \\ 5 & 0 & 5 & 2 & 2 & 2 \\ 2 & 5 & 2 & 4 & 7 & 3 \\ 3 & 2 & 4 & 2 & 5 & 2 \\ 7 & 2 & 7 & 5 & 4 & 5 \\ 4 & 2 & 3 & 2 & 5 & 2 \end{pmatrix}$$

(b) *Le coefficient* a_{ij} *de la matrice* $\mathcal{M}^3$ *situé à la i-ième ligne et à la j-ième colonne indique le nombre de chaînes composées de* 3 *arêtes reliant les sommets* i *et* j.

Puisque le coefficient a_{14} *de la matrice* $\mathcal{M}^3$ *est* $a_{14} = 3$, *il y a donc 3 chemins permettant d'aller de l'atelier A à l'atelier D en empruntant exactement trois canalisations.*

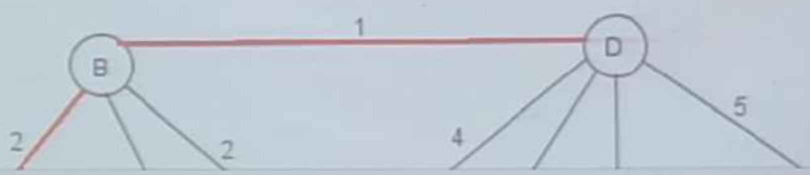
À l'aide du graphe, on trouve les chaînes : **A-B-C-D**, **A-E-C-D** *et* **A-F-E-D**.

(c) *Oui, vu que la matrice* $\mathcal{M}^3$ *ne comporte pas de* 0 *sauf sur la diagonale* ($a_{22} = 0$).

On souhaite déterminer la plus courte chaîne reliant les sommets A et G. Vu que les pondérations sont toutes positives on peut utiliser l'algorithme de Dijkstra à partir du sommet A qui est racine :

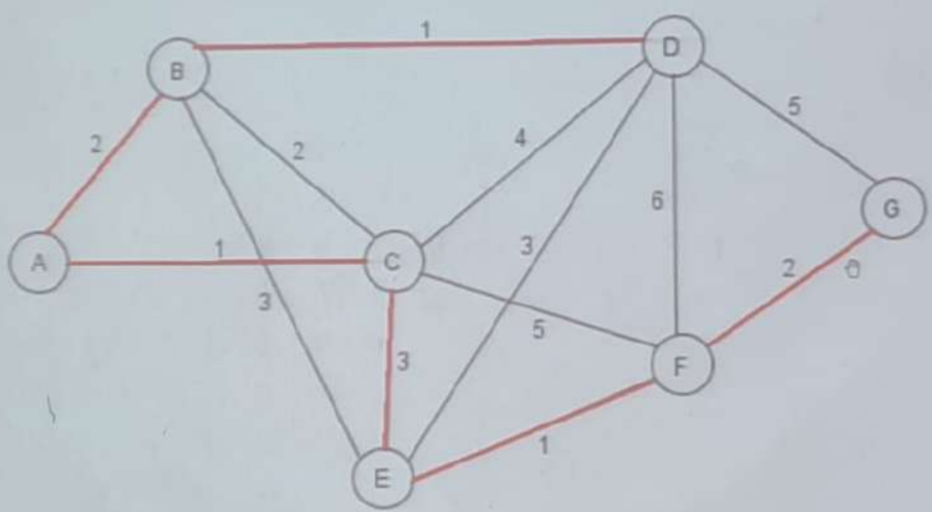
	L							P						
	A	B	C	D	E	F	G	A	B	C	D	E	F	G
A	0	2	1	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	A	A	-	-	-	-
AC	0	2	1	5	4	6	$+\infty$	A	A	A	C	C	C	-
ACB	0	2	1	3	4	6	$+\infty$	A	A	A	B	C	C	-
$ACBD$	0	2	1	3	4	6	8	A	A	A	B	C	C	D
$ACBDE$	0	2	1	3	4	5	8	A	A	A	B	C	E	D
$ACBDEF$	0	2	1	3	4	5	7	A	A	A	B	C	E	F
$ACBDEFG$	0	2	1	3	4	5	7	A	A	A	B	C	E	F

L'arbre des plus courtes chaînes reliant A aux autres sommets du graphe est :



ACBDEFG	*0*	*2*	*1*	*3*	*4*	*5*	*7*	*A*	*A*	*A*	*B*	*C*	*E*	*F*

L'arbre des plus courtes chaînes reliant A aux autres sommets du graphe est :



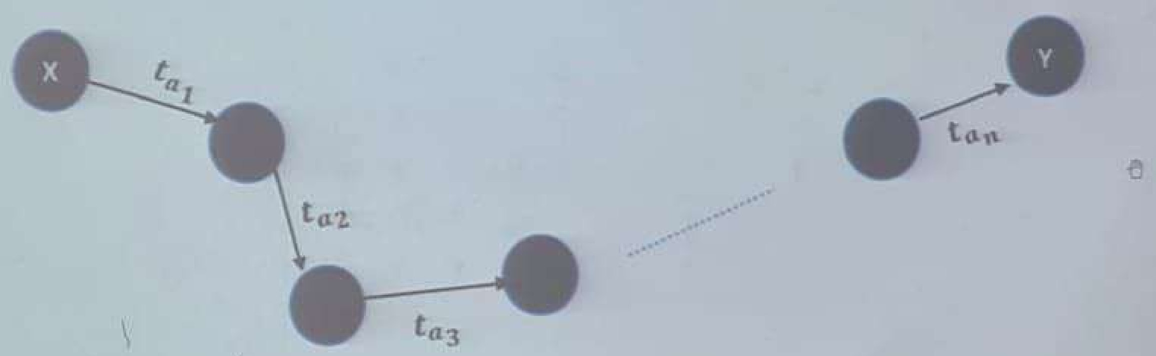
- *Le trajet* **A-C-E-F-G** *comporte le nombre minimum de 7 feux tricolores de A à G.*
- *Toute sous chaîne d'une plus courte chaîne est également plus courte. Donc la plus courte chaîne de F à C est* **F-E-C**.

Exercice 3

On a $\mathbf{p_i}$ *est la probabilité de se faire dépouiller sur le tronçon* $\mathbf{t_i}$.

On note par $\mathbf{q_i = 1 - p_i}$ *la probabilité de passer en sécurité sur le tronçon* $\mathbf{t_i}$.

On considère un chemin $\mathcal{C}$ *de la ville* **X** *vers la ville* **Y** *constitué des tronçons suivants :* $\mathbf{t_{a_1}}, \mathbf{t_{a_2}}, \mathbf{t_{a_3}}, \cdots, \mathbf{t_{a_n}}$



L'événement **D** : "**se faire dépouiller sur le chemin** $\mathcal{C}$", *se réalise si on se fait dépouiller sur au moins un tronçon de ce chemin.*

C'est pourquoi nous allons calculer la probabilité de passer sur le chemin $\mathcal{C}$ *en sécurité sans se faire dépouiller sur aucun tronçon de ce chemin. Donc la probabilité* $p(S)$ *de passer en sécurité sur le chemin* $\mathcal{C}$ *est (avec* $\mathbf{S} = \overline{\mathbf{D}}$*) :*

$$p(S) = \prod^{n} q_{a_k}.$$

tronçon de ce chemin.

C'est pourquoi nous allons calculer la probabilité de passer sur le chemin $\mathcal{C}$ en sécurité sans se faire dépouiller sur aucun tronçon de ce chemin. Donc la probabilité $p(S)$ *de passer en sécurité sur le chemin $\mathcal{C}$ est (avec* $\mathbf{S} = \overline{\mathbf{D}}$*) :*

$$p(S) = \prod_{\substack{k=1 \\ t_{a_k} \in \mathcal{C}}}^{n} q_{a_k}.$$

On introduit la fonction $-log$, on obtient :

$$-log(p(S)) = \sum_{\substack{k=1 \\ t_{a_k} \in \mathcal{C}}}^{n} -log(q_{a_k}).$$

Afin de chercher un chemin qui permet de minimiser la probabilité de se faire dépouiller entre deux ville, on modélise la situation par un graphe G tel que :

- *Les sommets sont les villes.*
- *Les arêtes représentent les tronçons reliant les villes.*
- *Chaque arête sera pondérée par le poids :* $\mathbf{-log(q_i)}$.

Vu que les $q_i \leq 1$, on aura $-log(q_i) \geq 0$. Donc toutes les pondérations du graphe G sont positives.

On introduit la fonction $-log$, on obtient :

$$-log(p(S)) = \sum_{\substack{k=1 \\ t_{a_k} \in \mathcal{C}}}^{n} -log(q_{a_k}).$$

Afin de chercher un chemin qui permet de minimiser la probabilité de se faire dépouiller entre deux ville, on modélise la situation par un graphe G tel que :

- *Les sommets sont les villes.*
- *Les arêtes représentent les tronçons reliant les villes.*
- *Chaque arête sera pondérée par le poids :* $\mathbf{-log(q_i)}$.

Vu que les $q_i \leq 1$, on aura $-log(q_i) \geq 0$. Donc toutes les pondérations du graphe G sont positives.

Ici on cherche à minimiser la probabilité de se faire dépouiller, $\mathbf{min\{p(D)\}}$. *Ce qui revient à chercher le* $\mathbf{max\{p(S)\}}$ *et par la suite le* $\mathbf{min\{-log(p(S))\}}$.

Donc, on peut appliquer l'algorithme Dijkstra à partir de X pour chercher le plus court chemin sur le graphe G partant de X vers les autres sommets.

séquence.

2. *Vu que le taux de change d'un chemin représente le produit des taux de changes de ses arcs, on va procéder à la modification du graphe G afin d'obtenir un nouveau graphe G' en remplaçant uniquement les poids des arcs $t_{i,j}$ par* $\mathbf{log(t_{i,j})}$.

   Du moment qu'on peut avoir certains $t_{i,j} < 1$ et d'autres $t_{i,j} \geq 1$ donc on peut avoir sur le graphe G' des pondérations positives comme on peut avoir d'autres strictement négatives.

   En plus dans ce problème, on cherche à maximiser la valeur du taux de change entre deux sommets donc cherche à maximiser la quantité :

$$[\mathrm{Max}]\ \mathbf{log(t)} = \sum_{i=1}^{k-1} \mathbf{log(t_{i,i+1})}.$$

   Vu les éléments précédent, on doit appliquer un algorithme permettant de déterminer **le plus long chemin**. *Pour cela, on peut appliquer l'algorithme de* **Bellman-Ford modifié** *comme suit :*

   - *On vérifie d'abord l'absence de circuit absorbant avec une valuation strictement positive.*
   - *Dans la phase initialisation de l'algorithme, on affecte la valeur $-\infty$ aux sommets différents du sommet s.*
   - *Dans la phase traitement d'un sommet x, on ne change la valeur de $L[y]$ d'un successeur y de x sauf si la nouvelle longueur trouvée pour atteindre le sommet y via le sommet x qui est $L[x] + p_{sx}$ est strictement supérieure à sa valeur actuelle $L[y]$: $L[x] + p_{sx} > L[y]$.*

3. *Quelqu'un peut devenir infiniment riche en changeant de l'argent si à partir d'une somme X d'une devise*

$$[\text{Max}] \quad \log(t) = \sum_{i=1}^{k-1} \log(t_{i,i+1}).$$

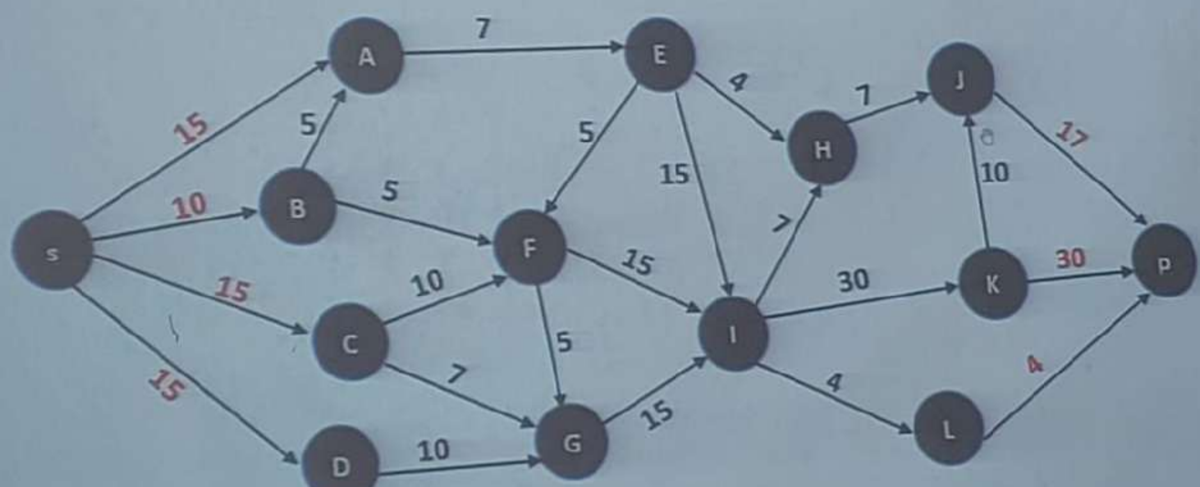
Vu les éléments précédent, on doit appliquer un algorithme permettant de déterminer **le plus long chemin.** *Pour cela, on peut appliquer l'algorithme de* **Bellman-Ford modifié** *comme suit :*

- *On vérifie d'abord l'absence de circuit absorbant avec une valuation strictement positive.*
- *Dans la phase initialisation de l'algorithme, on affecte la valeur $-\infty$ aux sommets différents du sommet s.*
- *Dans la phase traitement d'un sommet x, on ne change la valeur de $L[y]$ d'un successeur y de x sauf si la nouvelle longueur trouvée pour atteindre le sommet y via le sommet x qui est $L[x] + p_{xx}$ est strictement supérieure à sa valeur actuelle $L[y]$: $L[x] + p_{xx} > L[y]$.*

3. *Quelqu'un peut devenir infiniment riche en changeant de l'argent si à partir d'une somme X d'une devise initiale, il arrive à trouver une succession de conversions qui permettent de revenir vers la même devise initiale en obtenant une nouvelle somme X' telle que $X' > X$.*

   Ceci est possible lorsque le graphe G comporte un circuit avec un taux de change t_C qui vérifie : $t_C > 1$. Ce circuit est donc un **circuit absorbant** *dans le sens de maximisation vu que $\log(t_C) > 0$..*

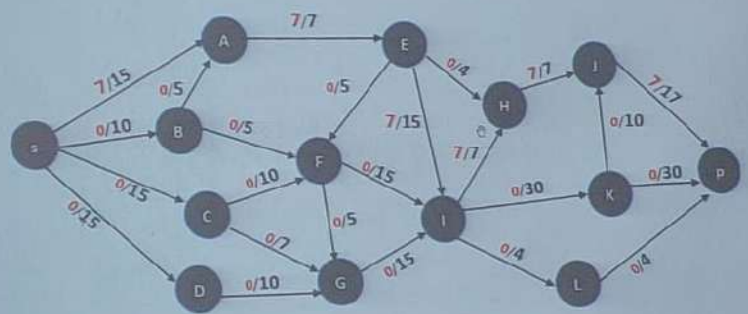
Exercice 5 1. *Le réseau de distribution d'eau de la ville comporte 4 sources et 3 destinations. On rajoute une source s et des arcs la reliant à A, B, C et D de capacité 15, 10, 15, 15. On rajoute un puits p et des arcs reliant J, K et L à p de capacité égale à la somme des capacités entrantes. Ainsi, on obtient un réseau de transport avec des capacités.*



On commence par un flot nul. On cherche à chaque fois un chemin augmentant, on trouve :

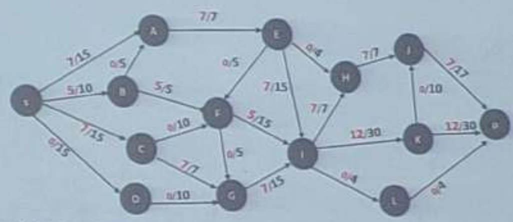
On commence par un flot nul. On cherche à chaque fois un chemin augmentant, on trouve :

- s-A-E-I-H-J-p : valeur +7

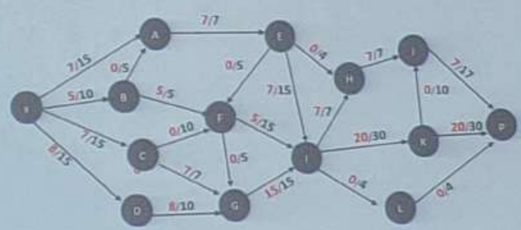


- s-B-F-I-K-p : valeur +5

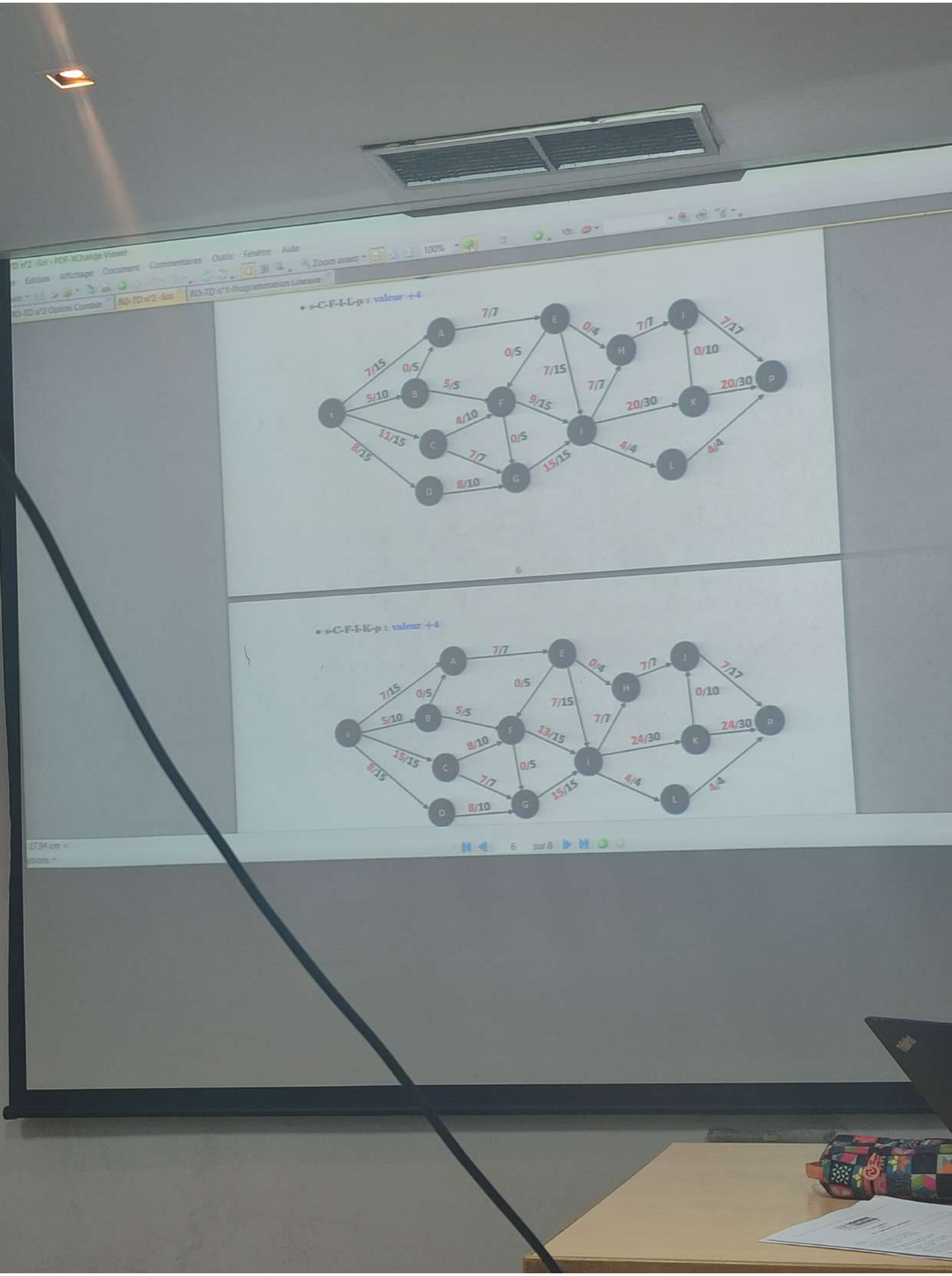
- s-C-G-I-K-p : valeur +7

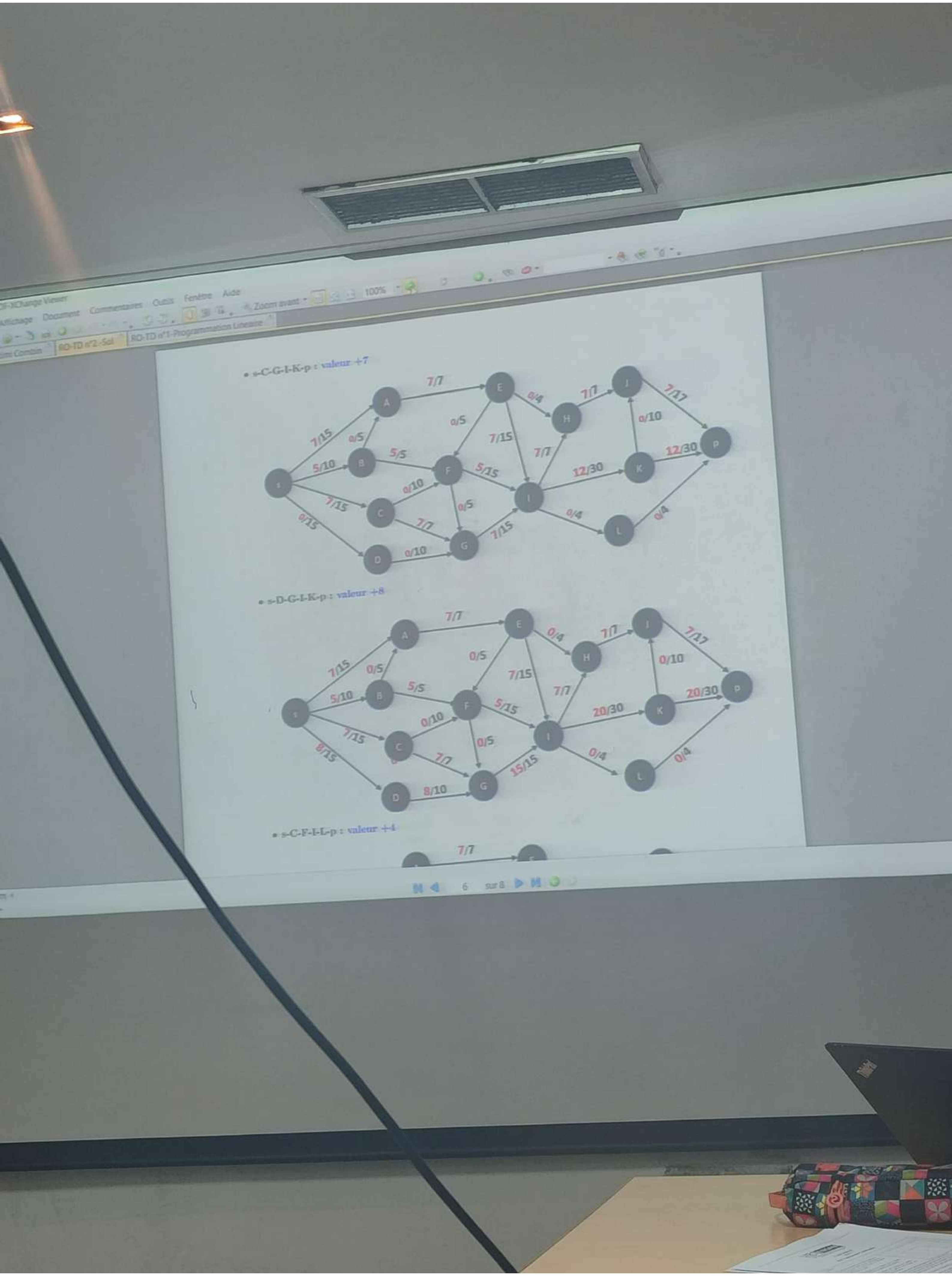


- s-D-G-I-K-p : valeur +8



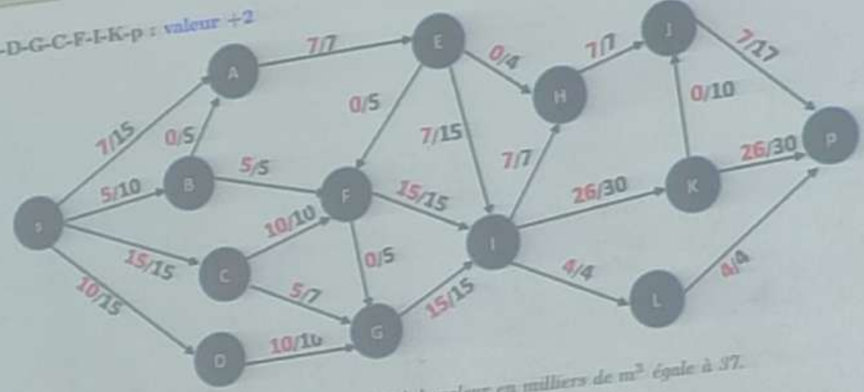
- s-C-F-I-L-p : valeur +4


• s-C-F-I-L-p : valeur +4
7/7
0/4
7/7
7/17
7/15
0/5
0/5
7/15
0/10
5/10
5/5
9/15
7/7
20/30
20/30
4/10
11/15
0/5
8/15
7/7
4/4
4/4
15/15
8/10
6
• s-C-F-I-K-p : valeur +4
7/7
0/4
7/7
7/17
0/5
7/15
7/15
0/10
0/5
5/10
5/5
13/15
7/7
24/30
24/30
8/10
15/15
0/5
8/15
7/7
4/4
4/4
15/15
8/10
6 sur 8

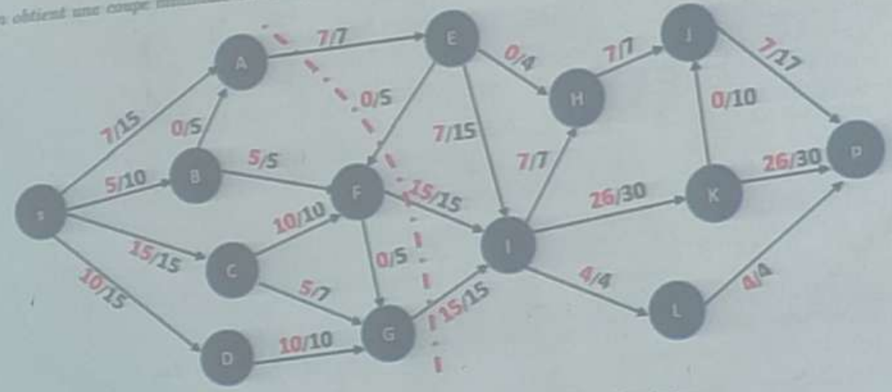


PDF-XChange Viewer
Affichage Document Commentaires Outils Fenêtre Aide
Zoom avant
100%
RO-TD n°2 -Sol
RO-TD n°1-Programmation Linéaire
• s-C-G-I-K-p : valeur +7
• s-D-G-I-K-p : valeur +8
• s-C-F-I-L-p : valeur +4
6 sur 8

- s-D-G-C-F-I-K-p : valeur +2



On obtient une coupe minimale donc de flot maximal de valeur en milliers de m^3 égale à 37.

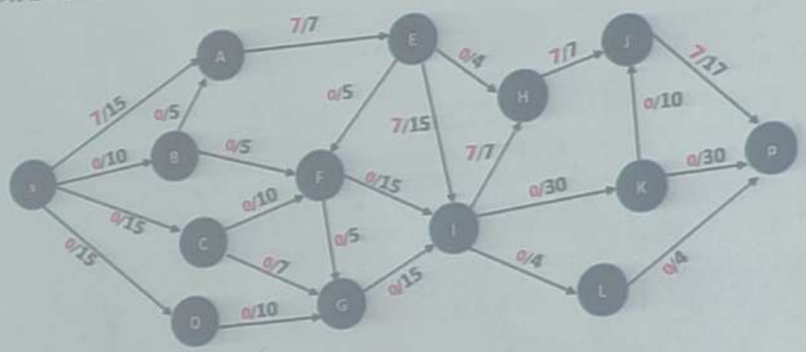


On peut déterminer le flot maximal en appliquant l'un des algorithmes suivants :

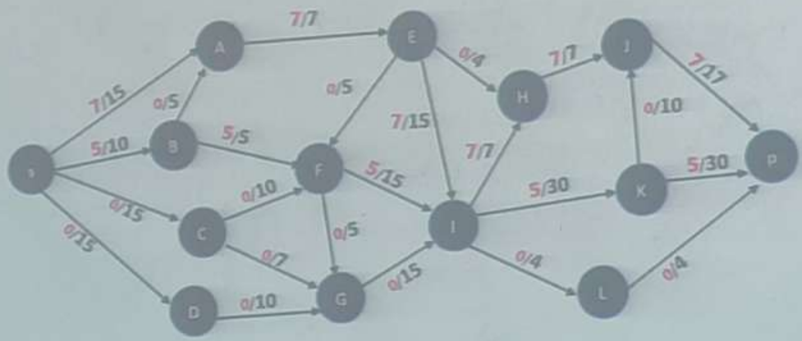
- *Algorithme de Ford-Fulkerson*
- *Algorithme d'Edmonds-Karp*
- *Algorithme de Dinic*
- *Algorithme de Preflow-Push (Goldberg-Tarjan)*

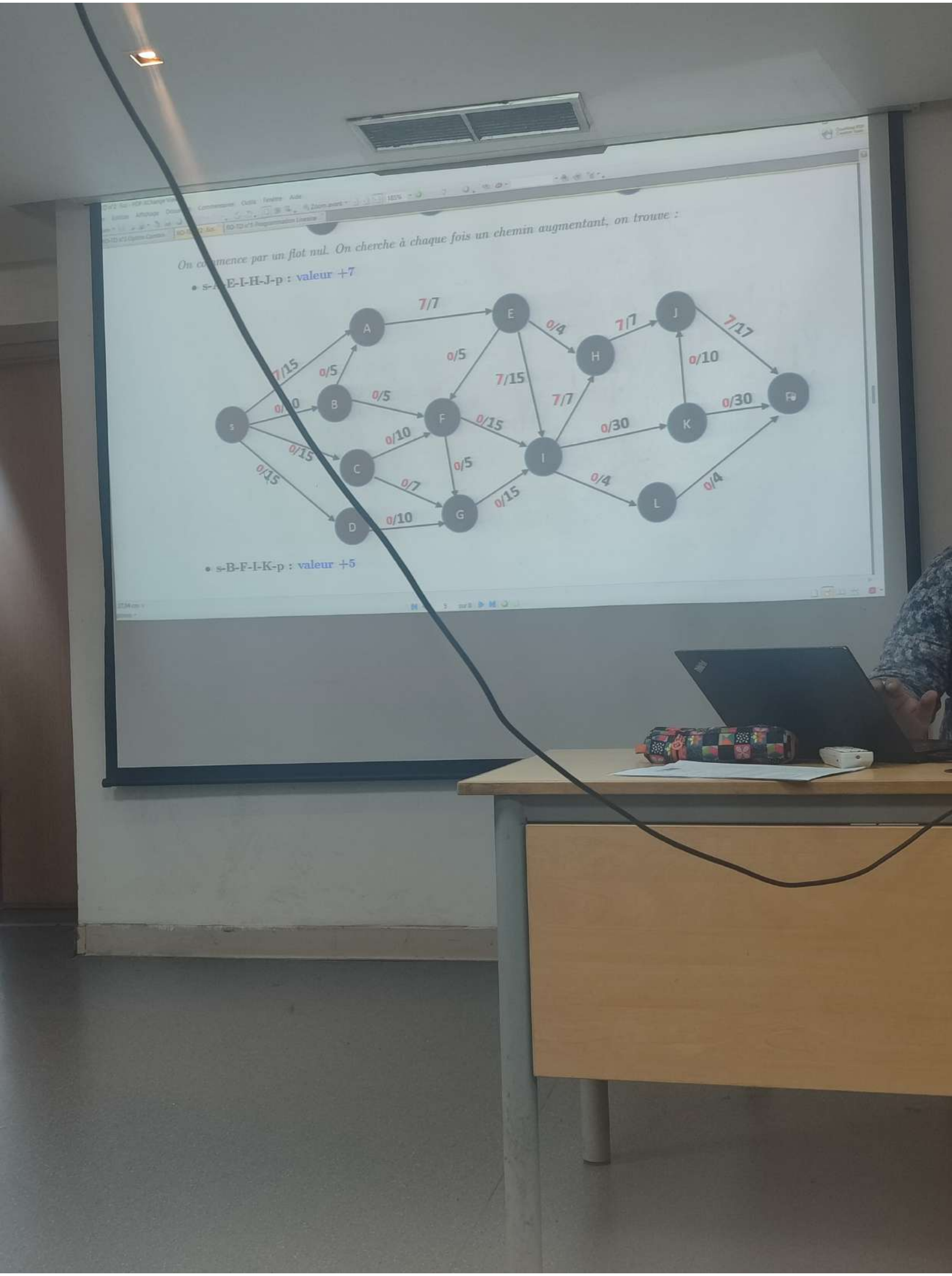
On commence par un flot nul. On cherche à chaque fois un chemin augmentant, on trouve :

- **s-A-E-I-H-J-p :** valeur +7

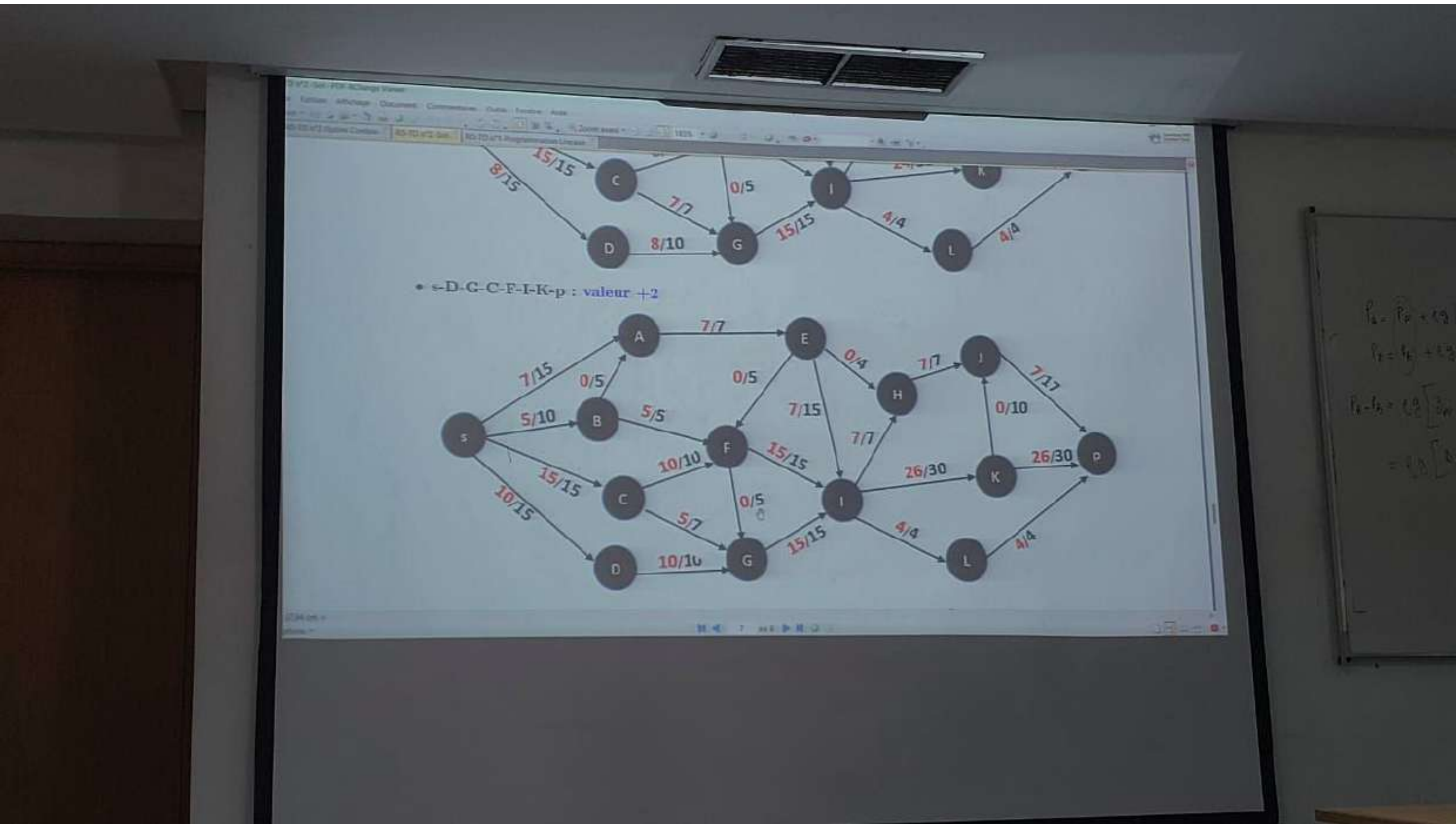


- **s-B-F-I-K-p :** valeur +5



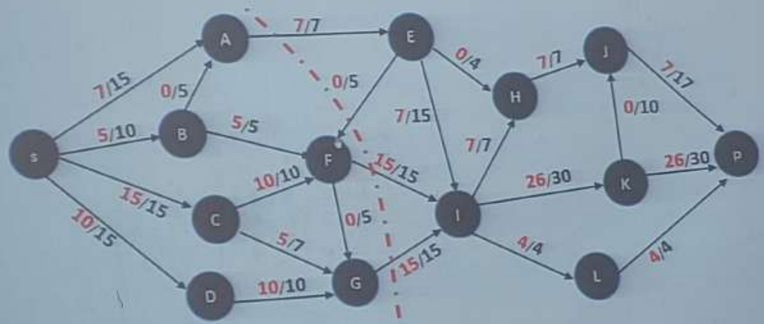


On co
mence par un flot nul. On cherche à chaque fois un chemin augmentant, on trouve :
• s-
-E-I-H-J-p : valeur +7
7/7
0/4
7/7
7/17
0/5
0/5
7/15
0/10
0/5
7/7
0/30
0/15
0/10
0/15
0/15
0/5
0/7
0/4
0/15
0/10
0/4
A
B
C
D
E
F
G
H
I
J
K
L
s
• s-B-F-I-K-p : valeur +5



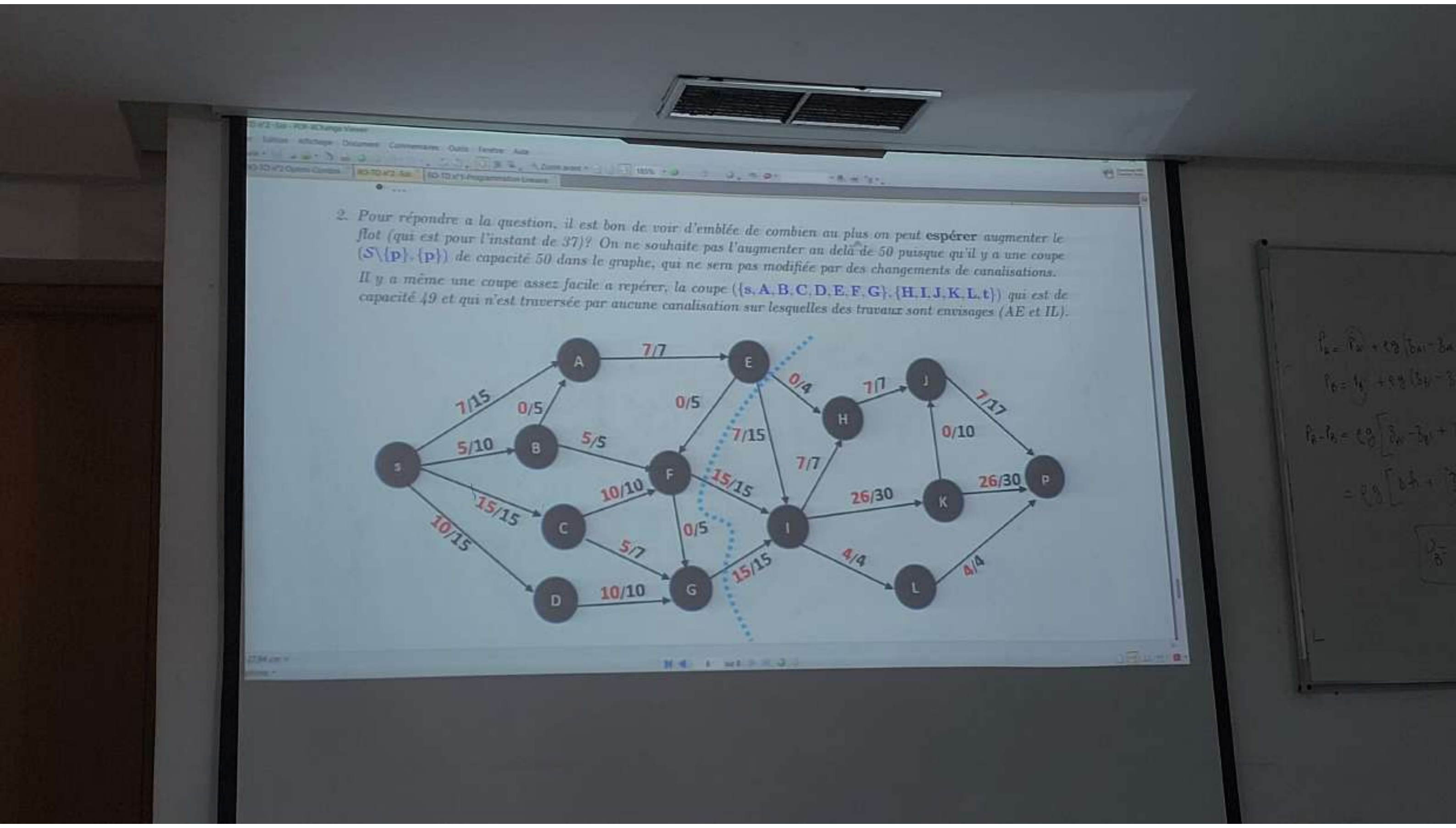
15/15
8/15
C
7/7
0/5
I
4/4
D
8/10
G
15/15
L
4/4
• s-D-G-C-F-I-K-p : valeur +2
A
7/7
E
0/4
7/7
J
7/15
0/5
0/5
7/15
H
7/17
s
5/10
B
5/5
0/10
7/7
F
10/10
15/15
15/15
26/30
K
26/30
p
10/15
C
0/5
I
5/7
4/4
D
10/10
G
15/15
L
4/4

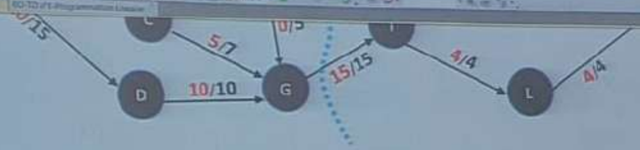
On obtient une coupe minimale donc de flot maximal de valeur en milliers de m^3 égale à 37.



On peut déterminer le flot maximal en appliquant l'un des algorithmes suivants :

- *Algorithme de Ford-Fulkerson*
- *Algorithme d'Edmonds-Karp*
- *Algorithme de Dinic*


2. Pour répondre a la question, il est bon de voir d'emblée de combien au plus on peut **espérer** augmenter le flot (qui est pour l'instant de 37)? On ne souhaite pas l'augmenter au delà de 50 puisque qu'il y a une coupe $(S\setminus\{\mathbf{p}\}, \{\mathbf{p}\})$ de capacité 50 dans le graphe, qui ne sera pas modifiée par des changements de canalisations.
Il y a même une coupe assez facile a repérer, la coupe $(\{\mathbf{s, A, B, C, D, E, F, G}\}, \{\mathbf{H, I, J, K, L, t}\})$ qui est de capacité 49 et qui n'est traversée par aucune canalisation sur lesquelles des travaux sont envisages (AE et IL).
7/7
7/15
0/5
5/10
5/5
0/5
7/15
0/4
7/7
7/17
0/10
15/15
10/10
15/15
10/15
0/5
7/7
26/30
26/30
5/7
15/15
10/10
4/4
4/4



On peut donc espérer augmenter le flot courant d'au plus 12 pour atteindre 49.

Cela exige alors d'augmenter la capacité de la coupe saturée qu'on a exhibée d'au moins 12. Faisons cela, c'est a dire augmentons la capacité de AE de 7 a 19 , et voyons si on peut augmenter d'autant le flot en jouant aussi sur IL.

On peut bien faire arriver 12 de flot supplémentaire en E en envoyant du flot supplémentairement depuis s : 8 selon l'arc sA et 4 selon le chemin s, B, A, par exemple. Ce flot se retrouve alors en E ou il peut traverser la coupe limitante de capacité 49, par l'arc EI pour arriver en I. On a alors un flot supplémentaire entrant sur I de 12. Pour augmenter la capacité de IL le moins possible, essayons d'acheminer ce flot supplémentaire de I vers t sans augmenter IL. Sans changer la capacité de IL, on peut envoyer au plus 2 de flot supplémentaire jusqu'à p car le flot traversant la coupe ($V\setminus\{L,t\},\{L,t\}$) est de 37 alors que la capacité de la coupe est de 39 . Et on peut bien en effet acheminer ce flot supplémentaire de 2 de I a p en passant par exemple par le chemin I,H,J,p. Il nous reste donc un flot supplémentaire de 10 en I. que l'on peut tout acheminer vers p par le chemin I,L,p si on augmente la capecite de IL de 10. L'augmentation maximale qu'on peut attendre pour le flot est donc de 12 (et le flot total de 49) et les augmentations minimales des capacités de AE et IL qu'il faut faire pour y parvenir sont de respectivement 12 et 10.