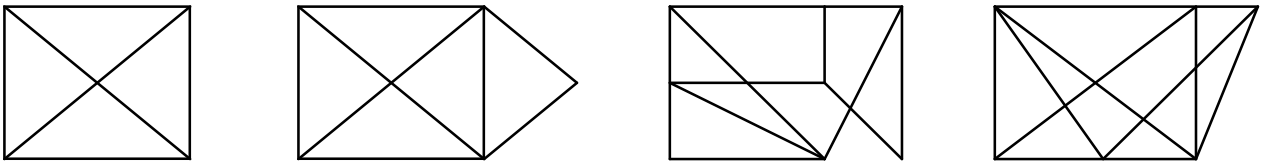
TD 2 –Plus courts chemins et flots (Annoté)

Questions de cours

Chemin Eulérien

Est-il possible de tracer les figures suivantes sans lever le crayon (et sans passer deux fois sur le même trait !…) ? Pourquoi ?



Corrigé :

De tels tracés sont possibles si le graphe correspondant admet un chemin eulérien, c'est-à-dire s'il contient exactement 0 ou 2 sommets de degré impair. La réponse est donc positive uniquement pour la deuxième figure…

Graphe Eulérien

Soit *G* un graphe non eulérien. Est-il toujours possible de rendre *G* eulérien en lui rajoutant un sommet et quelques arêtes ?

Corrigé :

Pour qu'un graphe soit eulérien, il faut et il suffit que tous ses sommets soient de degré pair. Si un graphe contient k sommets impairs, il est possible de rajouter un nouveau sommet x, relié à ces k sommets. Dans le graphe obtenu, les k sommets considérés sont devenus pairs… Cependant, le degré de x étant k, le graphe n'est toujours pas eulérien si k était impair…

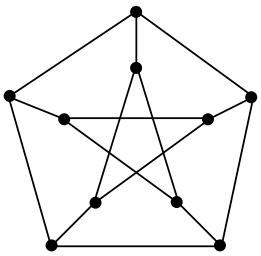
Remarquons qu'il est possible de rajouter des arêtes entre les sommets de degré impair dans le graphe d'origine… Mais l'ajout d'une telle arête, entre deux sommets impairs a et b par exemple, fait que le nombre de sommets impairs devient k-2, qui a la même parité que k…

La réponse est donc : ce n'est possible que si le nombre de sommets impairs est pair…

Graphes Hamiltoniens

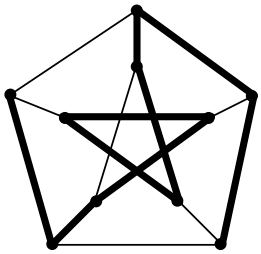
Est-il possible de parcourir le graphe ci-dessous

- en passant une et une seule fois par chacun des sommets et en revenant à son point de départ ?
- sans revenir nécessairement à son point de départ ?



Corrigé :

Le graphe en question (connu sous le nom de graphe de Petersen) n'est pas hamiltonien. On peut par contre trouver un chemin hamiltonien, comme l'indique la figure ci-dessous.



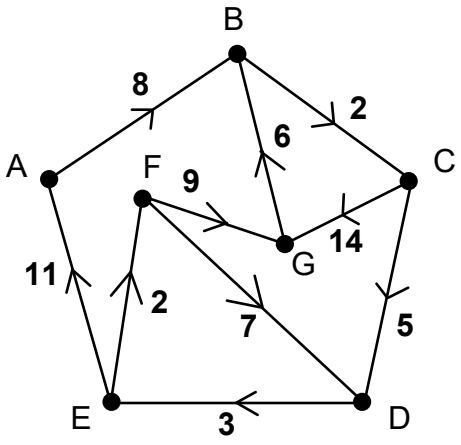
Exercice 1 (Plus court chemin - Dijkstra)

1. Remplir le tableau suivant qui, pour le graphe valué ci-dessous, donne la valeur du plus court chemin d'un sommet à un autre.

2. Exécuter l'algorithme de Dijkstra sur le graphe précédent, à partir du sommet C, puis à partir du sommet F.

3. Exécuter sur le même graphe les algorithmes de Floyd et de Bellman-Ford. Comparer entre les trois algorithmes en terme de complexité.

	A	**B**	**C**	**D**	**E**	**F**	**G**
A							
B							
C							
D							
E							
F							
G							



Corrigé :

Le calcul peut se faire directement… On obtient le tableau suivant :

	A	B	C	D	E	F	G
A	0	8	10	15	18	20	24
B	21	0	2	7	10	12	16
C	19	20	0	5	8	10	14
D	14	20	22	0	3	5	14
E	11	17	19	9	0	2	11
F	21	15	17	7	10	0	9
G	27	6	8	13	16	18	0

À partir du sommet C, nous obtenons :

Initial : poids = ∞, ∞, 0, ∞, ∞, ∞, ∞ venant de : -, -, C, -, -, -, - $\Pi = \varnothing$

choix de C : poids = ∞, ∞, 0, 5, ∞, ∞, 14 venant de : -, -, C, C, -, -, C $\Pi = \{C\}$

choix de D : poids = ∞, ∞, 0, 5, 8, ∞, 14 venant de : -, -, C, C, D, -, C Π = {C,D}

choix de E : poids = 19, ∞, 0, 5, 8, 10, 14 venant de : E, -, C, C, D, E, C Π = {C,D,E}

choix de F : poids = 19, ∞, 0, 5, 8, 10, 14 venant de : E, -, C, C, D, E, C Π = {C,D,E,F}

choix de G : poids = 19, 20, 0, 5, 8, 10, 14 venant de : E, G, C, C, D, E, C Π = {C,D,E,F,G}

choix de A : poids = 19, 20, 0, 5, 8, 10, 14 venant de : E, G, C, C, D, E, C Π = {A,C,D,E,F,G}

choix de B : poids = 19, 20, 0, 5, 8, 10, 14 venant de : E, G, C, C, D, E, C Π = {A,B,C,D,E,F,G}

fin de l'algorithme….

À partir du sommet F, nous obtenons :

Initial : poids = ∞, ∞, ∞, ∞, ∞, 0, ∞ venant de : -, -, -, -, -, F, - $\Pi = \varnothing$

choix de F : poids = ∞, ∞, ∞, 7, ∞, 0, 9 venant de : -, -, -, F, -, F, F $\Pi = \{F\}$

choix de D : poids = ∞, ∞, ∞, 7, 10, 0, 9 venant de : -, -, -, F, D, F, F $\Pi = \{D,F\}$

choix de G : poids = ∞, 15, ∞, 7, 10, 0, 9 venant de : -, G, -, F, D, F, F Π = {D,F,G}

choix de E : poids = 21, 15, ∞, 7, 10, 0, 9 venant de : E, G, -, F, D, F, F Π = {D,E,F,G}

choix de B : poids = 21, 15, 17, 7, 10, 0, 9 venant de : E, G, B, F, D, F, F Π = {B,D,E,F,G}

choix de C : poids = 21, 15, 17, 7, 10, 0, 9 venant de : E, G, B, F, D, F, F Π = {B,C,D,E,F,G}

choix de A : poids = 21, 15, 17, 7, 10, 0, 9 venant de : E, G, B, F, D, F, F Π = {A,B,C,D,E,F,G}

fin de l'algorithme….

Algorithme de Dijkstra

A partir de C

S	A	B	C	D	E	F	G
{}	∞	∞	**0**	∞	∞	∞	∞
{C}	∞	∞	-	**5**	∞	∞	14
{C,D}	∞	∞	-	-	**8**	∞	14
{C,D,E}	19	∞	-	-	-	**10**	14
{C, D, E, F}	19	∞	-	-	-	-	**14**
{C, D, E, F, G}	**19**	20	-	-	-	-	-
{A, C, D, E, F, G}	-	**20**	-	-	-	-	-
{A, B, C, D, E, F, G}	-	-	-	-	-	-	-

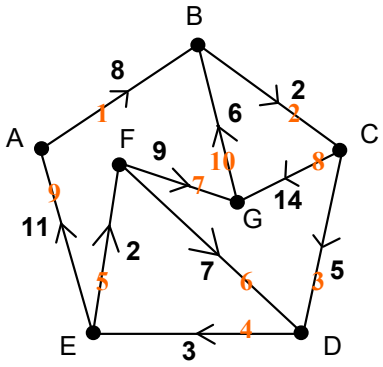
A partir de F

S	A	B	C	D	E	F	G
{}	∞	∞	∞	∞	∞	0	∞
{F}	∞	∞	∞	**7**	∞	-	9
{D, F}	∞	∞	∞	-	10	-	**9**
{D, F, G}	∞	15	∞	-	**10**	-	-
{D, E, F, G}	21	**15**	∞	-	-	-	-
{B, D, E, F, G}	21	-	**17**	-	-	-	-
{B, C, D, E, F, G}	**21**	-	-	-	-	-	-
{A, B, C, D, E, F, G}	-	-	-	-	-	-	-

On peut ici remarquer que toute sous-structure d'un plus court chemin est aussi un plus court chemin.

Algorithme de Bellman-Ford

D'abord numéroter les arcs selon un ordre quelconque.



Ensuite dérouler l'algorithme à partir de C et de F.

A partir de C

A	B	C	D	E	F	G	Itération i
∞	∞	0	∞	∞	∞	∞	***i=1***

∞	∞	0	5	∞	∞	∞	
∞	∞	0	5	8	∞	∞	
∞	∞	0	5	8	10	∞	
∞	∞	0	5	8	10	19	
∞	∞	0	5	8	10	14	
19	∞	0	5	8	10	14	
19	20	0	5	8	10	14	

A partir de F

A	B	C	D	E	F	G	Itération i
∞	∞	∞	∞	∞	0	∞	***i=1***
∞	∞	∞	7	∞	0	∞	
∞	∞	∞	7	∞	0	9	
∞	15	∞	7	∞	0	9	
∞	15	17	7	∞	0	9	***i=2***
∞	15	17	7	10	0	9	
∞	15	17	7	10	0	9	

Nous comparons ensuite la complexité des trois algorithmes

Analyse de l'algorithme de Dijkstra
Soit un Graphe G=(X, A) et Q la file contenant les élèments de X-S

Rappel de l'algorithme

While Q ≠∅ do
u←Extract-min (Q)
S←S ∪ {u}
For each v ∈Adjacent [u] do
If d[v] > d[u]+w(u,v)
Then d[v]←d[u]+w(u,v)

degré (u) fois

|X| fois

Handshaking Lemma : Θ (|A|) x Implicit Decrease Key

Temps= Θ(|X|. $T_{Extract\text{-}min}$ + ||. $T_{Decrease\ Key}$)
NB. Même formule utilisée dans l'analyse de l'algorithme de Prim d'arbre recouvrent minimal.

Rappels de la complexité des structures de données

Operation	List (Liste Chainée)	Binary Heap (Tas binaire)	Fibonacci heap
Insert	O (1)	O (log n)	O (1)
Access min	O (n)	O (1)	O (1)
Delete min	O (n)	O (log n)	O (log n)*
Decrease key	O (1)	O (log n)	O (1)*

Delete	**O (n)**	**O (log n)**	**O (log n)***
merge	**O (1)**	**O (m log (n+m))**	**O (1)**

Temps=Θ(|X|). $T_{Extract\text{-}min}$ + Θ (|A|). $T_{DecreaseKey}$

Q	$T_{Extract\text{-}min}$	$T_{DecreaseKey}$	Total
Tableau	**O (|X|)**	**O (1)**	**O ($|X|^2$)**
Tas Binaire	**O (log |X|)**	**O (log |X|)**	**O (|A| log |X|)**
Tas de Fibonnacci	**O (log |X|)**	**O (1)**	**O (|A| + |X| log |X|)**
	amortie	**amortie**	**Pire cas**

Lorsque le graphe est non valué l'algorithme de Dijkstra peut être amélioré par l'utilisation d'une simple file FIFO (au lieu de la file à priorité).

```
While Q ≠∅ do
        u←Dequeue (Q)
        S←S ∪ {u}
        For each v ∈Adjacent [u] do        } degré (u) fois        } |X| fois
                If d[v] = ∞
                Then d[v]←d[u]+1
                Enqueue (Q, v)
```

L'analyse du temps dans le cas non value: Θ (|X| + |A|)

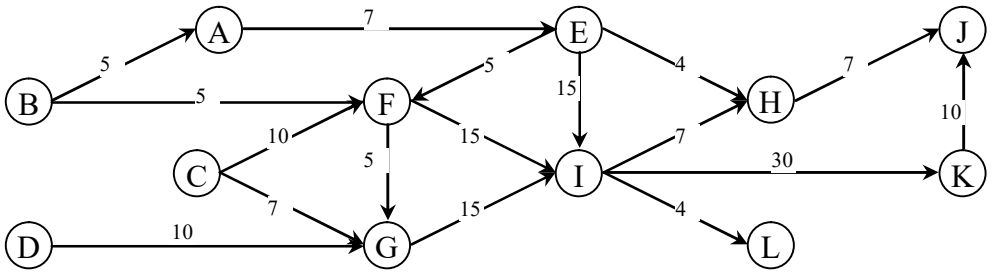
Single source Shortest Path

Cas d'arcs avec poids non négatifs : Algorithme de Dijkstra : O (|A| + |X| log |X|)
Cas général : Algorithme de Bellman Ford (BFA) : O (|X| . |A|)
Cas de graphes non valués (BFA): O (|X| + |A|)

All pairs Shortest Path
Cas de graphes non valués : |X| x BFS : O (|X| |A|)
Cas d'arcs avec poids non négatifs : |X| x Algorithme de Dijkstra : O(|X||A| + $|X|^2$ log |X|)
Cas général :
|X| x Algorithme de Bellman Ford (BFA) : O ($|X|^2$. |A|)
Dans un graphe dense : |A|=Θ($|X|^2$) → O ($|X|^4$)

Exercice 2 (Problèmes de flots)

Trois sommets J, K et L représentent trois stations relais alimentées par quatre stations de base A, B, C et D (Ces stations de base sont reliées via un réseau maillé de stations relais). Les capacités disponibles de ces stations de base sont de 15 Gbps pour A, C et D et de 10 Gbps pour B. Le réseau maillé de distribution, comprenant plusieurs stations relais fixes installées progressivement pour permettre d'augmenter la portée du réseau sans fil. Ce réseau est schématisé par le graphe ci-dessous (les débits maximaux, en Gbps, sont indiqués sur chaque arc). La capacité de chaque lien dépend fortement de la distance entre les relais et de l'environnement physique.



Les zones couvertes par les trois stations relais, sont en pleine évolution, désirent améliorer leur réseau d'alimentation afin de satisfaire des besoins futurs plus importants. Une étude a été faite et a permis de déterminer les demandes maximales probables, à savoir, pour le relais J, 15 Gbps, pour le relais K, 20 Gbps et 15 Gbps pour le relais L.

1. Déterminer la valeur du flot maximal pouvant passer dans le réseau actuel et donnez la coupe minimale correspondante

2. La valeur de ce flot est jugée nettement insuffisante, ainsi l'opérateur décide d'augmenter la capacité de la station de base A (en augmentant la capacité du lien AE) et du relais I (en augmentant la capacité du lien IL). Déterminer les capacités à prévoir pour ces deux liens et la valeur du nouveau flot optimal.

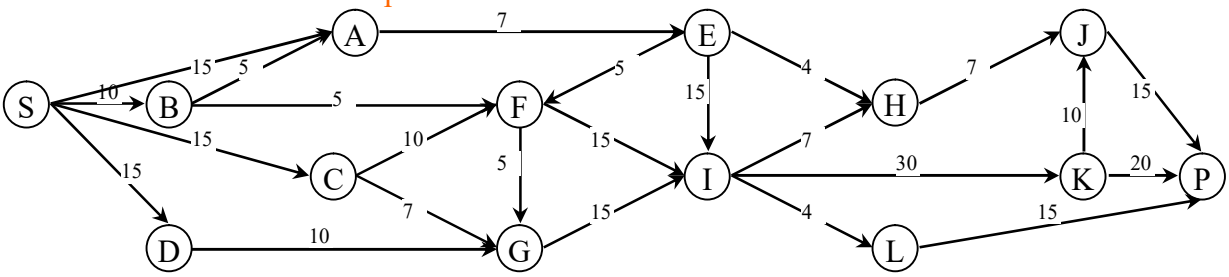
3. Devant le coût des travaux, l'opérateur décide de ne pas remplacer les deux stations en même temps. Dans quel ordre doit-on entreprendre le remplacement de façon à augmenter, après chaque tranche de travaux, la valeur du flot optimal passant dans le réseau ?

4. Quelles sont, après chaque tranche de travaux, les valeurs des flots optimaux ?

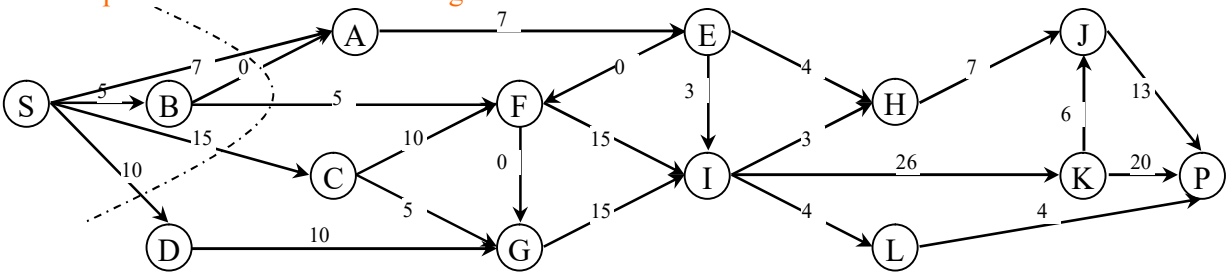
Corrigé :

1. On ajoute une source et un puit fictifs.

On identifie le réseau de transport suivant :



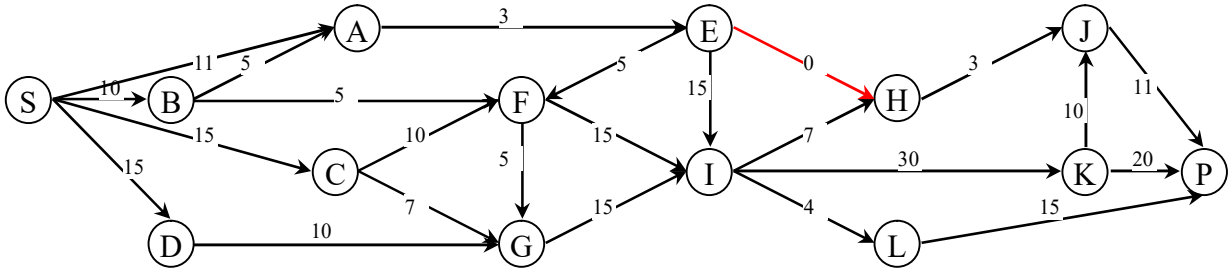
Le flot optimal obtenu en utilisant l'algorithme de Ford-Fulkerson :



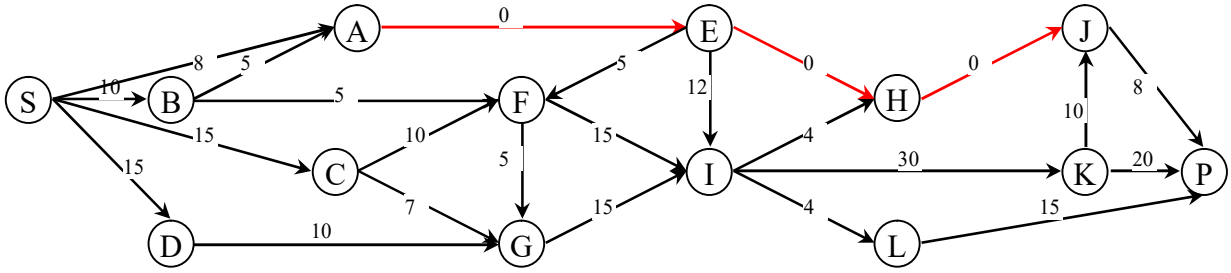
Le flot optimal a pour valeur 37. La coupe minimale est indiquée en traits pointillés

Ci-dessous les étapes d'application de Ford-Fulkerson.

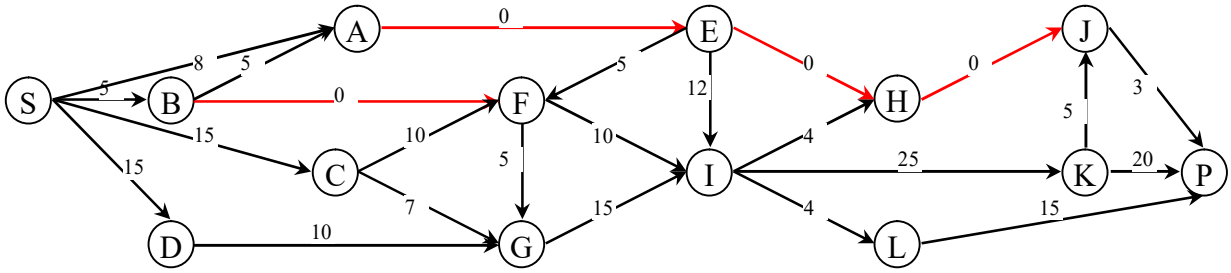
SAEHJP :4



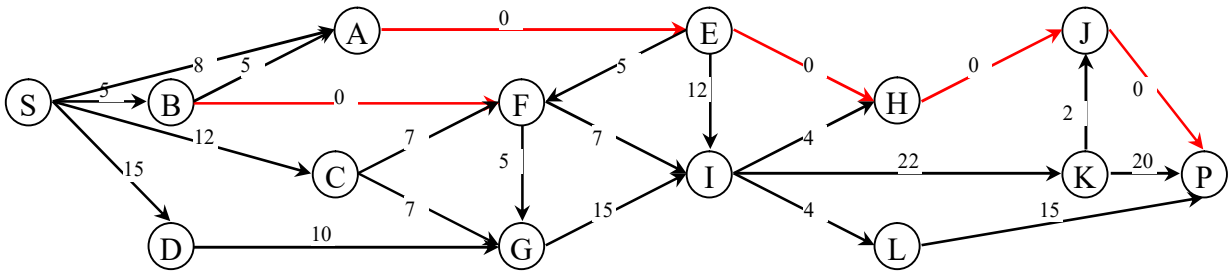
SAEIHJP : 3



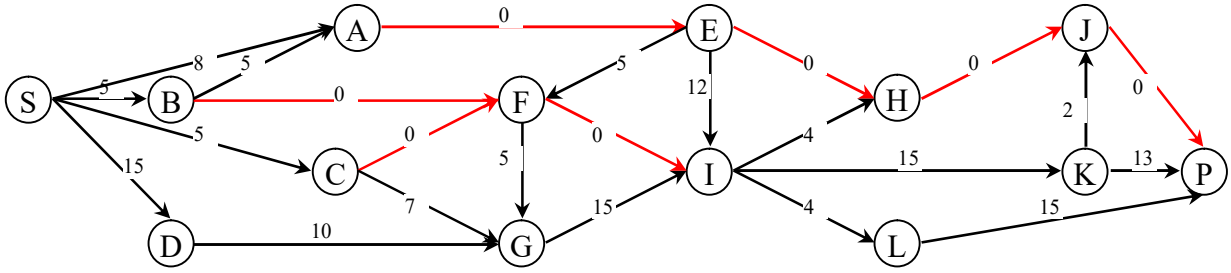
SBFIKJP :5



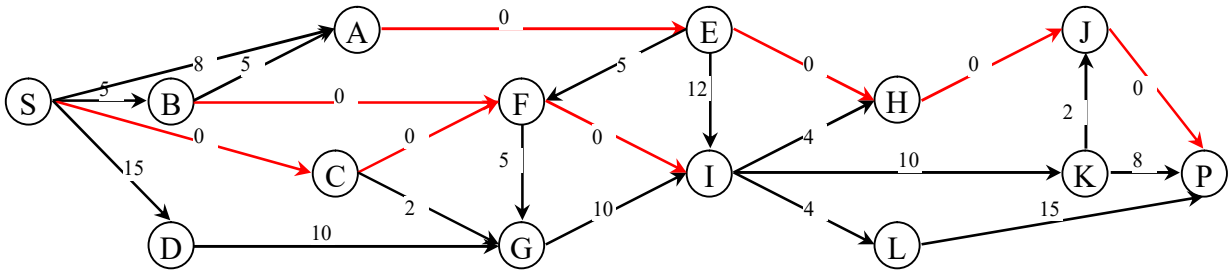
SCFIKJP :3



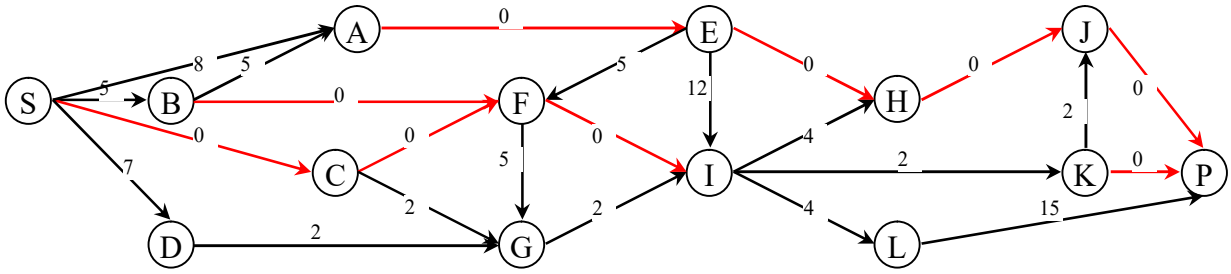
SCFIKP :7



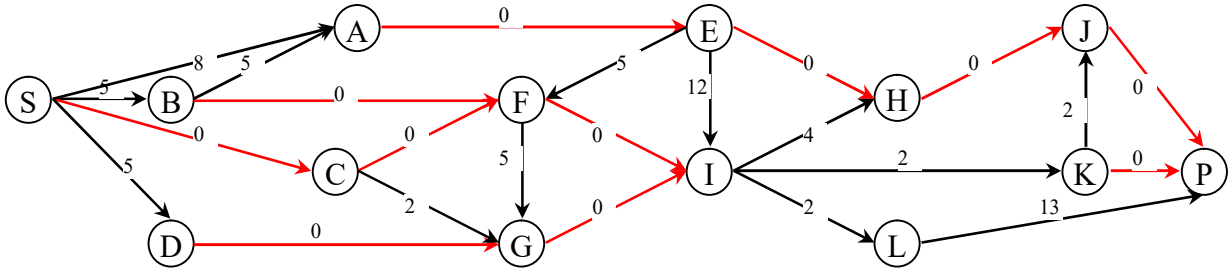
SCGIKP:5



SDGIKP:8



SDGILP:2



2. On désire augmenter les capacités des liens (AE) et (IL). Sachant que :

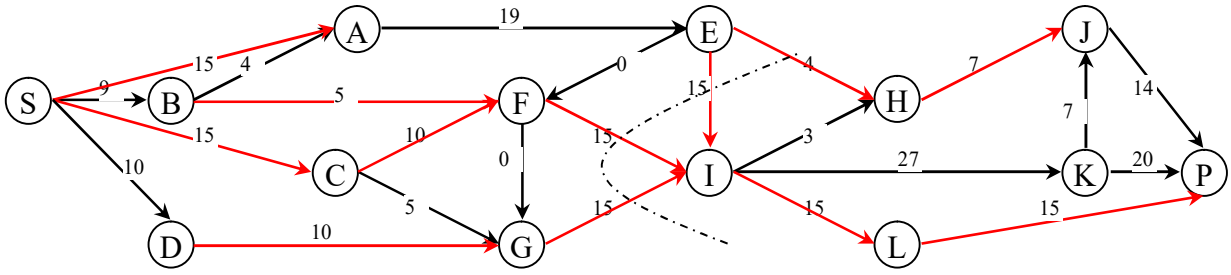
- Il ne peut arriver au maximum que 15+5= 20Gbps en A
- Il ne peut partir au maximum que 5+15+4=24 Gbps de E

Le flot sur l'arc AE pourra être égal au plus à min (20, 24) soit 20 Gbps.
De même,

- Il ne peut arriver au maximum que 15+15= 30Gbps en I
- Il ne peut partir au maximum que 15 Gbps de L

Le flot sur l'arc AE pourra être égal au plus à min (30, 15) soit 15 Gbps.

On cherche alors un nouveau flot maximal avec ces nouvelles capacités maximales.



Le nouveau flot optimal a pour valeur 49. La coupe minimale est indiquée en traits pointillés

3. Après l'augmentation de la capacité du lien (AE) à 20 Gbps, le flot optimal est de 39 Gbps et l'arc (IL) appartient à la coupe minimale. Après l'augmentation de la capacité du lien (IL) à 15 Gbps, le flot optimal est de 49 Gbps.