Théorie des graphes

Présenté par :
Pr.Abdelaziz Qaffou

EST-Beni Mellal – Université Sultan Moulay Slimane

DUT: GI et ARI
2ème année-S3-2023-2024

Plan du cours

Outline

Le ponts de Königsberg

Figure 1 – Königsberg en 1652

Introduction

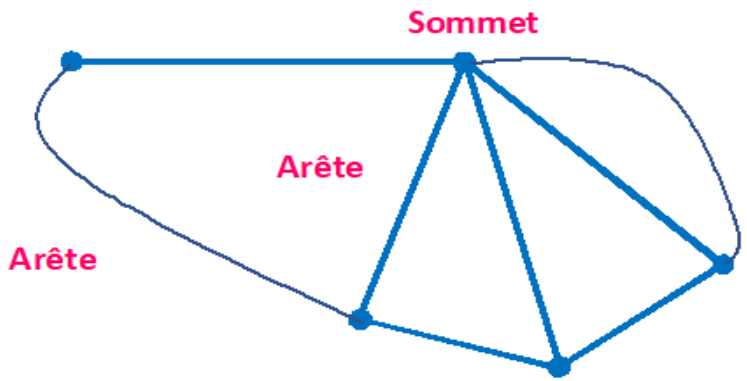
L'histoire de la théorie des graphes débute avec les travaux d'Euler au XVIIIe siècle et trouve son origine dans l'étude de certains problèmes, tels que celui des ponts de Königsberg (les habitants de Königsberg se demandaient s'il était possible, en partant d'un quartier quelconque de la ville, de traverser tous les ponts sans passer deux fois par le même et de revenir à leur point de départ), la marche du cavalier sur l'échiquier ou le problème de coloriage de cartes.
La théorie des graphes s'est alors développée dans diverses disciplines telles que la chimie, la biologie, la physique, les sciences sociales. Depuis le début du XXe siècle, elle constitue une branche à part entière des mathématiques, grâce aux travaux de König, Menger, Cayley puis de Berge et d'Erdös.

Introduction

De manière générale, un graphe permet de représenter la structure, les connexions d'un ensemble complexe en exprimant les relations entre ses éléments : réseau de communication, réseaux routiers, interaction de diverses espèces animales, circuits électriques,...
Les graphes constituent donc une méthode de pensée qui permet de modéliser une grande variété de problèmes en se ramenant à l'étude de sommets et d'arcs. Les derniers travaux en théorie des graphes sont souvent effectués par des informaticiens, du fait de l'importance qu'y revêt l'aspect algorithmique.

Définition

Un graphe est la donnée d'un certain nombre de points du plan, appelés sommets, certains étant reliés par des segments de droites ou de courbes appelés arêtes , la disposition des sommets et la forme choisie pour les arêtes n'intervenant pas.



Exemples des situations modélisées par un graphe

1. Transport :
   - Carte géographique : Recherche du chemin le plus court entre deux villes ;
   - Un réseau ferroviaire : chaque gare est un sommet, les voies entre deux gares sont les arêtes. Idem avec un réseau routier ;
   - Les lignes aériennes.
2. Informatique :
   - Le web : chaque page est un sommet du graphe, chaque lien hypertexte est une arête entre deux sommets ;
   - Le routage des réseaux informatiques ;
   - Un réseau social : les sommets sont les personnes, deux personnes sont adjacentes dans ce graphe lorsqu'elles sont "amies". Si la notion d'amitié n'est pas réciproque, le graphe est orienté.

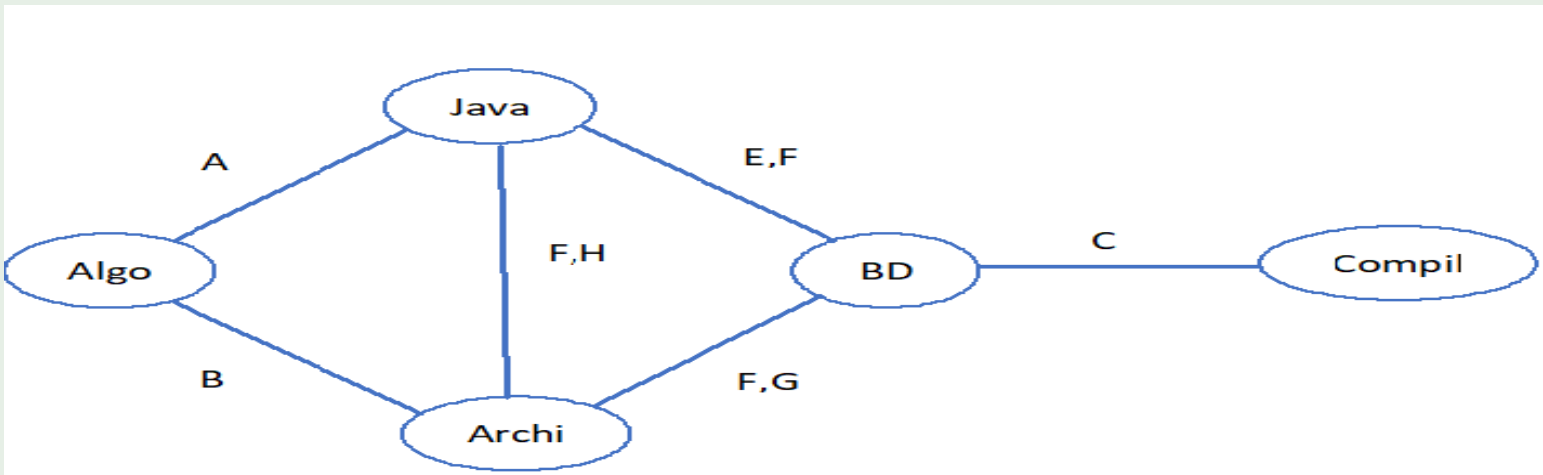
Exemples d'application 1

Des étudiants A, B, C, D, E et F doivent passer des examens dans différentes disciplines, chaque examen occupant une demi journée :

- Algorithmique : étudiants A et B.
- Compilation : étudiants C et D.
- Bases de données : étudiants C, E, F et G.
- Java : étudiants A, E, F et H.
- Architecture : étudiants B, F, G et H.

On cherche à organiser la session d'examen la plus courte possible.

Exemples d'application 1 : Solution

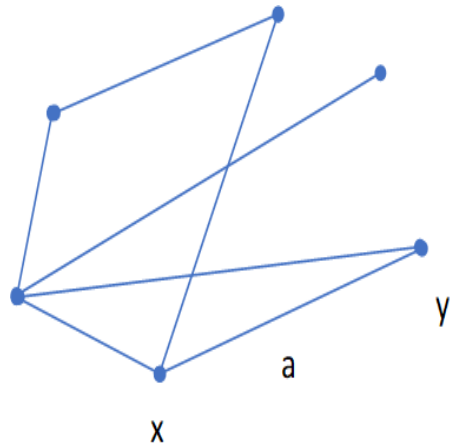


Exemples d'application 1 : Solution

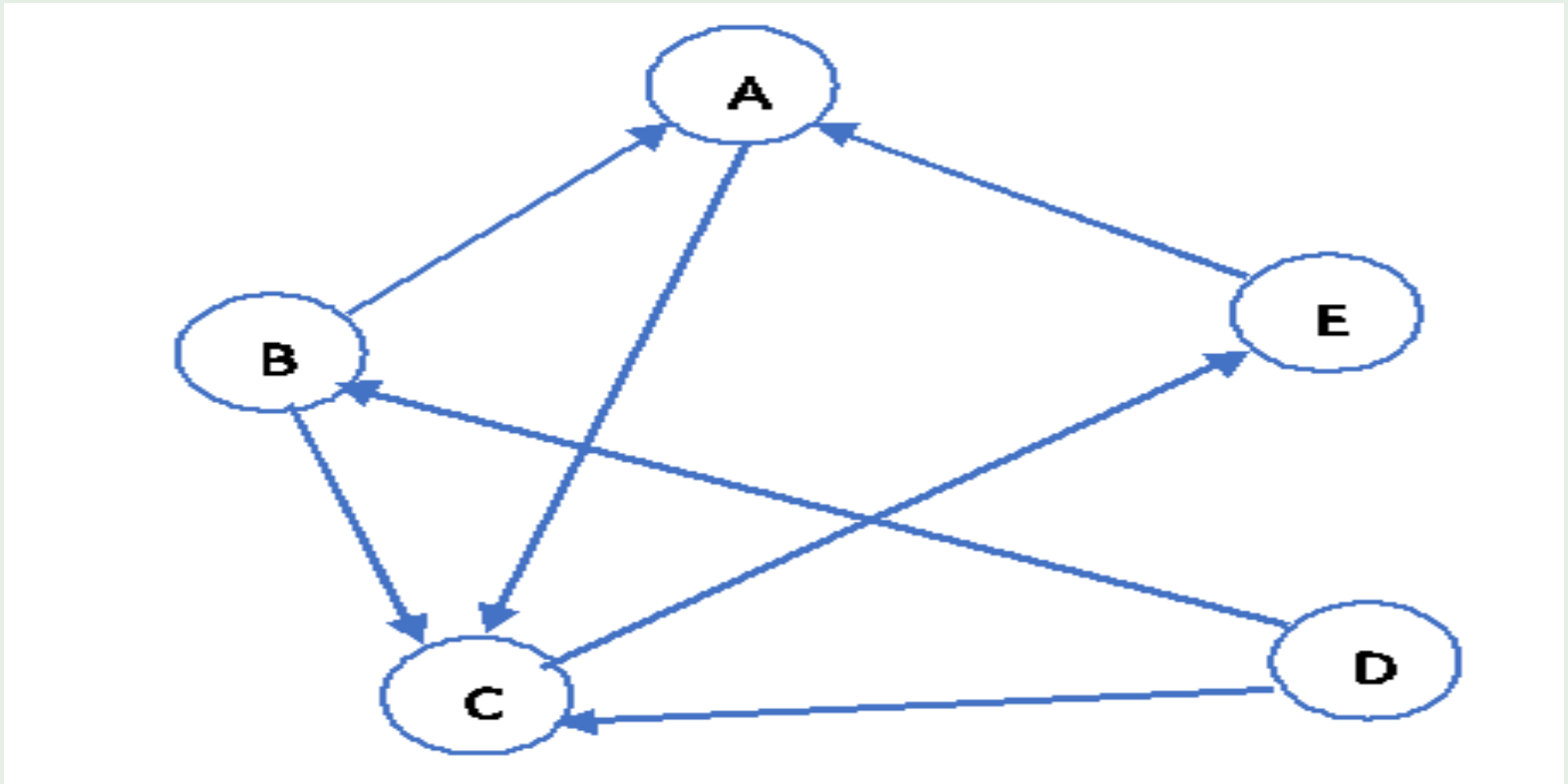
Session 1 :
Algo : A,B
BD : C,E,F,G
Session 2 :
Java : A,E,F,H
Compil : C,D
Session 3 :
Arch :B,F,G,H

Définition

Un graphe non orienté est un couple formé de deux ensembles finis : un ensemble $X = \{x_1, x_2, ..., x_n\}$ dont les éléments sont appelés sommets, et un ensemble fini $A = \{a_1, a_2, ..., a_m\}$ partie de l'ensemble des liens reliant deux éléments de X, appelé des arêtes. On notera $a = \{x, y\}$ lorsque a est l'arête reliant x à y (ou y à x). On dit aussi que l'arête a est d'extrémités x et y. les sommets x et y sont des adjacentes.

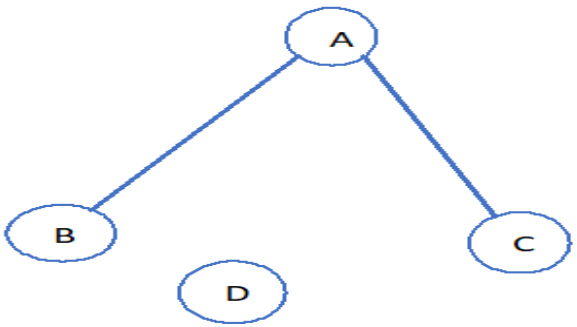


Graphe orienté

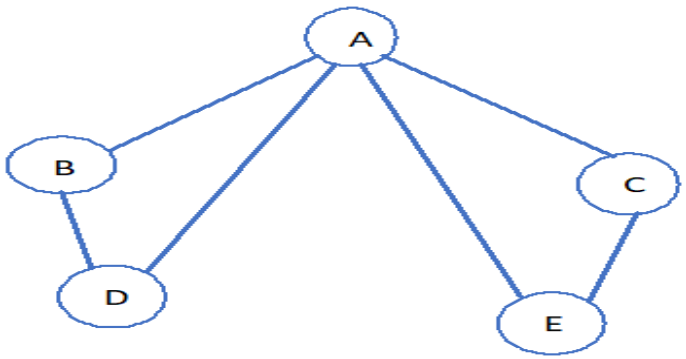


Terminologie de la théorie des graphes

L'ordre d'un graphe est le nombre de ses sommets.



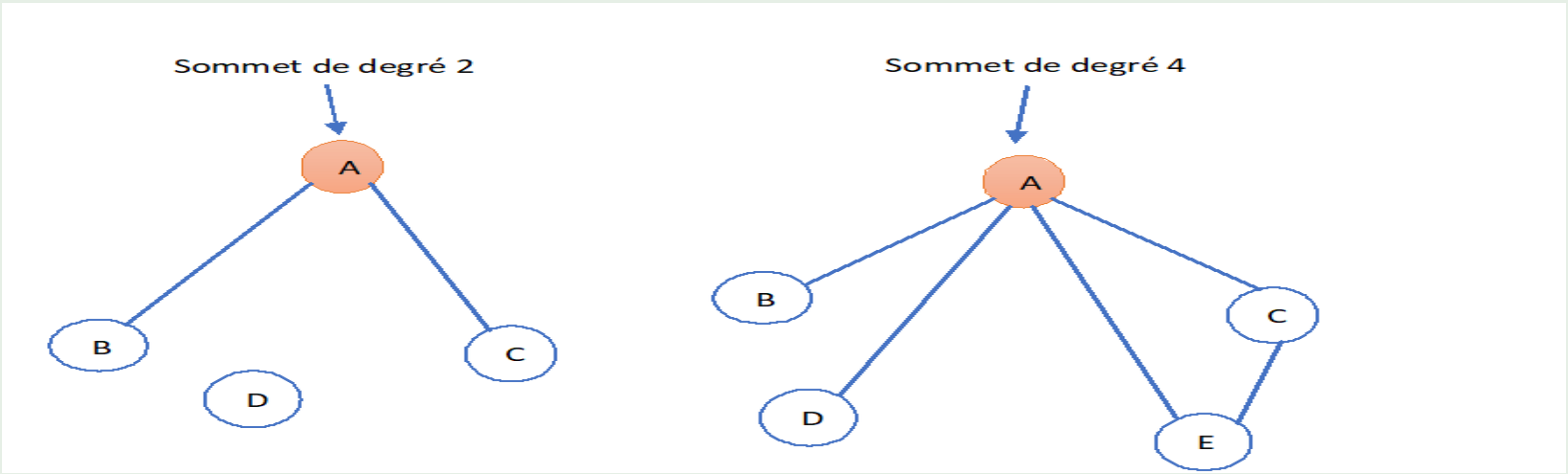
Graphe d'ordre 4



Graphe d'ordre 5

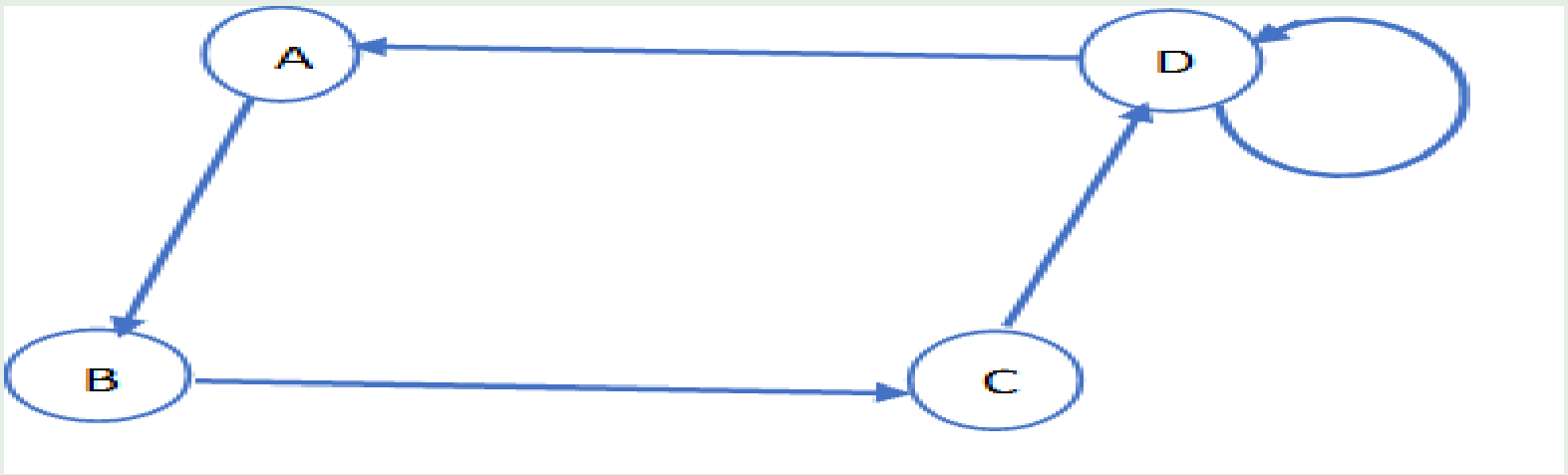
Terminologie de la théorie des graphes

Degré d'un sommet : nombre d'arêtes reliées à ce sommet.



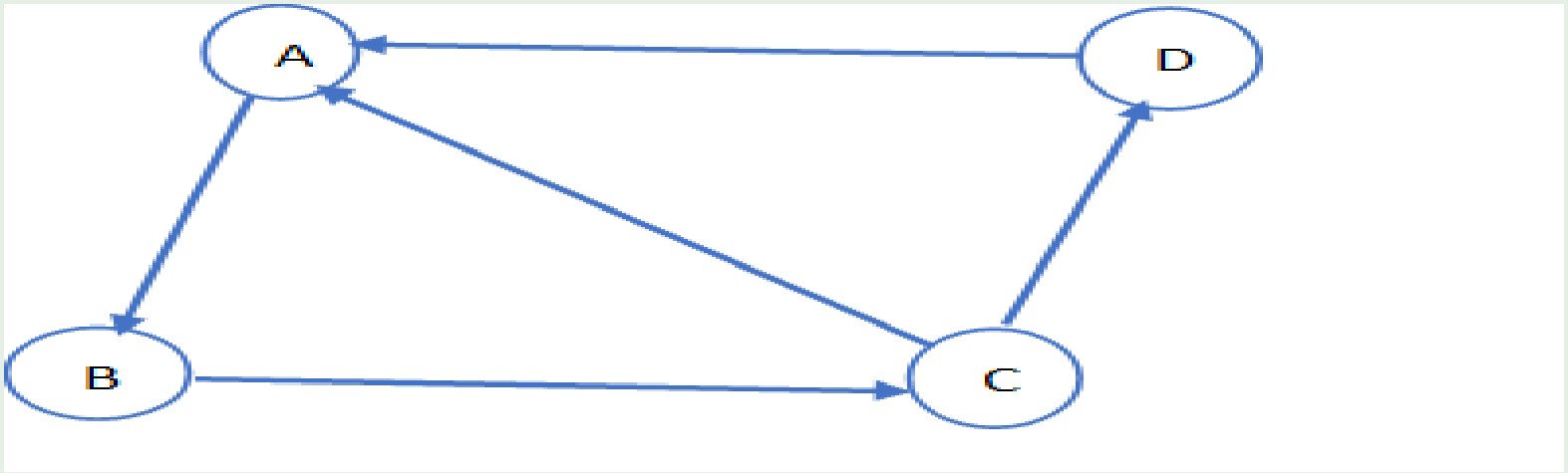
Terminologie de la théorie des graphes

Un arc boucle : est un arc qui part d'un sommet vers le même sommet.
L'arc <D,D> est une boucle.



Terminologie de la théorie des graphes

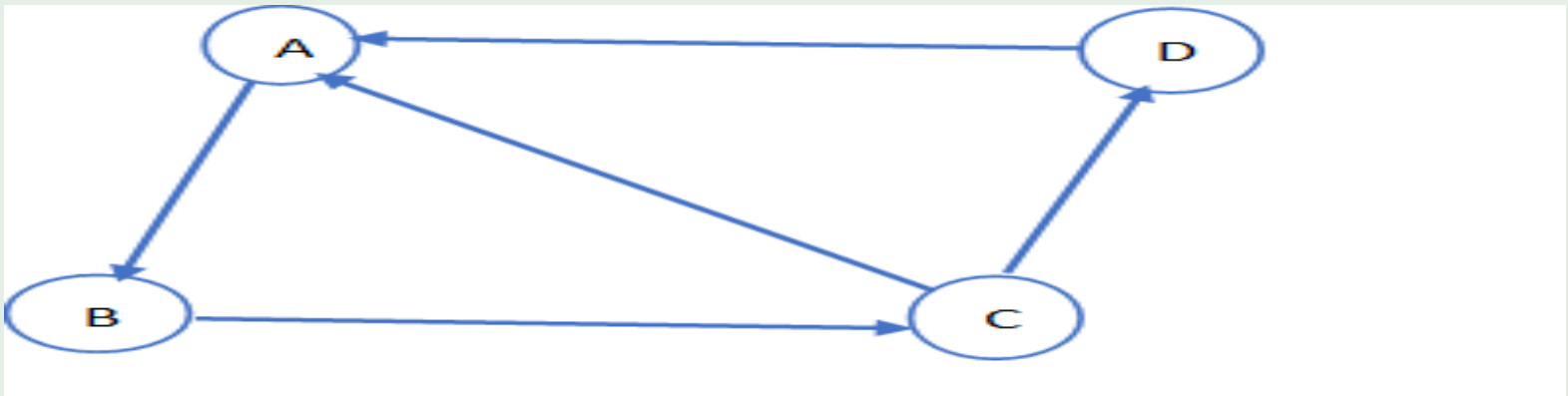
Chaîne : Une chaine de longueur n est une suite de n arêtes qui relient un sommet i à un autre j ou à lui même.



Des chaînes de longueur 2 : A D C et A B C
Des chaînes de longueur 3 : A B C D et B A D C
Des chaînes de longueur 4 : A B C D A et A B C A D

Terminologie de la théorie des graphes

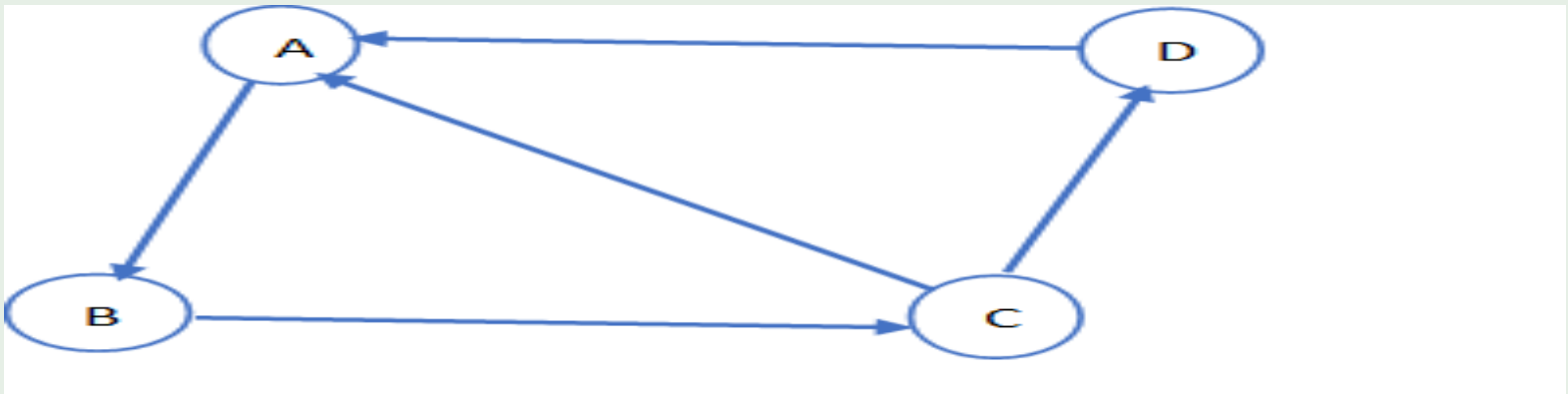
Cycle : Un cycle est une chaine qui permet de partir d'un sommet et revenir a ce sommet en parcourant une et une seule fois les autres sommets.



A B C D A est un cycle
A B C A est un cycle
D C A D est un cycle
A B C A D C A n'est un cycle

Terminologie de la théorie des graphes

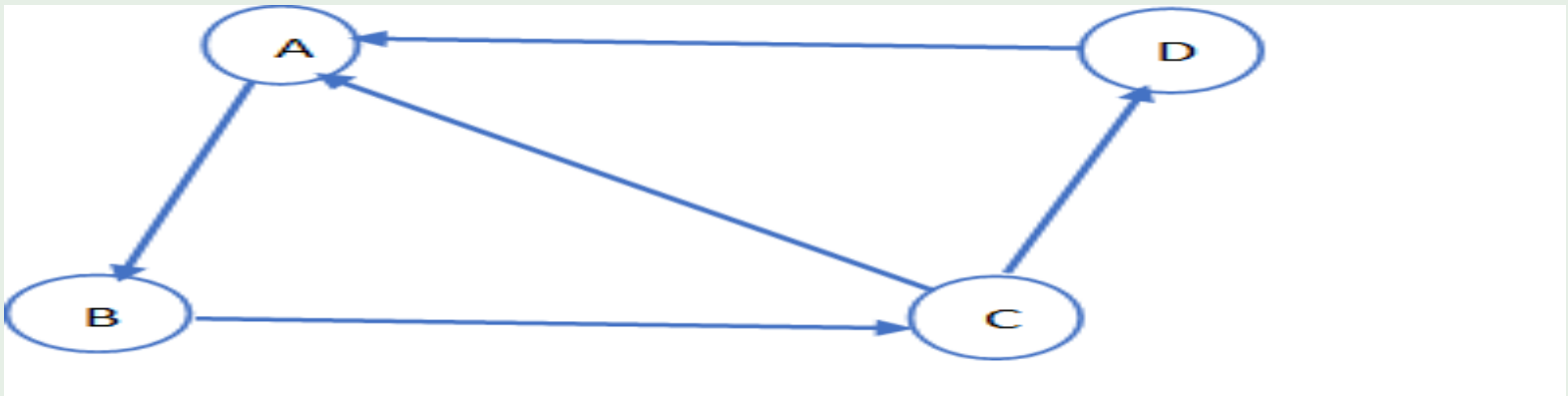
Chemin : c'est une chaine bien orientée



A B C D est un chemin
A D C A B n'est pas un chemin
A B C A D n'est pas un chemin

Terminologie de la théorie des graphes

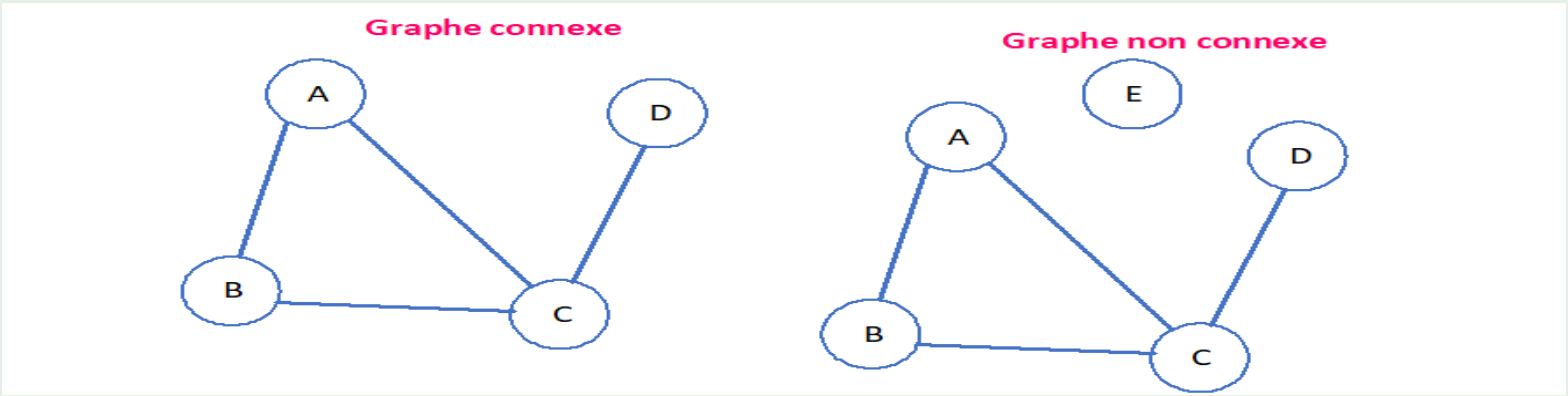
Circuit : est un cycle "bien orienté", à la fois cycle et chemin.



A B C A est un circuit
A B C D A es un circuit
A D C A est un cycle mais pas un circuit

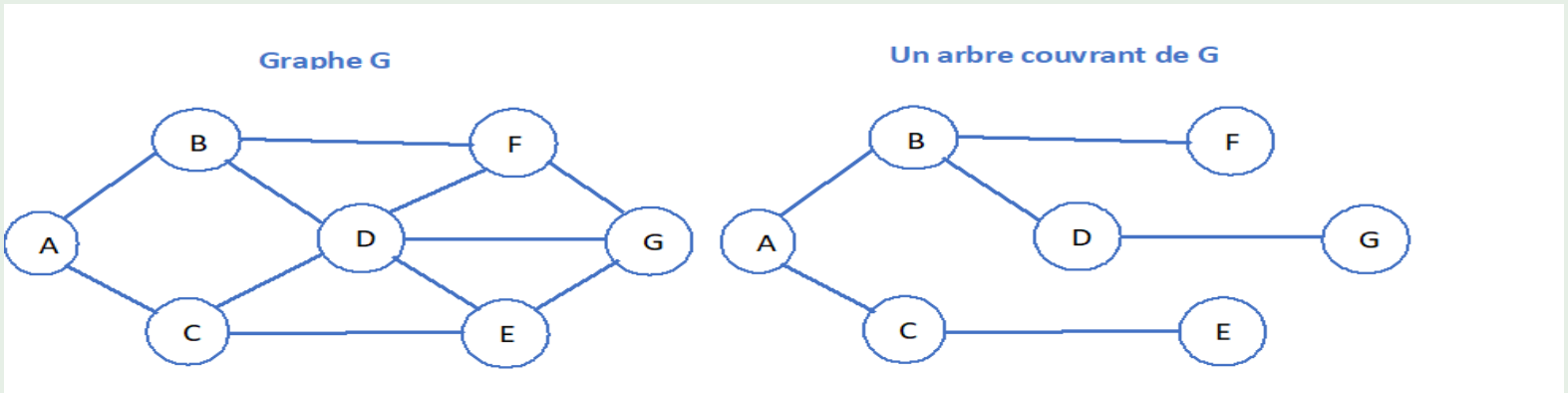
Terminologie de la théorie des graphes

Graphe Connexe : Un graphe connexe est un graphe dont tout couple de sommets peut être relie par une chaine de longueur $n \geq 1$



Terminologie de la théorie des graphes

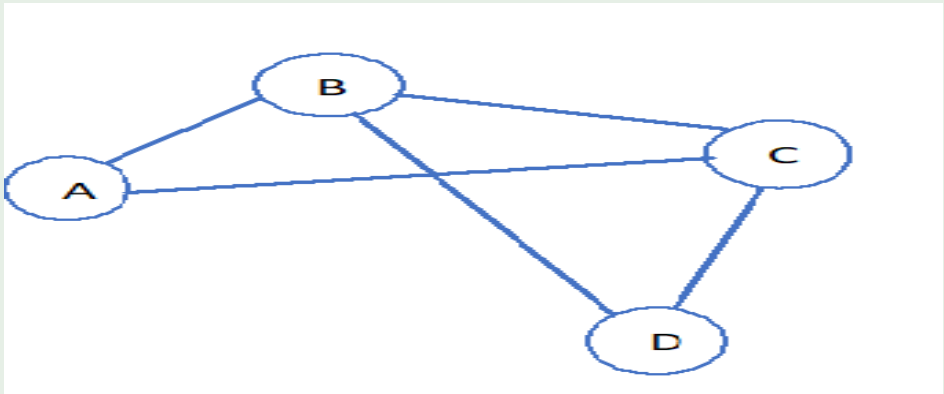
Un graphe est connexe s'il possède un arbre couvrant.



Algorithme de recherche d'un arbre minimal d'un graphe :
Algorithme de Kuskal
Algorithme de Prime

Terminologie de la théorie des graphes

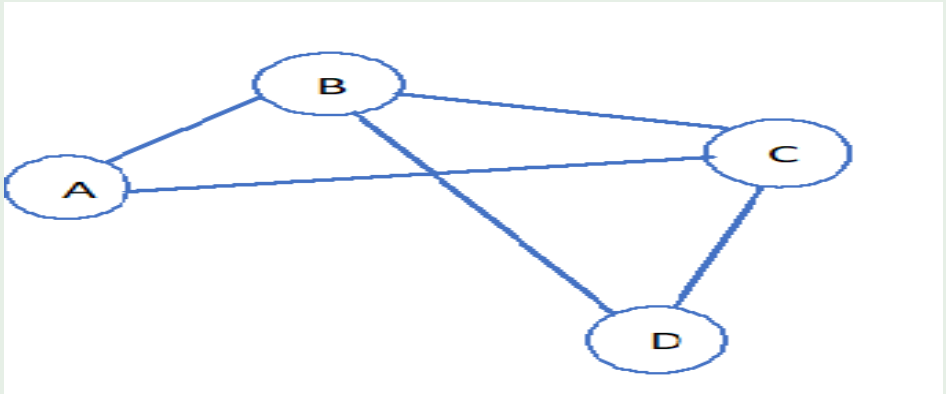
Chaîne Hamiltonienne : Chaîne passant une seule fois par tous les sommets d'un graphe.



Exemples : ABCD, ABDC, ACBD, ACDB

Terminologie de la théorie des graphes

Chaîne Eulérienne : Chaîne passant une seule fois par toutes les arêtes d'un graphe.

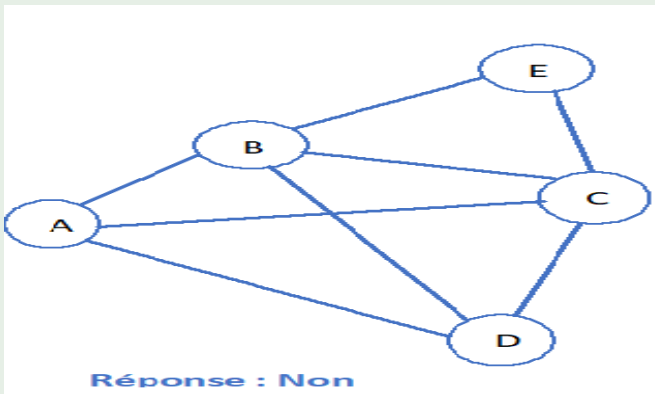


Exemples :
BACBDC est une chaîne Eulérienne
ACDBACB n'est pas une chaîne Eulérienne

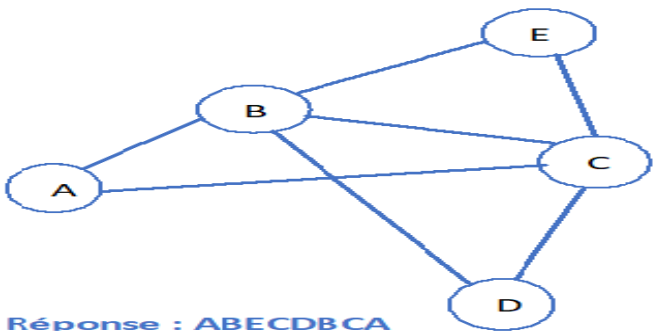
Terminologie de la théorie des graphes

- Cycle Hamiltonien : passant une seule fois par tous les sommets d'un graphe et revenant au sommet de départ.
- Cycle Eulérien : passant une seule fois par toutes les arêtes d'un graphe et revenant au sommet de départ.

Exemple : Existe t il un cycle Eulérien ?



Réponse : Non

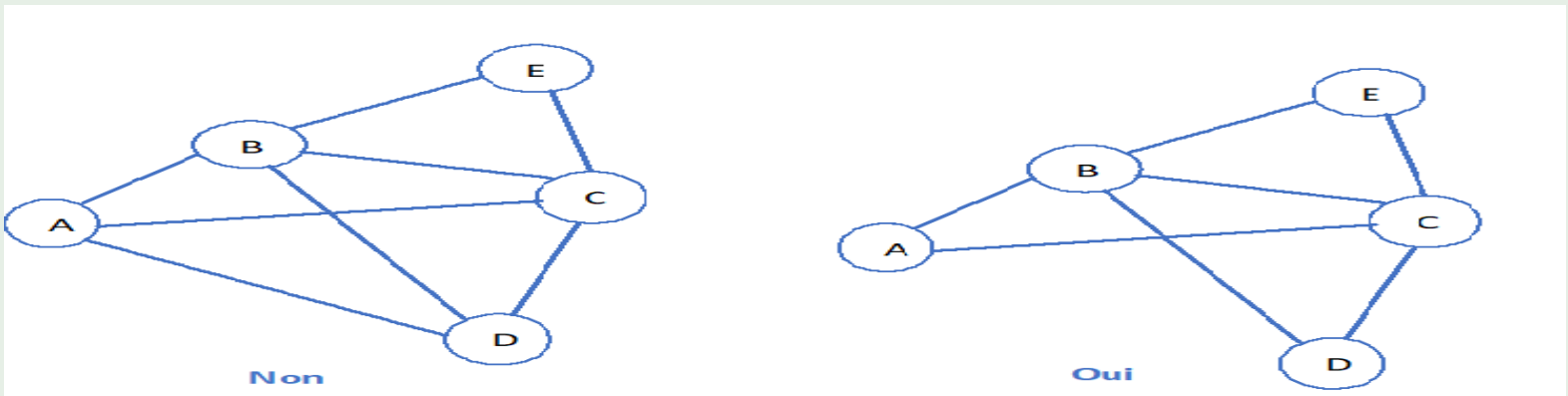


Réponse : ABECDBCA

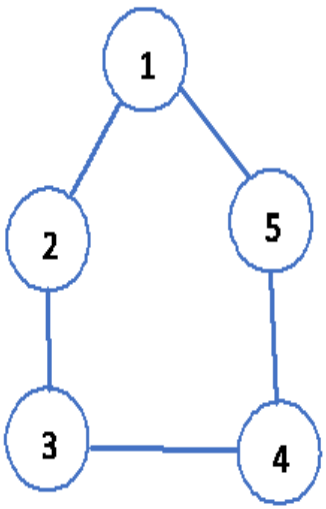
Terminologie de la théorie des graphes

- Graphe Hamiltonien : Graphe qui possède au moins un cycle Hamiltonien.
- Graphe Eulérien : Graphe qui possède au moins un cycle Eulérien.

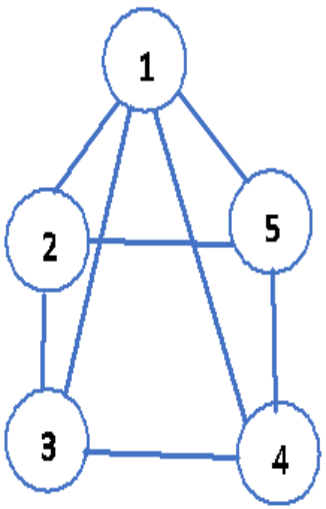
Théorème d'Euler (1766) : Un graphe est eulérien si tous les sommets du graphe ont un degré pair.



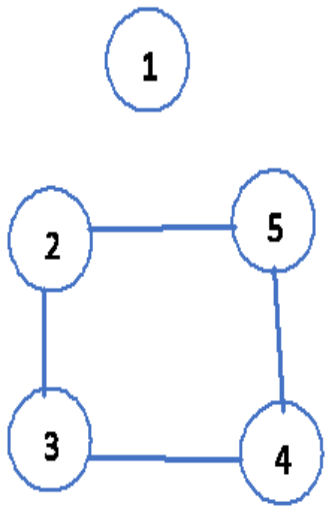
Exemples de graphe Hamiltonien, Eulérien, non Hamiltonien et non Eulérien



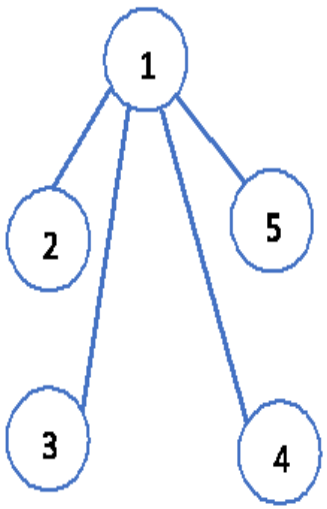
Hamiltonien et Eulérien



Hamiltonien et non Eulérien



Non Hamiltonien et Eulérien



Non Hamiltonien et non Eulérien

Terminologie de la théorie des graphes

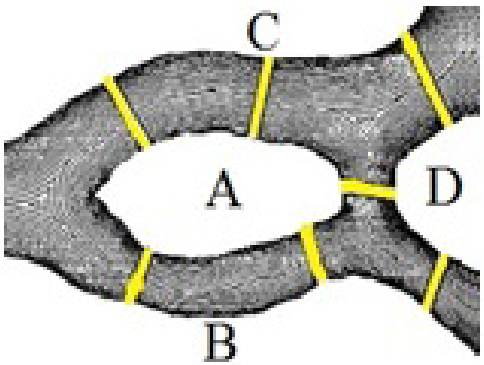
- Graphe valué : graphe où des réels sont associés aux arêtes (dans ce cours les réels sont positifs).
- Longueur d'une chaîne : nombre des arêtes qui composent la chaîne.
- Valeur d'une chaîne : somme des valeurs des arêtes (arcs) d'une chaîne d'un graphe valué.
- Distance entre deux sommets : longueur de la plus courte chaîne joignant ces deux sommets.
- Diamètre d'un graphe : maximum des distances entre les sommets d'un graphe.

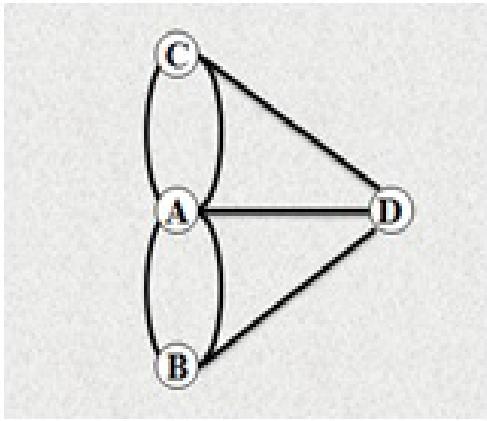
Retour au problème des ponts de Königsberg

La ville de Königsberg en Prusse (maintenant Kaliningrad) comprenait 4 quartiers, séparés par les bras du Prégel. Les habitants de Königsberg se demandaient s'il était possible, en partant d'un quartier quelconque de la ville, de traverser tous les ponts sans passer deux fois par le même et de revenir à leur point de départ.

Modélisation du problème des ponts de Königsberg par un graphe

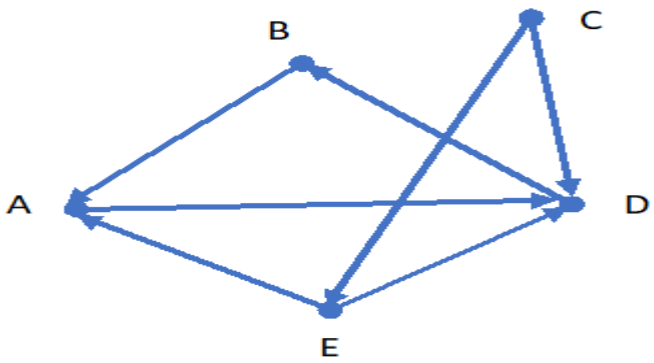
Le plan de la ville peut se modéliser à l'aide du graphe ci-dessous, les quartiers sont représentés par les 4 sommets, les 7 ponts par des arêtes : La question posée devient alors : ce graphe est-il eulérien ? Le théorème d'Euler répond immédiatement de façon négative aux habitants de Königsberg.



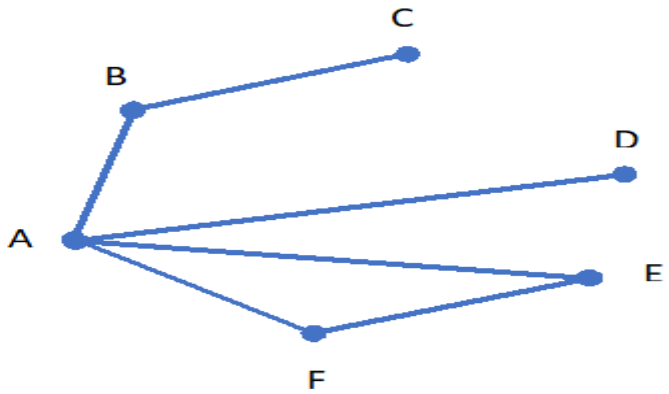


Outline

Les croix désignant les arcs (ou arêtes) entre sommets. Elles peuvent être remplacées par des valeurs désignant les longueurs des arcs (ou arêtes). Les tableaux suivants représentent les graphes précédents :



Graphe 1



Graphe 2

→	A	B	C	D	E
A				×	
B	×				
C				×	×
D		×			
E	×			×	

Graphe 1

→	A	B	C	D	E	F
A		×		×	×	×
B	×		×			
C		×				×
D	×					
E	×				×	
F	×		×		×	

Graphe 2

Il dresse pour chaque point les points qui le précédent dans le graphe (ou qui le succède), cette représentation n'a aucun sens pour les graphes non orientés. Pour le Graphe 1 cette représentation est la suivante :

X	Successeurs
A	D
B	A
C	D, E
D	B
E	A, D

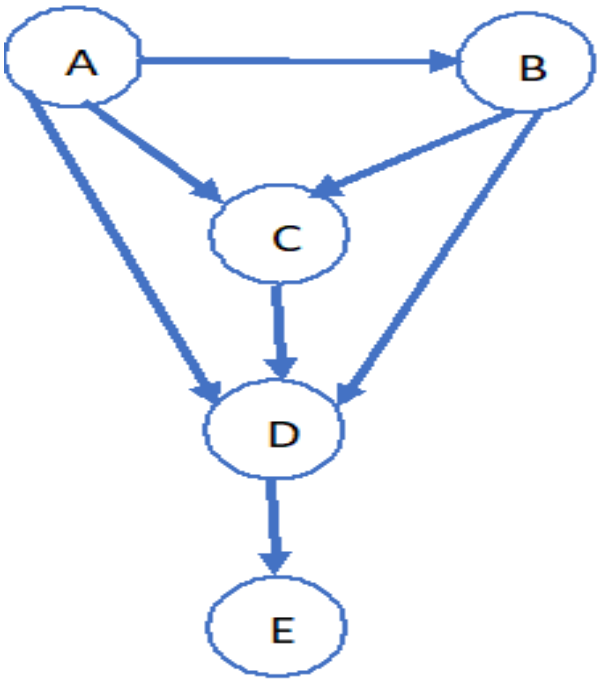
Graphe 1

X	Prédécesseurs
A	B, E
B	D
C	-
D	A, C, E
E	C
F	x

Graphe 1

Les deux tableaux sont équivalents.

On peut représenter un graphe simple par une matrice d'adjacences, c'est une matrice carrée où les termes désignent la présence d'un arc (flèche) entre deux sommets donnés, un " 1 " à la position (i,j) signifie que le sommet i est adjacent au sommet j.
Exemple d'un graphe orienté et sa matrice d'adjacences :



→	A	B	C	D	E
A	0	1	1	0	0
B	0	0	1	1	0
C	0	0	0	1	0
D	1	0	0	0	1
E	0	0	0	0	0

La matrice d'adjacences est :

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Dans cet exemple, le sommet A est de degré 3. Le sommet D est de degré 4.
La matrice d'adjacences a plusieurs caractéristiques :

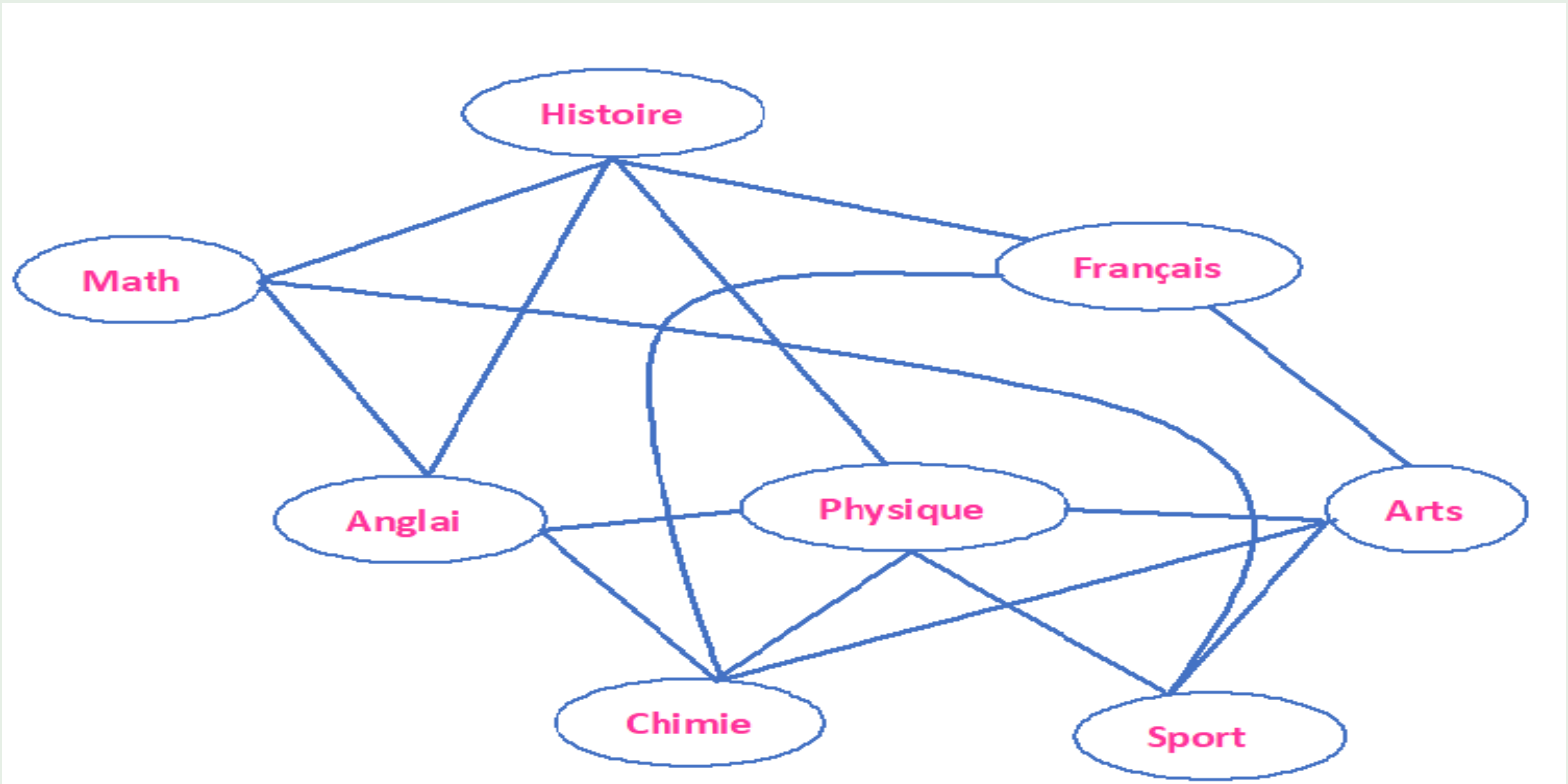
1. Elle est carrée.
2. Il n'y a que des zéros sur la diagonale. Un "1" sur la diagonale indiquerait une boucle.
3. Une fois que l'on fixxe l'ordre des sommets, il existe une matrice d'adjacences unique pour chaque graphe. Celle-ci n'est la matrice d'adjacences d'aucun autre graphe.

Outline

Une coloration des sommets d'un graphe est une partition de ses sommets en ensembles stables. En d'autres termes, il s'agit d'attribuer une couleur à chaque sommet de telle sorte qu'aucune arête du graphe n'ait ses extrémités de même couleur. Une coloration des arêtes d'un graphe est une partition de ses arêtes en couplages. En d'autres termes, il s'agit d'attribuer une couleur à chaque arête du graphe de telle sorte qu'aucun sommet ne soit incident à deux arêtes de même couleur.
Le nombre chromatique est le nombre minimal de couleurs différentes nécessaires pour colorer tous les sommets de ce graphe sans que deux sommets adjacents aient la même couleur.
Ce nombre est égal au plus grand degré des sommets du graphe diminué de 1.

1. Énumérer tous les sommets du graphe et les placer en ordre décroissant de degré. Pour deux sommets ayant le même degré, l'ordre n'a pas d'importance,
2. Attribuer une première couleur au sommet ayant le plus grand degré,
3. Appliquer cette même couleur à tous les sommets de degré inférieur qui ne sont pas reliés à ce sommet et qui ne sont pas reliés entre eux,
4. Répéter les étape 3 et 4 avec de nouvelles couleurs jusqu'à ce que tous les sommets soient colorés,
5. Le nombre chromatique du graphe est égal au nombre de couleurs utilisées.

Exemple 1



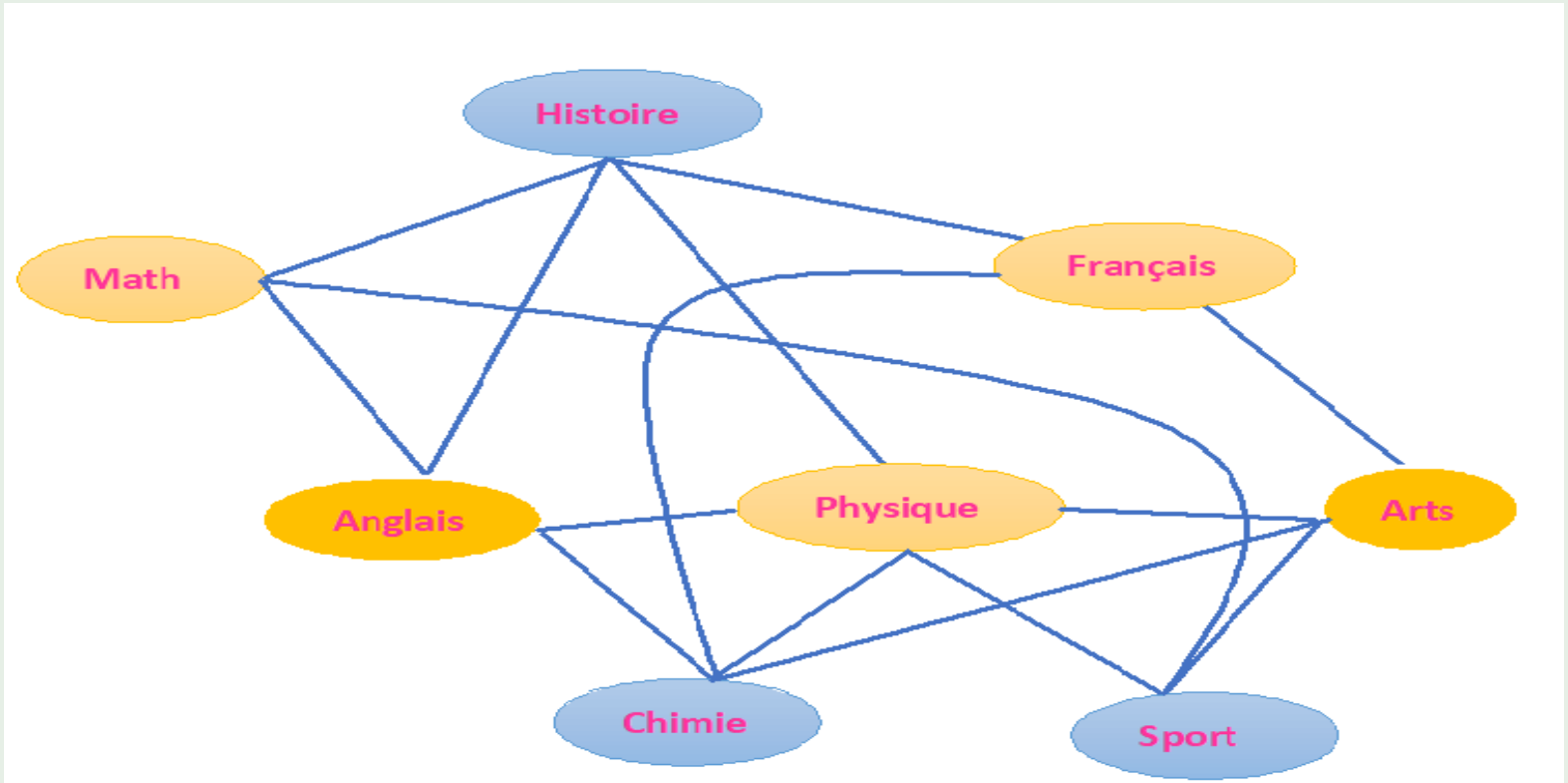
Exemple 1

Voici un graphe qui représente les tests à faire passer aux élèves d'une école secondaire. Les tests se déroulent dans des locaux différents.
Les sommets sont occupés par des matières (ou disciplines). Chaque arête signifie que les deux tests peuvent impliquer un même étudiant. Le lien «chimie–arts» signifie qu'un élève aura à passer les deux tests, de chimie et d'arts.
On sait qu'un élève ne peut pas passer deux tests ou plus simultanément. Ainsi toutes les arêtes représentées sur le graphe ne doivent pas exister parceque ce sont les sources du conflit d'horaires.
Comment doit-on donc procéder pour que chaque élève puisse passer tous ses tets sans conflit d'horaires ?
La méthode de coloriage répond à la question.

Voici le tableau des sommets énumérés en ordre décroissant de degré :

Sommet	Degré
Physique	5
Histoire	4
Arts	4
Chimie	4
Anglais	4
Sport	3
Français	3
Maths	3

On attribue une couleur (orange) pour le sommet de degré le plus haut (physique,5),
On applique cette couleur orange aux autres sommets qui ne sont pas adjacents à «physique» et ne sont pas adjacents l'un de l'autre. Ces sommets sont : «maths» et « français».



Maintenant, on applique une autre couleur (blue) à l'un des sommets de degré immédiatement inférieur qui est 4, soit le sommet «chimie »

On applique cette couleur blue aux autres sommets qui ne sont pas adjacents à «chimie» et ne sont pas adjacents l'un de l'autre. Ces sommets sont : «histoire» et «sport».
Maintenant, on applique une autre couleur (jaune) à l'un des sommets de degré immédiatement inférieur qui est 3, soit le sommet «anglais».
On applique cette couleur jaune aux autres sommets qui ne sont pas adjacents à «anglais» et ne sont pas adjacents l'un de l'autre. Il ne reste que le sommet «arts».
Trois couleurs ont été utilisées sur le graphe. Le nombre chromatique est donc égal à 3.

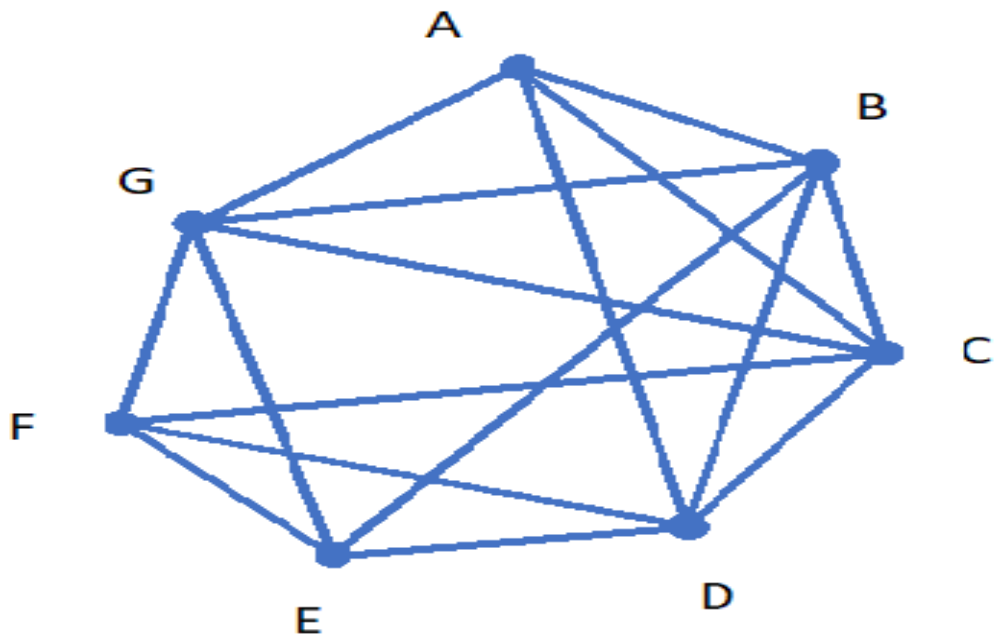
Conclusion :
Trois couleurs ont été utilisées. Il faut donc faire passer les tests à trois séances séparées, dont l'ordre n'a aucune importance.
Au court de la première séance, on fait passer des tests en physique, maths, et français ,
Au court de la deuxième séance, on fait passer des tests en chimie, histoire et sport ,
Au court de la troisième séance, on fait passer des tests sur les deux matières qui restent : anglais et arts .

Exemple 2 : Problème d'emploi du temps

Une université doit organiser les horaires des examens. On suppose qu'il y a 7 épreuves à planifier, correspondant aux cours numérotés A, B, C, D, E, F, G et que les paires de cours suivantes ont des étudiants communs : (A et B), (A et C), (A et D), (A et G), (B et C), (B et D), (B et E), (B et G), (C et D), (C et F), (C et G), (D et E), (D et F), (E et F), (E et G) et (F et G). Comment organiser ces épreuves de façon qu'aucun étudiant n'ait à passer deux épreuves en même temps et cela sur une durée minimale ?

Solution : Problème d'emploi du temps

Construisons le graphe G dont les sommets A, B, C, D, E, F, F, G, une arête relie deux de ses sommets lorsque les deux cours correspondant possèdent des étudiants communs :



Solution : Problème d'emploi du temps

1) On range les sommets du plus haut degré au plus petit (on ne tient pas compte de l'ordre pour les sommets de même degré).

Sommet	B	C	D	G	A	E	F
Degré	5	5	5	5	4	4	4

On choisit une couleur pour le premier sommet de la liste, Le rouge par exemple :

Sommet	B	C	D	G	A	E	F
Degré	5	5	5	5	4	4	4

On colorie en rouge les sommets non adjacents à B et non adjacents entre eux : F

Sommet	B	C	D	G	A	E	F
Degré	5	5	5	5	4	4	4

Solution : Problème d'emploi du temps

2) On réitère le procédé vu au 1) en prenant une autre couleur pour le premier sommet non colorié de la liste. On colorie C en bleu.

Sommet	B	C	D	G	A	E	F
Degré	5	5	5	5	4	4	4

On colorie ensuite E en bleu.

Sommet	B	C	D	G	A	E	F
Degré	5	5	5	5	4	4	4

Solution : Problème d'emploi du temps

3) On réitère le procédé. On colorie D en vert puis G.

Sommet	B	C	D	G	A	E	F
Degré	5	5	5	5	4	4	4

Solution : Problème d'emploi du temps

4) Enfin A en noire.
Donc on a 4 stables $S_1 = \{B, F\}$, $S_2 = \{C, E\}$, $S_3 = \{D, G\}$ et $S_4 = \{A\}$
Les examens peuvent être répartis en 4 périodes, de la manière suivante :
la 1*re* période, épreuves des cours 2 et 6,
la 2*me* période, épreuve du cours 3 et 5,
la 3*me* période, épreuves des cours 4 et 7,
la 4*me* période, épreuves des cours 1.

Exemple 3 : Problème de wagons

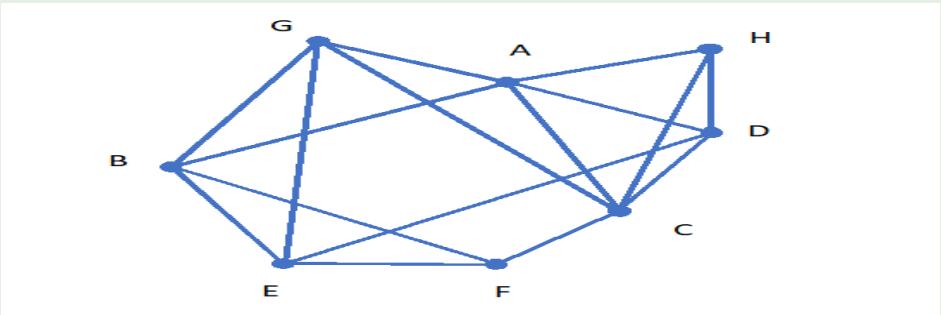
On veut transporter des produits chimiques par le rail. A, B, C, D, E, F, G et H désignent huit produits chimiques. Dans le tableau ci-dessous, une croix signifie que les produits ne peuvent pas être entreposés dans le même wagon, car il y aurait risque d'explosion :

	A	B	C	D	E	F	G	H
A		×	×	×			×	×
B	×				×	×	×	
C	×			×		×	×	×
D	×		×		×			×
E		×		×		×	×	
F		×	×		×			
G	×	×	×		×			
H	×		×	×				

Quel nombre minimum de wagons faut-il ?

Solution : Problème de wagons

Construisons le graphe G dont les sommets sont les huit produits chimiques tel que deux de ses sommets sont reliés lorsque les produits associés à ces sommets ne peuvent pas être entreposés dans le même wagon. Le nombre minimum de wagons est égal au nombre chromatique de ce graphe.



La partition minimale des sommets est :

$$S_1 = \{A, E\}, S_2 = \{B, C\}, S_3 = \{D, F, G\}, S_4 = \{H\}$$

Il faut donc 4 wagons.

Outline

Beaucoup de problèmes peuvent être modélisés en utilisant des graphes valués. Les problèmes de cheminement dans les graphes, en particulier la recherche du plus court chemin, comptent parmi les problèmes les plus anciens de la théorie des graphes et les plus importants par leurs applications : coût de transport, temps de parcours, problème de trafic,. . .Les algorithmes de recherche de plus court chemin seront différents selon les caractéristiques du graphe.
Il y a deux méthodes principales : L'algorithme de Ford-Bellman et l'algorithmes de Dijkstra.
La méthode qui est largement utilisée pour trouver le plus court chemin dans un graphe est celle présentée par Ford-Bellman. Cette méthode a été initialement consacrée aux problèmes économiques, mais elle peut être appliquée à d'autres domaines.

Pour déterminer le plus court chemin dans un graphe l'algorithme de Ford-Bellman consiste à suivre les étapes suivantes :

1. Les sommets x_0 sans précédents sont dit du 'premier niveau' ou niveau de depart : on leur affecte une fonction coût (ou distance), notée m, égale à 0 : $m(x_0) = 0$
2. Examiner les noeuds x_1 adjacents aux noeuds x_0 (ceux de premier niveau). Pour chacun, la fonction m se calcul alors en ajoutant la distance à partir du noeud source par :
$m(x_1) = d(x_1, x_0) + m(x_0) = d(x_1, x_0)$
3. Au niveau $i+1$, on continue l'examination des noeuds adjacents à ceux visités dans le niveau i précedent. Quand un noeud $x(i+1)$ a des liaisons avec plusieurs précédents, la valeur de $m(x_{i+1})$ se calcule alors en retenant le noeud x_{i+1} le plus proche, en d'autre terme :
$m(x_{i+1}) = min\{d(x_{i+1}, x_i) + m(x_i),\ x_i \text{ noeud précédent à } x_{i+1}\}$
4. Répéter l'étape 3, niveau après niveau jusqu'à ce que le noeud de destination soit atteint.

La Figure ci-dessous montre la distance entre 10 villes voisines avec l'unité de (1=100 kilomètre). Le problème est de trouver le plus court chemin entre la ville de départ (a) et la ville de destination (j).

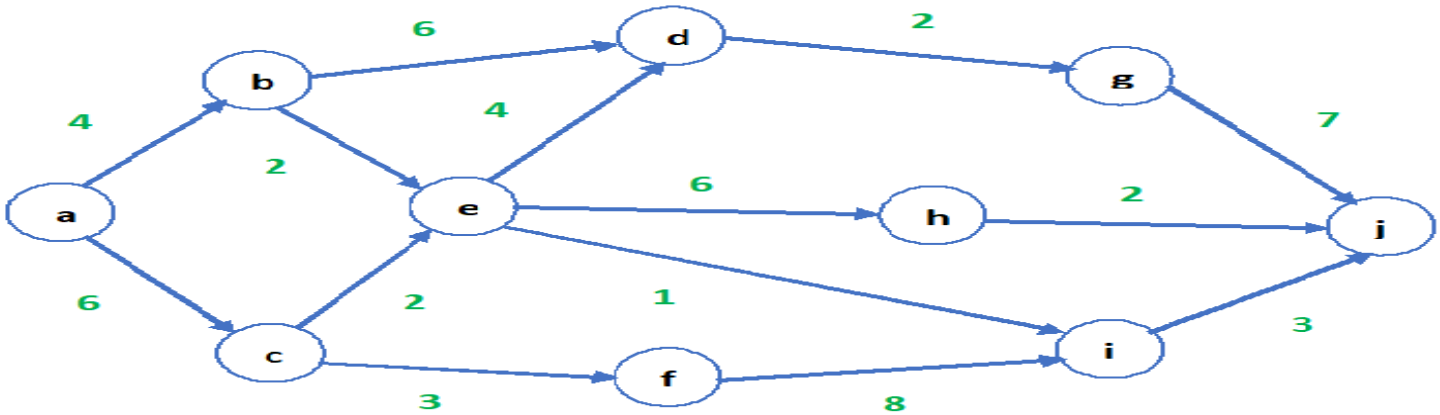


Figure 2 – Distances entre les villes

Ce problème est souvent rencontré dans d'autres situations différentes, par exemple on peut associer à chaque arc de graphe le montant à payer au point de péage, et le but dans ce cas est de trouver le trajet dont le coût est minimal.

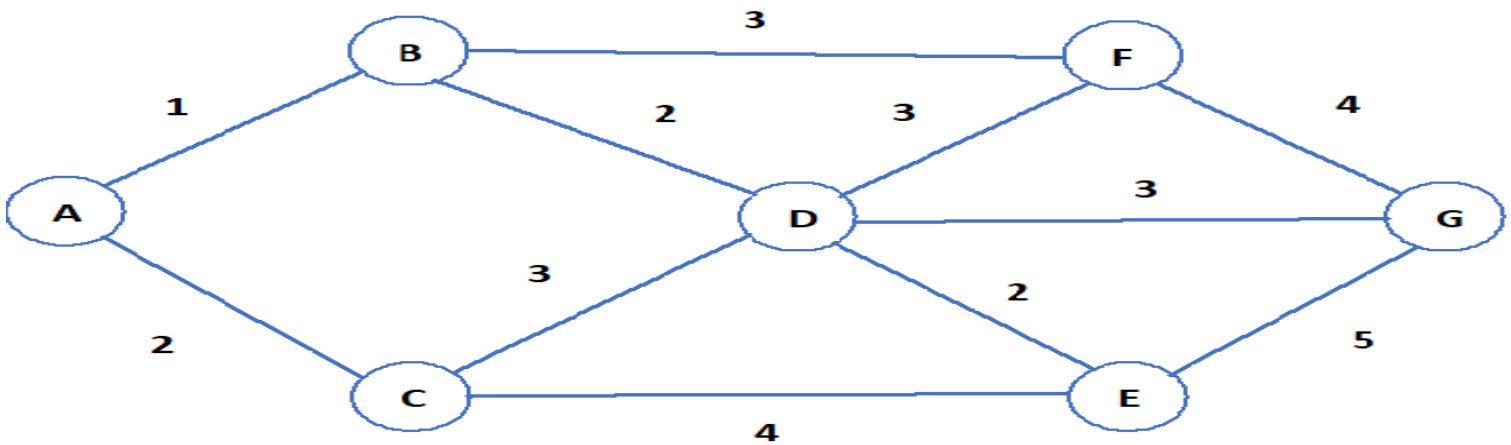
Niveau 1 : Point de départ est (a) donc $m(a) = 0$
Niveau 2 : sommets (a) et (b)
Les sommets b et c sont adjacents à a. Pour ces noeuds on trouve la distance par l'ajout de $m(a)$ avec la distance des noeuds à partir de a. On a alors :
$m(b) = m(a) + d(a,b) = 0 + 4 = 4$
$m(c) = m(a) + d(a,c) = 0 + 6 = 6$
Niveau 3 : sommets (d), (e) et (f)
$m(d) = m(b) + d(b;d) = 4 + 6 = 10$
$m(e) = \min\{m(b) + d(b,e), m(c) + d(c,e)\} = \min\{4+2, 6+2\} = 6$
$m(f) = m(c) + d(c,f) = 6 + 3 = 9$
Niveau 4 : sommets (g), (h) et (i)
$m(g) = m(d) + d(d,g) = 10 + 2 = 12$
$m(h) = m(e) + d(e,h) = 6 + 6 = 12$
$m(i) = \min\{m(e) + d(e,i), m(f) + d(f,i)\} = \min\{6+1, 9+8\} = 7$
Niveau 5 : sommets (j)
$m(j) = \min\{m(g) + d(g,j), m(h) + d(h,j), m(i) + d(i,j)\} =$
$\min\{12+7, 12+2, 7+3\} = 10$

Le plus court chemin à partir de a vers j est reconstitué de la fin vers le début comme suit : Pour le niveau 5 la ville la plus proche de (j) qui appartient au plus court chemin est (i) (la distance minimal pour les trois trajets).
Pour le niveau 4 la ville la plus proche de (i) dans le plus court chemin est (e).
Pour le niveau 3 la ville la plus proche de (e) dans le plus court chemin est (b).
Pour le niveau 1 la ville la plus proche de (b) dans le plus court chemin est (a).
Le plus court chemin est donc a, b, e, i, j
La distance minimal d'aller de (a) à (j) est de $10 \times 100 = 1000$km.

En théorie des graphes, l' algorithme de Dijkstra sert à résoudre le problème du plus court chemin.
Il s'applique à un graphe connexe dont le poids lié aux arêtes est un réel positif.
Applications :

1. Chemin le plus court entre deux villes.
2. Chemin le plus rapide entre deux poins en ville.
3. Routage d'information optimal.
4. ...

Exemple : Trouver le chemin optimal entre A et G.



Initialisation de l'algorithme :
Étape 1 : On affecte le poids 0 au sommet origine (E) et on attribue provisoirement un poids ∞ aux autres sommets.
Répéter les opérations suivantes de l'étape 2 et 3 tant que le sommet de sortie (s) n'est pas affecté d'un poids définitif :
Étape 2 : Parmi les sommets dont le poids n'est pas définitivement fixé choisir le sommet X de poids p minimal. Marquer définitivement ce sommet **X** affecté du poids **p(X)**.
Étape 3 : Pour tous les sommets Y qui ne sont pas définitivement marqués, adjacents au dernier sommet fixé X :
Calculer la somme s du poids de X et du poids de l'arête reliant X à Y.
Si la somme s est inférieure au poids provisoirement affecté au sommet Y, affecter provisoirement à Y le nouveau poids s et indiquer entre parenthèses le sommet X pour se souvenir de sa provenance.
Quand le sommet s est définitivement marqué, le plus court chemin de E à S s'obtient en écrivant de gauche à droite le parcours en partant de la fin S.

Algorithme de Dijkstra :

Règles pour remplir les cases de chaque cellule :

Soit **e** le sommet de départ.

1. La valeur de chaque cellule est un couple : (d(v), p(v)).
   Où : d(v) est la distance minimale du sommet de départ **e** au somment **v** et p(v) représente le sommet précédent du chemin optimal reliant e et v.
2. d(e)=0 (e est le sommet de départ).
3. d(v)=∞ si la distance est non encore calculé.

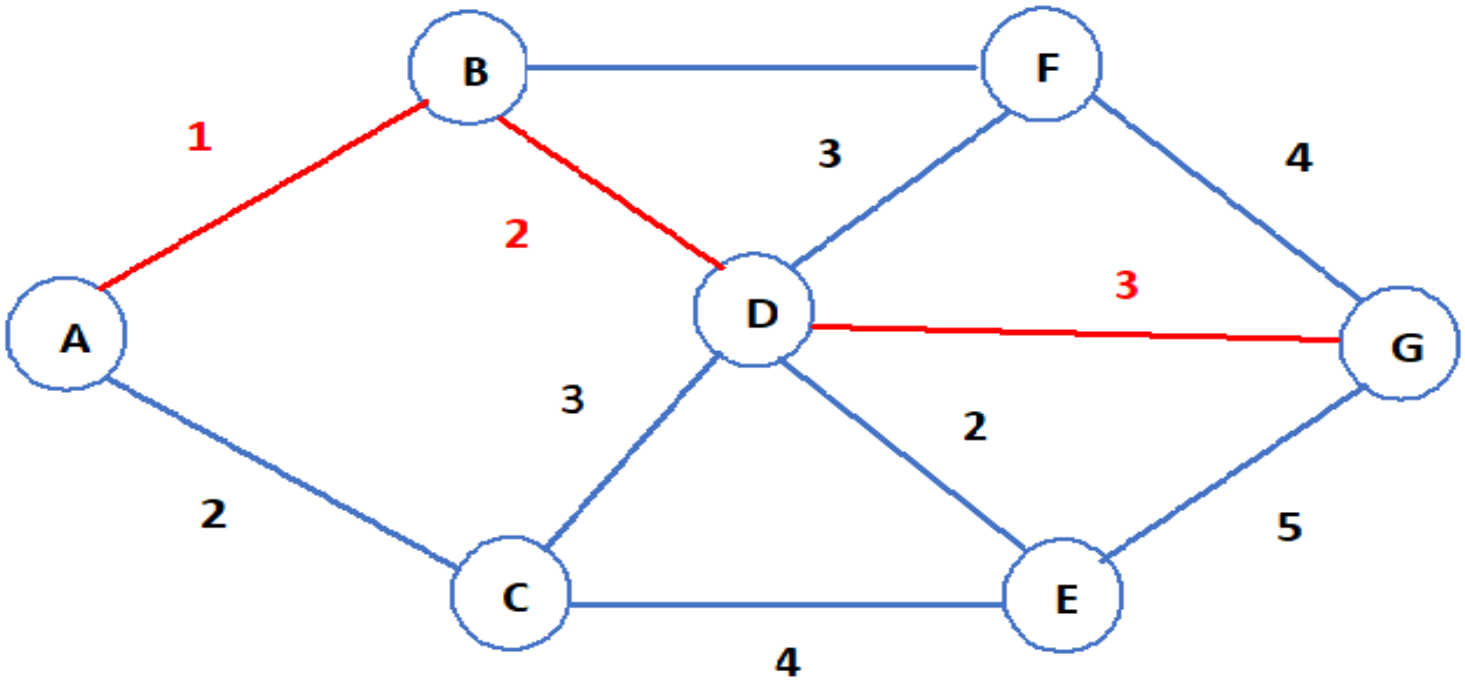
Etapes de l'algorithme de Dijkstra :

	A	B	C	D	E	F	G
Etape 1	0	∞	∞	∞	∞	∞	∞
Etape 2	$\times$	(1,A)	(2,A)	∞	∞	∞	∞
Etape 3	$\times$	$\times$	(2,A)	(3,B)	∞	(4,B)	∞
Etape 4	$\times$	$\times$	$\times$	(3,B)	(6,C)	(4,B)	∞
Etape 5	$\times$	$\times$	$\times$	$\times$	(5,D)	(4,B)	(6,D)
Etape 6	$\times$	$\times$	$\times$	$\times$	(5,D)	$\times$	(6,D)
Etape 7	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	(6,D)

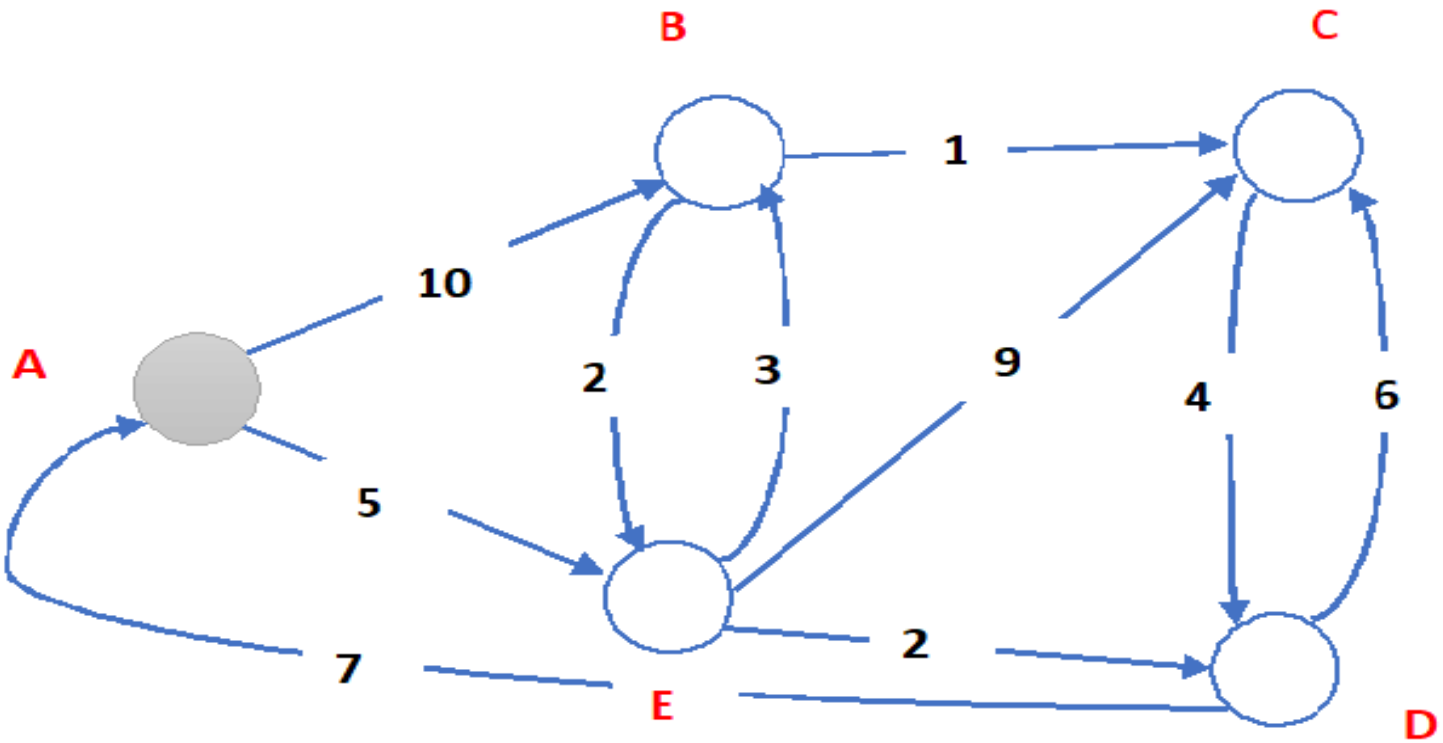
Etapes de l'algorithme de Dijkstra :

	A	B	C	D	E	F	G
Etape 1	0	∞	∞	∞	∞	∞	∞
Etape 7	(0,A)	(1,A)	(2,A)	(3,B)	(5,D)	(4,B)	(6,D)

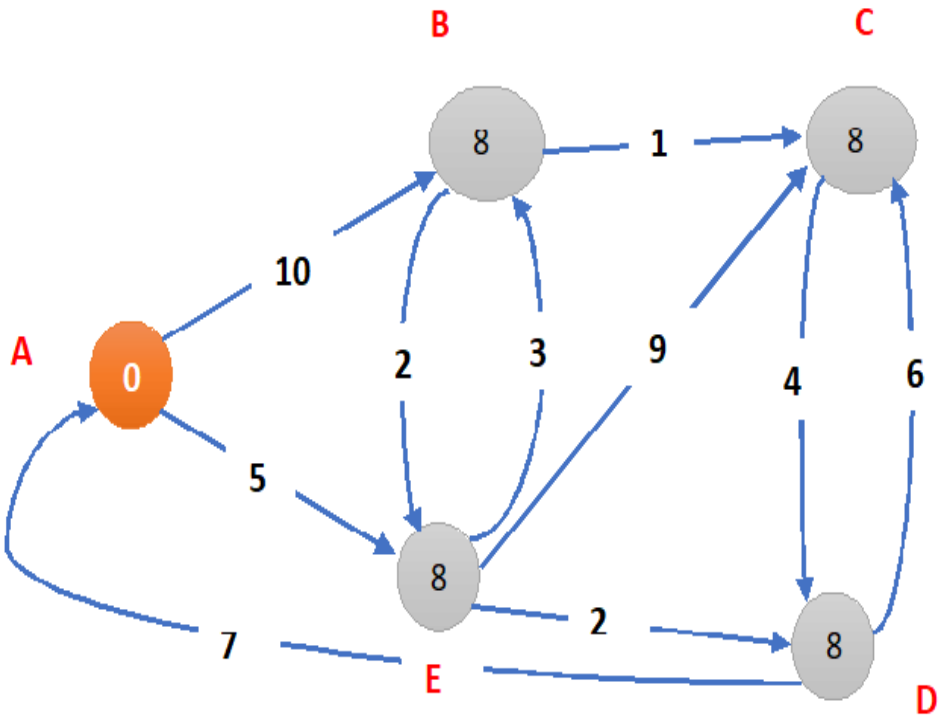
Le chemin optimal est : ABDG de coût égal à 6.



Chercher les plus courts chemins d'origine A dans ce graphe :

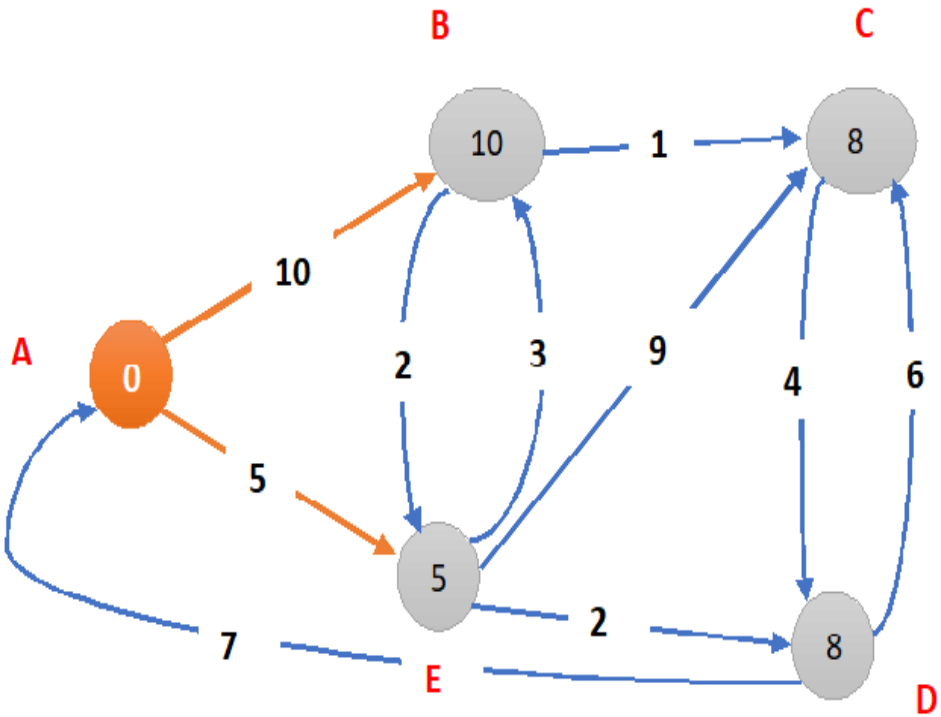


On se place au sommet de plus petit poids, ici le sommet A.



A	B	C	D	E
0	8	8	8	8
*				
*				
*				
*				
*				

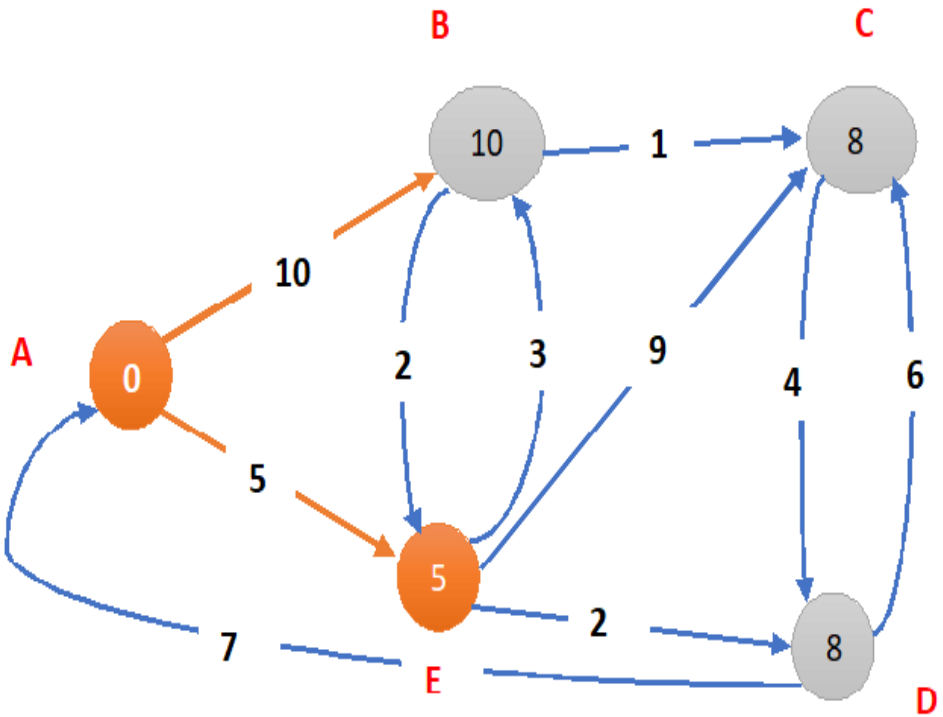
On étudie chacune des arêtes partant du sommet choisi.



A	B	C	D	E
0	8	8	8	8
*	10_A	8	8	5_A
*				
*				
*				
*				

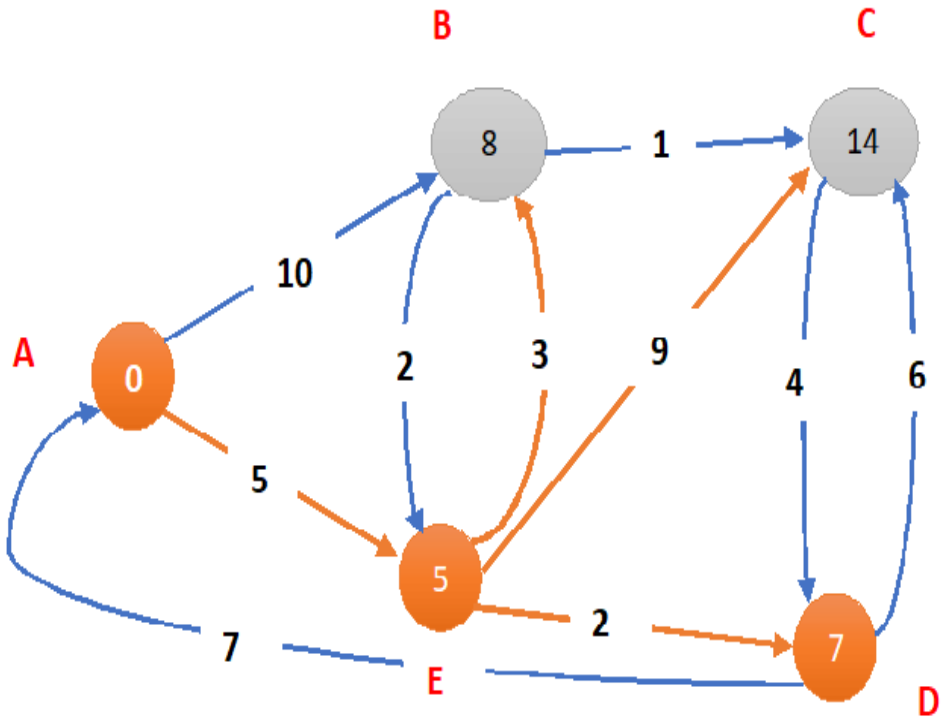
Dans les colonnes, on mets la distance à A, et le sommet d'où l'on vient.

On se place de nouveau au sommet de plus petit poids, ici E.

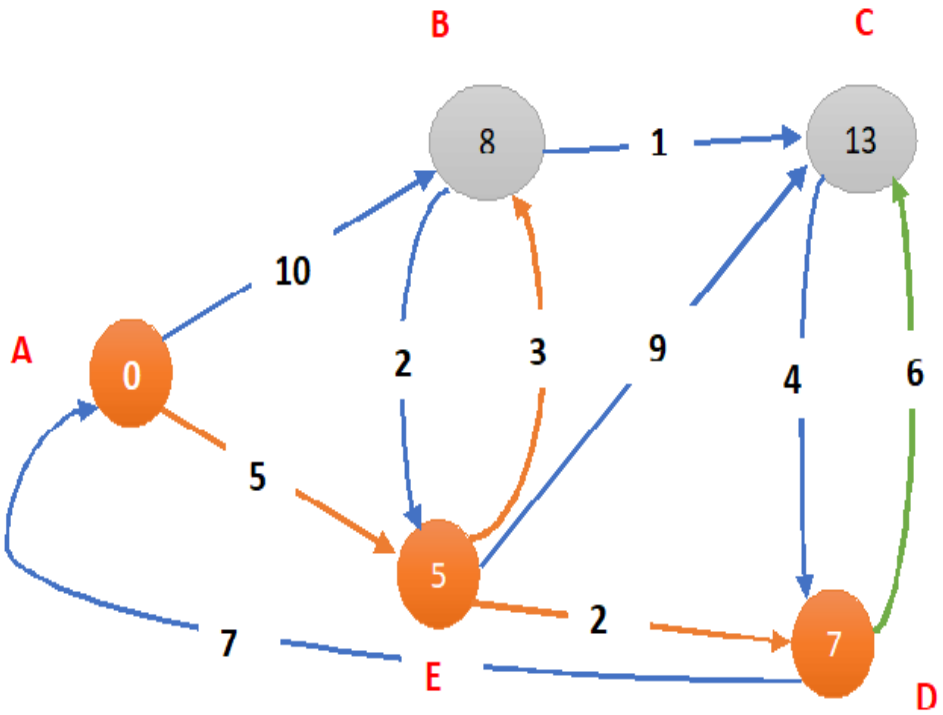


A	B	C	D	E
0	8	8	8	8
*	10_A	8	8	5_A
*				*
*				*
*				*
*				*

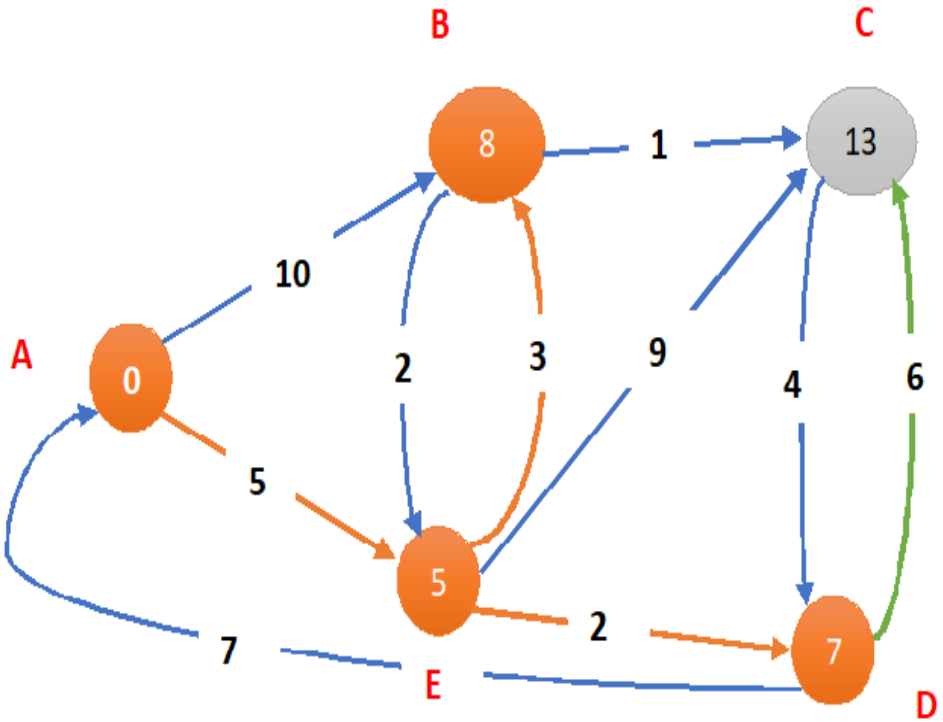
et ainsi de suite



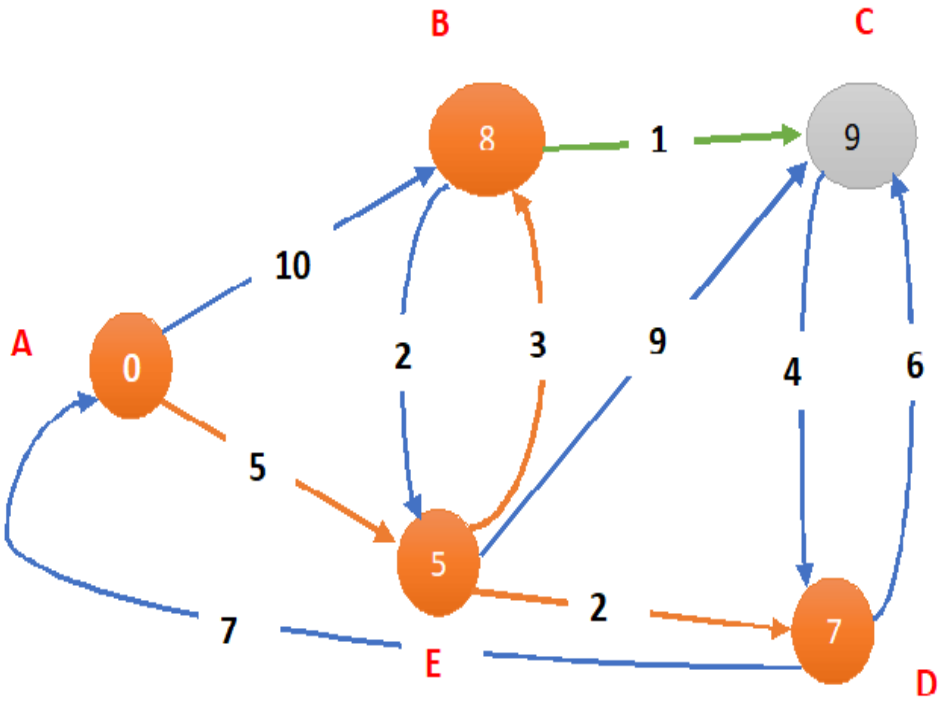
A	B	C	D	E
0	8	8	8	8
*	10_A	8	8	5_A
*	8_E	14_E	7_E	*
*			*	*
*			*	*
*			*	*



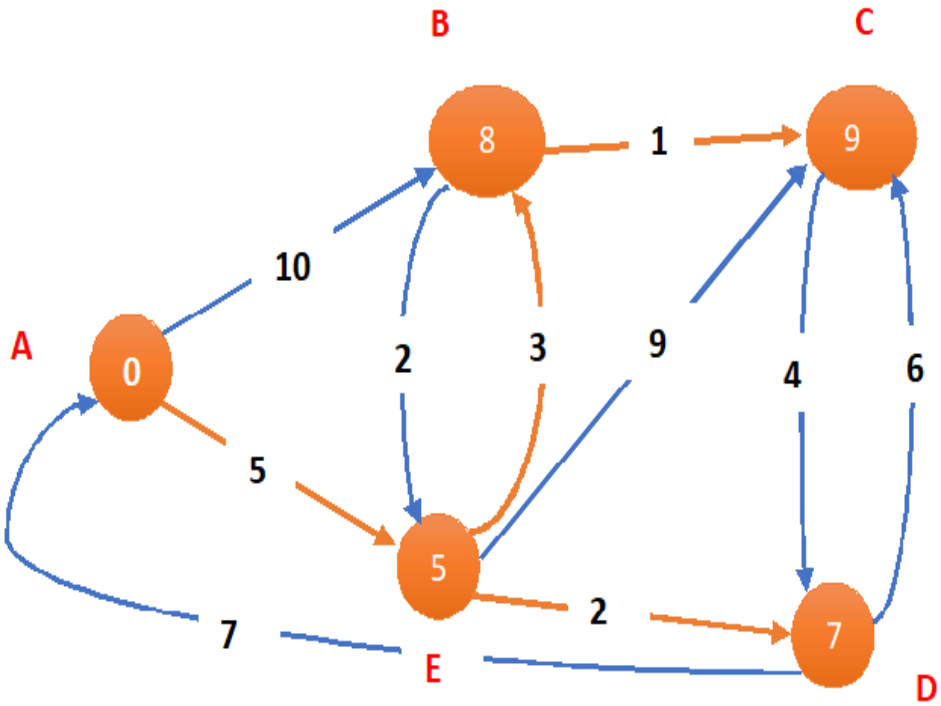
A	B	C	D	E
0	8	8	8	8
*	10_A	8	8	5_A
*	8_E	14_E	7_E	*
*	8_E	13_D	*	*
*			*	*
*			*	*



A	B	C	D	E
0	8	8	8	8
*	10_A	8	8	5_A
*	8_E	14_E	7_E	*
*	8_E	13_D	*	*
*	*		*	*
*	*		*	*



A	B	C	D	E
0	8	8	8	8
*	10_A	8	8	5_A
*	8_E	14_E	7_E	*
*	8_E	13_D	*	*
*	*	9_B	*	*
*	*	*	*	*



A	B	C	D	E
0	8	8	8	8
*	10_A	8	8	5_A
*	8_E	14_E	7_E	*
*	8_C	13_D	*	*
*	*	9_B	*	*
*	*	*	*	*

Si l'on ne considère que les flèches soulignées, on obtient un arbre, un graphe sans cycle.

Outline

La WBS (Work Breakdown Structure)

Dès la conception préliminaire du projet. Il s'agit de décomposer de façon structurée et précise le projet en sous ensembles, de manière à visualiser l'ensemble du projet. Il se fait par niveau successifs jusqu'à un degré optimum de détail, afin d'éviter les oublis, et de permettre la consolidation des informations.
Le découpage de produit, ou " Product Breakdown Structure " (P.B.S) établit l'arborescence des différentes composantes du projet. La décomposition des produit est effectuée par niveaux, selon un principe de " filiation " : pour tout produit de niveau 'n', doivent apparaître les produits le constituant à niveau '$n+1$'.
Pour planifier un projet, il est donc nécessaire de le décomposer en sous-ensembles, tâches ou activités. Cette opération est appelée décomposition structurée des tâches (WBS). La WBS doit inclure toutes les composantes nécessaires à la réalisation d'un projet et qui influent sur sa durée et son coût : tâches de conception, fabrication, matières consommables, équipements, rapports,...

La WBS (Work Breakdown Structure)

On peut associer directement à une WBS les coûts de chaque tâche, les ressources prévues ainsi que les ressources alternatives. L'élaboration d'une WBS simplifie considérablement :

- La planification
- L'identification précoce des activités critiques
- Le contrôle des coûts
- L'élaboration des budgets

La WBS (Work Breakdown Structure)

Si le nombre de tâches auquel on parvient est trop élevé et par conséquent si cela nuit à la clarté de la lecture du planning, on peut, une fois effectue ce découpage minutieux, rassembler différents tâches sous un même terme et constituer ainsi des " work packages " ou lots de travaux. En découpant le projet en éléments successifs depuis le haut de l'arborescence, il est possible d'identifier des éléments de plus en plus simples, dont les coûts, délais et ressources deviennent plus faciles à estimer.
Le degré optimal de décomposition est atteint lorsque les trois critères sont remplis :

- La possibilité de maîtriser la durée d'une activité
- La connaissance des ressources requises
- La possibilité de connaitre le coût de l'activité

Plus la décomposition sera fine et plus la planification et le contrôle seront précis mais plus la coordination entre tâches sera ardue.
On distingue alors :

- Projet
  - Un seul début et une seule fin
  - Début et fin identifiés en tant qu'événements (décision, revue...)
- Sous-projet
  - Projet contenu dans le projet principal
  - Lié à un objet ou un dérivable partiel du projet
- Phase (étape)
  - Ensemble d'actions qui marque un avancement significatif
  - Lié à un type de compétence et à un degré dans la progression du projet
- Tâche
  - Maille la plus fine de la planification
  - Action exécutable par une seule ressource (ou un seul ensemble de ressource)

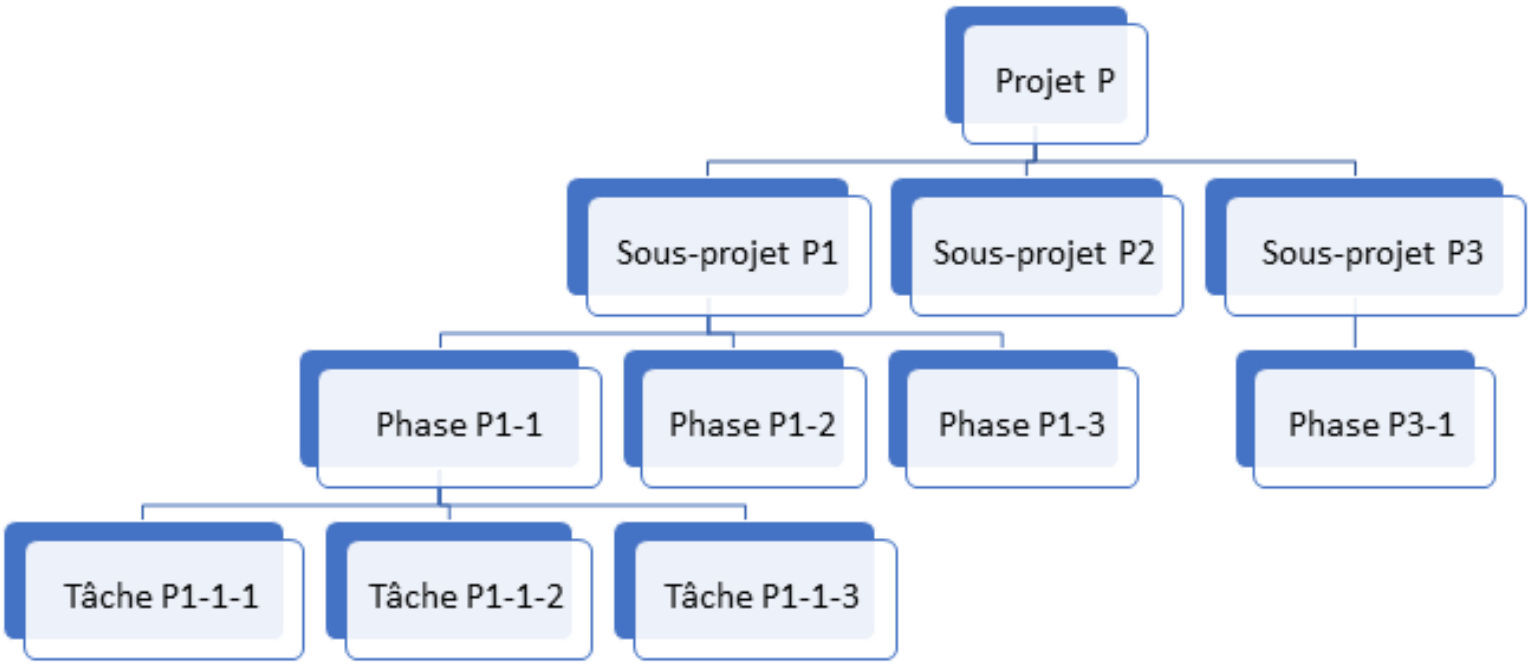


Figure 3 – Schéma général d'une décomposition d'un projet en WBS

Contraintes d'ordonnancement

Les problèmes d'ordonnancement sont apparus au départ dans la planification de grands projets. Le but était de gagner du temps sur leur réalisation. De tels projets sont constitués étapes, également appelées tâches. Des relations temporelles existent entre dernières. Par exemple :

- Une étape doit commencer à une date précise ;
- Un certain nombre de tâches doivent être terminées pour pouvoir en démarrer une autre ;
- Deux tâches ne peuvent être réalisées en même temps (elles utilisent une même machine par exemple) ;
- Chaque tâche nécessite une quantité de main d'oeuvre. Il faut donc éviter, à chaque instant, de dépasser la capacité totale de main d'oeuvre disponible.

Contraintes d'ordonnancement

Toutes ces contraintes ne sont pas simples à prendre en compte dans la résolution de problème. Ici, nous allons nous intéresser uniquement aux deux premiers types de contraintes.
On cherche à déterminer une planification, un ordonnancement des étapes qui minimise le temps total de réalisation de projet. A partir de cette planification, nous verrons que le temps de certaines étapes peut éventuellement être modifié sans entraîner un retard du projet, alors que d'autre, les tâches dites " critiques ", retardent entièrement le projet au moindre retard local.

La réalisation d'un projet nécessite souvent une succession des tâches auxquelles s'attachent certaines contraintes :

- De temps : délais à respecter pour l'exécution des tâches ;
- D'antériorité : certaines tâches doivent s'exécuter avant d'autres ;
- De production : temps d'occupation du matériel ou des hommes qui l'utilisent...

Les techniques d'ordonnancement dans le cadre de la gestion d'un projet ont pour objectif de répondre au mieux aux besoins exprimés par un client, au meilleur coût et dans les meilleurs délais, en tenant compte des différentes contraintes.
L'ordonnancement se déroule en trois étapes :

- La planification : qui vise à déterminer les différentes opérations à réaliser, les dates correspondantes, et les moyens matériels et humains à y attacher.
- L'exécution : qui consiste à la mise en oeuvre des différentes opérations définies dans la phase de planification.
- Le contrôle : qui consiste à effectuer une comparaison entre planification et exécution, soit au niveau des coûts, soit au niveau des dates de réalisation.

Il existe trois méthodes d'ordonnancement : la méthode PERT (Program Evaluation and Research Task), la méthode MPM (Méthode des Potentiels Métra) et le diagramme de Gantt.

L'objectif d'une technique d'ordonnancement est de visualiser sous forme d'un graphe (ou réseau d'ordonnancement) l'enchaînement des tâches de projet, d'estimer la durée global du projet, de déterminer les tâches ne tolèrent pas le retard, les retards tolérées pour certaines tâches, etc. Voici une technique la plus utilisée en pratique de la gestion des projets :

La méthode PERT (Program Evaluation and Research Task)

Les objectifs de la méthode PERT sont : Ordonnancer le projet, calculer la durée du projet, déterminer les tâches critiques, etc.

a-Le réseau PERT

La méthode PERT (Program Evaluation and Research Task) : c'est la méthode la plus connue. Elle a été développée par la marine américaine dans les années 1950 pour coordonner l'action de près de 6000 constructeurs pour la réalisation de missiles à ogives nucléaires POLARIS. C'est aussi la plus "compliquée" à mettre en oeuvre.
Le réseau PERT est un graphe permettant de visualiser et coordonner les l'enchaînement des tâches. Il permet également de focaliser l'attention sur des points (ou étapes) où peuvent être prises des mesures pour respecter les délais et optimiser les coûts. Il est utilisé chaque fois que nous nous trouvons en présence d'activités simultanées avec ou sans partage de ressources. Sa conception s'appuie bien entendu sur la WBS.

a-Le réseau PERT

Le réseau PERT est constitue d'étapes et de tâches élémentaires ou opérations :
Une étape est le commencement ou la fin d'une ou plusieurs tâches. Elle est représentée graphiquement par une cellule dans laquelle sont indiqués le " départ au plus tôt " et le " départ au plus tard " des tâches reliées à ce sommet.
La tâche élémentaire est une action bien déterminée s'inscrivant dans la réalisation du projet. Elle a une durée et consomme de la main d'oeuvre, des matières, des équipements ou d'autres ressources. Elle est représentée graphiquement par une flèche. Une tâche fait évoluer l'ouvrage vers son état final.

a-Le réseau PERT

Dans un graphe PERT :

- Chaque tâche est représentée par une flèche (ou arc). Dans la figure ci-dessous, la tâche est mentionnée par la lettre T et sa durée par le nombre d souvent en unité de temps (seconde, minute, heure, jour...).
- Entre les flèches figurent des cercles appelées " sommets " ou " étapes " qui marquent l'aboutissement d'une ou plusieurs tâches. Ces cercles sont numérotés afin de suivre de succession des diverses étapes du projet (la figure ci-dessous montre deux étapes i et j).

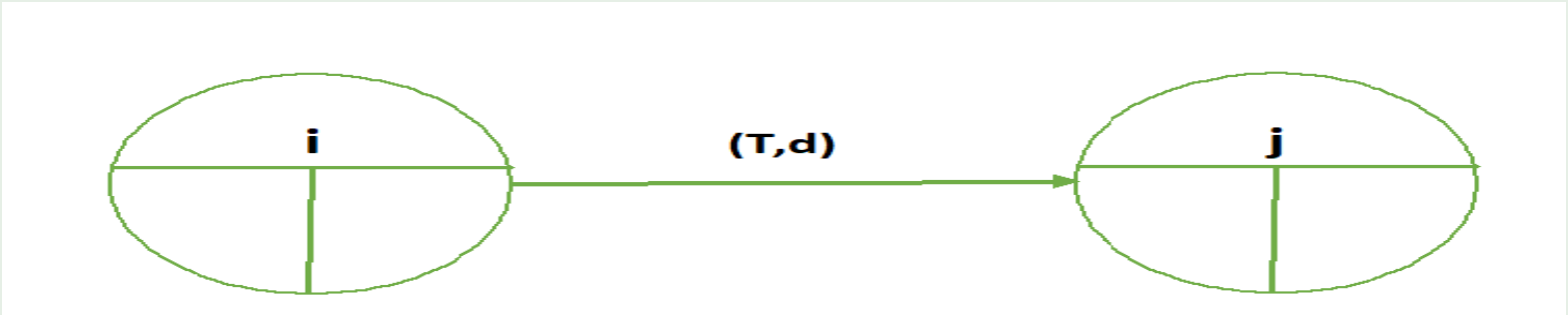


Figure 4

a-Le réseau PERT

Pour construire un graphe PERT, on établit d'abord un tableau donnant les tâches et, pour chacune d'entre elles, les opérations pré-requises. Puis, on utilise la méthode des niveaux qui consiste à :

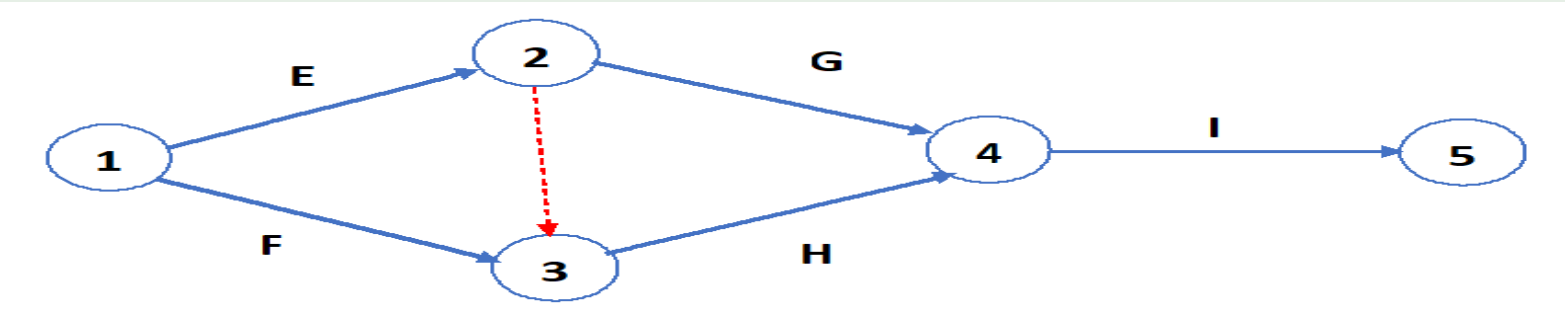
- Déterminer les tâches sans antécédents, qui constituent le niveau 1.
- Identifier les tâches dont les antécédents sont exclusivement du niveau 1. Ces tâches constituent le niveau 2, et ainsi de suite...

Exemple

Soit l'exemple suivant :

Tâches	Durée	Tâches antérieures
E	3	néant
F	5	néant
G	6	E
H	7	E, F
I	5	G, H

Le niveau 1 : E, F, **Le niveau 2** : G, H, **Le niveau 3** : I
Sur le graphique, les tâches d'un même niveau se retrouvent sur une même verticale.



Remarque

Il a été nécessaire d'introduire une tâche fictive de durée égale à 0, pour représenter la relation antérieur d'antériorité entre E, H : en effet, 2 tâches ne peuvent être identifiées par 2 flèches ayant la même origine et la même extrémité.
Ainsi si 2 tâches sont simultanées, elles seront représentées par 2 flèches différentes en partant de la même origine :

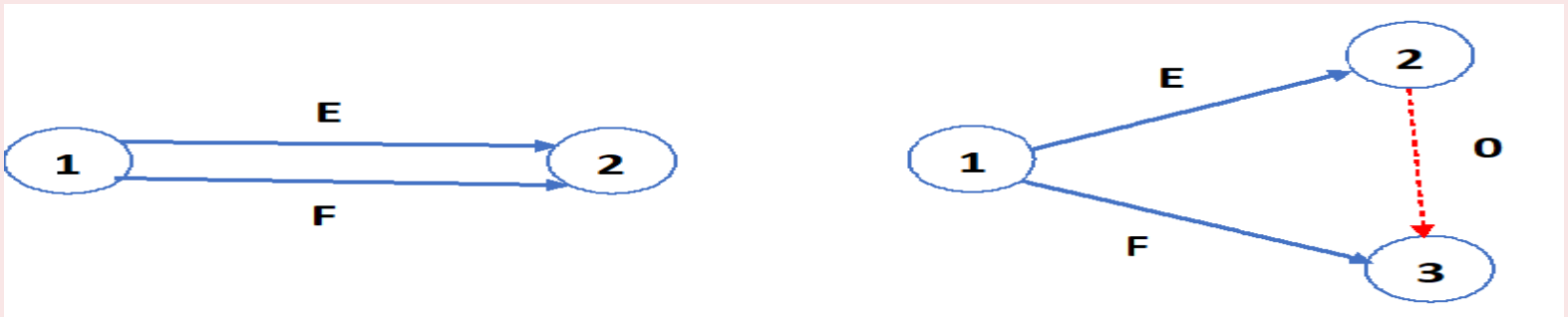


Figure 6

Notons qu'il est très utile, dans le cas général, de déterminer les tâches précédentes directes de chaque tâche. Ceci permettra de savoir quand il est nécessaire d'introduire des tâches fictives : en effet, chaque fois qu'une tâche admet deux (ou plus) précédentes directes qui ont même sommet de début, il est nécessaire de faire dévier une des deux par une flèche fictive. (c'est la cas de la tâche H qui a deux précédentes directes E et F qui ont le même départ.
Enfin les boucles sont interdites ; elles sont contraires à la logique du système. C'est un point qu'il faut toujours vérifier car, à la suite de modification dans les antériorités, on peut créer des boucles de manière insidieuse.

b-Calcul des dates des tâches

Ayant estimé les durées de toutes les tâches constitutives du réseau, on calcule **les dates de début au plus tôt** et **les dates de fin au plus tard** respectivement, pour chaque tâche du projet. Comme il montre la figure ci-dessus, on a :

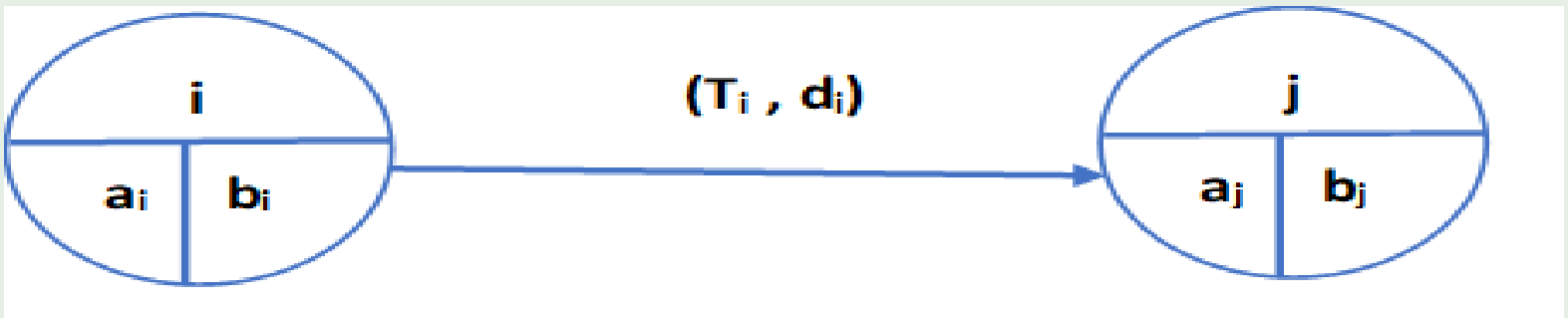


Figure 7

T_{ij} une tâche commence à l'étape i (événement i) et se termine à l'étape j (événement j) de durée d_i (en unité de temps).

a_i : La date de début au plus tôt de la tâche T_i.

Elle est égale au temps cumulé le plus long à partir de l'origine en considérant tous les chemins qui aboutissent à la dite tâche. (c'est la date à laquelle toutes les tâches précédentes de cette tâche sont terminées).Pour la calculer, il faut ajouter à la date au plus tôt de chacune des étapes immédiatement précédentes de l'étape considérée la durée de cette tâche. Parmi les valeurs obtenues, on choisit la plus élevée. On initialise le somme "Début" avec une date au plus tôt = 0.

b_i : La date de début au plus tard de la tâche T_i.

C'est la date limite à laquelle cette tache doit débutée au plus tard afin de ne pas causer de retard dans la réalisation totale du projet. Pour la calculer, on part de la fin du projet en remontant à son début en suivant les liaisons. Elle est égale au temps le plus faible obtenu en remontant de la dernière tâche vers la première. On initialise à l'étape terminale, le dernier sommet "Fin" par la date au plus tard = date au plus tôt.

Calcul des dates au plus tôt des taches

On cherche à quelles dates, au plus tôt, peuvent être exécutées les différentes tâches du projet. Dans une première phase, on établit les dates au plus tôt de chaque étape, début et fin de tâche. La technique est la suivante :
On initialise à 0 l'étape de début, numéroté souvent par 1.
On parcourt le réseau à partir de l'origine dans le sens des flèches. Pour chaque tâche T_i on a : $a_j = a_i + d_i$
Date de fin au plus tôt de tâche=Date de début au plus tôt de tâche+durée de cette tâche.
Si plusieurs chemins aboutissent à une même étape ou noeud, on suit chacun des chemins possibles et on prend pour date au plus tôt de l'étape la plus élevée de toutes les dates obtenues en suivant les différents chemins.

$$a_j = \max\{a_i + d_{ij} : \text{ l'étape } i \text{ prédécesseur de } j\}$$

On arrive ainsi à dater au plus tôt l'étape finale.

Calcul des dates au plus tard des taches

Partant de cette date finale, on procède au datage au plus tard, qui correspond à un calage de toutes les tâches par l'aval. On remonte le réseau dans le sens inverse des flèches. Pour chaque étape i début d'une tâche T_i on a :

$$b_i = b_j - d_{ij}$$

Date de début au plus tard de tâche=Date de fin au plus tard de tâche-durée de cette tâche.
Si plusieurs chemins partent de cette étape, on suit chacun des chemins possibles dans le sens inverse des flèches et on prend pour date au plus tard de cette étape la moins élevée de toutes les dates obtenues en parcourant les différents chemins.

$$b_i = \min\{b_j - d_{ij} : \text{ l'étape } j \text{ successeur de } i\}$$

Calcul des dates au plus tard des taches

On inscrit date au plus tôt et date au plus tard de chaque étape. Si date au plus tôt et date au plus tard sont identiques, la tâche est dite critique. Si les dates sont différentes, il y a un battement ou marge. Il convient d'analyser cette marge, qui peut être une marge libre pour une tâche ou une marge totale pour un chemin.
On appelle chemin critique une succession de tâches critiques du début de projet à la fin du projet. Il représente le chemin le plus long du projet au sens des durées. Pour toutes les tâches du chemin critique, les dates au plus tôt et au plus tard coïncident.

c- Marge libre et marge totale

La marge libre : C'est le retard que l'on peut prendre dans la réalisation d'une tâche sans retarder la date de début au plus tôt de tout autre tâche qui suit.

$$ML(t) = a_j - a_i - d$$

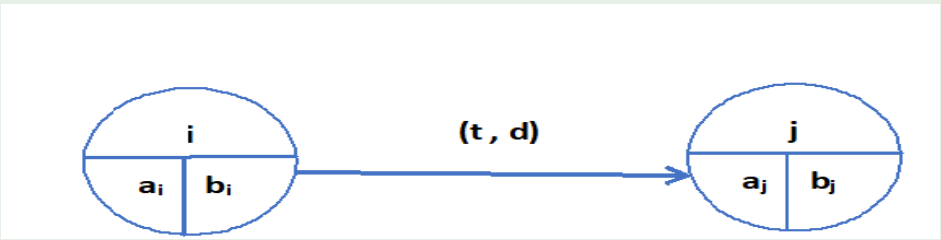


Figure 8

c- Marge libre et marge totale

La marge totale d'une tâche : C'est le retard que l'on peut prendre dans la réalisation de cette tâche sans retarder l'ensemble du projet, elle est obtenue par la différence entre la date au plus tard d'une tâche et la date au plus tôt.

$$MT = b_j - a_i - d$$

Pour toute tâche du chemin critique, la marge totale et la marge libre est nulle.

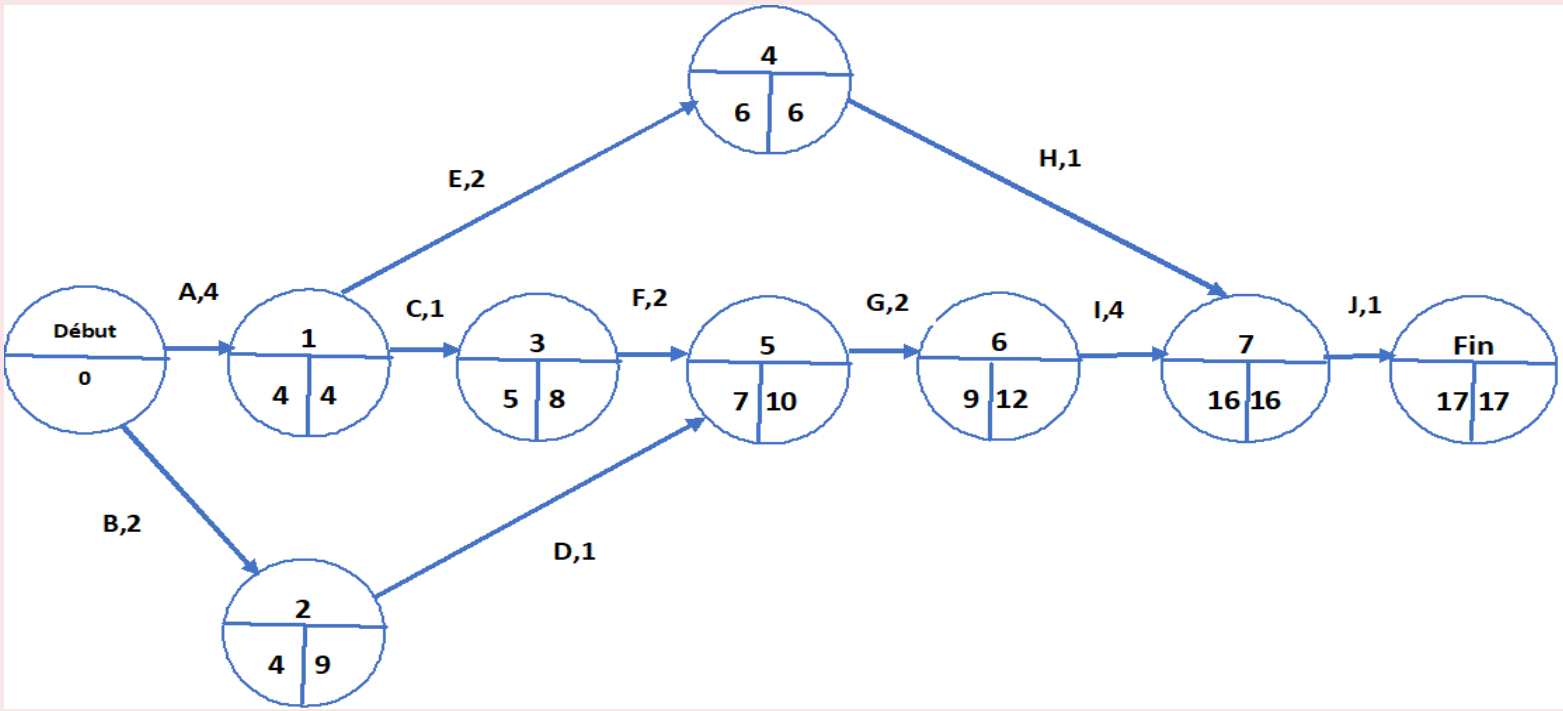
Example d'application

Nous devons déterminer la durée maximale des travaux nécessaires à la construction d'un entrepôt.

Tâches	Tâches antérieures	Durée
A : Etude, réalisation et application des plans	-	4
B : Préparation du terrain	-	2
C : Commande matériaux (bois, briques,. . .)	A	1
D : Creusage des fondations	A, B	1
E : Commandes portes, fenêtres	A	2
F : Livraison des matériaux	C	2
G : Coulage des fondations	D, F	2
H : Livraison portes, fenêtres	E	10
I : Construction des murs, du toit	G	4
J : Mise en place portes et fenêtres	H, I	1

Example d'application

Tracé du réseau PERT La figure ci-dessous représente le graphe PERT du projet, les dates au plus tôt et au plus tard sont calculées pour chaque tâche, on prend par exemple : La date de début au plus tôt de la tâche D est le max de 4+0,0+2 est égal à 4.
La date de début au plus tard de tâche C est le min de 6-2,8-1 est égal à 4.



Example d'application

Le **chemin critique** du graphe est construit par les tâches A-E-H-J a pour durée 17 unité du temps.

Calcul des marges libres et des marges totales

Pour la tâche B par exemple, on a :

La marge libre est ML(B) = 4-0-2 = 2

La marge total est MT(B) = 9-0-2 = 7

le tableau suivant montre la marge libre et total de chaque tâche du projet :

Tâches	A	B	C	D	E	F	G	H	I	J
Marge libre	0	2	0	2	0	0	0	0	3	0
Marge totale	0	7	3	5	0	3	3	0	3	0

Merci pour votre attention