Introduction:

Ce problème du voyageur de commerce est un problème de recherche opérationnelle et n'est pas sans rappeler les cycles hamiltoniens. En effet, le nom donné à ce problème du voyageur de commerce a pour origine le fait qu'un circuit hamiltonien est celui que suivrait un voyageur de commerce partant d'une ville pour traverser une seule et unique fois n autres villes et revenir à son point de départ et ce à cause de son travail. Son objectif est rationnellement d'effectuer ces trajets, ce circuit en un temps minimal, une distance minimale ou un coût minimal. Pour parvenir à ses fins, le voyageur commercial peut bien sûr effectuer des essais successifs différents en notant à chaque essai, le temps nécessaire, le coût et la distance. Il pourra certainement trouver la solution mais cela pourra lui prendre d'autant plus de temps que le nombre de villes à traverser est lui aussi grand. Cette façon de procéder devient donc vite irréalisable.

Mais avant de poursuivre sur le problème, on abordera dans un premier temps les origines du problème avec Sir William Rowan Hamilton. Ensuite par poser le problème et la problématique sous jacente puis on énoncera des méthodes de résolution. Et pour finir, on va donner des exemples qui appliquent une de ces méthodes de résolution sur quelques exemples ou exercices.

i. **Les origines du problème du voyageur de commerce**

ii. **Le problème du voyageur de commerce**

iii. **Méthodes de résolution**

iv. **Exemples de résolution du problème du voyageur de commerce**

-

a. Les origines du problème du voyageur de commerce

C’est en 1859 que la notion de cycle hamiltonien est apparue et ce à la suite du lancement par l’illustre mathématicien Sir William Rowan Hamilton d’un casse tête, celui du voyage fermé autour du monde, principalement composé d’un dodécaèdre régulier en bois. C’est l’un des soldes de Platon, un polyèdre dont les 12 faces sont des pentagones réguliers et dont 3 arêtes de ces pentagones se rencontrent à chacun des sommets.

Dans son casse tête, Hamilton a donné à chacun des 20 sommets du dodécaèdre le nom d’une grande ville du monde : Bruxelles, Francfort, Delhi etc. ce qui n’est pas sans faire penser au tour du monde en 80 jours écris par Jules Verne en 1873. L’objectif était alors de passer une seule et unique fois dans chaque ville et ce en n’utilisant que les arêtes du dodécaèdre mais en partant d’une ville pour au final y revenir. Cela revient donc à chercher un cycle Hamiltonien dans la figure car le dodécaèdre étant assez difficilement manipulable, Sir Hamilton la remplacer par un graphe planaire isomorphe. En effet, la difficulté de manipulation venait du fait que pour noter les passages, on pouvait planter un clou sur chaque sommet et faire passer une ficelle.

Rien ne peut amener la preuve du succès de ce dodécaèdre auprès du public mais les mathématiciens ont immortalisés son souvenir en définissant un cycle hamiltonien comme étant un cycle élémentaire qui passe une fois et une seule par chacun des sommets d’un graphe (généralement, il ne passe pas par toutes les arêtes, il n’en emprunte que deux à chaque sommet).

Et c’est donc de ce problème que le problème du voyageur de commerce est tiré.

Définissons ce que sont les circuits et cycles hamiltoniens.

Soit G = [X, U] un graphe connexe d’ordre N

On appelle chemin hamiltonien (chaîne hamiltonienne) un chemin (une chaîne) passant une fois, et une seule fois seulement, par chacun des sommets de G.

Un chemin hamiltonien (une chaîne hamiltonienne) est donc élémentaire et de longueur N-1.

Un circuit hamiltonien (un cycle hamiltonien) est un circuit (un cycle) qui passe une fois, et une seule fois seulement, par chacun des sommets de G.

Un cycle hamiltonien est donc un cycle élémentaire de longueur N.

On dit qu’un graphe G est hamiltonien s’il contient un cycle hamiltonien (cas non orienté) ou un circuit hamiltonien (cas orienté).

2. Le problème du voyageur de commerce

Le problème du voyageur de commerce ou problème de circuits hamiltoniens (cycle hamiltonien) de longueur minimale se rencontre dans de très nombreux et dans des domaines variés : ramassage scolaire, ordonnancement de production, câblage de circuits, synthèse de circuits logiques séquentiels etc. mais il peut également être étendu au cas multiple où plusieurs voyageurs de commerce doivent couvrir une certaine zone. Toutefois, dans le cas d'un seul voyageur de commerce, il faut faire la distinction entre deux cas de figure, le cas non orienté et le cas orienté lequel peut d'ailleurs se ramener à un problème non orienté.

Le cas orienté est caractérisé par le fait que les arcs séparant des sommets vont représenter des contraintes d'antériorités c'est à dire que le début d'exécution d'une tache j ne pourra se faire après une certaine durée suivant le début d'exécution d'une tache i de sorte que par exemple. La tache j ne pourra débuter après que i soit exécutée.

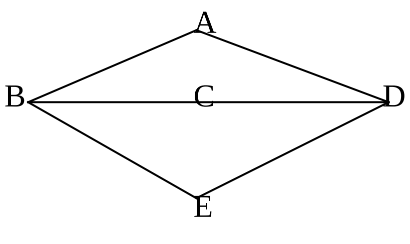
Dans le problème du voyageur de commerce, on a donc un graphe connexe G = [X,U] dans lequel chaque arc u= (i,j) est muni d'une longueur L(i,j) et ce graphe est d'ordre N.

X est l'ensemble des sommets.
U est l'ensemble des arcs.

Dès lors que l'on a un problème à résoudre et que l'on a un graphe associé, pour pouvoir trouver le chemin de longueur minimal passant par tous les sommets c'est à dire une chaîne hamiltonienne, il faut que des conditions nécessaires et suffisantes soient validées afin que la résolution puisse être mise en œuvre. Mais il n'existe pas de conditions nécessaires et suffisantes mais seulement des conditions nécessaires et des conditions suffisantes.

- Voyons tout d'abord les conditions nécessaires (CN).

Une CN pour qu'un graphe G=X, U non orienté soit hamiltonien est que ce graphe soit 2-connexe c'est à dire que dans le graphe, on peut trouver deux chaînes allants de n'importe quel sommet à n'importe quel autre. Ceci est évident mais cette condition n'est pas suffisante. C'est une CN car s'il n'y avait qu'une seule chaîne, le chemin minimal se réduirait à cette chaîne et cette solution serait non seulement évidente mais inévitable et de plus il n'y aurait plus de critère de minimisation.



C'est un graphe 2-connexe mais il ne contient pas de cycle hamiltonien car en effet, on ne peut passer chaque sommet une seule et unique fois en partant d'un point pour y revenir. On est obligé de passer 2 fois par 2 sommets.

Une autre condition nécessaire pour qu'un plus court chemin existe est que le graphe associé au problème ne comporte pas de longueur d'arc négative.

Et dans le cas du problème du voyageur de commerce, il faut que le graphe associé au problème soit complet c'est à dire que tous les couples de sommet sont reliés au moins par un arc donc si (i,j) n'existe pas alors (j,i) existe.

Il existe d'autres CN mais ne connaissant pas de conditions nécessaires et suffisantes voyons plutôt des conditions suffisantes CS.

CS dans le cas non orienté :

Proposition 1 : le graphe complet d'ordre N, K_N est hamiltonien.
Si le graphe est complet alors tous les couples de sommets sont reliés au moins par un arc et s'il est d'ordre N alors il y a N arcs donc il est forcement hamiltonien.

Soit G=[X,U] un graphe non orienté d'ordre N et supposons qu'il existe deux sommets s et t non reliés par un arc tel que :

$$d_G(s) + d_G(t) \geq N$$

Soit G +(s,t) la notation pour le graphe obtenue en rajoutant l'arc (s,t) à G

Proposition 2 : si G +(s,t) est hamiltonien, G est lui même hamiltonien.
On dit alors que la propriété P : « être hamiltonien » est stable par la transformation
$G \rightarrow G+(s,t)$

Soit G un graphe simple d'ordre N et k un nombre entier 1 k N.

On considère la procédure récursive qui suit : à partir de G on ajoute progressivement des arêtes joignant des sommets non adjacents dont la somme des degrés est supérieure ou égale à k.

Ce graphe ainsi obtenue qui contient le graphe G sera appelé la k-fermeture de G et sera noté $[G]_N$

Théorème (Bondy, Chvatal 1976) : pour que G soit hamiltonien, il faut et il suffit que $[G]_N$ soit hamiltonien.

Théorème (Erdös- Chvatal 1972) : soit G un graphe h-connexe ($h \geq 2$). Si le nombre de stabilité $\alpha(G)$ vérifie : $\alpha(G) \leq h$, alors G est hamiltonien.

Lemme : si G est h-connexe et si $A=\{a1,\ldots,ah\} \subset X$ est un ensemble de h sommets distincts ; pour tout sommet $x \in X-A$, il existe h chaînes élémentaires joignant x à chacun des ai et ayant x comme seul sommet commun.

Corollaire : si $\alpha([G]_N) \leq h([G]_N)$, alors G est hamiltonien.

Voyons maintenant le cas orienté.

Théorème (Meyniel 1973) : soit G = [X,U] un graphe fortement connexe sans boucle d'ordre N tel que :
$(i,j) \notin U$ et $(j,i) \notin U \Rightarrow d_G(i) + d_G(j) \geq 2N-1$
alors G contient un circuit hamiltonien.

3. Méthodes de résolution

En 1963, J.D.C. Little, Lawler, E.L. et Wood, D.E. ont présente une méthode rigoureuse d'optimisation pour ce problème

Dans le problème du voyageur de commerce, on a donc un graphe connexe G = [X,U] dans lequel chaque arc u= (i,j) est muni d'une longueur L(i,j) et ce graphe est d'ordre N où X est l'ensemble des sommets et u l'ensemble des ars. Et on va chercher un chemin minimal passant par tous les sommets une seule et unique fois sauf pour le sommet de départ sur lequel on revient. On cherche donc à minimiser la somme des distances parcourues entre les sommets d'une chaîne élémentaire.

$$\text{Soit} \begin{cases} \underset{i,j\ (i\neq j)}{\text{Min}} \ \Sigma\, L(i,j) \\ i,j \in X \text{ mais tel qu'on ne prenne qu'une seule et unique fois chaque sommet.} \end{cases}$$

Etant donne que c'est un problème d'ordonnancement, on va procéder par étape car le problème ne pourrait être résolu directement. On va donc initialiser et prendre le sommet de départ s'il est donné comme sommet initial puis on va effectuer des itérations, au nombre de N. On va donc prendre les sommets un à un et minimiser un critère à chaque étape en fonction des étapes antérieures.

On va démarrer à un sommet particulier noté t qui sera alors l'extrémité du parcours. De là pour tout ensemble de sommet $S \subset X$ et contenant t, on désigne par F(S,k) la longueur d'un plus court chemin d'origine k ($k \in S$) et d'extrémité t et passant une seule et unique fois par tous les sommets de S. par convention $F(S,k) = +\infty$ si un tel chemin n'existe pas.

$$(1) \begin{cases} F(S,k) = \underset{k' \in \Gamma(k) \cap S}{\text{Min}} \{ L(k,k') + F(S-\{k\},k')\} \\ F(\{t\},t) = 0 \\ F(\{t\},k) = +\infty \ \forall k \neq t \end{cases}$$

On voit que pour tout ensemble S de cardinalité p, les valeurs F(S,k) peuvent être déterminée à partir des valeurs F(S',k') pour tous les ensembles S' de cardinalité p-1. la valeur optimale du problème est :

$$\underset{k' \in \Gamma(k) \cap S}{\text{Min}} \{ L(k,k') + F(S-\{k\},k')\}$$

Le processus de résolution par programmation dynamique comporte N étapes (où N est le nombre de sommet du graphe), et l'ensemble des états à chaque étape i correspond à tous les couples (S,k) où S est un sous ensemble de X contenant t et de cardinalité i, et $k \in S$.

Le nombre d'états augmente donc exponentiellement en fonction de la taille du problème c'est à dire le nombre de sommets du graphe et c'est la raison pour laquelle le problème ne peut être généralement résolu par l'équation (1).

Pour le résoudre, il existe des techniques de réduction des calculs en programmation dynamique. Pour des problèmes conduisant à un espace d'état de cardinalité très élevée et en particulier lorsque le nombre de contraintes c'est à dire la dimension de l'espace d'état est élevée, des techniques ont été développées soit pour réduire la dimension de l'espace d'état, soit pour éliminer du calcul une fraction importante des états a priori possibles. Parmi celles ci, il y a :

- La technique des multiplicateurs de Lagrange (relaxation lagrangienne des contraintes), Bellman (1956) et Everett(1963).
- Les méthodes de séparation évaluation (Branch and Bound)
- Algorithme de recherche admissible utilisé pour la recherche de plus court ou de plus long chemin dans des graphes de grande dimension (Hart, Nilson et Raphael 1968, Nilson 1971, Martelli 1977)
- Les méthodes de relaxation de l’espace d’état (Christofidis, Minjozzi et Toth 1979, 1981)

Voyons donc le principe de l’algorithme développe par Little, algorithme permettant de déterminer un circuit hamiltonien minimal.
A chaque arc (i, j) est attaché une valeur vij ≥ 0 et pour chaque couple sans arc, est attachée la valeur ∝

b) Enlever à chaque ligne et / ou chaque colonne de la matrice [vij], le plus petit élément de la ligne(respectivement de la colonne) jusqu’à épuisement. Ceci donne une nouvelle matrice [vij’] qui contient au moins un zéro par ligne et par colonne
c) Calculer la somme de tous les éléments soustraits des lignes et /ou colonnes de [vij]. Ceci donne une borne inférieure de la racine de l’arborescence, racine correspondant à l’ensemble de toutes les solutions
d) Choisir (k, l) pour réaliser une bipartition(branchement) de telle sorte que :

$$\theta\,(k, l) = \max_{i,j} \Omega\,(i, j)$$

où Ω (i, j) est la somme du plus petit éléments de la ligne i en omettant v(i,j) et le plus petit éléments de la colonne j en omettant v(i,j), ce Ω (i, j) étant calculé pour tous les vij=0

e) Mettre en place un sommet et un arc dans l’arborescence correspondant à la propriété Pij* : les circuits ne passent pas par (k,l). Donner à ce sommet une borne égale à la borne du sommet antérieur plus la valeur θ (k, l)
f) Mettre en place un sommet et un arc dans l’arborescence correspondant à la propriété Pij : les circuits passent par (k,l). supprimer de la matrice la ligne k et la colonne l. placer ∝ une valeur dans toutes les cases correspondant à un arc qui réaliserait un circuit de longueur inférieur à n où n est le nombre de sommet initialement donné.

PS : avec ∝ dans case, on empêche que la solution ne soit pas conforme à l’hypothèse : tout circuit doit être hamiltonien.

g) Enlever à chaque ligne et / ou chaque colonne de la matrice le plus petit éléments de la ligne(respectivement de la colonne) jusqu'à épuisement. Ceci donne une matrice qui contient au moins un zéro par ligne et par colonne.

h) Calculer la somme de tous les éléments soustraits des lignes et /ou colonnes de la matrice. Ajouter cette quantité à la borne obtenue au sommet antérieur de l'arborescence :on obtient alors la borne relative au sommet de l'arborescence où la propriété Pij est introduite.

i) La matrice obtenue est elle d'ordre 1*1. si oui, les calculs sont terminés, c'est un circuit hamiltonien minimal. Sinon, on passe à l'étape i)

j) Examiner la valeur des bornes obtenues pour tous les sommets produits et sélectionner la plus petite borne (s'il existe plusieurs solutions, on réalise un choix arbitraire)

k) La borne choisie en i) correspond elle à un sommet pour lequel la propriété de bipartition est une propriété Pij ou une propriété Pij* (le circuit hamiltonien passe par (i,j) et Pij* la propriété contraire) ?

   Si sommet possède propriété Pij, on passe en c)

   Si sommet possède propriété Pij*, on passe en k

l) Dans la matrice équivalent à ce sommet, mettre une valeur dans la case (i,j), où i et j sont ceux de la propriété Pij*. Enlever à la ligne i et à la colonne j leurs plus petits éléments respectifs. Passer maintenant en c)

a. Exemples de résolution du problème du voyageur de commerce

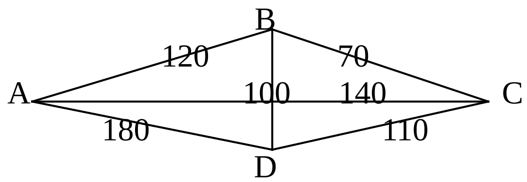
Exercice : un voyageur de commerce vit dans une ville A et doit visiter B, C et D. et on a les données suivantes :

AB=120 AC=140 AD=180

BC=70 BD=100 CD=110

On cherche donc le trajet le plus court qui part de A et y revient après avoir passé les 3 autres points. On a donc la matrice suivante :

	A	B	C	D
A	∝	120	140	180
B	120	∝	70	100
C	140	70	∝	110
D	180	100	110	∝



Ce graphe est 3-connexe donc une CN est verifiée. Mais ce graphe est également complet et aucun arc ne comporte de valeurs négatives donc plusieurs CN sont vérifiées. Vérifions maintenant des CS.

On retire donc 120 à la première ligne, 70 à la seconde et à la troisième et 100 à la dernière mais comme chaque colonne ne contient pas au moins un zéro, on retire alors 50 à la première colonne et 30 à la dernière. Et on obtient la matrice suivante :

	A		B		C		D			element soustrait
A	∝		0		20		30			120
B	0	20	∝		0	10	0	10		70
C	20		0	20	∝		10			70
D	30		0	10	10		∝			100
										somme
element soustrait	50		0		0		30			440

Dans cette matrice, on a également effectué la somme des plus petits éléments de chaque ligne et colonne pour toutes les cases où sont inscrits des zéros. 440 est donc la borne inférieure pour l'ensemble X.

On cherche donc le maximum parmi ces éléments c'est à dire que l'on cherche :
Max Ω (i, j) où est la somme du plus petit éléments de la colonne j en omettant v(i, j) et le plus petit éléments i de la ligne en omettant v(i, j) et on a :

Max Ω (i, j) = 20 et je choisis AB* dont la borne vaut 440+20=460

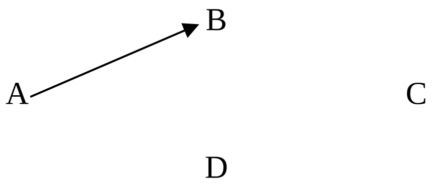
On supprime la ligne A et la colonne B

	A		C		D	
B	0		0		0	
C	20		∝		10	
D	30		10		∝	

Puis on enlève à chaque ligne et/ou colonne le plus petit éléments et on obtient la matrice suivante, et ce après avoir inscrit un ∝ dans la case BA car ce chemin AB serait de longueur inférieur à 4 étant le nombre de sommet. On enlève dont 10 à la deuxième ligne ainsi qu'à la troisième et à la première colonne:

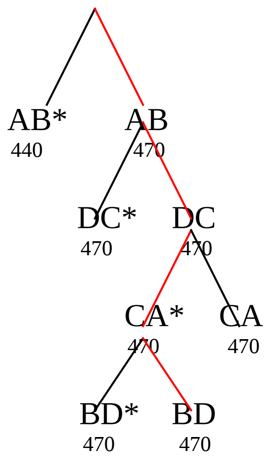
	A		C		D			element soustrait
B	∝		0		0			0
C	0		∝		0			10
D	10		0		∝			10
								somme
element soustrait	10		0		0			30

La borne de AB vaut 440+30 = 470 et on débute par ce sommet car il est imposé de débuter par A. on inscrit ceci dans l'arborescence.



La matrice n'est pas 1*1 donc on reprend l'algorithme à la recherche du Max Ω (i, j) où est la somme du plus petit éléments de la colonne j en omettant v(i, j) et le plus petit éléments i de la ligne en omettant v(i, j).

Arborescence :



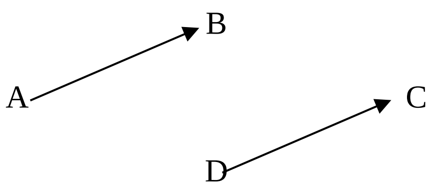
	A		C		D			element soustrait
B	∝		0	0	0	0		0
C	0	10	∝		0	0		0
D	10		0	10	∝			0
								somme
element soustrait	0		0		0			0

Max Ω (i, j) = 10 donc je choisis DC* dont la borne vaut 470+10 = 480

On supprime la ligne D et la colonne C puis on enlève à chaque ligne et/ou colonne le plus petit éléments et on obtient :

	A		D			
B	∝		0			
C	0		∝			0
						somme
						0

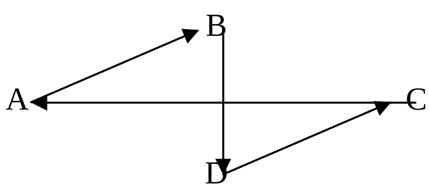
La borne de DC vaut 470 donc on choisit DC qui a la propriété Pij. On enlève de ce fait à chaque ligne et/ou colonne le plus petit éléments ce qui revient à garder la matrice telle qu'elle est .



	A		D			
B	∝		0			
C	0		∝			0
						somme
						0

Je choisis CA* car pour chacun des deux sommets, la valeur maximale est la même et à CA* est associé la borne 470+0=470 et je choisis le sommet associé à la propriété Pij* car il faut retourner en A. Je supprime la colonne A et la ligne C et on obtient une matrice 1*1. c'est fini et la solution du plus court chemin est alors :

	D	
B	0	



{A,B,D,C,A) dont la valeur est de 470 km

	A	B	C	D
A	∝	120	140	180
B	120	∝	70	100
C	140	70	∝	110
D	180	100	110	∝

Prenons un second exemple :

Soit la matrice :

	A	B	C	D	E	F	G
A	∝	5	9	6	3	5	9
B	8	∝	8	8	5	9	2
C	6	9	∝	1	6	7	3
D	7	11	4	∝	4	2	9
E	4	6	3	2	∝	2	8
F	5	2	2	8	4	∝	3
G	8	1	3	16	5	3	∝

Le graphe associé à cette matrice est un graphe orienté qui est h-connexe ($h \geq 2$) et fortement connexe. En effet, dans ce graphe on peut trouver une chaîne allant de n'importe quel sommet à n'importe quel autre et il existe un chemin entre chacun des sommets et ce dans les deux sens. Donc la CN est vérifiée de même que la CS. Le graphe associé possède donc un circuit hamiltonien.

On retire donc 3 à la première ligne, 2 à la seconde, à la quatrième, la cinquième et la sixième et 1 à la troisième et à la dernière mais comme chaque colonne ne contient pas au moins un zéro, on retire alors 2 à la première colonne. Et on obtient la matrice suivante :

	A		B		C		D		E		F		G		
A	∝		2		6		3		0	4	2		6		3
B	4		∝		6		6		3		7		0	4	2
C	3		8		∝		0	2	5		6		2		1
D	3		9		2		∝		2		0	2	7		2
E	0	1	4		1		0	0	∝		0	0	6		2
F	1		0	2	0	1	6		2		∝		1		2
G	5		0	2	2		15		4		2		∝		1
															somme
	2		0		0		0		0		0		0		15

Dans cette matrice, on a également effectué la somme des plus petits éléments de chaque ligne et colonne pour toutes les cases où sont inscrits des zéros. 15 est donc la borne inférieure pour l'ensemble X.

On cherche donc le maximum parmi ces éléments c'est à dire que l'on cherche :

Max Ω (i, j) où est la somme du plus petit éléments de la colonne j en omettant v(i, j) et le plus petit éléments i de la ligne en omettant v(i, j) et on a :

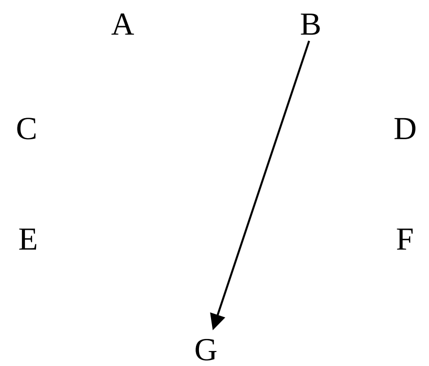
Max Ω (i, j) = 4 et je choisis BG* dont la borne vaut 15+4=19

On supprime la ligne B et la colonne G et on obtient la matrice suivante, et ce après avoir inscrit un ∝ dans la case GB car ce chemin serait de longueur inférieur à 7 étant le nombre de sommet.

	A		B		C		D		E		F			
A	∝		2		6		3		0		2			0
C	3		8		∝		0		5		6			0
D	3		9		2		∝		2		0			0
E	0		4		1		0		∝		0			0
F	1		0		0		6		2		∝			0
G	3		∝		0		13		2		0			2
														somme
	0		0		0		0		0		0			2

La borne relative à BG est 15+2=17

L’ordre de la matrice n’est pas 1*1 donc on prend la plus petite borne soit 17 ce qui correspond à BG qui a la propriété Pij.



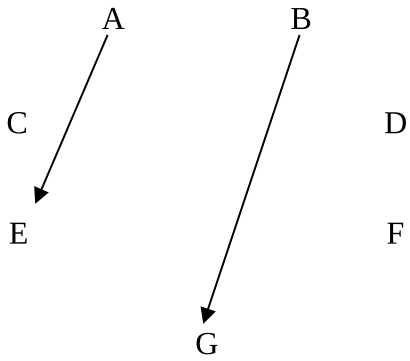
	A		B		C		D		E		F			
A	∝		2		6		3		0	4	2			0
C	3		8		∝		0	3	5		6			0
D	3		9		2		∝		2		0	2		0
E	0	1	4		1		0	0	∝		0	0		0
F	1		0	2	0	1	6		2		∝			0
G	3		∝		0	1	13		2		0	0		0
														somme
	0		0		0		0		0		0			0

Max Ω (i, j) = 4 et je choisis AE* dont la borne vaut 17+4=21

On supprime la ligne A et la colonne E et on obtient la matrice suivante, et ce après avoir inscrit un ∝ dans la case EA car ce chemin serait de longueur inférieur à 7 étant le nombre de sommet.

	A		B		C		D		F			
C	2		8		∝		0		6			0
D	2		9		2		∝		0			0
E	∝		4		1		0		0			0
F	0		0		0		6		∝			0
G	2		∝		0		13		0			0
												somme
	1		0		0		0		0			1

La borne relative à AE est 17+1=18



L’ordre de la matrice n’est pas 1*1 donc on prend la plus petite borne soit 18 et le sommet associé AE a la propriété Pij.

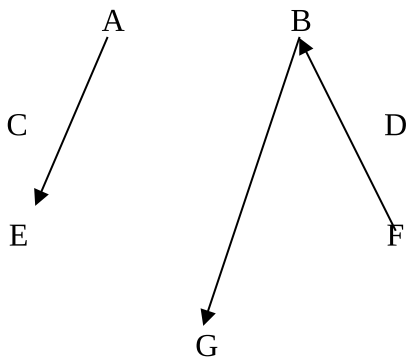
	A		B		C		D		F			
C	2		8		∝		0	3	6			0
D	2		9		2		∝		0	2		0
E	∝		4		1		0	0	0	0		0
F	0	2	0	4	0	0	6		∝			0
G	2		∝		0	0	13		0	0		0
												somme
	0		0		0		0		0			0

Max Ω (i, j) = 4 et je choisis FB* dont la borne vaut 18+4=22

On supprime la ligne F et la colonne B et on obtient la matrice suivante :

	A		C		D		F			
C	0		∝		0		6			0
D	0		2		∝		0			0
E	∝		1		0		0			0
G	0		0		13		0			0
										somme
	2		0		0		0			2

La borne relative à FB est 18+2=20



L’ordre de la matrice n’est pas 1*1 donc on prend la plus petite borne soit 19 et le sommet associé BG* a la propriété Pij*. On enlève donc à la ligne B son plus petit élément et on enlève à la colonne G son plus petit élément.

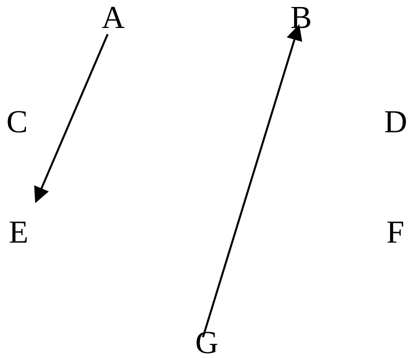
	A		B		C		D		E		F			
A	∝		2		6		3		0	4	2			0
C	3		8		∝		0	3	5		6			0
D	3		9		2		∝		2		0	2		0
E	0	1	4		1		0	0	∝		0	0		0
F	1		0	2	0	0	6		2		∝			0
G	3		∝		0	0	13		2		0	0		0
														somme
	0		0		0		0		0		0			0

Max Ω (i, j) = 4 et je choisis EA* dont la borne vaut 19+4=23

On supprime la ligne E et la colonne A et on obtient la matrice suivante et ce après avoir mis une valeur ∞ à AE car sinon ce chemin serait de longueur inférieur à 7 étant le nombre de sommet.

	B	C	D	E	F		
A	0	4	1	∝	0		2
C	8	∝	0	5	6		0
D	9	2	∝	2	0		0
F	0	0	6	2	∝		0
G	∝	0	13	2	0		0
							somme
	0	0	0	0	0		2

La borne relative à EA est 19+2=21



L'ordre de la matrice n'est pas 1*1 donc on prend la plus petite borne soit 20 et le sommet associé est FB qui a la propriété Pij.

	A		C		D		F			
C	0	0	∝		0	0	6			0
D	0	0	2		∝		0	0		0
E	∝		1		0	0	0	0		0
G	0	0	0	1	13		0	0		0
										somme
	0		0		0		0			0

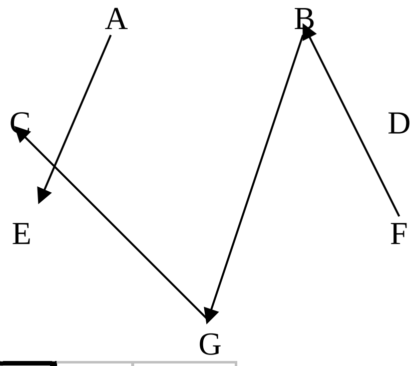
Max Ω (i, j) = 1 soit GC* dont la borne vaut 20+2=22

On supprime la ligne G et la colonne C et on obtient la matrice suivante et ce après avoir inscrit un ∝ dans la case CF car ce chemin serait de longueur inférieur à 7 étant le nombre de sommet.:

	A		D		F			
C	0		0		∝			0
D	0		∝		0			0
E	∝		0		0			0
								somme
	0		0		0			0

La borne relative à GC est 20+0=20

L'ordre de la matrice n'est pas 1*1 donc on prend la plus petite borne soit 20 et le sommet associé est GC qui a la propriété Pij.



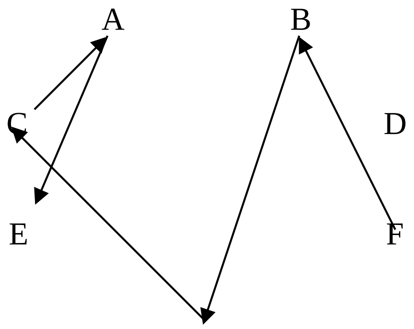
	A		D		F			
C	0	0	0	0	∝			0
D	0	0	∝		0	0		0
E	∝		0	0	0	0		0
								somme
	0		0		0			0

Max Ω (i, j) = 0 et je choisis CA* dont la borne vaut 20+0=20

On supprime la ligne C et la colonne A et on obtient la matrice suivante

	D		F			
D	∝		0			0
E	0		0			0
						somme
	0		0			0

La borne relative à CA est 20+0=20



G

L’ordre de la matrice n’est pas 1*1 donc on prend la plus petite borne soit 20 et je choisis le sommet associé est CA qui a la propriété Pij.

	D		F			
D	∝		0	0		0
E	0	0	0	0		0
						somme
	0		0			0

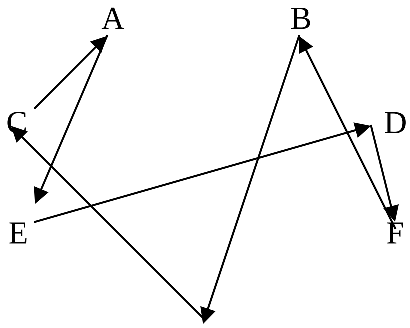
Max Ω (i, j) = 0 et je choisis DF* dont la borne vaut 20+0=20

On supprime la ligne D et la colonne F et on obtient la matrice suivante

	D			
E	0			0
				somme
	0			0

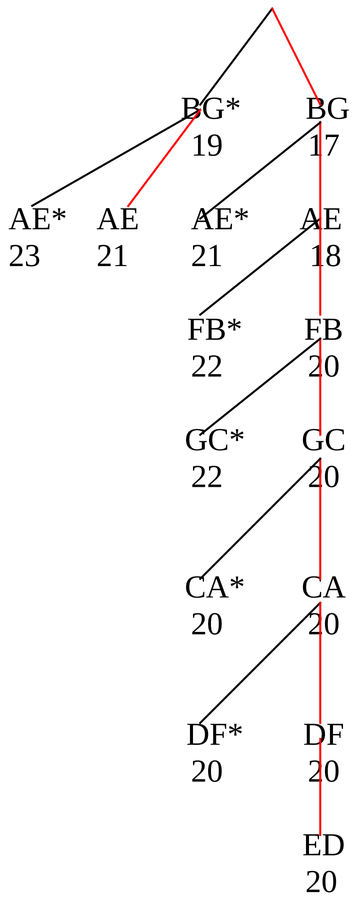
La borne relative à DF est 20+0=20

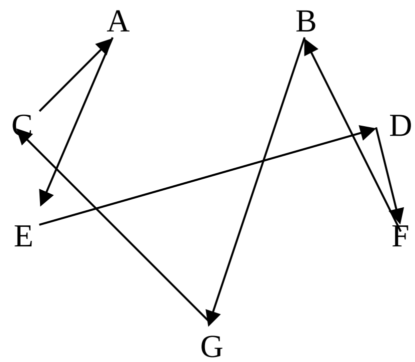
L’ordre de la matrice est d’ordre 1*1 donc c’est fini et on prend la plus petite borne soit 20 et je choisis le sommet associé est DF qui a la propriété Pij.



	A		B		C		D		E		F		G			
A	∝		5		9		6		3		5		9			
B	8		∝		8		8		5		9		2			
C	6		9		∝		1		6		7		3			
D	7		11		4		∝		4		2		9			
E	4		6		3		2		∝		2		8			
F	5		2		2		8		4		∝		3			
G	8		1		3		16		5		3		∝			
	6		2		3		2		3		2		2		somme	20

Arborescence :





Cet algorithme de Little peut également servir à la recherche d’un chemin minimal :

- recherche d’un chemin minimal d’origine Xa
  1) placer ∞ dans la diagonale principale
  2) vij = 0 dans colonne Xa sauf vaa = ∞
  3) opérer ensuite en recherchant circuit hamiltonien minimum
- si on veut connaître chemin minimal quel que soit son origine, on adjoindra au graphe un sommet S relié à chaque Xi par des arcs (S,Xi) et (Xi,S) de valeur 0 puis recherche d’un circuit hamiltonien.