
Université Ibn Zohr

Faculté des Sciences Juridiques Économiques et Sociales

Recherche Opérationnelle

Mohamed HACHIMI

FILIÈRE SCIENCES ÉCONOMIQUES ET GESTION
TROISIÈME ANNEE

EG

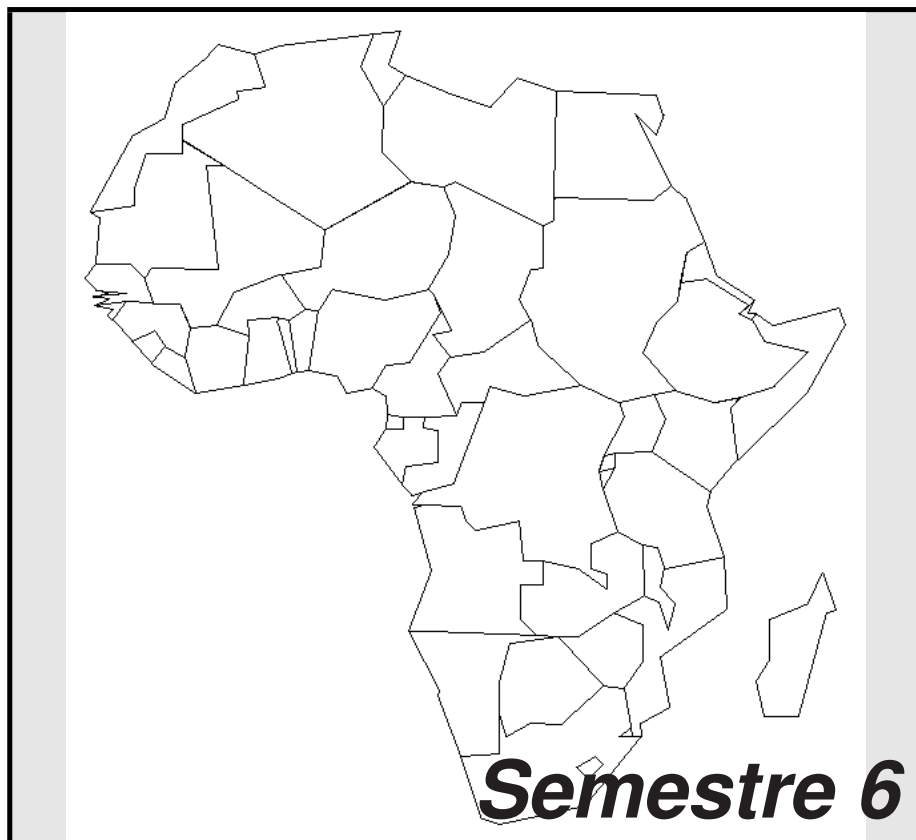


Table des matières

1	Éléments de théorie des graphes	3
1.	La notion de graphe	3
2.	Parcours eulériens et hamiltoniens	13
3.	Coloration des sommets d'un graphe	16

2	Décomposition des graphes	20
1.	Introduction	20
2.	Décomposition basée sur la matrice d'adjacence	21
3.	Décomposition basée sur la matrice de la fermeture transitive	24
4.	Application aux arcs	28

3	Problèmes d'ordonnement	31
1.	Introduction	31
2.	Modélisation par un graphe orienté	32
3.	Construction du graphe PERT	35
4.	Résolution du graphe PERT	38
5.	Diagramme Gantt	48

4	Problème du plus court chemin	49
1.	Introduction	49
2.	Algorithme de Ford	50
3.	Algorithme de Bellman	50
4.	Algorithme de Dijkstra	50

1

Éléments de théorie des graphes

1. La notion de graphe

Devant un grand nombre de situations, le mathématicien, comme d'ailleurs le planificateur ou l'économiste, a été amené à tracer sur le papier des points (représentant des nombres, des individus, des localités, des opérations) et des lignes continues reliant certaines paires de ces points et symbolisant une relation, une route, une préférence, etc.

Pour raisonner sur de tels schémas, on a convenu d'appeler *sommets* ces points, *arcs* ou *arêtes* ces lignes (suivant qu'elles sont orientées ou non), et d'en étudier les propriétés combinatoires.

Il convient de distinguer les graphes *orientés* de ceux qui ne le sont pas. Nous nous intéressons dans ce chapitre aux premiers et nous donnerons quelques exemples et propriétés des seconds dans la section

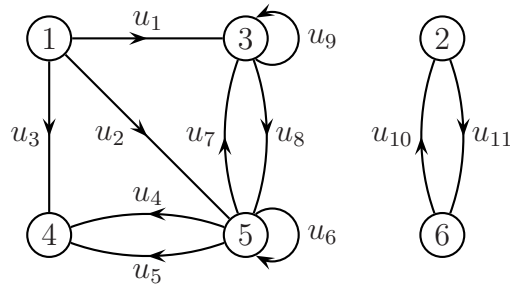
1.1. Graphe orienté

Les graphes orientés dont nous traiterons dans ce cours peuvent être considérés comme des schémas représentant simplement la structure d'un problème. Ils sont généralement déterminés par la donnée de

- 1) un ensemble fini $X = \{x_1, x_2, \dots, x_n\}$ d'éléments appelés *sommets*. Si X possède n éléments distincts, le graphe sera dit d'*ordre* n .
- 2) un ensemble fini $U = \{u_1, u_2, \dots, u_m\}$ dont les éléments sont des couples ordonnés de sommets appelés *arcs*. U est donc une *famille* d'éléments du produit cartésien $X \times X$. Notons qu'un élément (x_i, x_j) de $X \times X$ peut apparaître plusieurs fois dans U . Il nous arrivera parfois dans la suite, par souci de simplification, de noter un arc (x_i, x_j) par (i, j) .

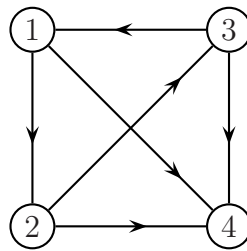
Définition 1 On appelle graphe le couple $G = (X, U)$ formé de deux ensembles : un ensemble X dont les éléments sont appelés sommets, et un ensemble U qui est une famille d'éléments de $X \times X$, dont les éléments sont appelés arcs.

Un exemple de graphe d'ordre 6 est donné par la figure 1 ci-dessous :



Exemple :

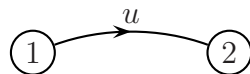
Le graphe $G = (X, U)$ suivant présente les résultats d'un tournoi d'échecs : Les sommets x_1, x_2, x_3 et x_4 représentent les quatres concurrents et l'existence de l'arc (x_i, x_j) indique que le joueur x_i a remporté la partie unique l'opposant à x_j .



$$\begin{cases} X = \{x_1, x_2, x_3, x_4\} \\ U = \{(1, 2), (1, 4), (2, 3), (2, 4), (3, 1), (3, 4)\} \end{cases}$$

Voici quelques définitions de base relatives aux arcs et aux sommets d'un graphe.

Extrémités d'un arc. Pour un arc $u = (x_i, x_j)$, x_i est l'extrémité initiale, x_j l'extrémité finale (ou bien origine et destination). L'arc u part de x_i et arrive à x_j .



Boucle. On appelle boucle un arc dont l'extrémité initiale est égale à son extrémité finale. Par exemple, pour la figure 1, les arcs $u_6 = (5, 5)$ et $u_9 = (3, 3)$ sont des boucles.



Adjacence. Deux sommets reliés par un arc sont dits adjacents. De même, deux arcs ayant au moins une extrémité en commun sont dits adjacents.

Successeur. On dit que x_j est successeur de x_i si $(x_i, x_j) \in U$ (c-à-d s'il existe un arc ayant x_i comme extrémité initiale et x_j comme extrémité finale). L'ensemble des successeurs de x_i est noté $\Gamma^+(x_i)$.

$$\Gamma^+(x_i) = \{x_j \in X \mid u = (x_i, x_j) \in U\}$$

Par exemple, pour la figure 1, $\Gamma^+(x_1) = \{x_3, x_4, x_5\}$ et $\Gamma^+(x_3) = \{x_3, x_5\}$.

Prédécesseur. On dit que x_j est prédécesseur de x_i si $(x_j, x_i) \in U$ (c-à-d s'il existe un arc ayant x_j comme extrémité initiale et x_i comme extrémité finale). L'ensemble des successeurs de x_i est noté $\Gamma^-(x_i)$.

$$\Gamma^-(x_i) = \{x_j \in X \mid u = (x_j, x_i) \in U\}$$

Par exemple, pour la figure 1, $\Gamma^-(x_1) = \emptyset$ et $\Gamma^-(x_3) = \{x_1, x_3, x_5\}$.

Sommets adjacents. On dit que x_i est adjacent à x_j s'il est prédécesseur ou successeur de x_j . Ces sommets peuvent aussi être dits *voisins*. L'ensemble des voisins de x_i est noté $\Gamma(x_i)$.

$$\Gamma(x_i) = \Gamma^+(x_i) \cup \Gamma^-(x_i)$$

Ainsi sur la figure 1 on voit que : $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ et

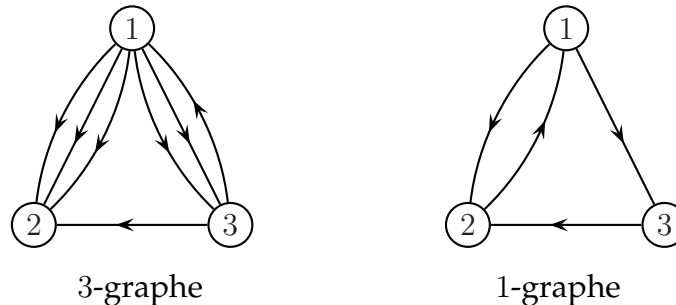
$\Gamma^+(x_1) = \{x_3, x_4, x_5\}$	$\Gamma^-(x_1) = \emptyset$	$\Gamma(x_1) = \{x_3, x_4, x_5\}$
$\Gamma^+(x_2) = \{x_6\}$	$\Gamma^-(x_2) = \{x_6\}$	$\Gamma(x_2) = \{x_6\}$
$\Gamma^+(x_3) = \{x_3, x_5\}$	$\Gamma^-(x_3) = \{x_1, x_3, x_5\}$	$\Gamma(x_3) = \{x_1, x_3, x_5\}$
$\Gamma^+(x_4) = \emptyset$	$\Gamma^-(x_4) = \{x_1, x_5\}$	$\Gamma(x_4) = \{x_1, x_5\}$
$\Gamma^+(x_5) = \{x_3, x_4, x_5\}$	$\Gamma^-(x_5) = \{x_1, x_3, x_5\}$	$\Gamma(x_5) = \{x_1, x_3, x_4, x_5\}$
$\Gamma^+(x_6) = \{x_2\}$	$\Gamma^-(x_6) = \{x_2\}$	$\Gamma(x_6) = \{x_2\}$

Degré. Soit x_i un sommet du graphe G .

- Le demi-degré extérieur de x_i , noté $d^+(x_i)$, est le nombre d'arcs ayant x_i comme extrémité initiale.
- Le demi-degré intérieur de x_i , noté $d^-(x_i)$, est le nombre d'arcs ayant x_i comme extrémité finale.
- Le degré de x_i est $d(x_i) = d^+(x_i) + d^-(x_i)$.

Remarque : dans le cas d'un 1-graphe, on a $d^+(x_i) = |\Gamma^+(x_i)|$ et $d^-(x_i) = |\Gamma^-(x_i)|$.

p -graphe. Dans la séquence U , il peut y avoir plusieurs couples identiques. Un p -graphe est un graphe dans lequel il n'existe jamais plus de p arcs de la forme (i, j) entre deux sommets quelconques.



Remarque : Sur la figure 1, par exemple, on voit que $u_4 = (5, 4)$ et $u_5 = (5, 4)$. Le graphe G de la figure 1 est donc au moins un 2-graphe. Comme il n'y a pas d'autres couples identiques, G est un 2-graphe.

Sauf spécification contraire, tous les graphes considérés dans la suite seront supposés être des 1-graphes. Dans ce cas, on convient d'assimiler U à un ensemble de couples (x_i, x_j) , c-à-d une partie du produit cartésien $X \times X$.

1.2. Matrice d'un graphe

Un 1-graphe G peut être défini par le couple (X, U) . Il est aussi déterminé par la donnée de X et l'application Γ^+ (ou Γ^-). Il peut enfin être caractérisé par sa matrice d'adjacence (ou matrice associée) $M = (a_{ij})$, carrée d'ordre n , où

$$a_{ij} = \begin{cases} 1 & \text{si } (x_i, x_j) \in U \\ 0 & \text{si } (x_i, x_j) \notin U \end{cases}$$

Ainsi, dans l'exemple du tournoi d'échecs, nous avons

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

Matrices booléennes

Définition 2 Une matrice $M = (a_{ij})$ est booléenne si tous ses éléments a_{ij} ne sont que des 0 et des 1.

Les matrices associées aux graphes d'ordre n (de n sommets) sont donc des matrices booléennes d'ordre n (de n lignes et n colonnes).

Rappelons la définition de la somme booléenne \oplus et du produit booléen \otimes :

$$\begin{array}{ll} 0 \oplus 0 = 0 & 0 \otimes 0 = 0 \\ 1 \oplus 0 = 1 & 1 \otimes 0 = 0 \\ 0 \oplus 1 = 1 & 0 \otimes 1 = 0 \\ 1 \oplus 1 = 1 & 1 \otimes 1 = 1 \end{array}$$

Addition et produit de matrices booléennes Soit $\mathbf{A} = (a_{ij})$ et $\mathbf{B} = (b_{ij})$ deux matrices booléennes d'ordre n

Définition 3 La somme booléenne $\mathbf{A} \oplus \mathbf{B}$ des deux matrices \mathbf{A} et \mathbf{B} est la matrice $\mathbf{C} = (c_{ij})$ où $c_{ij} = a_{ij} \oplus b_{ij}$

par exemple :
$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \oplus \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Les propriétés des deux matrices particulières $\mathbf{\Omega} = (\alpha_{ij})$ où $\alpha_{ij} = 0$ et $\mathbf{\Psi} = (\beta_{ij})$ où $\beta_{ij} = 1$ sont évidentes :

- $\mathbf{A} \oplus \mathbf{\Omega} = \mathbf{A}$, $\mathbf{\Omega}$ est la matrice nulle (élément neutre)
- $\mathbf{A} \oplus \mathbf{\Psi} = \mathbf{\Psi}$, $\mathbf{\Psi}$ est absorbante

Définition 4 Le produit booléen des matrices s'effectue comme le produit habituel sauf que les sommes et produits de réels sont remplacés par des sommes et produits booléens.

par exemple :
$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

La matrice $\mathbf{I} = (\sigma_{ij})$

$$\sigma_{ij} = \begin{cases} 0 & \text{si } i \neq j \\ 1 & \text{si } i = j \end{cases}$$

est élément neutre du produit booléen.

Définition 5 Une puissance d'ordre p de la matrice booléenne \mathbf{A} sera notée

$$\mathbf{A}^{[p]} = \underbrace{\mathbf{A} \otimes \mathbf{A} \otimes \cdots \otimes \mathbf{A}}_{p \text{ fois}}$$

Par exemple, si $\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$, alors $\mathbf{A}^{[2]} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$

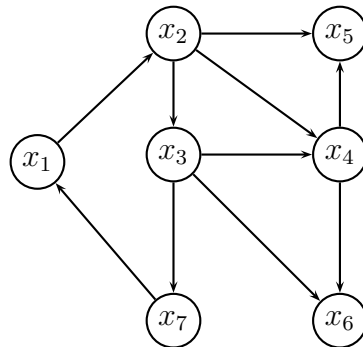
$$\mathbf{A}^{[3]} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

On en déduit que pour tout $p \geq 2$, $\mathbf{A}^{[p]} = \mathbf{A}^{[2]}$.

Existence d'un chemin de longueur donnée Soit G un 1-graphe d'ordre n et \mathbf{M} sa matrice associée. La matrice $\mathbf{M}^{[p]}$ ($p \leq n$) permet de déterminer l'existence d'un chemin de longueur p d'un sommet x_i à un sommet x_j : L'élément a_{ij} de la matrice $\mathbf{M}^{[p]}$ vaut

$$a_{ij} = \begin{cases} 0 & \text{s'il n'existe pas de chemin de longueur } p \text{ de } x_i \text{ vers } x_j \\ 1 & \text{s'il existe un chemin de longueur } p \text{ de } x_i \text{ vers } x_j \end{cases}$$

Considérons le graphe suivant :



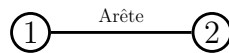
On a

$$\mathbf{M}^{[1]} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{M}^{[2]} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{M}^{[3]} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

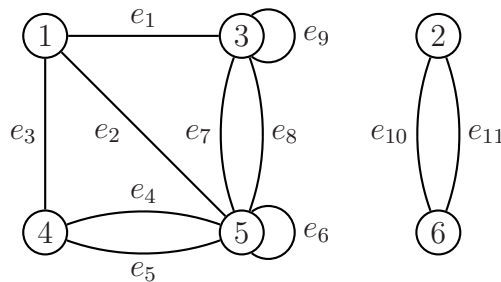
$$\mathbf{M}^{[4]} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \quad \mathbf{M}^{[5]} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \quad \mathbf{M}^{[6]} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

1.3. Graphe non orienté

Lors de l'étude de certaines propriétés, il arrive que l'orientation des arcs ne joue aucun rôle. On s'intéresse simplement à l'existence de *lignes continues* entre deux sommets. Une ligne continue sans orientation, joignant x_i et x_j , est appelé *arête* qu'on note $e = [x_i, x_j]$.

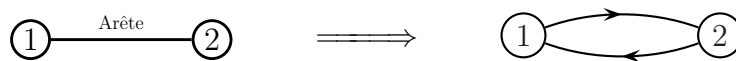


Si E est l'ensemble des arêtes, on parle alors du *multigraphe* $G = (X, E)$. Un multigraphe est par ailleurs appelé un *graphe simple* s'il n'a pas de boucle et s'il existe au maximum une arête entre deux sommets.



Pour une arête $e = [x_i, x_j]$, on dit que u est *incidente* aux sommets x_i et x_j . Le degré d'un sommet est le nombre d'arêtes qui lui sont incidentes.

Remarque : 1° Un arc non orienté peut toujours être transformé en une situation où l'on n'a que des arcs orientés.



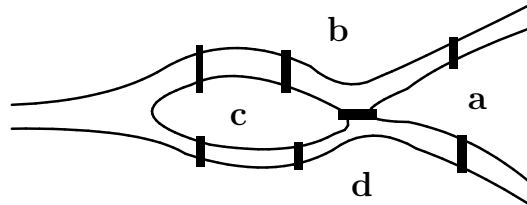
2° A tout graphe orienté $G = (X, U)$, on associe le multigraphe (X, E) où :

$$[x_i, x_j] \in E \iff [(x_i, x_j) \in U \text{ ou } (x_j, x_i) \in U]$$

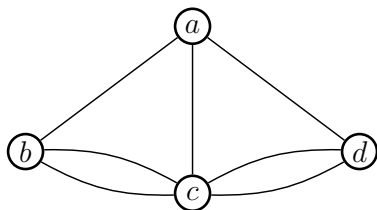
Le multigraphe de la figure 10 est associé au graphe orienté de la figure 1.

Exemple :

Les sept ponts de Königsberg La ville de Königsberg (aujourd'hui Kaliningrad) est traversée par la rivière Pregel qui la divise en quatre régions. Ces régions sont reliées entre elles à l'aide de sept ponts, comme le montre le dessin suivant :



Un piéton désire se promener en empruntant une et une seule fois chaque pont. Pour simplifier ce problème, on ramène le dessin au schéma suivant :



Les quatre régions sont représentées par des sommets, et les sept ponts par des arêtes. Ainsi on obtient un multigraphe.

Lemme 1 *Poignées de mains* Soit $G = (X, E)$ un graphe simple, alors

$$\sum_{x \in X} d(x) = 2|E|$$

En effet, chaque paire $[x, y]$ de E est comptée deux fois, une fois pour $d(x)$ et une seconde fois pour $d(y)$.

Remarque : Le lemme des poignées de mains reste valable pour les multigraphes avec boucles en *convenant* qu'une boucle contribue pour 2 dans le calcul du degré d'un sommet.

Corollaire 1 *Un graphe simple a un nombre pair de sommets de degré impair.*

Notons P l'ensemble des sommets de degré pair et I l'ensemble des sommets de degré impair d'un graphe simple $G = (X, E)$. P et I forment une partition de X , d'après le lemme des poignées de mains, on a :

$$\sum_{x \in X} d(x) = 2|E| = \sum_{x \in P} d(x) + \sum_{x \in I} d(x)$$

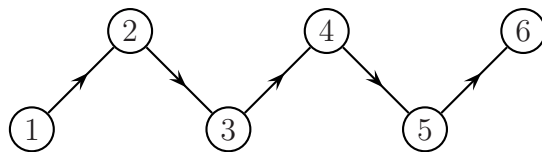
Or $2|E|$ et $\sum_{x \in P} d(x)$ sont des entiers pairs, on en déduit alors que $\sum_{x \in I} d(x)$ est également pair, comme différence de deux entiers pairs. Chaque terme de cette dernière somme est impair, elle ne peut donc être paire que si et seulement si le nombre de termes est pair, on a donc montré que I est un entier pair.

Exemple :

Plusieurs personnes assistent à une réunion. Personne ne serre sa propre main et deux personnes quelconques de l'assemblée se serrent la main au plus une fois. Alors, le nombre de personnes ayant serré la main *un nombre impair de fois* est pair.

1.4. Chemin et chaîne

Chemin. On appelle chemin une suite ordonnée d'arcs u_1, u_2, \dots, u_p telle que l'extrémité finale de chaque arc coïncide avec l'extrémité initiale de l'arc suivant.

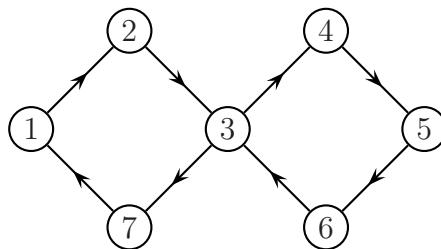


Autrement dit, un chemin μ est une succession d'arcs parcourus dans le même sens. Si un chemin rencontre successivement les sommets $x_1, x_2, \dots, x_p, x_{p+1}$, on peut aussi le noter :

$$\mu = [x_1, x_2, \dots, x_p, x_{p+1}]$$

La *longueur* du chemin est égale au nombre d'arcs qui le composent.

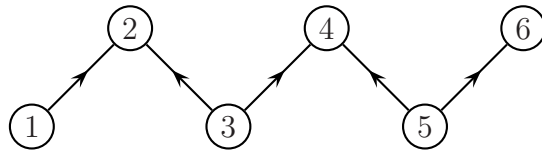
Circuit. Un circuit est un chemin pour lequel l'extrémité finale du dernier arc est confondue avec l'extrémité initiale du premier arc.



Autrement dit, on parle de circuit lorsqu'un chemin revient à son point de départ.

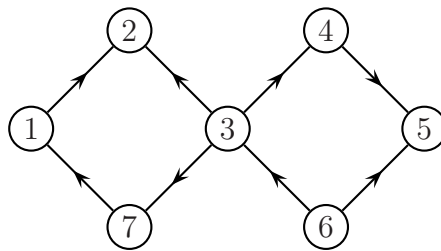
Lorsque ces arcs sont tous distincts, on dit que le chemin ou le circuit est *simple*, quand ils ont pour extrémité finale (ou initiale) des sommets tous différents, on dit qu'il est *élémentaire*.

Chaîne. On appelle chaîne une suite ordonnée d'arcs u_1, u_2, \dots, u_p telle que chaque arc est rattaché à l'arc qui le précède par une extrémité et à celui qui le suit par l'autre extrémité.



Généralement, on parle de chaîne si l'on ne tient pas compte de la direction des arcs.

Cycle. Un cycle est une chaîne pour laquelle l'extrémité finale du dernier arc est confondue avec l'extrémité initiale du premier arc.



Lorsque ces arcs sont tous distincts, on dit que la chaîne ou le cycle est *simple*, quand ils ont pour extrémité finale (ou initiale) des sommets tous différents, on dit qu'elle ou qu'il est *élémentaire*.

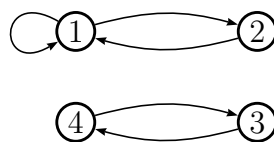
1.5. Types de graphes

Voici quelques types de graphes jouissant de propriétés particulières :

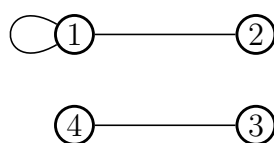
Graphe symétrique. Un graphe est dit symétrique si deux sommets adjacents sont toujours reliés par deux arcs (un dans chaque sens) :

$$(x_i, x_j) \in U \implies (x_j, x_i) \in U$$

Un graphe symétrique a donc une allure particulière. Pour chaque arc « aller » (x_i, x_j) , il y a un arc « retour » (x_j, x_i) .



Dans ce cas, il est plus commode de représenter ce graphe en remplaçant les doubles arcs par une simple arête.



Graphe anti-symétrique. Un graphe est dit anti-symétrique si deux sommets adjacents ne sont jamais reliés par deux arcs (un dans chaque sens) :

$$(x_i, x_j) \in U \implies (x_j, x_i) \notin U$$

Dans un graphe anti-symétrique, pour chaque arc « aller » (x_i, x_j) , il n'y a pas d'arc « retour » (x_j, x_i) .

Graphe complet. Un graphe est dit complet si deux sommets quelconques sont toujours adjacents :

$$(x_i, x_j) \notin U \implies (x_j, x_i) \in U$$

Autrement dit, un graphe est complet si tout couple de sommets est relié dans au moins un des deux sens.

Graphe transitif. Un graphe est dit transitif si pour trois sommets quelconques on a :

$$[(x_i, x_k) \in U \text{ et } (x_k, x_j) \in U] \implies (x_i, x_j) \in U$$

Autrement dit, un graphe est transitif s'il existe toujours un arc allant de l'origine d'un chemin quelconque à son extrémité.

Graphe connexe. Un graphe est dit connexe s'il existe au moins une chaîne entre deux sommets quelconques de G .

2. Parcours eulériens et hamiltoniens

L'étude des problèmes eulériens - ou hamiltoniens - (recherche d'une chaîne ou d'un cycle passant exactement une fois par chaque arête - ou par chaque sommet - remonte aux origines de la théorie des graphes.

L'intérêt porté aujourd'hui à ces problèmes s'explique par leurs nombreuses applications : tournées de distribution, tracé automatique sur ordinateur, problèmes d'ordonnement d'atelier, etc.

2.1. Chaînes et cycles eulériens

Il s'agit là d'une généralisation du jeu bien connu consistant à dessiner toutes les arêtes d'un graphe avec un crayon sans jamais le soulever, ni passer deux fois sur la même arête.

Soit $G = (X, U)$ un graphe orienté.

Chaîne eulérienne. Une chaîne eulérienne est une chaîne empruntant une fois et une fois seulement chaque arc de G .

Cycle eulérien. Un cycle eulérien est une chaîne eulérienne dont les extrémités coïncident.

Un graphe possédant un cycle eulérien est appelé *graphe eulérien*.

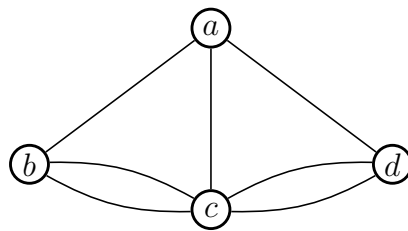
Le problème de l'existence et de la détermination d'un cycle eulérien (d'une chaîne eulérienne) dans un graphe non orienté a été posé la première fois et résolu par Euler en 1736 à propos du célèbre problème des ponts de Königsberg. Euler prouva l'impossibilité de l'obtention d'une solution en démontrant le théorème suivant :

Théorème 1

- a) Un graphe non orienté connexe possède une chaîne eulérienne si et seulement si le nombre de sommets de degré impair est égal à 0 ou 2.
- b) Un graphe non orienté connexe admet cycle eulérien si et seulement si tous ses sommets ont un degré pair.

Exemple :

Le problème des sept ponts de Königsberg peut maintenant être posé en ces termes : existe-t-il une chaîne ou un cycle eulérien dans le graphe suivant :



Il est facile de vérifier que les quatre sommets sont de degré impair : il n'existe donc pas de solution au problème!

Le problème d'Euler peut aussi être considéré avec des « sens uniques ».

Chemin eulérien. Un chemin dans un graphe orienté $G = (X, U)$ est dit eulérien s'il passe exactement une fois par chaque arc de G .

Théorème 2

- a) Un graphe orienté connexe admet un chemin eulérien (mais pas de circuit eulérien) si, et seulement si, pour tout sommet sauf deux (x_d et x_f), le demi-degré intérieur est égal au demi-degré extérieur et

$$d^-(x_d) = d^+(x_d) - 1 \quad \text{et} \quad d^-(x_f) = d^+(x_f) + 1$$

- b) Un graphe orienté connexe admet un circuit eulérien si, et seulement si, pour tout sommet, le demi-degré intérieur est égal au demi-degré extérieur.

2.2. Chaînes et cycles hamiltoniens

Soit $G = (X, U)$ un graphe connexe d'ordre n .

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

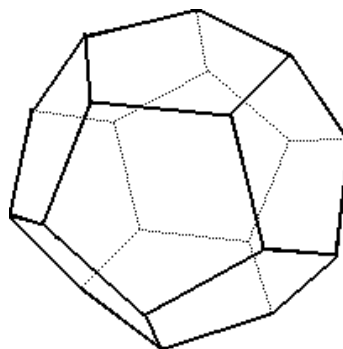
Un chemin hamiltonien (une chaîne hamiltonienne) est donc un chemin (une chaîne) élémentaire de longueur $n - 1$.

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

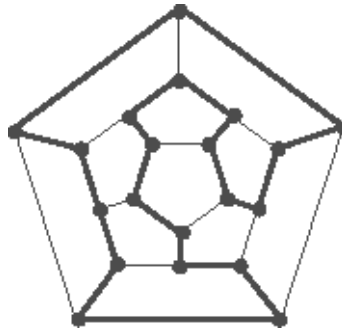
On dit qu'un graphe G est hamiltonien s'il contient un cycle hamiltonien (cas non orienté) ou un circuit hamiltonien (cas orienté).

Exemple :

1° La notion de cycle hamiltonien trouve son origine dans le jeu inventé, en 1859, par Hamilton : ce jeu consiste en un dodécaèdre régulier en bois (polyèdre à 12 faces et 20 sommets), chaque face étant un pentagone régulier comme le montre la figure suivante :



En suivant les arêtes du dodécaèdre, le joueur doit passer une et une seule fois par les vingt sommets du dodécaèdre et, éventuellement, revenir à son point de départ.



2° Un représentant de commerce doit rendre visite à n clients x_1, x_2, \dots, x_n en partant d'une ville x_0 et revenir à son point de départ. Il connaît les distances d_{0j} qui séparent le dépôt x_0 de chacun de ses clients x_j , ainsi que la distance d_{ij} entre deux clients quelconques x_i et x_j .

Dans quel ordre doit-il rendre visite à ses clients pour que la distance totale parcourue soit minimale ? Ce problème revient à chercher un cycle hamiltonien de longueur totale minimale dans le graphe complet G construit sur l'ensemble des sommets $X = \{x_1, x_2, \dots, x_n\}$, les arêtes étant munies des longueurs d_{ij} .

3. Coloration des sommets d'un graphe

Définition 6 Soit $G = (X, E)$ un graphe non orienté sans boucle. Une partie $S \subset X$ est dite stable s'il ne comprend que des sommets non adjacents deux à deux :

$$\forall x_i, x_j \in S \implies [x_i, x_j] \notin E$$

Comme tout sous-ensemble d'un ensemble stable est un ensemble stable, il est naturel de chercher le cardinal maximum d'un ensemble stable. Ce nombre, noté $\alpha(G)$, est le nombre de stabilité.

Définition 7 Soit G un graphe non orienté sans boucle. Une coloration de G est une application $\gamma : X \longrightarrow C$ de l'ensemble des sommets de G dans un ensemble fini C de couleurs, telle que si une arête e est incidente à x et y alors $\gamma(x) \neq \gamma(y)$.

Le nombre chromatique $\gamma(G)$ est défini comme le nombre minimum de couleurs distinctes nécessaires à la coloration des sommets de G .

Exemple :

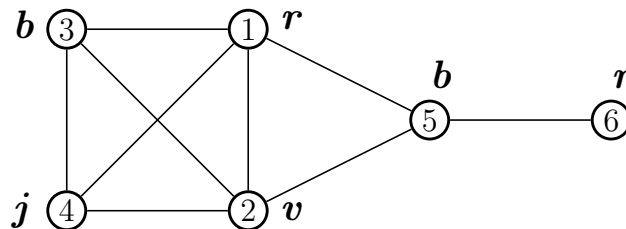
Quel est le nombre chromatique $\gamma(G)$ du graphe $G = (X, E)$ donné par $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ et

$$E = \{[x_1, x_2], [x_1, x_3], [x_1, x_4], [x_2, x_3], [x_2, x_4], [x_3, x_4], [x_1, x_5], [x_2, x_5], [x_5, x_6]\}$$

Les quatre sommets x_1, x_2, x_3, x_4 doivent avoir des couleurs différentes, car ils sont voisins deux à deux. Donc $\gamma(G) \geq 4$. D'autre part, l'application $\gamma : X \rightarrow \{r, v, b, j\}$, où

$$\gamma(x_1) = r, \quad \gamma(x_2) = v, \quad \gamma(x_3) = b, \quad \gamma(x_4) = j, \quad \gamma(x_5) = b, \quad \gamma(x_6) = r.$$

est une coloration de G , donc $\gamma(G) \leq 4$.



Un graphe G tel que $\gamma(G) = k$ qui est coloriable en k couleurs est dit k -chromatique. Une k -coloration des sommets est une partition $\{S_1, S_2, \dots, S_k\}$ de l'ensemble des sommets en k ensembles stables.

Proposition 1 Soit $G = (X, E)$ un graphe simple d'ordre n . On a l'encadrement suivant :

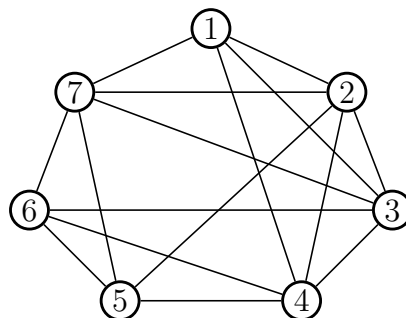
$$\left\lceil \frac{n}{\alpha(G)} \right\rceil \leq \gamma(G) \leq r + 1$$

où r est le plus grand degré des sommets du graphe G .

Exemple :

Un 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 de 1 à 7 et que les paires de cours suivantes ont des étudiants communs : 1 et 2, 1 et 3, 1 et 4, 1 et 7, 2 et 3, 2 et 4, 2 et 5, 2 et 7, 3 et 4, 3 et 6, 3 et 7, 4 et 5, 4 et 6, 5 et 6, 5 et 7 et 6 et 7. Comment organiser ces épreuves de façon qu'aucun étudiant n'ait à passer deux épreuves en même temps et cela sur une durée minimale ?

Construisons le graphe G dont les sommets sont les épreuves numérotées de 1 à 7, une arête relie deux de ses sommets lorsque les deux cours correspondant possèdent des étudiants communs :



Planifier les examens en un temps minimal consiste à déterminer une k -coloration de G , avec $k = \gamma(G)$.

G possède un sous-graphe complet d'ordre 4 (de sommets 1,2,3,4), donc $\gamma(G) \geq 4$. Déterminons une partition des sommets de G en sous-ensembles stables :

$$S_1 = \{1, 6\}$$

$$S_2 = \{2\}$$

$$S_3 = \{3, 5\}$$

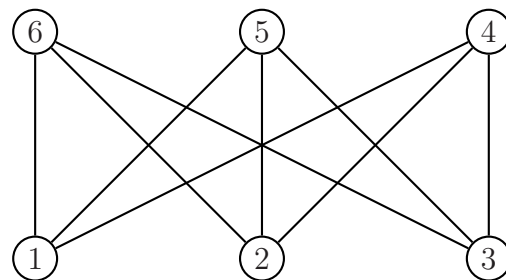
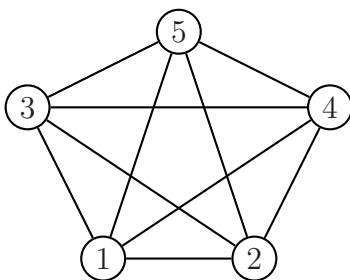
$$S_4 = \{4, 7\}$$

d'où $\gamma(G) \leq 4$, et finalement $\gamma(G) = 4$. Les examens peuvent être répartis en 4 périodes, de la manière suivante :

- 1^{re} période, épreuves des cours 1 et 6
- 2^e période, épreuve du cours 2
- 3^e période, épreuves des cours 3 et 5
- 4^e période, épreuves des cours 4 et 7

Définition 8 *Un graphe planaire est un graphe que l'on peut dessiner sur une surface plate sans que ses arêtes se croisent. Les graphes que l'on ne peut pas dessiner sans croisements sont dits non planaires.*

Tout graphe contenant l'un ou l'autre des deux sous-graphes ci-dessous est non planaire :



Théorème 3 (théorème des quatre couleurs) *On peut colorer les sommets d'un graphe planaire (sans boucles) en utilisant au plus quatre couleurs de telle sorte que toutes les arêtes aient des extrémités de couleurs différentes.*

Cette conjecture a été formulée pour la première fois en 1852. Il était alors question de coloration de carte de géographie (voir l'exemple suivant). La preuve de ce théorème n'arriva qu'en 1976. La démonstration fit grand bruit car ce fut le premier théorème de l'histoire des mathématiques qui a nécessité l'usage systématique de l'ordinateur.

Exemple :

Colorez la carte ci-dessous en utilisant le moins de couleurs possibles, de sorte que deux pays voisins aient des couleurs différentes.

Construisez d'abord un graphe associé à cette carte, puis colorez-en les sommets



2

Décomposition des graphes

1. Introduction

Le but de ce chapitre est de présenter quelques méthodes utilisées pour la décomposition d'un graphe sans circuit en niveaux. Cette décomposition consiste à partitionner l'ensemble X des sommets du graphe $G = (X, U)$ d'ordre n ($n = |X|$), en sous ensembles :

$$X(0), X(1), X(2), \dots, X(p) \quad (p \leq n)$$

appelées niveaux, de telle façon que :

- les sommets du niveau $X(0)$ n'ont pas de prédécesseurs et ce sont les seuls ;
- les sommets du niveau $X(p)$ n'ont pas de successeurs ;
- aucun sommet d'un niveau $X(i)$ ($i = 1, \dots, p$) ne possède de successeurs dans les niveaux $X(i-1), \dots, X(0)$
- il n'y a pas d'arc entre deux sommets d'un même niveau

Il est clair que :

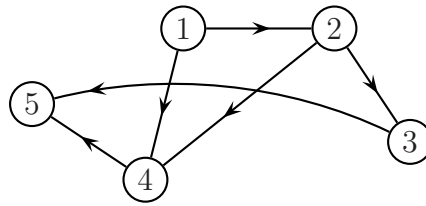
$$\left\{ \begin{array}{l} X(0) = \{x \mid x \in X, \Gamma^-(x) = \emptyset\}; \\ X(1) = \{x \mid x \in (X \setminus X(0)), \Gamma^-(x) \subset X(0)\}; \\ X(2) = \{x \mid x \in (X \setminus (X(0) \cup X(1))), \Gamma^-(x) \subset (X(0) \cup X(1))\}; \\ \vdots \\ X(p) = \{x \mid x \in (X \setminus (X(0) \cup X(1) \cup \dots \cup X(p-1))), \\ \Gamma^-(x) \subset (X(0) \cup X(1) \cup \dots \cup X(p-1))\} \end{array} \right.$$

Proposition 2 Pour qu'un graphe soit décomposable en niveaux, il faut et il suffit qu'il soit sans circuit.

Remarque : Si le graphe G est sans circuit, pour $i = 1, 2, \dots, n$, on a :

$$X(i) = \{x \in X \mid \text{le chemin de longueur maximale arrivant à } x \text{ contient } i \text{ arcs}\}$$

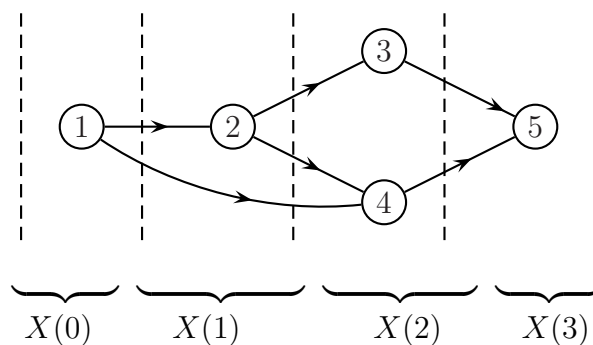
Exemple : Considérons par exemple le graphe G_1 suivant :



On vérifie que G_1 est sans circuit et on trouve

$$X(0) = \{x_1\}, \quad X(1) = \{x_2\}, \quad X(2) = \{x_3, x_4\}, \quad X(3) = \{x_5\}, \quad X(4) = X(5) = \emptyset$$

G_1 peut alors être représenté sous la forme suivante :



on a quatre niveaux.

Soit un graphe sans circuit, présenté pêle mêle, nous allons présenter trois méthodes pour l'ordonner en niveaux de manière à ce qu'on ne peut passer à niveau sans passer par ceux qui le précèdent :

- Décomposition basée sur la matrice d'adjacence
- Décomposition basée sur la matrice de la fermeture transitive du graphe

2. Décomposition basée sur la matrice d'adjacence

Soit $G = (X, U)$ un graphe sans circuit d'ordre n et M sa matrice d'adjacence (Matrice des prédécesseurs). Appelons C_1, C_2, \dots, C_n les colonnes représentant les colonnes de la matrice M . Le principe de la méthode de décomposition de G en niveaux, basée sur la matrice d'adjacence, est le suivant :

- a) Sélectionner tous les sommets de X n'ayant aucun prédécesseur : pour cela calculer le vecteur V_0 défini par :

$$P_0 = C_1 + C_2 + \dots + C_n$$

Il est évident que ces sommets correspondent uniquement aux valeurs **nulles** du vecteur P_0 . Ils forment le sous-ensemble $X(0)$, d'où son nom de niveau **0** ;

- b) Sélectionner tous les sommets de $X \setminus X(0)$ n'ayant de prédécesseurs que dans $X(0)$: cela revient à calculer P_1 en retranchant du vecteur P_0 les colonnes correspondant aux sommets de $X(0)$:

$$P_1 = P_0 - C_{i_1} - C_{i_2} - \dots - C_{i_p}$$

où $C_{i_1}, C_{i_2}, \dots, C_{i_p}$ sont les colonnes qui correspondent aux sommets du niveau 0 et à rechercher les sommets qui correspondent uniquement aux valeurs **nulles** dans la nouvelle colonne P_1 . On obtient ainsi le niveau 1.

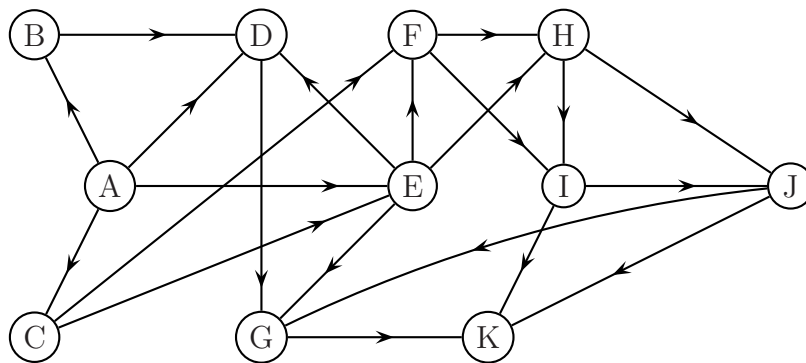
- c) Itérer le processus comme suit : le dernier niveau obtenu étant **N-1**, adopter comme niveau **N** le sous-ensemble formé par les sommets correspondant aux valeurs **nulles** dans le vecteur P_N défini par :

$$P_N = P_{N-1} - C_{j_1} - C_{j_2} - \dots - C_{j_k}$$

où $C_{j_1}, C_{j_2}, \dots, C_{j_k}$ sont les colonnes qui correspondent aux sommets du niveau **N-1**. On obtient ainsi le niveau **N**.

- d) Itérer le processus c) jusqu'à épuisement de tous les sommets de X .

Exemple : Considérons, à titre d'exemple, le graphe G_2 suivant :



Considérons la matrice d'adjacence M de G_2 et appelons $C_A, C_B, C_C, \dots, C_K$ les vecteurs représentant les colonnes de cette matrice.

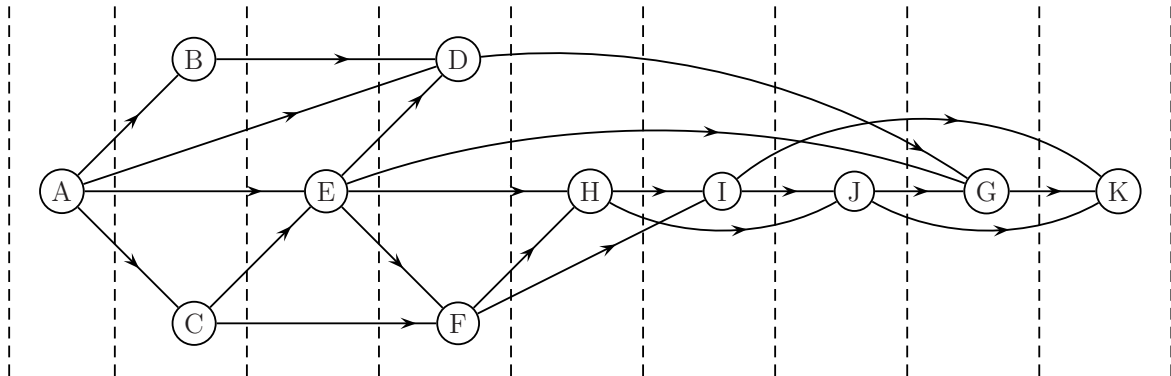
- 1- Calculons le vecteur $P_0 = C_A + C_B + \dots + C_K$. Portons ce résultat à droite de la matrice comme le montre le tableau ci-après. Le vecteur P_0 contient un **zéro** qui correspond au sommet A . Ceci signifie que A n'a pas de prédécesseur, d'où son nom de niveau 0.
- 2- Calculons maintenant le vecteur P_1 : $P_1 = P_0 - C_A$ (élimination de la colonne de A). Deux nouveaux 0 apparaissent en B et C formant le niveau 1.
- 3- Calculons maintenant le vecteur P_2 : $P_2 = P_1 - C_B - C_C$. Un nouveau 0 apparaît en E formant le niveau 2.

	A	B	C	D	E	F	G	H	I	J	K	P_0
A												0
B	1											1
C	1											1
D	1	1			1							3
E	1		1									2
F			1		1							2
G				1	1			1		1		4
H					1	1						2
I						1		1				2
J								1	1			2
K							1		1	1		3

- 4- Calculons le vecteur $P_3 : P_3 = P_2 - C_E$. Deux nouveaux **0** apparaissent pour D et F formant le niveau **3**.
- 5- Calculons le vecteur $P_4 : P_4 = P_3 - C_D - C_F$. Un **zéro** apparaît en H formant le niveau **4**.
- 6- Calculons le vecteur $P_5 : P_5 = P_4 - C_H$. Un **zéro** apparaît en I formant le niveau **5**.
- 7- Calculons le vecteur $P_6 : P_6 = P_5 - C_I$. Un **zéro** apparaît en J formant le niveau **6**.
- 8- Calculons le vecteur $P_7 : P_7 = P_6 - C_J$. Un **zéro** apparaît en G formant le niveau **7**.
- 9- Calculons le vecteur $P_8 : P_8 = P_7 - C_G$. On arrête à P_8 qui n'est formé que de 0 correspondant à K et constituant le niveau **8**.

	A	B	C	D	E	F	G	H	I	J	K	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	
A												0	×	×	×	×	×	×	×	×	×
B	1											1	0	×	×	×	×	×	×	×	×
C	1											1	0	×	×	×	×	×	×	×	×
D	1	1			1							3	2	1	0	×	×	×	×	×	×
E	1		1									2	1	0	×	×	×	×	×	×	×
F			1		1							2	2	1	0	×	×	×	×	×	×
G				1	1			1		1		4	4	4	3	2	1	1	0	×	×
H					1	1						2	2	2	1	0	×	×	×	×	×
I						1		1				2	2	2	2	1	0	×	×	×	×
J								1	1			2	2	2	2	2	1	0	×	×	×
K							1		1	1		3	3	3	3	3	3	2	1	0	0
Niveau												0	1	2	3	4	5	6	7	8	
Sommet												A	B	E	D	H	I	J	G	K	
													C		F						

Les niveaux obtenus sont alors :

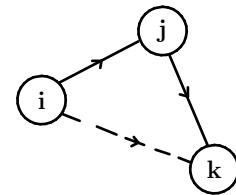


3. Décomposition basée sur la matrice de la fermeture transitive

Définition 9 A tout graphe $G = (X, U)$ on peut associer de façon unique un graphe transitive $\widehat{G} = (X, \widehat{U})$, appelé fermeture transitive de G , où \widehat{U} est défini par

$$(x, y) \in \widehat{U} \iff \text{il existe dans } G \text{ un chemin de } x \text{ vers } y$$

Schématiquement, on construit \widehat{U} en ajoutant à U un arc (x_i, x_k) qui n'appartient pas à U si $(x_i, x_j) \in U$ et $(x_j, x_k) \in U$ pour au moins un j , et ce de façon itérative jusqu'à l'obtention d'un graphe \widehat{G} transitif.



Définition 10 On appelle fermeture transitive d'un sommet x_i d'un graphe, le sous ensemble des sommets du graphe y compris x_i lui même pour lesquels il existe un chemin de x_i à chacun d'eux.

Considérons la p -ième puissance booléenne, de la matrice M associée à G :

$$M^{[p]} = M \otimes M \otimes \dots \otimes M \quad (p \text{ fois})$$

Rappelons que $M^{[p]}$ représente la matrice d'adjacence d'un graphe où un arc de x_i vers x_j signifie qu'il existe dans G un chemin de longueur p allant de x_i vers x_j .

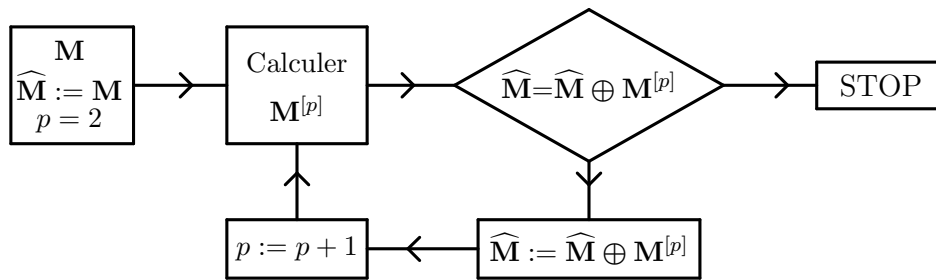
On en déduit la matrice d'adjacence \widehat{M} de \widehat{G} :

$$\widehat{M} = M \oplus M^{[2]} \oplus \dots \oplus M^{[n-1]}$$

Remarque : Soit G un graphe et \widehat{G} sa fermeture transitive. On voit facilement que

- 1- G est transitif si et seulement si la matrice d'adjacence M associée à G est égale à la matrice d'adjacence \widehat{M} associée à \widehat{G} : $\widehat{M} = M$
- 2- G est sans circuit si et seulement si la matrice d'adjacence \widehat{M} associée à \widehat{G} ne possède aucun 1 sur la diagonale. Cette propriété fournit un premier algorithme de reconnaissance d'un graphe sans circuit.

Organigramme de l'algorithme d'obtention de la fermeture transitive d'un graphe



Remarquons que si

$$\mathbf{M} \oplus \mathbf{M}^{[2]} \oplus \dots \oplus \mathbf{M}^{[p]} = \mathbf{M} \oplus \mathbf{M}^{[2]} \oplus \dots \oplus \mathbf{M}^{[p]} \oplus \mathbf{M}^{[p+1]}$$

alors $\widehat{\mathbf{M}} = \mathbf{M} \oplus \mathbf{M}^{[2]} \oplus \dots \oplus \mathbf{M}^{[p]}$ (à expliquer).

Exemple la fermeture transitive \widehat{G}_1 du graphe G_1 .

$$\widehat{G}_1 = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Pour la commodité des calculs, on utilise souvent la matrice \mathbf{M}^* de la fermeture reflexive transitive définie par :

$$\mathbf{M}^* = \mathbf{I} \oplus \mathbf{M} \oplus \mathbf{M}^{[2]} \oplus \dots \oplus \mathbf{M}^{[p]}$$

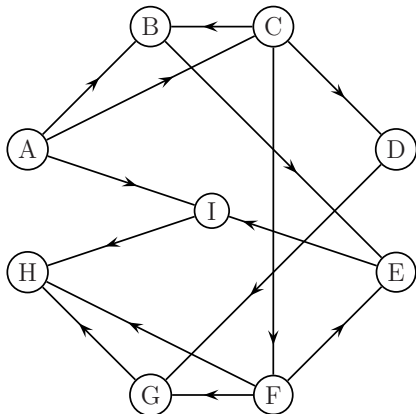
Règle de calcul de la matrice \mathbf{M}^*

a) Le calcul de la fermeture transitive d'un graphe représenté par une matrice d'adjacence se fait en ajoutant les matrices booléennes \mathbf{I} , \mathbf{M} , $\mathbf{M}^{[2]}$, $\mathbf{M}^{[3]}$... jusqu'à obtenir une matrice dont les éléments ne changent plus.

(le nombre de termes de cette somme ne dépasse pas le nombre de sommets du graphe).

b) Mieux on calcule $(\mathbf{M} + \mathbf{I}) \otimes (\mathbf{M} + \mathbf{I}) \otimes \dots \otimes (\mathbf{M} + \mathbf{I})$ jusqu'à obtenir une matrice dont les éléments ne changent plus.

Exemple : Considérons le graphe G_3 suivant :



$$M = \begin{array}{c|cccccccccc} & A & B & C & D & E & F & G & H & I \\ \hline A & & & & & & & & & \\ B & 1 & & 1 & & & & & & \\ C & 1 & & & & & & & & \\ D & & & 1 & & & & & & \\ E & & 1 & & & & 1 & & & \\ F & & & 1 & & & & & & \\ G & & & & 1 & & 1 & & & \\ H & & & & & & 1 & 1 & & 1 \\ I & 1 & & & & 1 & & & & \end{array}$$

► On ajoute I à M puis on calcule $(M \oplus I)^{[2]}$

$$M \oplus I = \begin{array}{c|cccccccccc} & A & B & C & D & E & F & G & H & I \\ \hline A & 1 & & & & & & & & \\ B & 1 & 1 & 1 & & & & & & \\ C & 1 & & 1 & & & & & & \\ D & & & 1 & 1 & & & & & \\ E & & 1 & & & 1 & 1 & & & \\ F & & & 1 & & & 1 & & & \\ G & & & & 1 & & 1 & 1 & & \\ H & & & & & & 1 & 1 & 1 & 1 \\ I & 1 & & & & 1 & & & & 1 \end{array}$$

$$(M \oplus I)^{[2]} = \begin{array}{c|cccccccccc} & A & B & C & D & E & F & G & H & I \\ \hline A & 1 & & & & & & & & \\ B & 1 & 1 & 1 & & & & & & \\ C & 1 & & 1 & & & & & & \\ D & 1 & & 1 & 1 & & & & & \\ E & 1 & 1 & 1 & & 1 & 1 & & & \\ F & 1 & & 1 & & & 1 & & & \\ G & & & 1 & 1 & & 1 & 1 & & \\ H & 1 & & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ I & 1 & 1 & & & 1 & 1 & & & 1 \end{array}$$

► Comme $(M \oplus I)^{[2]} \neq (M \oplus I)$, on calcule $(M \oplus I)^{[3]}$.

$$(M \oplus I)^{[3]} = \begin{array}{c|cccccccccc} & A & B & C & D & E & F & G & H & I \\ \hline A & 1 & & & & & & & & \\ B & 1 & 1 & 1 & & & & & & \\ C & 1 & & 1 & & & & & & \\ D & 1 & & 1 & 1 & & & & & \\ E & 1 & 1 & 1 & & 1 & 1 & & & \\ F & 1 & & 1 & & & 1 & & & \\ G & 1 & & 1 & 1 & & 1 & 1 & & \\ H & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ I & 1 & 1 & 1 & & 1 & 1 & & & 1 \end{array}$$

► Comme $(M \oplus I)^{[3]} \neq (M \oplus I)^{[2]}$, on calcule $(M \oplus I)^{[4]}$. On constatera que l'on obtient la même matrice obtenue précédemment. Donc :

$$M^* = (M \oplus I)^{[3]}.$$

Méthode de décomposition d'un graphe en niveaux

Le principe de la méthode de décomposition de G en niveaux, basée sur la matrice de la fermeture transitive, est le suivant :

- a) Sélectionner les sommets pour lesquels les lignes correspondantes de M^* contiennent un et un seul 1 : ces sommets constituent le niveau 0 ;
- b) Supprimer les lignes et colonnes de M^* correspondant aux sommets du niveau 0 : on forme niveau 1 avec les sommets correspondant aux colonnes contenant un et un seul 1 dans la matrice amputée ;
- c) Itérer le processus b) jusqu'à épuisement de tous les sommets de G .

Exemple : Reprenons le graphe G_3 et considérons la matrice M^* de sa fermeture reflexive transitive :

1- Cherchons les lignes contenant un et un seul nombre 1, il y en a un, c'est la ligne de A (voir Tab. 1). Il constituera le niveau 0.

2- Supprimons maintenant la ligne et la colonne correspondantes au sommet A et cherchons les lignes contenant un et un seul 1, on note celui de C (voir Tab. 2). Donc C va former le niveau 1.

	A	B	C	D	E	F	G	H	I
→ A	1								
B	1	1	1						
C	1		1						
D	1		1	1					
E	1	1	1		1	1			
F	1		1			1			
G	1		1	1		1	1		
H	1	1	1	1	1	1	1	1	1
I	1	1	1		1	1			1

Tab. 1

	B	C	D	E	F	G	H	I
B	1	1						
→ C		1						
D		1	1					
E	1	1		1	1			
F		1			1			
G		1	1		1	1		
H	1	1	1	1	1	1	1	1
I	1	1		1	1			1

Tab. 2

3- Supprimons maintenant la ligne et la colonne correspondantes au sommet C et cherchons les lignes contenant un et un seul 1, on note ceux de B , D et F (voir Tab. 3). Donc B , D et F vont former le niveau 2.

4- Supprimons maintenant les lignes et la colonnes correspondantes au sommets B , D et F et cherchons les lignes contenant un et un seul 1, on note ceux de E et G (voir Tab. 4). Donc E et G vont former le niveau 3.

5- Supprimons maintenant les lignes et les colonnes correspondantes aux sommets E et G et cherchons les lignes contenant un et un seul 1, on note celui de I (voir Tab. 5). Donc I va former le niveau 4.

	B	D	E	F	G	H	I
B	1						
D		1					
E	1		1	1			
F				1			
G		1		1	1		
H	1	1	1	1	1	1	1
I	1		1	1			1

Tab. 3

	E	G	H	I
E	1			
G		1		
H	1	1	1	1
I	1			1

Tab. 4

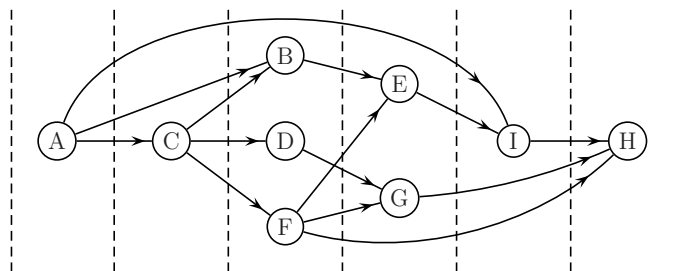
	H	I
H	1	1
I		1

Tab. 5

	H
H	1

Tab. 6

6- STOP : épuisement de tous les sommets du graphe. On obtient alors la décomposition en niveaux suivante :



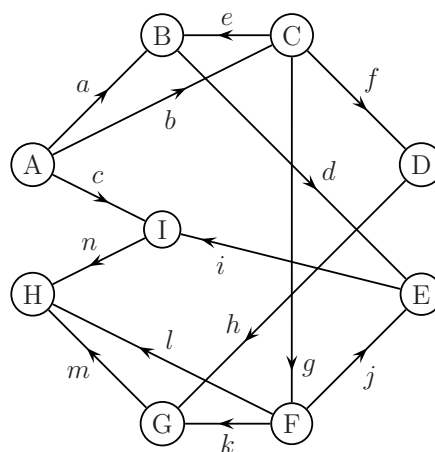
4. Application aux arcs

Les mêmes méthodes étudiées précédemment peuvent être utilisées pour l'ordonnancement des arcs.

Désignons les arcs par des lettres latines minuscules et utilisons l'une des méthodes déjà vues, la méthode basée sur la matrice d'adjacence par exemple pour le cas vu précédemment. La matrice d'adjacence M se définit comme suit :

$$M = (a_{ij}) \text{ où : } a_{ij} = \begin{cases} 1 & \text{si l'arc } i \text{ succède l'arc } j \\ 0 & \text{si l'arc } i \text{ ne succède pas l'arc } j \end{cases}$$

Exemple : Reprenons le graphe G_3 :



La matrice d'incidence se présente comme suit :

	a	b	c	d	e	f	g	h	i	j	k	l	m	n
a														
b														
c														
d	1				1									
e		1												
f		1												
g		1												
h						1								
i				1						1				
j							1							
k							1							
l							1							
m								1			1			
n			1						1					

Appelons C_a, C_b, \dots, C_m les vecteurs représentant les colonnes de cette matrice.

1- Calculons :

$$p_0 = C_a + C_b + \dots + C_m$$

Portons ce résultat à droite de la matrice comme le montre le tableau ci-dessus.

Le vecteur p_0 contient des **zéros** qui correspondent aux arcs a, b et c . Ceci signifie que a, b et c n'ont pas de prédécesseurs, donc ils forment le niveau **0**.

2- Calculons maintenant le vecteur p_1 :

$$p_1 = p_0 - C_a - C_b - C_c$$

Il apparaît trois nouveaux **zéros** pour e, f et g qui seront dit niveau **1**.

3- Calculons maintenant le vecteur p_2

$$p_2 = p_1 - C_e - C_f - C_g$$

Cinq nouveaux **zéros** apparaissent en d, h, j, k et l formant le niveau **2**.

4- Calculons le vecteur p_3

$$p_3 = p_2 - C_d - C_h - C_j - C_k - C_l$$

Deux nouveaux **zéros** apparaissent pour i et m formant le niveau **3**.

5- Calculons le vecteur p_4

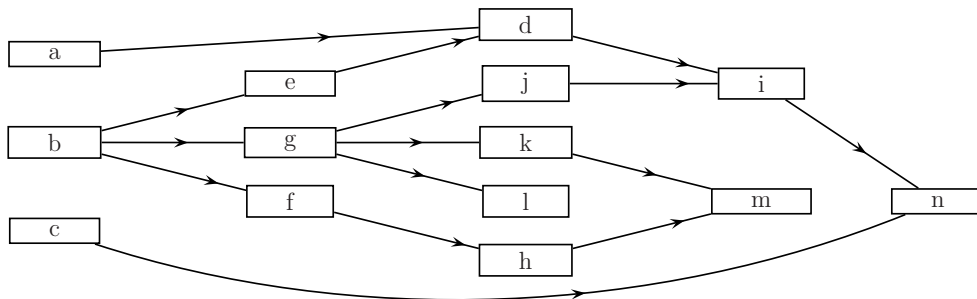
$$p_4 = p_3 - C_i - C_m$$

Un nouveau **zéro** apparaît pour n qui forme le niveau **4**.

6– STOP, il n’y a plus d’arcs.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	p_0	p_1	p_2	p_3	p_4
a															0	×	×	×	×
b															0	×	×	×	×
c															0	×	×	×	×
d	1				1										2	1	0	×	×
e		1													1	0	×	×	×
f		1													1	0	×	×	×
g		1													1	0	×	×	×
h						1									1	1	0	×	×
i				1						1					2	2	1	0	×
j							1								1	1	0	×	×
k							1								1	1	0	×	×
l							1								1	1	0	×	×
m								1			1				2	2	2	0	×
n			1						1						2	1	1	1	0
Niveau															0	1	2	3	4
Arcs															a	e	d	i	n
															b	f	h	m	
															c	g	j		
																	k		
																	l		

La décomposition en niveaux des arcs est suivante :



3

Problèmes d'ordonnancement

1. Introduction

Lors de tout projet de grande envergure (construction d'une maison, d'un bateau, d'un avion,...), un problème crucial qui se pose est celui du calendrier d'exécution des tâches. Le problème est de déterminer dans quel ordre doivent s'enchaîner les diverses tâches de manière à minimiser le temps total d'exécution du projet.

Prenons un exemple. On veut construire une nouvelle maison de manière à pouvoir déménager au plus tôt. Certaines tâches ne peuvent s'exécuter qu'après que d'autres soient terminées. Par exemple, on ne peut commencer les fondations que lorsque la mise en place du chantier est finie. On ne peut monter les gros œuvre que lorsque les fondations sont terminées. D'autres tâches peuvent s'exécuter simultanément. Par exemple, la réalisation des plans et les formalités d'achat du terrain peuvent être menées de pair. Les données sont reprises au tableau ci-dessous pour cet exemple.

Code	Nom de la tâche	durée	antériorités
<i>a</i>	Réalisation des plans	8	–
<i>b</i>	Formalités d'achat du terrain	6	–
<i>c</i>	Obtention du permis de construire	9	<i>a, b</i>
<i>d</i>	Branchements et mise en place du chantier	4	<i>c</i>
<i>e</i>	Fondations	6	<i>d</i>
<i>f</i>	Gros oeuvre	24	<i>e</i>
<i>g</i>	Livraison de matériel de plomberie	27	<i>c</i>
<i>h</i>	Revêtements extérieurs	3	<i>f</i>
<i>i</i>	Plomberie, électricité, menuiserie	30	<i>f, g</i>
<i>j</i>	Plâtres	4	<i>i</i>
<i>k</i>	Finitions intérieures	3	<i>j</i>
<i>l</i>	Finitions extérieures	2	<i>h</i>
<i>m</i>	Clôture du chantier	1	<i>k, l</i>

On doit tenir compte, dans les problèmes d'ordonnement, de divers types de contraintes.

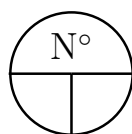
- Les **contraintes de localisation temporelle** expriment la localisation d'une tâche dans le temps : une tâche ne peut commencer avant une telle date, ou après une telle date (par exemple, en raison des conditions climatiques).
- Les **contraintes de succession temporelle** expriment les relations d'antériorité entre les tâches : une telle tâche ne peut commencer avant la fin d'une autre (par exemple, on ne coule pas les fondations si le chantier n'est pas mis en place).
- Les **contraintes disjonctives** expriment le fait que deux tâches ne peuvent avoir lieu en même temps sans que l'on puisse dire laquelle doit être effectuée avant l'autre (par exemple, une même grue est utilisée sur deux chantiers).

2. Modélisation par un graphe orienté

La présentation des techniques de planification de tâches s'appuie sur une représentation des tâches en utilisant la méthode PERT (ou « potentiel événements ») qui est la plus couramment utilisée. La méthode PERT repose sur une modélisation du projet qui s'appuie sur un graphe orienté. Certains logiciels utilisent aussi une autre représentation dite « potentiel tâches ». Cette dernière diffère de la méthode PERT par la construction des graphes associés aux problèmes d'ordonnement. Dans ce cours, nous voyons uniquement la méthode PERT.

Dans ce chapitre, nous allons adapter les concepts des graphes afin d'aborder les problèmes d'ordonnement de projets à l'aide de la méthode PERT :

Sommets : Un sommet (ou noeud) d'un graphe correspond dans la méthode PERT à un événement ou à une étape. C'est un jalon matérialisant la chaîne de succession des tâches concrétisant l'avancement ou l'enchaînement sur le plan de réalisation des travaux : Il marque le début ou la fin d'un travail, d'une opération, d'une tâche. En plus, il ne consomme ni temps, ni ressources physiques, ni ressources financières.



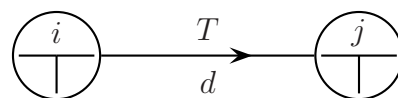
Représentation d'un événement avec son numéro

Numérotation des sommets : Soit un réseau PERT d'ordre n . Ce réseau obéit aux conventions de numérotation suivantes :

- les n sommets (ou événements) sont numérotés de 1 à n
- les sommets 1 et n représentent respectivement le démarrage et le parachèvement du projet :
 - le sommet 1 n'est terminal pour aucune tâche,
 - le sommet n n'est initial pour aucune tâche,
 - tout autre sommet est à la fois terminal pour au moins une tâche et initial pour une ou plusieurs autres tâches.
- le numéro attribué au sommet initial (le début) d'une tâche est inférieur à celui attribué à son sommet terminal (sa fin).

Arcs : Un arc d'un graphe représente dans la méthode PERT une tâche ou une opération. Une tâche est délimitée par deux sommets (ou événements) qui représentent le début et la fin de la tâche. Une tâche est une activité qui consomme du temps, nécessite des ressources physiques et des dépenses.

On symbolise un tâche par une flèche, au-dessus du quelle on inscrit le code de la tâche et en dessous sa durée. La longueur de la flèche n'est pas proportionnelle au délai de la tâche, elle dépend du tracé général du graphe.

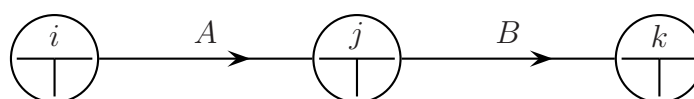


Représentation d'une tâche avec son code et sa durée

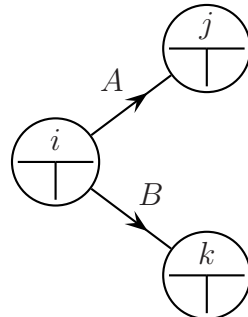
- la lettre T définit le code de la tâche ;
- le chiffre d correspond à sa durée (seconde, minute, heure, jour, . . .) ;
- la flèche indique le sens de l'exécution.

Comment représenter les différentes tâches ? Il est commode de représenter graphiquement les tâches et leurs ordonnancement en s'appuyant sur la liste des prédécesseurs immédiats de chaque tâche.

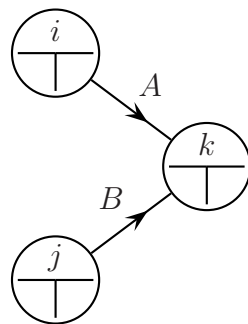
a) Les tâches successives : Pour indiquer que la tâche A est prédécesseur immédiat de la tâche B , l'arc correspondant à B prend son départ là où aboutit l'arc associé à A



b) Les tâches simultanées : Les tâches qui peuvent être exécutées en même temps, elles partent du même sommet : deux tâches A et B qui commencent en même temps se représentent ainsi :

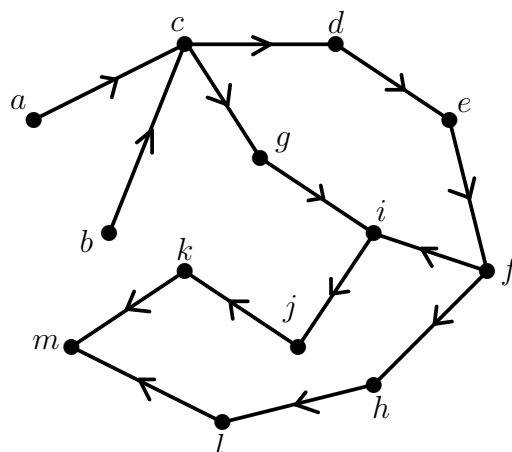


c) Les tâches convergentes : Les tâches qui sont prédécesseurs immédiats de la même tâche, vont vers un événement commun : deux tâches A et B qui précèdent le même événement se représentent ainsi :



Le graphe associé au projet doit représenter l'enchaînement des tâches composant le projet tout en respectant les antériorités et les délais de réalisation existant entre ces tâches.

En ce qui concerne le projet de la construction de la nouvelle maison, le dessin ci-dessous présente un graphe représentant l'enchaînement des tâches.



Exposé sous cette façon, il ne présente aucune utilité pour la planification vu sa complexité.

3. Construction du graphe PERT

3.1. Obtention des niveaux

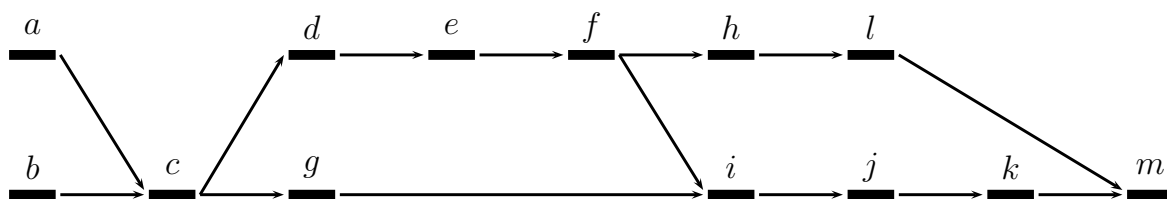
A titre d'illustration, nous allons déterminer les niveaux pour le graphe associé à l'exemple de la construction de la nouvelle maison et ceci selon la méthode basée sur la matrice d'adjacence. La matrice d'adjacence M associée au projet se définit comme suit :

$$M = (a_{ij}) \text{ où : } a_{ij} = \begin{cases} 1 & \text{si la tâche } i \text{ succède la tâche } j \\ 0 & \text{si la tâche } i \text{ ne succède pas la tâche } j \end{cases}$$

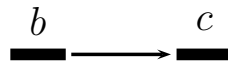
En utilisant le tableau des antériorités immédiates, le calcul est présenté ci-dessous :

	a	b	c	d	e	f	g	h	i	j	k	l	m	v ₀	v ₁	v ₂	v ₃	v ₄	v ₅	v ₆	v ₇	v ₈						
a														0	×	×	×	×	×	×	×	×	×					
b														0	×	×	×	×	×	×	×	×	×					
c	1	1												2	0	×	×	×	×	×	×	×	×					
d			1											1	1	0	×	×	×	×	×	×	×					
e				1										1	1	1	0	×	×	×	×	×	×					
f					1									1	1	1	1	0	×	×	×	×	×					
g			1											1	1	0	×	×	×	×	×	×	×					
h						1								1	1	1	1	1	0	×	×	×	×					
i						1	1							2	2	2	1	1	0	×	×	×	×					
j									1					1	1	1	1	1	1	0	×	×	×					
k										1				1	1	1	1	1	1	1	0	×	×					
l								1						1	1	1	1	1	1	0	×	×	×					
m											1	1		2	2	2	2	2	2	2	1	0	0					
Niveau														0	1	2	3	4	5	6	7	8						
Arcs														a	c	d	e	f	h	j	k	m	b	g	i	l		

on aboutit ainsi à un graphe facile à manipuler à des fins de PRP comme le montre la figure ci-dessous. L'ordonnement des arcs est alors :



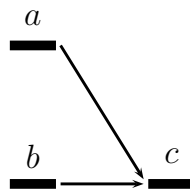
Les flèches indiquent les relations d'antériorité. Le schéma



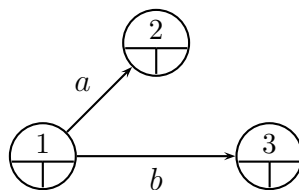
signifie que la tâche c ne peut démarrer avant le parachèvement de la tâche b .

3.2. La construction du graphe PERT

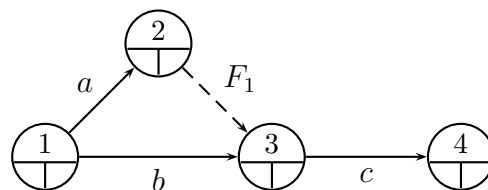
Revenons maintenant au projet de la construction de la nouvelle maison. Considérons le sous projet contenant les tâches a , b et c :



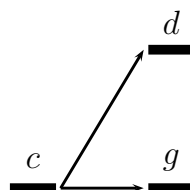
Puisque les tâches a et b n'ont pas de prédécesseurs, elles commencent du sommet 1. Le tracé du graphe des tâches débute comme suit :



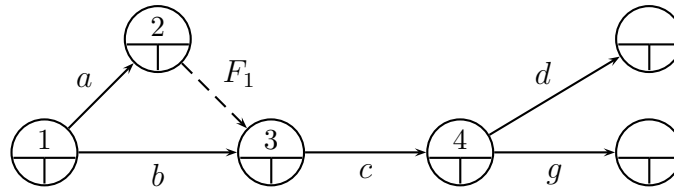
La tâche c admet comme prédécesseurs immédiats les tâches a et b . Il faudrait donc que a et b partagent le même sommet final. Donc a et b sont deux tâches parallèles joignant un même couple de sommets. par conséquent, on ajoute entre les sommets 2 et 3 une tâche fictive F_1 :



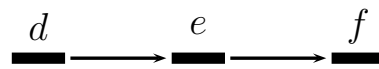
Considérons maintenant le sous projet constitué des tâches c , d et g :



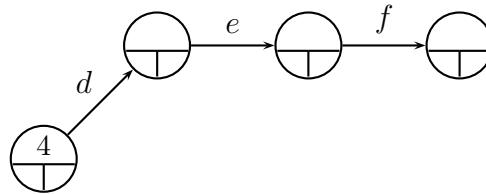
Les tâches d et g partent du même sommet (début de la tâche c). Elles sont des tâches simultanées.



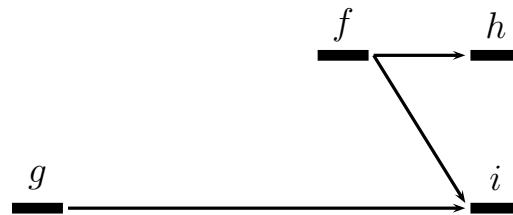
Considérons maintenant le sous projet formé des tâches d, e et f :



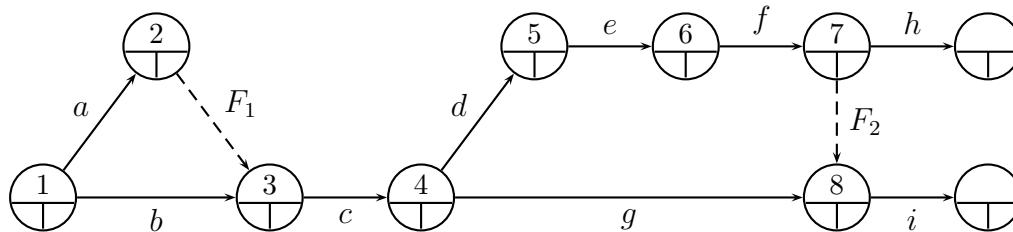
Les tâches d, e et f sont des tâches successives, elles se représentent ainsi :



Le sous projet formé des tâches f, g, h et i présente une difficulté.

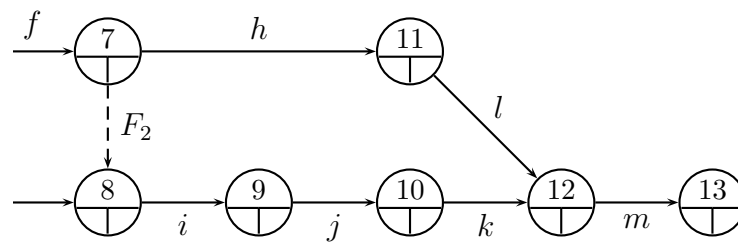


En effet, les tâches f et g sont deux tâches convergentes. Par conséquent elles devraient partager le même sommet final, qui serait le début de i . D'autre part, i et h sont deux tâches simultanées. Elles devraient partager le même sommet initial, qui serait la fin de f . Ce qui implique que g précède h , bien que g ne fasse pas partie de la liste des prédécesseurs immédiats de h . Pour empêcher cela, il a fallu ajouter un arc représentant une tâche fictive F_2 (donc de durée 0) reliant la fin de la tâche f au début de la tâche i .

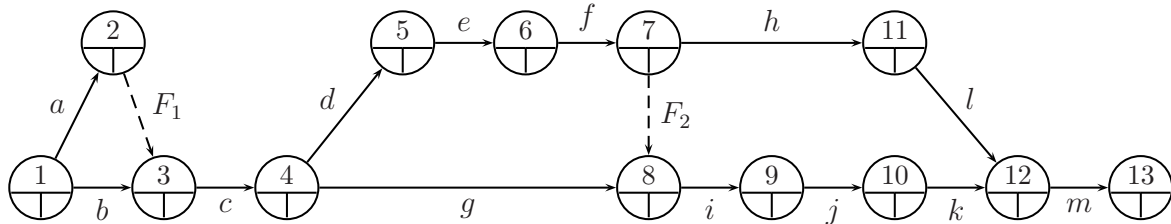


Notons donc que l'existence d'un successeur immédiat h , propre à la seule tâche f , interdit de donner à f et g un même sommet final même si elles sont convergentes.

Pour compléter le graphe PERT du projet, il suffit d'ajouter les tâches j, k, l et m , en respectant les règles déjà vues.



La figure suivante illustre le graphe complet du projet de la construction du nouvelle maison.



4. Résolution du graphe PERT

La technique PERT-Temps consiste en une suite de quatre étapes : les dates « au plus tôt », les dates « au plus tard », les « marges » et le « chemin critique ».

4.1. Calcul des dates au plus tôt

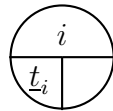
Les dates au plus tôt sont obtenues en traitant le réseau logique sur une échelle de temps qui démarre à une date $t_0 = 0$, et se déroule vers l'avenir.

Le calcul des dates au plus tôt correspond intuitivement à la question : si le projet démarre à $t_0 = 0$, quand se terminera-t-il ? et quelles en seront les dates intermédiaires ?

Nous allons maintenant voir un algorithme de calcul des dates au plus tôt. Le calcul des dates au plus tôt détermine les dates d'atteinte au plus tôt des étapes, notées t_i , en partant du sommet de début du projet (étape 1).

Illustrons les choses sur l'exemple de la construction de la nouvelle maison. En commençant par l'étape 1 (sommet 1), nous allons calculer pour chaque étape la date au plus tôt, c'est-à-dire la première date où il sera possible d'atteindre l'étape. Les dates au plus tôt sont associées aux sommets représentant les débuts ou fins de tâches.

La date au plus tôt est positionnée en général dans la partie se trouvant à droite du cercle matérialisant le sommet (l'événement ou l'étape).



Représentation d'une étape i avec sa date au plus tôt \underline{t}_i .

Pourvu que la numérotation des sommets respecte les conventions mentionnées à la section 2, le calcul des \underline{t}_i se fera selon l'ordre croissant des numéros. Tout d'abord le sommet 1 correspond au début du projet,

$$\underline{t}_1 = 0$$

Pour obtenir la date au plus tôt d'une étape j , il faut comparer pour chaque étape précédente i , la somme :

$$(\text{date au plus tôt de l'étape } i) + (\text{durée de la tâche entre } i \text{ et } j)$$

La date au plus tôt est alors la plus grande de ces sommes. De manière générale la date au plus tôt se calcule comme suit :

Une seule tâche avant une étape	Plusieurs tâches avant une étape
$\underline{t}_j = \underline{t}_i + d_{ij}$	$\underline{t}_k = \max\{\underline{t}_i + d_{ik}, \underline{t}_j + d_{jk}\}$

Déterminons maintenant \underline{t}_2 : la tâche a est la seule qui aboutisse en 2 ; cette tâche dure 8, par conséquent,

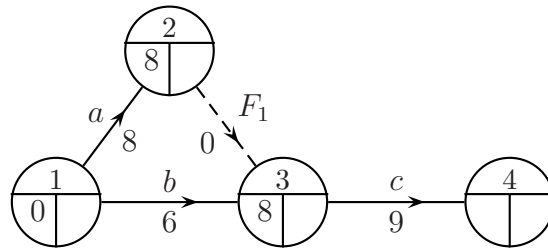
$$\underline{t}_2 = \underline{t}_1 + (\text{durée de } a) = 0 + 8 = 8.$$

Pour atteindre le sommet 3, il faut que les tâches b et F_1 soient parachevées ; d'où \underline{t}_3 serait le plus grand de temps suivant :

$$\underline{t}_1 + (\text{durée de } b) = 0 + 6 = 6$$

$$\underline{t}_2 + (\text{durée de } F_1) = 8 + 0 = 8$$

le date au plutôt à retenir est : $\underline{t}_3 = \max\{6, 8\} = 8.$



Cette date au plus tôt de 8 est donc impliquée par l'antériorité de la tâche fictive F_1 .

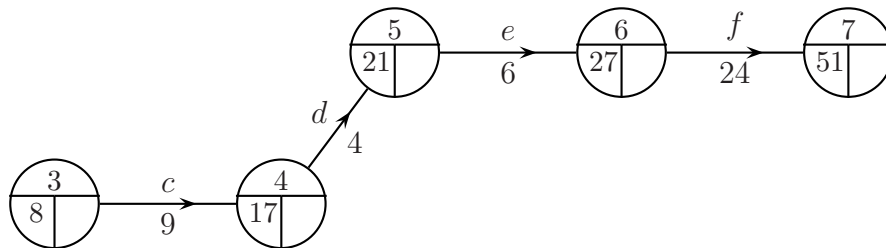
De même,

$$t_4 = t_3 + (\text{durée de } c) = 8 + 9 = 17$$

$$t_5 = t_4 + (\text{durée de } d) = 17 + 4 = 21$$

$$t_6 = t_5 + (\text{durée de } e) = 21 + 6 = 27$$

$$t_7 = t_6 + (\text{durée de } f) = 27 + 24 = 51$$



Pour le reste

$$t_8 = \max\{t_4 + (\text{durée de } g), t_7 + (\text{durée de } F_2)\} = \max\{17 + 27, 51 + 0\} = 51$$

$$t_9 = t_8 + (\text{durée de } i) = 51 + 6 = 57$$

$$t_{10} = t_9 + (\text{durée de } j) = 57 + 4 = 61$$

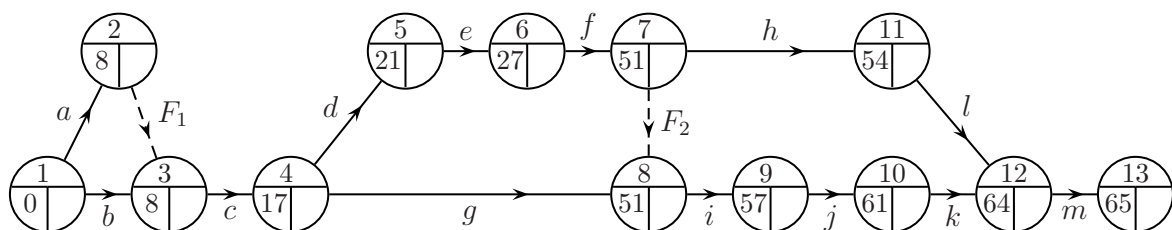
$$t_{11} = t_7 + (\text{durée de } h) = 51 + 3 = 54$$

$$t_{12} = \max\{t_{11} + (\text{durée de } l), t_{10} + (\text{durée de } k)\} = \max\{54 + 2, 61 + 3\} = 64$$

$$t_{13} = t_{12} + (\text{durée de } m) = 64 + 1 = 65.$$

Le calcul des dates au plus tôt une fois terminé, le graphe de la page 40 est obtenu ; nous savons maintenant que le projet ne peut se terminer en moins de 65 semaines.

C'est la **durée minimale du projet**.



4.2. Calcul des dates au plus tard

Certaines étapes sont telles que si on retarde leur date au plus tôt, cela aura des répercussions sur la date de fin du projet. Par exemple, si on retarde la date au plus tôt de l'étape 12, cela va directement retarder la date de fin du projet. De même, si on retarde l'étape 10, cela va retarder la date au plus tôt de l'étape 12 qui elle-même retarde la date de fin du projet.

Par contre, si on retarde la date au plus tôt de l'étape 11, cela n'aura pas de répercussion, car ce n'est pas à partir de cette étape que son successeur (étape 12) a été marqué mais bien à partir de l'étape 10. On voit donc que l'on peut retarder la date au plus tôt de l'étape 11 sans conséquence sur la date de fin de chantier jusqu'à un certain point. Autrement dit, la date d'atteinte de l'étape 11 peut être retardée jusqu'à la valeur :

$$\underline{t}_{12} - (\text{durée de } l) = 64 - 2 = 62$$

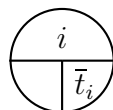
sans retarder la date d'atteinte de l'étape 12. On dit que 62 est la date au plus tard de l'étape 11. C'est-à-dire que l'étape 11 peut être atteinte à cette date au plus tard sans allonger la durée totale minimale des travaux.

Dans la pratique, il est intéressant de connaître la marge de manœuvre dont on dispose pour atteindre de chaque étape. Cette marge sera connue quand nous disposerons des dates au plus tard. La **date au plus tard d'une étape** est la dernière date possible pour atteindre de cette étape sans que cela implique un retard pour le projet.

Les dates au plus tard sont obtenues en traitant le réseau logique sur une échelle de temps qui démarre à une date t_f , et se déroule vers **le passé**.

Le calcul des dates au plus tard correspond intuitivement à la question : si le projet doit se terminer à t_f , quand doit-il démarrer ? et quelles en seront les dates intermédiaires ?

On notera une date au plus tard par \bar{t}_i . La date au plus tard est positionnée en général dans la partie se trouvant à gauche du cercle matérialisant le sommet (l'événement ou l'étape).



Représentation d'un étape i avec sa date au plus tard \underline{t}_i .

On peut calculer les dates au plus tard de la manière suivante :

- Partant du sommet final du projet, pour lequel la date au plus tard coïncide avec la date au plus tôt, puisqu'il ne peut y avoir de marge de manœuvre pour la fin du projet sans que cela ne la retarde.

$$\bar{t}_f = \underline{t}_f = t_f = 64.$$

- Pour obtenir la date au plus tard d'une étape i , il faut comparer pour chaque étape j qui lui succède, la différence :

$$(\text{date au plus tard de l'étape } j) - (\text{durée de la tâche entre } i \text{ et } j)$$

La date au plus tard est alors la plus petite de ces différences. De manière générale la date au tard se calcule comme suit :

Une seule tâche après une étape	Plusieurs tâches après une étape
$\bar{t}_i = \bar{t}_j - d_{ij}$	$\bar{t}_i = \min\{\bar{t}_j - d_{ij}, \bar{t}_k - d_{ik}\}$

Pour notre exemple, on a :

$$\bar{t}_{13} = \underline{t}_{13} = 65$$

$$\bar{t}_{12} = \bar{t}_{13} - (\text{durée de } m) = 65 - 1 = 64$$

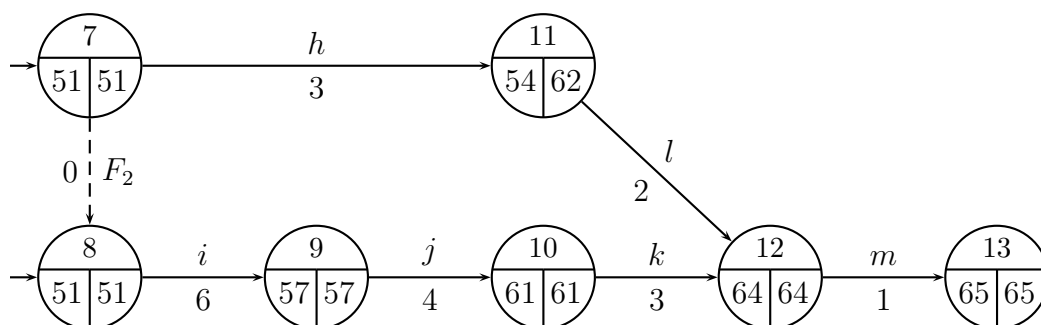
$$\bar{t}_{11} = \bar{t}_{12} - (\text{durée de } l) = 64 - 2 = 62$$

$$\bar{t}_{10} = \bar{t}_{11} - (\text{durée de } k) = 62 - 3 = 61$$

$$\bar{t}_9 = \bar{t}_{10} - (\text{durée de } j) = 61 - 4 = 57$$

$$\bar{t}_8 = \bar{t}_9 - (\text{durée de } i) = 57 - 6 = 51$$

$$\bar{t}_7 = \min\{\bar{t}_8 - (\text{durée de } F_2), \bar{t}_{11} - (\text{durée de } h)\} = \min\{51 - 0, 62 - 3\} = 51$$



$$\bar{t}_6 = \bar{t}_7 - (\text{durée de } f) = 51 - 24 = 27$$

$$\bar{t}_5 = \bar{t}_6 - (\text{durée de } e) = 27 - 6 = 21$$

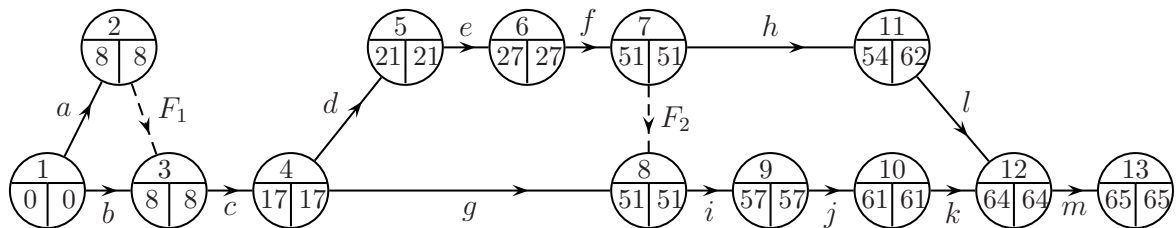
$$\bar{t}_4 = \min\{\bar{t}_5 - (\text{durée de } d), \bar{t}_5 - (\text{durée de } d)\} = \min\{21 - 4, 51 - 27\} = 17$$

$$\bar{t}_3 = \bar{t}_4 - (\text{durée de } c) = 17 - 9 = 8$$

$$\bar{t}_2 = \bar{t}_3 - (\text{durée de } F_1) = 8 - 0 = 8$$

$$\bar{t}_1 = \min\{\bar{t}_2 - (\text{durée de } c), \bar{t}_3 - (\text{durée de } b)\} = \min\{8 - 8, 8 - 6\} = 0$$

Le calcul des dates au plus tard une fois terminé, le graphe de la ci-après donne pour chaque sommet les dates au plus tôt et les dates au plus tard.



remarque : Il est à remarquer que ces dates aux plus tard des *étapes* \bar{t}_i ne correspondent pas dans tous les cas aux dates au plus tard des tâches qui suivent le sommet ! Prenons l'exemple de la tâche b , de durée 6. La date de début au plus tard de son successeur direct, la tâche c est de 8. La date de début au plus tard de la tâche b est donc de

$$8 - 6 = 2$$

alors que la date de début au plus tard du sommet précédent la tâche dans le graphe est de 0. Ce 0 provient en fait de la tâche a , $8 - 8 = 0$, qui est critique (voir plus loin).

4.3. Les marges

On appelle **marge d'une tâche** le retard qu'il est possible de tolérer dans la réalisation de celle-ci, sans que la durée optimale prévue du projet global en soit affectée. Il est possible de calculer trois types de marges : la marge totale, la marge libre et la marge indépendante.

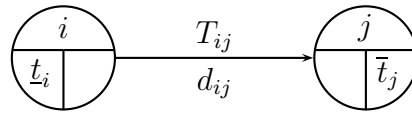
a) Marge totale

La marge totale d'une tâche T_{ij} indique le retard maximal que l'on peut admettre dans sa réalisation (sous réserve qu'elle ait commencé à sa date au plus tôt) sans

allonger la durée optimale du projet.

Elle se calcule en retirant la durée T_{ij} de la tâche en question à l'écart qu'il peut y avoir entre sa date au plus tôt de début \underline{t}_i et sa date au plus tard de fin \bar{t}_j :

$$M_t = \bar{t}_j - \underline{t}_i - d_{ij}$$



Sauf cas particulier, un retard correspondant à la marge totale d'une tâche se traduit par une modification des dates au plus tôt des tâches qui lui succèdent. Il n'est donc pas possible de cumuler des retards correspondant à leur marge totale sur plusieurs tâches successives, sans remettre en cause la durée optimale prévue pour le projet.

On peut aussi écrire :

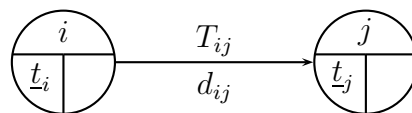
$$M_t = \bar{t}_j - (\underline{t}_i + d_{ij})$$

Autrement dit, la marge total est la différence entre la date au plus tard de l'étape j et l'arrivée au plus tôt à cette étape pour la tâche T_{ij} qui peut partir au plus tôt en \underline{t}_i de l'étape i . On obtient alors les **dates au plus tard des tâches** en additionnant à la date au plus tôt du sommet de départ, la marge de la tâche.

b) Marge libre

La marge libre d'une tâche T_{ij} indique le retard que l'on peut admettre dans sa réalisation (sous réserve qu'elle ait commencé à sa date au plus tôt) sans modifier les date au plus tôt des tâches suivantes et sans allonger la durée optimale du projet. Elle se calcule en retirant la durée d_{ij} de la tâche en question à l'écart qu'il peut y avoir entre ses dates au plus tôt de début \underline{t}_i et de fin \underline{t}_j :

$$M_l = \underline{t}_j - \underline{t}_i - d_{ij}$$



Un retard correspondant à la marge libre d'une tâche reste sans conséquence sur les marges des tâches qui lui succèdent. Il est donc possible de cumuler des retards, s'inscrivant dans leur marge libre, pour plusieurs tâches successives, sans remettre en cause la durée optimale prévue pour le projet.

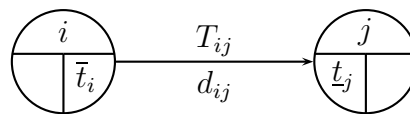
c) Marge indépendante

La marge indépendante (ou certaine) d'une tâche T_{ij} indique le retard que l'on peut admettre dans sa réalisation (quelle que soit sa date de début) sans allonger la durée optimale du projet.

Elle se calcule en retirant la durée d_{ij} de la tâche en question à l'écart qu'il peut y avoir entre sa date au plus tard de début \bar{t}_i et sa date au plus tôt de fin \underline{t}_j :

$$M_i = \max [0, (\underline{t}_j - \bar{t}_i - d_{ij})]$$

D'après cette formule, la marge indépendante est considérée comme nulle lorsque son calcul donne un nombre négatif



Un retard correspondant à la marge indépendante d'une tâche reste sans conséquence sur les marges des tâches qui lui succèdent, même si elle commence à sa date au plus tard. Il est donc possible de cumuler des retards, s'inscrivant dans leur marge indépendante, pour plusieurs tâches successives, même si elles commencent à leur date au plus tard, sans remettre en cause la durée optimale prévue pour le projet.

Remarque : On peut aussi calculer la marge totale d'une tâche T_{ij} par la formule :

$$M_t = M_l + M_e$$

où M_e est la marge liée (elle est propre à une étape). Elle est égale à la différence entre la date au plus tôt et la date au plus tard de l'étape j (fin de la tâche T_{ij}).

Ainsi, dans notre exemple (projet CNM) :

Tâche	Marge totale M_t	Marge libre M_l	Marge indépendante M_i
<i>a</i>	0	0	0
<i>b</i>	2	2	2
<i>c</i>	0	0	0
<i>d</i>	0	0	0
<i>e</i>	0	0	0
<i>f</i>	0	0	0
<i>g</i>	7	7	7
<i>h</i>	8	0	0
<i>i</i>	0	0	0
<i>j</i>	0	0	0
<i>k</i>	0	0	0
<i>l</i>	8	8	0
<i>m</i>	0	0	0

4.4. Le chemin critique

On appelle **tâche critique** toute tâche T_{ij} dont la durée, ajoutée à la date au plus tôt de son sommet de début, égale la date au plus tard de son sommet de fin. Autrement dit, toute tâche dont la **marge totale** est *nulle* est dite critique.

Le fonctionnement sans retard des tâches critiques est essentiel à l'aboutissement normal du projet. En effet, la marge de manoeuvre dont on dispose pour l'exécution d'une tâche est la différence entre sa date au plus tard et sa date au plus tôt ; quand celle-ci est nulle, la marge de manoeuvre est nulle : tout retard dans la tâche entraîne alors un retard de même importance pour l'ensemble du projet.

Dans tout projet, il existe au moins un **chemin critique**, c'est-à-dire un chemin formé de tâches critiques allant de la première jusqu'à la dernière étape du graphe PERT. La somme des durées des tâches qui définissent un chemin critique, parfois appelée longueur du chemin critique, coïncide avec la durée minimale du projet.

Dans notre exemple (projet CNM), le chemin suivant :

$$P = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13]$$

constitue un chemin critique. Le chemin critique est tracé en traits gras sur le graphe PERT.

Le chemin n'est pas forcément *unique*. Si par exemple la durée de la tâche b avait aussi été de 8, un second chemin critique apparaît dans notre exemple :

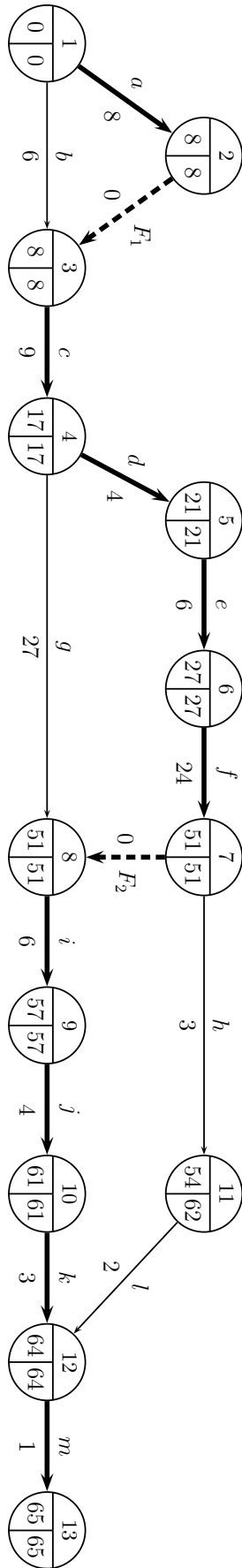
$$P_2 = [1, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13]$$

Il en serait de même pour le chemin

$$P_3 = [1, 2, 3, 4, 5, 6, 7, 11, 12, 13]$$

si, toutes choses restant égales par ailleurs, la durée de la tâche h passait à 11.

Remarquez également que, pour réduire la durée minimum de réalisation du projet, il faut réduire la durée d'une tâche dans chaque chemin critique. En effet, il ne sert à rien de réduire la durée d'un chemin critique si un autre chemin reste critique avec une valeur supérieure. Remarquez que l'on peut soit réduire la durée d'une tâche dans chaque chemin, soit réduire la durée d'une tâche commune aux chemins dans le cas où une telle tâche existe.



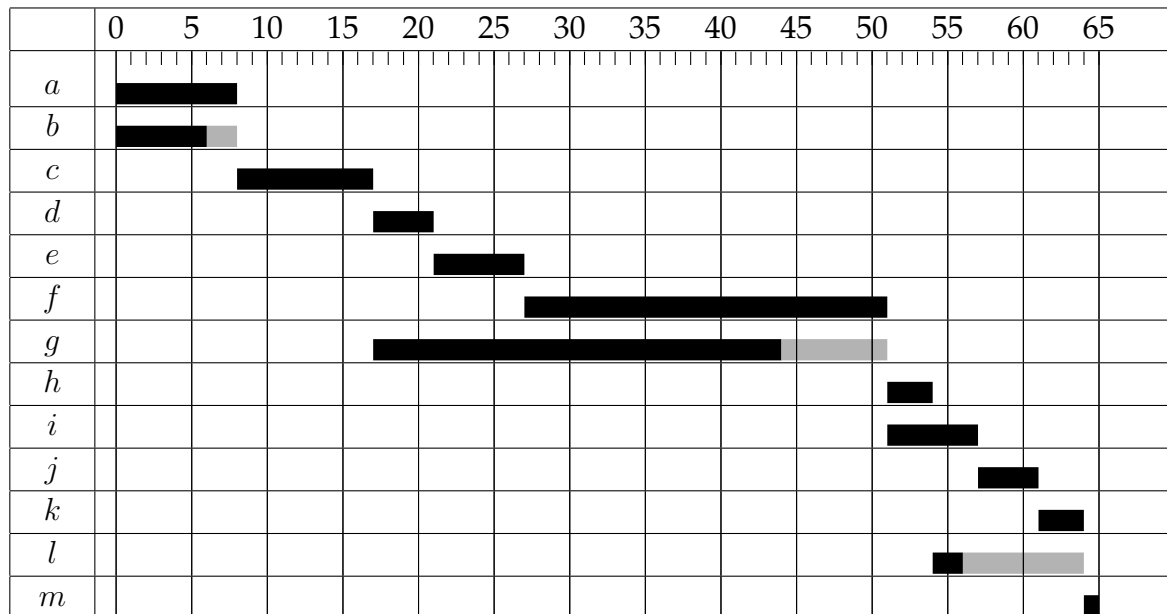
Graphe pour le projet de la construction de la nouvelle maison :
Le **chemin critique**

5. Diagramme Gantt

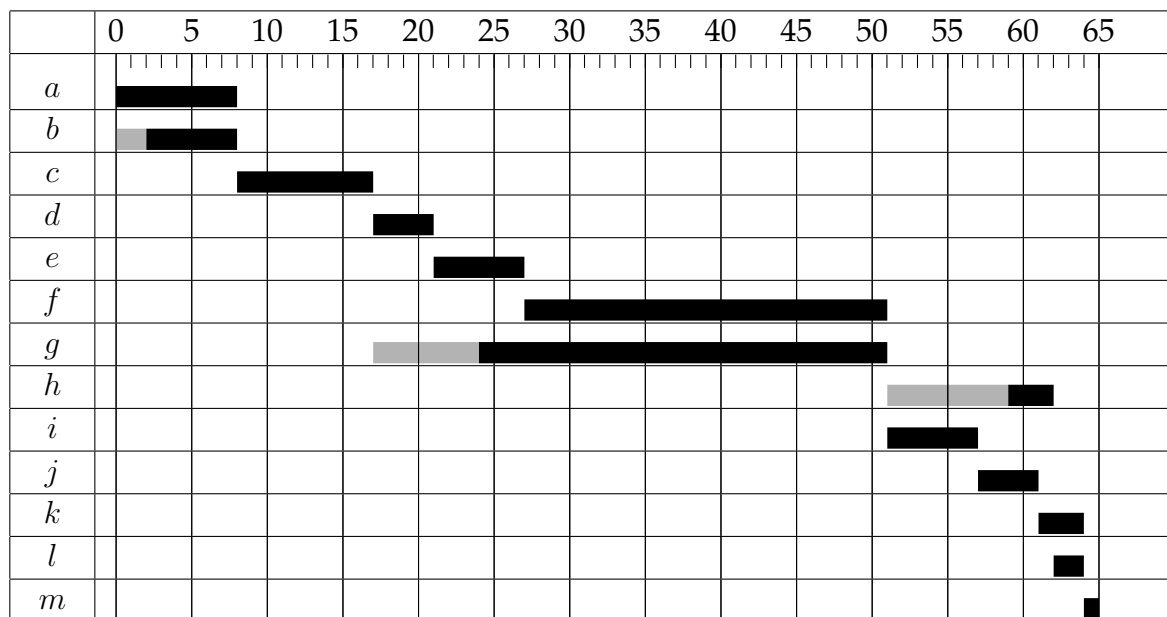
C'est un outil permettant de modéliser la planification de tâches nécessaires à la réalisation d'un projet. Il s'agit d'un outil inventé en 1917 par Henry Gantt.

Ce type de modélisation est particulièrement facile à mettre en œuvre avec un simple tableur mais il existe des outils spécialisés dont le plus connu est Microsoft Project. Il existe par ailleurs des équivalents libres (et gratuits) de ce type de logiciel.

5.1. Diagramme au plus tôt



5.2. Diagramme au plus tard



4

Problème du plus court chemin

1. Introduction

Un problème d'un réel intérêt, pour les applications, est celui qui consiste à chercher un chemin de longueur minimale (ou maximale) entre deux sommets donnés d'un graphe orienté ou non.

Etant donné un graphe $G = (X, U)$, on associe à chaque arc u un nombre $\ell(u)$ appelé « longueur de l'arc u ». Cette valeur $\ell(u)$ peut être, par exemple, un coût, une durée, une distance. La longueur d'un chemin μ quelconque, notée $\ell(\mu)$, est alors définie comme la somme des longueurs des arcs qui le composent :

$$\ell(\mu) = \sum_{u \in \mu} \ell(u)$$

Un chemin joignant un sommet x_a à un sommet x_b est dit de longueur minimale (ou maximale) s'il minimise (ou maximise) la longueur $\ell(\mu)$ dans l'ensemble de tous les chemins μ joignant x_a à x_b . Un tel chemin est appelé parfois le plus court (ou long) chemin de x_a à x_b .

Pour résoudre un tel problème, on peut toujours penser à la méthode énumérative qui consiste à établir la liste de tous les chemins possibles et calculer leurs longueurs respectives. Seulement, cette méthode est en général longue et difficile à mettre en œuvre, d'autant plus qu'elle est impossible pour des cas de taille considérable.

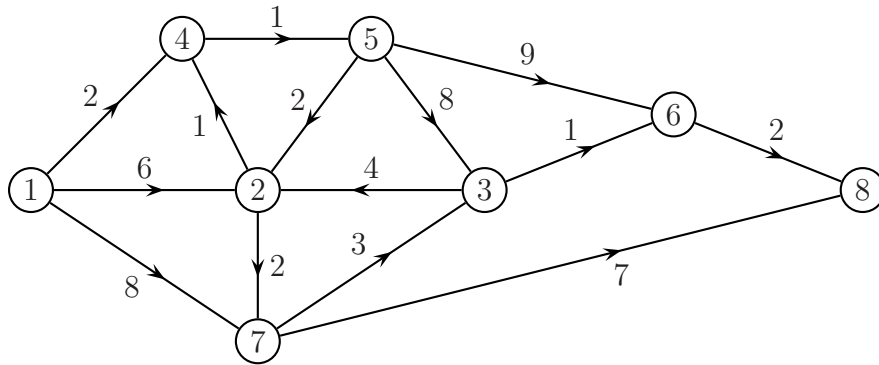
Nous présentons dans ce chapitre trois algorithmes permettant d'obtenir des chemins de longueur minimale ou maximale, selon le cas. Une renumérotation des sommets étant toujours possible, nous ne considérons que des chemins de x_1 à x_n .

Il est utile d'introduire une matrice L , dite matrice des longueurs, dont l'élément ℓ_{ij} se définit pour le problème du plus court chemin par :

$$\ell_{ij} = \begin{cases} \ell(u) & \text{si } u = (x_i, x_j) \in U \\ +\infty & \text{si } (x_i, x_j) \notin U \\ 0 & \text{si } i = j \end{cases}$$

2. Algorithme de Ford

$$L = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{matrix} & \left(\begin{array}{cccccccc} 0 & 6 & \infty & 2 & \infty & \infty & 8 & \infty \\ \infty & 0 & \infty & 1 & \infty & \infty & 2 & \infty \\ \infty & 4 & 0 & \infty & \infty & 1 & \infty & \infty \\ \infty & \infty & \infty & 0 & 1 & \infty & \infty & \infty \\ \infty & 2 & 8 & \infty & 0 & 9 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 & \infty & 2 \\ \infty & \infty & 3 & \infty & \infty & \infty & 0 & 7 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 \end{array} \right) \end{matrix}$$



3. Algorithme de Bellman

4. Algorithme de Dijkstra