



Université A.Mira de Béjaïa  
Faculté des Sciences Exactes

## Département de Recherche Opérationnelle

Mémoire Préparé

En Vue de l'Obtention du Diplôme de Master en Recherche Opérationnelle

*Spécialité* : Modélisation Mathématique et Evaluation des Performances des Réseaux

*Thème*

---

---

### Quelques Techniques de Planification des Tâches

---

---

Présenté par :

M<sup>lle</sup> Chalabi Samia

Devant le jury composé de :

Président : M<sup>r</sup> Talem Djamel  
Encadré par : M<sup>r</sup> Kabyl Kamal  
Examineur : M<sup>me</sup> Younsi Leila

Promotion 2019/2020

## *Remerciements*

*Louange A Dieu, le miséricordieux, sans Lui rien de tout cela  
n'aurait pu être*

Je tiens tout d'abord à remercier M<sup>r</sup> K. Kabyl, pour l'honneur qu'il m' a fait en acceptant de m'encadrer. Ses conseils précieux ont permis une bonne orientation dans la réalisation de ce modeste travail.

Je tiens également à remercier les membres du jury pour l'honneur qu'il m'a fait en acceptant de juger ce travail, et d'avoir consacrer leurs temps pour sa lecture. Je tiens à exprimer ma profonde gratitude à l'ensemble du corps

enseignant qui a contribué à ma formation. Enfin je tiens à rendre hommage à

toutes ma famille et mes amis pour le soutien qu'ils m'a apportè durant toutes ces années d'études .

A la mémoire de mon frère jumeau *Yacine*

# Table des matières

<b>Introduction Générale</b>	<b>1</b>
<b>1 Notions de bases et Notations</b>	<b>3</b>
1.1 Graphe . . . . .	3
1.1.1 Graphes orientés . . . . .	3
1.1.2 Graphes non orientés . . . . .	3
1.1.3 Graphe simple . . . . .	5
1.1.4 Graphe multiple . . . . .	5
1.1.5 Graphe partiel . . . . .	5
1.1.6 Sous-graphe . . . . .	5
1.1.7 Sous-graphe partiel . . . . .	5
1.1.8 Clique . . . . .	6
1.1.9 Stable . . . . .	6
1.2 Connexité dans les graphes . . . . .	7
1.2.1 Connexité . . . . .	7
1.2.2 Forte connexité . . . . .	8
1.2.3 Graphe réduit . . . . .	8
1.2.4 Chaînes et cycles . . . . .	8
1.2.5 Chemins et circuits . . . . .	9
1.3 Représentations non graphiques d'un graphe . . . . .	9
1.3.1 Matrices d'adjacences . . . . .	9
1.3.2 Matrice d'incidence d'un graphe . . . . .	11
1.4 Quelques graphes particuliers . . . . .	12
1.4.1 Graphe complet . . . . .	12
1.4.2 Graphe biparti . . . . .	12
1.4.3 Graphe biparti complet . . . . .	13
1.4.4 Graphe biparti équilibré . . . . .	13
1.4.5 Graphes planaires . . . . .	14
1.4.6 Graphes Pondérés . . . . .	14
1.4.7 Réseaux . . . . .	15

1.4.8	Arbre . . . . .	15
1.4.9	Graphe d'intervalle . . . . .	16
<b>2</b>	<b>Complexité Algorithmique</b>	<b>17</b>
2.1	Introduction . . . . .	17
2.2	Complexité d'un algorithme . . . . .	17
2.3	Problème de reconnaissance . . . . .	19
2.4	Problèmes faciles et problèmes difficiles . . . . .	19
2.4.1	La classe NP . . . . .	19
2.4.2	La classe P . . . . .	20
2.4.3	La classe NP-Complet . . . . .	20
2.4.4	La classe NP-Difficile . . . . .	20
<b>3</b>	<b>Gestion et Planification de projets</b>	<b>22</b>
3.1	Quelques définitions . . . . .	22
3.2	Gestion de projet et planification . . . . .	25
3.2.1	Le cycle de vie d'un projet . . . . .	25
3.2.2	Gestion de projet . . . . .	25
3.2.3	Les étapes de la gestion de projet . . . . .	26
3.2.4	Les facteurs critiques de succès en gestion de projet . . . . .	26
3.2.5	Planification de projet . . . . .	27
3.3	Problème d'ordonnancement . . . . .	27
3.3.1	Les contraintes potentielles . . . . .	28
3.3.2	Les contraintes disjonctives . . . . .	28
3.3.3	Les contraintes cumulatives . . . . .	28
3.4	Modélisation d'un problème d'ordonnancement via les graphes . . . . .	29
3.5	Les outils de la planification de projet . . . . .	30
3.5.1	Structure de répartition de travail WBS . . . . .	30
3.5.2	Structure Organisationnelle (OBS) . . . . .	30
3.5.3	Matrice RACI . . . . .	30
3.5.4	Méthode P.E.R.T . . . . .	31
3.5.5	<b>Elaboration et exploitation de réseau PERT</b> . . . . .	32
3.5.6	Règles de construction d'un réseau P.E.R.T : . . . . .	35
3.5.7	Diagramme de Gantt . . . . .	36
3.5.8	Méthode du Potentiel . . . . .	37
<b>4</b>	<b>Résolution de quelques problèmes concrets</b>	<b>39</b>
4.1	Résolution d'un problème d'ordonnancement : . . . . .	39
4.1.1	Problème 1 . . . . .	39
4.1.2	Dates de début au plus tôt : . . . . .	41
4.1.3	Dates de début au plus tard . . . . .	42

4.1.4	Marges totales et chemin critique . . . . .	43
4.2	Problème 2 . . . . .	44
4.2.1	Construction d'un réseau PERT . . . . .	44
4.2.2	Etablir une liste précise des tâches . . . . .	44
4.2.3	Déterminer les tâches antérieures . . . . .	45
4.2.4	Exploitation du réseau PERT . . . . .	47
4.2.5	Calcul des marges . . . . .	50
4.2.6	Détermination du chemin critique . . . . .	51
4.3	Exemple 3 . . . . .	52
4.4	Calcul des dates et des marges . . . . .	53
<b>5</b>	<b>Application sous C++</b>	<b>56</b>
5.1	Présentation de c++ . . . . .	56
5.1.1	Planifier un projet par les activités . . . . .	56
5.1.2	Application sous C++ . . . . .	57
	<b>Conclusion Générale</b>	<b>60</b>
	<b>Bibliographie</b>	<b>i</b>
	<b>Résumé</b>	<b>ii</b>

# Table des figures

1.1	Un exemple d'un graphe orienté . . . . .	4
1.2	Graphe à 5 sommets et 7 arêtes . . . . .	5
1.3	Graphe partiel et sous-graphe d'un graphe G . . . . .	6
1.4	Clique . . . . .	6
1.5	Un exemple d'un stable . . . . .	7
1.6	Graphe connexe à 4 sommets et 3 arêtes . . . . .	7
1.7	Deux composantes fortement connexes . . . . .	8
1.8	Graphe réduit à deux composantes fortement connexes . . . . .	8
1.9	Chaîne dans un graphe . . . . .	9
1.10	Un graphe G et sa matrice d'adjacence . . . . .	11
1.11	Exemple d'une matrice d'incidence . . . . .	11
1.12	Exemple de graphe complet $k_5$ . . . . .	12
1.13	Graphe biparti . . . . .	12
1.14	Le graphe biparti complet $k_{2,3}$ . . . . .	13
1.15	Un graphe biparti équilibré $K_{3,3}$ . . . . .	14
1.16	Exemple de graphe planaire . . . . .	14
1.17	Graphe Pondéré . . . . .	15
1.18	Réseau . . . . .	15
2.1	Exemple de 3 algorithmes avec leurs complexités. . . . .	18
2.2	Tableau récapitulatif de classification des problèmes combina- toires . . . . .	21
3.1	Triangle de la triple contrainte . . . . .	24
3.2	Cycle de vie d'un projet . . . . .	25
3.3	Etapes de la gestion de projet . . . . .	26
3.4	Exemple de contraintes cumulatives . . . . .	28
3.5	Représentation AoN . . . . .	29
3.6	représentation AoA . . . . .	30
3.7	Schéma WBS . . . . .	31
3.8	Tableau d'ordonnancement . . . . .	36
3.9	Représentation graphique de la méthode PERT . . . . .	36

3.10	Diagramme de Gantt . . . . .	37
3.11	Tableau d'ordonnancement . . . . .	38
3.12	Représentation graphique de la méthode du potentiel . . . . .	38
3.13	Schéma planification de projet . . . . .	38
4.1	Tableau d'ordonnancement . . . . .	40
4.2	Diagramme de Gantt . . . . .	40
4.3	Réseau Pert . . . . .	41
4.4	Les dates de début au plus tôt . . . . .	42
4.5	Les dates de début au plus tard . . . . .	43
4.6	Marges totales et chemin critique . . . . .	44
4.7	Tableau d'ordonnancement . . . . .	45
4.8	Tableau d'ordonnancement . . . . .	45
4.9	Tableau d'ordonnancement . . . . .	46
4.10	réseau PERT . . . . .	46
4.11	calcul de la date au plus tôt dans le cas d'une seule tâche . . . . .	47
4.12	calcul de la date au plus tôt dans le cas de plusieurs tâches . . . . .	48
4.13	calcul des dates au plus tôt . . . . .	48
4.14	calcul de la date au plus tard dans le cas d'une seule tâche . . . . .	49
4.15	calcul des dates au plus tard dans le cas des plusieurs tâches . . . . .	50
4.16	Réseau PERT avec les dates au plus tôt et au plus tard . . . . .	50
4.17	calcul de la marge relative à une tâche . . . . .	51
4.18	détermination du chemin critique . . . . .	52
4.19	Un exemple du graphe MPM . . . . .	53
4.20	Un exemple du graphe PERT . . . . .	53
4.21	Solution avec la méthode MPM . . . . .	54
4.22	Solution avec la méthode PERT . . . . .	54
5.1	Tableau des tâches . . . . .	57
5.2	Les successeurs et la durée de chaque tâche . . . . .	57
5.3	Les successeues et la durée de chaque tâche . . . . .	58
5.4	Les dates de debut au plus tôt . . . . .	58
5.5	Les dates de debut au plutard . . . . .	58
5.6	Calcul les marges et le chemin critique . . . . .	59

# Introduction Générale

La recherche opérationnelle peut être définie comme l'ensemble des méthodes et techniques d'analyse des problèmes de nature combinatoire liés à des phénomènes observés pour élaborer de meilleurs politiques de décisions menant à leur résolution, elle comprend plusieurs branches comme la programmation dynamique, la théorie des graphes, la programmation non linéaire, l'optimisation combinatoire, etc.

Dans le cadre de la préparation de ce mémoire de fin d'étude on'a choisi d'exposer quelques problèmes de la vie professionnelle, dont la résolution fait appel aux techniques de la recherche opérationnelle. Un projet est un ensemble d'activités se succédant et ayant pour but la réalisation d'un objectif unique, bien déterminé. Avant sa concrétisation, un projet passe par plusieurs phases appelées "Cycle de vie" du projet : Conception, Planification et contrôle. Le concept de gestion de projets englobe des techniques et méthodes organisationnelles, des outils de planification et de suivi de projet (contrôle). Le principal objectif est d'optimiser des paramètres propres au projet c'est-à-dire : Durées, coûts et qualités. La planification, représente une étape très importante lors de l'élaboration d'un projet, c'est durant cette étape que sont estimées les durées des activités ainsi que le nombre de ressources nécessaires pour leur déroulement. L'ordonnancement intervient dans le séquençement des activités du projet, il fixe l'ordre dans lequel elles doivent être réalisées et calcule les besoins en ressources à chaque étape de l'ordonnancement. C'est dans cette étape également qu'intervient le calcul des dates au plus tôt et au plus tard des tâches.

La phase de contrôle a aussi son importance dans le suivi du projet, elle permet de situer, de savoir où en est à la réalisation par rapport à ce qui a été préalablement planifié et faire le point sur l'état d'avancement du projet. Ce travail est ainsi organisé :

- Dans le premier chapitre on présentera quelques notions sur la théorie des graphes.
- Dans le deuxième chapitre est consacré à la complexité algorithmique.
- Dans le troisième chapitre, on a expliqué brièvement l'ordonnancement de

projet, outils et méthodes .

- Le quatrième chapitre on a fait appel à la modélisation graphique dont le but de minimiser la durée d'un projet grâce à la méthode PERT .
- Le cinquième chapitre est consacré à programmer la méthode Pert en utilisant le langage c.

# Chapitre 1

## Notions de bases et Notations

Dans ce chapitre nous allons présenter certaines notions de base de la théorie des graphes que nous allons utiliser par la suite dans notre travail.

### 1.1 Graphe

**Définition 1.1.1.** *On appelle graphe  $G = (V, E)$  la donnée d'un ensemble  $X$  dont les éléments sont appelés sommets et d'une partie de  $E$  symétrique  $(v, e) \in A \iff (e, v) \in E$  dont les éléments sont appelés arêtes [1].*

#### 1.1.1 Graphes orientés

En donnant un sens aux arêtes d'un graphe, on obtient un digraphe (ou graphe orienté). Le mot « digraphe » est la contraction de l'expression anglaise « directed graph ». Un digraphe fini  $G = (V, E)$  est défini par l'ensemble fini  $\{ V = v_1, v_2, \dots, v_n \}$  dont les éléments sont appelés sommets, et par l'ensemble fini  $\{ E = e_1, e_2, \dots, e_m \}$  dont les éléments sont appelés arcs. Un arc  $e$  de l'ensemble  $E$  est défini par une paire ordonnée de sommets. Lorsque  $e = (u, v)$ , on dit que l'arc  $e$  va de  $u$  vers  $v$ . On dit aussi que  $u$  est l'extrémité initiale (prédécesseur de  $v$ ,  $\Gamma^-(v)$ ) et  $v$  l'extrémité finale de  $e$  (successeur de  $u$ ,  $\Gamma^+(u)$ ). Soit  $v$  un sommet d'un graphe orienté. On note  $d^+(v)$  le degré extérieur du sommet  $v$ , c'est-à-dire le nombre d'arcs ayant  $v$  comme extrémité initiale. On note  $d^-(v)$  le degré inférieur du sommet  $v$ , c'est-à-dire le nombre d'arcs ayant  $v$  comme extrémité finale. On définit le degré :

$$\boxed{d(v) = d^+(v) + d^-(v)}$$

La figure 1.1 montre un graphe orienté à 6 sommets et 8 arcs .

#### 1.1.2 Graphes non orientés

Un graphe  $G = (V, E)$  est défini par l'ensemble fini  $V = \{v_1, v_2, \dots, v_n\}$  dont les éléments sont appelés sommets (Vertices en anglais), et par l'ensemble fini  $E = \{e_1, e_2, \dots, e_m\}$  dont les éléments sont appelés arêtes (Edges en anglais).

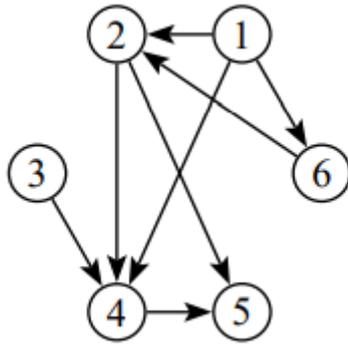


FIGURE 1.1 – Un exemple d'un graphe orienté

Une arête  $e$  de l'ensemble  $E$  est définie par une paire non ordonnée de sommets, appelés les extrémités de  $e$ . Si l'arête  $e$  relie les sommets  $a$  et  $b$ , on dira que ces sommets sont adjacents, ou incidents avec  $e$ , ou bien que l'arête  $e$  est incidente avec les sommets  $a$  et  $b$ .

On appelle ordre d'un graphe le nombre de sommets  $n$  de ce graphe.

On appelle degré du sommet  $v$ , et on note  $d(v)$ , le nombre d'arêtes incidentes à ce sommet.

Le degré d'un graphe est le degré maximum de tous ses sommets. [2].

Un graphe non orienté à 5 sommets et 7 arêtes est illustré dans la figure 1.2

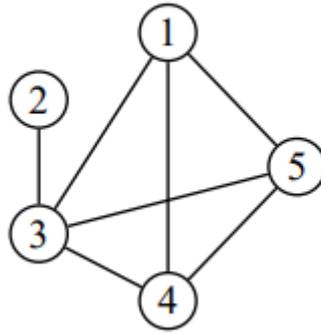


FIGURE 1.2 – Graphe à 5 sommets et 7 arêtes

### 1.1.3 Graphe simple

Un graphe est dit simple s'il n'admet aucune arête multiple et aucune boucle.

### 1.1.4 Graphe multiple

Un graphe est dit multiple s'il admet des arêtes multiples et des boucles. Le graphe complémentaire de  $G$ , noté  $\bar{G}$  est le graphe dont l'ensemble de sommets est  $V(G)$  et deux sommets distincts de  $\bar{G}$  sont adjacents si et seulement s'ils ne sont pas adjacents dans  $G$ .

### 1.1.5 Graphe partiel

Soit  $G = (V, E)$  un graphe. Le graphe  $G' = (V, E')$  est un graphe partiel de  $G$ , si  $E'$  est inclus dans  $E$ . Autrement dit, on obtient  $G'$  en enlevant une ou plusieurs arêtes au graphe  $G$ .

### 1.1.6 Sous-graphe

Pour un sous-ensemble de sommets  $A$  inclus dans  $V$ , le sous-graphe de  $G$  induit par  $A$  est le graphe  $G_A = (A, E(A))$  dont l'ensemble des sommets est  $A$  et l'ensemble des arêtes  $E(A)$  est formé de toutes les arêtes de  $G$  ayant leurs deux extrémités dans  $A$ . Autrement dit, on obtient  $G_A$  en enlevant un ou plusieurs sommets au graphe  $G$ , ainsi que toutes les arêtes incidentes à ces sommets.

Le graphe de la figure 1.3 (b) est un graphe partiel de graphe de la figure 1.3 (a) et le graphe de la figure 1.3 (c) est un sous graphe de graphe de la figure 1.3 (a).

### 1.1.7 Sous-graphe partiel

Un sous-graphe partiel est un graphe partiel d'un sous graphe.

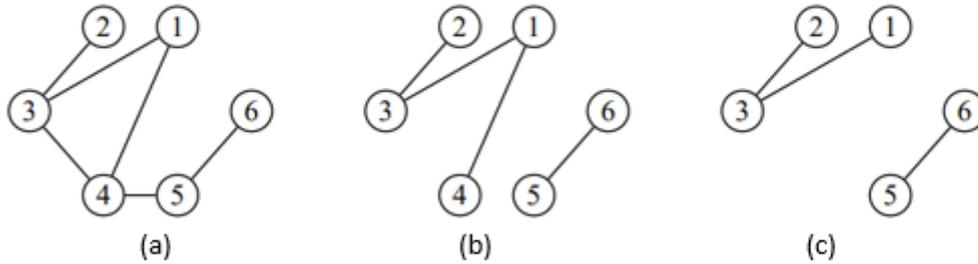


FIGURE 1.3 – Graphe partiel et sous-graphe d'un graphe G

### 1.1.8 Clique

Étant donné un graphe non orienté G, une clique est un sous-ensemble de sommets qui sont tous connectés deux à deux par des arêtes. En d'autre terme, il s'agit d'un sous-graphe complet.

Le graphe de la figure 1.4 représente une clique  $\{x_1, x_2, x_3\}$ [3].

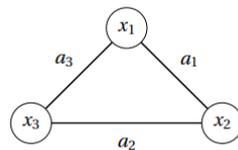


FIGURE 1.4 – Clique

### 1.1.9 Stable

Considérons un graphe G non orienté . Un ensemble stable de G est un ensemble de sommets non adjacents, qui forme d un sous-graphe  $G'$  sans arêtes. Un ensemble stable est donc un sous-ensemble de sommets du graphe G tel que tout couple de sommets de ce sous-ensemble ne soit pas adjacents (voisins). Pour chaque couple de sommet de ce sous-ensemble, il ne doit donc pas exister d'arêtes les reliant.

L'ensemble des sommets  $\{a,b,e,g\}$  de graphe de la figure 1.5 forme un ensemble stable.

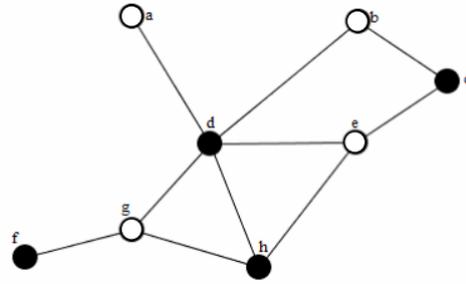


FIGURE 1.5 – Un exemple d'un stable

## 1.2 Connexité dans les graphes

### 1.2.1 Connexité

Un graphe  $G = (V(G), E(G))$  est dit connexe si pour toute paire de sommets  $(u, v)$  de  $V(G) \times V(G)$ , il existe une chaîne reliant  $u$  et  $v$ . Si  $G$  n'est pas connexe, alors il va avoir des sous-graphes connexes, ces derniers sont appelés composantes connexes de  $G$ .

La figure 1.6 présente un graphe connexe à 4 sommets et 3 arêtes.

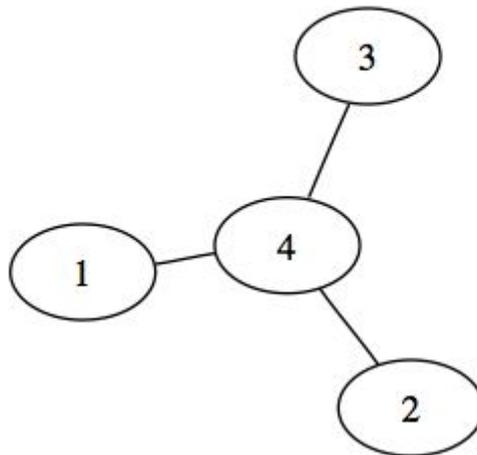


FIGURE 1.6 – Graphe connexe à 4 sommets et 3 arêtes

### 1.2.2 Forte connexité

Un graphe  $G$  est dit fortement connexe si  $\forall(x, y) \in X^2$  il existe un chemin de  $x$  à  $y$  et un autre chemin de  $y$  à  $x$ .

Le graphe de la figure 1.7 contient deux composantes fortement connexes.

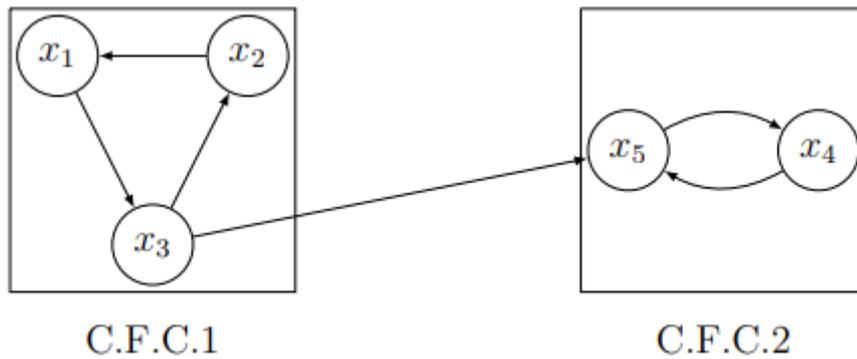


FIGURE 1.7 – Deux composantes fortement connexes

### 1.2.3 Graphe réduit

On appelle le graphe réduit du graphe  $G = (X, E)$  le graphe  $G_r = (X_r, E_r)$  tel que : Les sommets de  $G_r$  sont les composantes fortement connexes de  $G$ . S'il existe un arc  $(x, y)$  dans le graphe  $G$  avec :  $x \in c.f.c_i$  et  $y \in c.f.c_j$  alors il existera un arc de  $c.f.c_i$  à  $c.f.c_j$  dans le graphe  $G_r, i, j \in (1 \dots n)$  le graphe de la figure 1.8 représente le graphe réduit de l'exemple précédent.



FIGURE 1.8 – Graphe réduit à deux composantes fortement connexes

### 1.2.4 Chaînes et cycles

Une chaîne dans  $G$ , est une suite ayant pour éléments alternativement des sommets et des arêtes, commençant et se terminant par un sommet, et telle que chaque arête est encadrée par ses extrémités.

On dira que la chaîne relie le premier sommet de la suite au dernier sommet. En plus, on dira que la chaîne a pour longueur le nombre d'arêtes de la chaîne. Le graphe de la figure 1.9 contient entre autres les chaînes  $(v_1, e_1, v_2, e_2, v_3, e_5, v_5)$  et  $(v_4, e_4, v_3, e_2, v_2, e_1, v_1)$ .

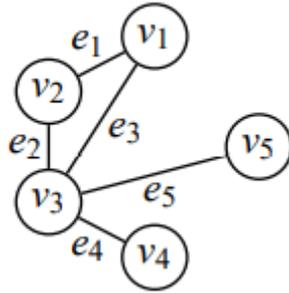


FIGURE 1.9 – Chaîne dans un graphe

Une chane simple dont les extrimités initiale et finale sont confondues est appelée cycle.

### 1.2.5 Chemins et circuits

Un chemin conduisant du sommet a au sommet b est une suite ayant pour éléments alternativement des sommets et des arcs, commençant et se terminant par un sommet, et telle que chaque arc est encadré à gauche par son sommet origine et à droite par son sommet destination. On ne peut donc pas prendre les arc à rebours. Sur le digraphe de la figure 1.9 , L'ensemble des sommets et arcs  $\{v_3, e_2, v_2, e_1, v_1\}$  est un chemin .

Un circuit est un chemin dont les sommets de départ et de fin sont les mêmes. Le digraphe de la figure 1.10 ne contient pas un chemin [2]..

## 1.3 Représentations non graphiques d'un graphe

### 1.3.1 Matrices d'adjacences

On peut représenter un graphe simple par une matrice d'adjacence. Une matrice  $(n \times m)$  est un tableau de n lignes et m colonnes.  $(i,j)$  désigne l'intersection de la ligne i et de la colonne j. Dans une matrice d'adjacence, les lignes et les colonnes représentent les sommets du graphe. Un « 1 » à la position  $(i,j)$  signifie que le sommet i est adjacent au sommet j.

Cette matrice a plusieurs caractéristiques :

1. Elle est carrée : il y a autant de lignes que de colonnes.
2. Il n'y a que des zéros sur la diagonale allant du coin supérieur gauche au coin inférieur droit. Un « 1 » sur la diagonale indiquerait une boucle.
3. Elle est symétrique :  $m_{ij} = m_{ji}$  . On peut dire que la diagonale est un axe de symétrie.
4. Une fois que l'on fixe l'ordre des sommets, il existe une matrice d'adjacence unique pour chaque graphe. Celle-ci n'est la matrice d'adjacence d'aucun autre graphe .

La matrice d'adjacence associée au graphe de la figure 1.10 (a) est illustré dans la figure 1.10 (b).

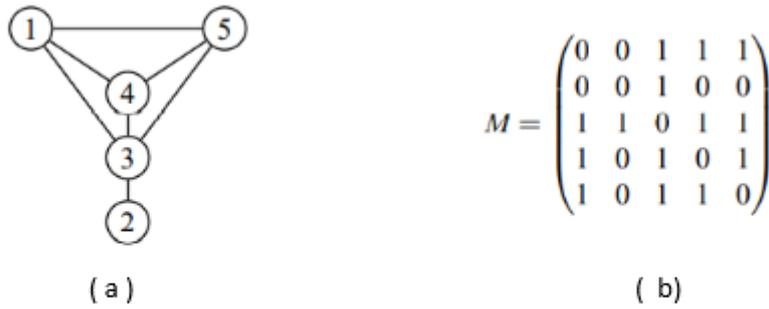


FIGURE 1.10 – Un graphe G et sa matrice d'adjacence

### 1.3.2 Matrice d'incidence d'un graphe

Soit G un graphe orienté qui possède n sommets numérotés de 1 à n, et m arcs numérotés de 1 à m. On appelle matrice d'incidence du graphe la matrice  $A = (a_{i,j})$  comportant n lignes et m colonnes telle que :

$$a_{i,j} = \begin{cases} +1 & \text{si } \Gamma^-(j) = i; \\ -1 & \text{si } \Gamma^+(j) = i; \\ 0 & \text{sinon.} \end{cases}$$

La matrice d'incidence associée au graphe de la figure 1.11 (a) est donnée par la figure 1.11 (b) .

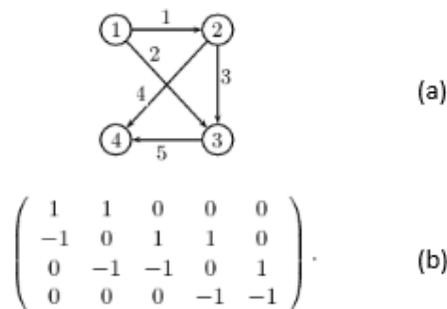


FIGURE 1.11 – Exemple d'une matrice d'incidence

## 1.4 Quelques graphes particuliers

### 1.4.1 Graphe complet

Un graphe  $G$  est dit complet si chaque sommet de  $G$  est relié directement à tous les autres sommets. Un graphe complet à  $n$  sommets est noté par  $k_n$ . Le graphe complet  $k_5$  est illustré dans la figure 1.12 .

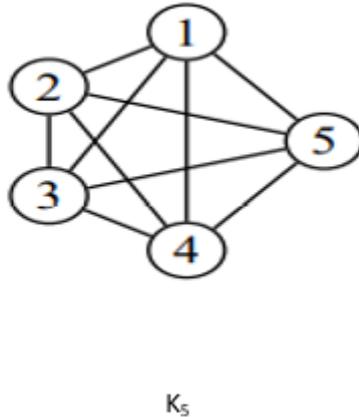


FIGURE 1.12 – Exemple de graphe complet  $k_5$

### 1.4.2 Graphe biparti

Un graphe  $G$  est dit biparti si ses sommets peuvent être partitionnés en deux ensembles  $X$  et  $Y$  , de sorte que toutes les arêtes de graphe relient un sommet dans  $X$  à un sommet dans  $Y$ , et que les sommets de même ensemble ne sont pas reliés entre eux.

Un graphe biparti est noté par  $G = (X \cup Y, E)$  . La figure 1.13 montre un graphe biparti à 5 sommets et 5 arêtes .

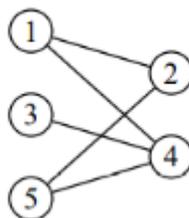


FIGURE 1.13 – Graphe biparti

### 1.4.3 Graphe biparti complet

Un graphe  $G$  est dit biparti complet (ou encore est appelé une biclique) si chaque sommet de  $X$  est relié à chaque sommet de  $Y$ .

Si  $|X| = p$  et  $Y = q$ , alors le graphe biparti complet est noté par  $k_{p,q}$ . La figure 1.14 montre un graphe biparti complet  $K_{2,3}$ .

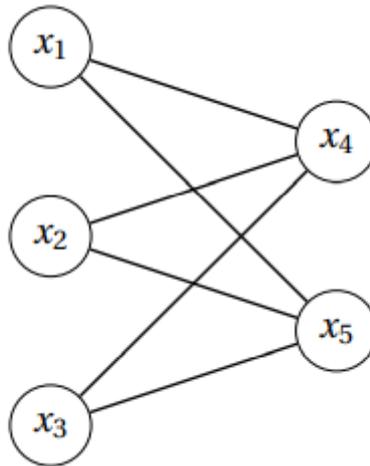


FIGURE 1.14 – Le graphe biparti complet  $k_{2,3}$

### 1.4.4 Graphe biparti équilibré

Un graphe biparti  $G$  est dit équilibré si chaque ensemble de la bipartition est de même taille. Les graphes ayant des cycles de longueur paire, c'est-à-dire les graphes bipartis complets, pour lesquels  $|X1| = |X2|$ , sont des ensembles de graphes équilibrés. La figure 1.15 donne un graphe biparti équilibré  $K_{3,3}$ .

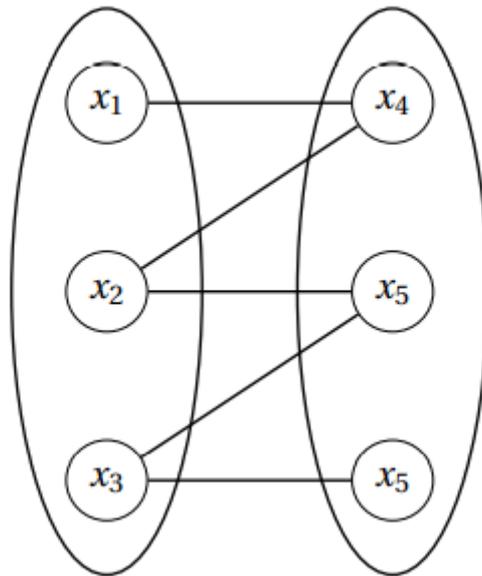


FIGURE 1.15 – Un graphe biparti équilibré  $K_{3,3}$

#### 1.4.5 Graphes planaires

On dit qu'un graphe  $G$  est planaire si on peut le dessiner dans le plan de sorte que ses arêtes ne se croisent pas en dehors de leur extrémités. La figure 1.16 montre un graphe planaire [3]. .

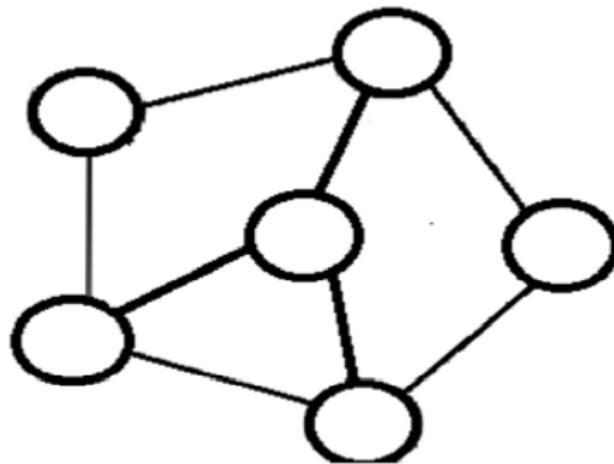


FIGURE 1.16 – Exemple de graphe planaire

#### 1.4.6 Graphes Pondérés

On appelle graphe pondéré, un graphe dont les arêtes sont affectées d'un nombre positif appelé poids. La figure 1.17 présente un graphe pondéré .

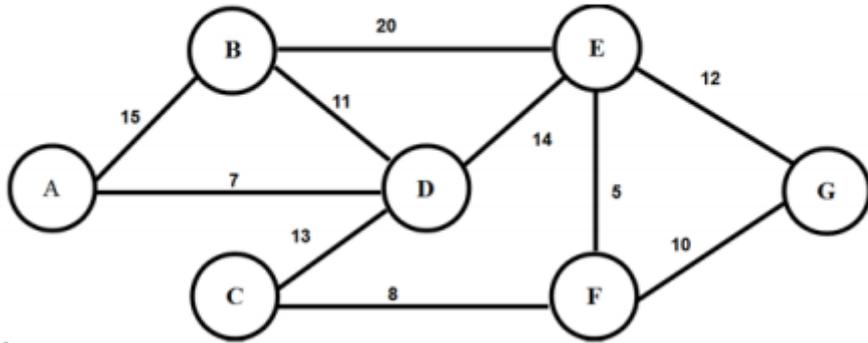


FIGURE 1.17 – Graphe Pondéré

### 1.4.7 Réseaux

Un réseau est un graphe connexe sans boucles telle que pour chaque arc de  $G$  on lui valué par un nombre positif  $c(u)$  appelé capacité de l'arc  $u$ . Ce réseau comporte un sommet sans prédécesseurs appelé (entrée du réseau) ou la ( source ) et un autre sommet sans successeur appelé la (sortie du réseau) ou ( le puits). On note un réseau par  $R = (V, A, C)$  . La figure 1.18 montre un réseau .

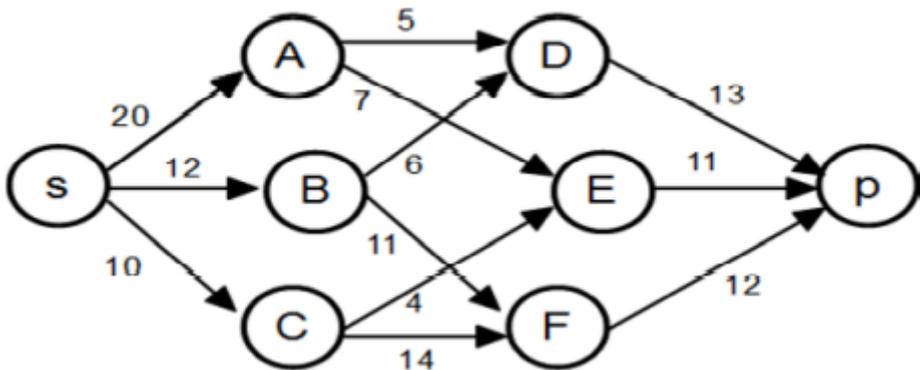


FIGURE 1.18 – Réseau

### 1.4.8 Arbre

On dit qu'un graphe  $G$  est un arbre si et seulement s'il est connexe et n'admet pas un cycle. Etant donné un graphe non orienté  $G = (X,A)$  d'ordre  $n$ , les affirmations suivantes sont équivalentes :

1.  $G$  est un arbre.
2.  $G$  est sans cycles et connexe.

3.  $G$  est sans cycles et comporte  $n - 1$  arêtes.
4.  $G$  est connexe et comporte  $n - 1$  arêtes.
5.  $G$  est connexe, et la suppression d'une arête le déconnecte.

#### 1.4.9 Graphe d'intervalle

On construit un graphe  $G$  à partir des intervalles de la droite réelle  $I_1, \dots, I_n$ , où les sommets de  $G$  sont numérotés de 1 à  $n$ . Dans un graphe d'intervalle, il existe une arête entre les sommets  $i$  et  $j$ ,  $i \neq j$ , si et seulement si  $I_i \cap I_j \neq \emptyset$ . Autrement dit, deux sommets sont reliés si et seulement si les deux intervalles correspondants se chevauchent [1].

## Chapitre 2

# Complexité Algorithmique

### 2.1 Introduction

A première vue, comment définir un algorithme efficace? Pour un problème donné, chercher un algorithme efficace, veut dire trouver un algorithme où le temps nécessaire à son exécution ne soit pas trop important.

Un problème est dit facile si on peut le résoudre facilement, c'est-à-dire s'il ne fait pas trop de temps pour arriver à la solution. Donc, s'il existe un algorithme efficace pour un problème donné, alors ce dernier est dit facile.

Un problème pour lequel on ne connaît pas d'algorithme efficace, est ce qu'il est facile ou difficile?

De nombreux chercheurs se sont penchés sur ce genre de problèmes et ils ont développé une théorie dite de la complexité. Nous n'allons pas détailler cette théorie, mais nous allons, quand même donner une idée globale du sujet en question.

### 2.2 Complexité d'un algorithme

La définition d'un modèle de calcul a pour but de fournir un mode d'expression à une méthode spécifique de résolution des problèmes.

**Définition 2.2.1.** *La complexité d'un algorithme  $A$  est une fonction  $C(A_N)$ , donnant le nombre d'instructions caractéristiques exécutées par  $A$  dans le pire des cas, pour une donnée de taille  $N$  [7].*

Certains pourraient demander ce qu'est précisément une opération. En général, on considère comme étant une opération élémentaire une affectation, une addition, un test... Mais cela est discutable puisque selon le langage et le compilateur, une opération sera exécutée plus ou moins vite, une addition s'exécutera plus ou moins vite qu'un test... Le nombre d'opérations varie selon les données qu'on a en entrée.

Prenons l'exemple d'un algorithme de tri d'un tableau à une dimension. Il est bien clair que le temps nécessaire pour exécuter cet algorithme avec 5 éléments

n'est pas le même avec 500 éléments.

présente un autre facteur déterminant dans le temps d'exécution d'un algorithme est la taille des données en entrée. Un tableau à deux dimensions prend plus de temps qu'un tableau à une seule dimension.

Enfin, un dernier facteur considéré dans la complexité est celui de la situation de la structure de données. Par exemple, l'exécution d'un tableau trié au départ n'est pas la même que celui du même tableau désordonné. Donc, la meilleure solution pour ce cas est de prendre le nombre d'opérations dans le pire des cas. Le critère de rapidité n'est pas toujours celui qui nous intéresse. On peut aussi vouloir estimer la place utilisée par un algorithme dans la mémoire de l'ordinateur. Dans ce cas, on parle de complexité spatiale .

### Algorithme efficace

La complexité d'un problème est la complexité minimale dans le pire des cas d'un algorithme qui le résout . est notée par  $O$  . . Ainsi, si le nombre d'itérations nécessaires pour obtenir une solution optimale est décrit par  $4n^3 - 6n^2 + 5n - 1$ , l'algorithme est dit en  $O(n^3)$  , ce qui caractérise une complexité polynomiale en la taille du problème. Par contre si la complexité est de l'ordre  $O(2^n)$  on parle de complexité exponentielle. Un algorithme sera dit efficace si sa complexité est bornée par un polynôme ayant la taille des données comme variable.

Considérons les algorithmes A, B et C. Leurs complexités ( $C$ ) sont donnés dans la figure 2.1 .

	Complexité	N 'ombre d'opération pour N=4	N'ombre d'opération pour n=20
A	$70*n$	280	1400
B	$9*n^2$	144	3600
C	$n!$	24	$2.4*10^{18}$

FIGURE 2.1 – Exemple de 3 algorithmes avec leurs complexités.

On remarque que le plus efficace pour 4 éléments est donc C, mais tout de suite, il ne devient plus utilisable. Par contre, A et B restent applicables.

Maintenant, avec 100 éléments, il faut respectivement 8000 et 100000 opérations . Bien évidemment, l'algorithme A est le plus performant, mais l'algorithme B reste applicable. Le nombre d'opérations de C devient presque inimaginable.

Cet exemple justifie la notion d'efficacité. Si le nombre d'opérations n'explose pas avec une augmentation de la taille des données, l'algorithme est considéré efficace.

Cependant, il faut bien comprendre que l'on s'intéresse au comportement général de l'algorithme face à des problèmes de grande taille.

Ainsi, ce n'est pas utile de compter toutes les opérations dans le détail, ni de

considérer le langage de programmation. Dans notre exemple, les coefficients 70 et 9 ne sont pas très importants, dès que la taille augmente, on s'aperçoit que c'est le terme en  $n$  qui prime. Ceci explique la difficulté à déterminer la complexité d'un algorithme. Il faut être très précis dans la démarche mais ne pas se soucier de la valeur exacte en terme de temps d'exécution de chaque opération.

## 2.3 Problème de reconnaissance

Un problème d'optimisation combinatoire (POC) consiste à chercher le minimum  $s^*$  d'une fonction application  $f$ , le plus souvent à valeurs entières, sur un ensemble fini  $S$  :

$$F(s^*) = \min_{s \in S} f(s)$$

Un problème d'existence (PE) consiste à chercher dans un ensemble fini  $S$  s'il existe un élément  $s$  vérifiant une certaine propriété  $P$ .

**Définition 2.3.1.** *Un problème de reconnaissance ou problème de décision est un problème qui consiste à apporter une réponse "oui" ou "non" à une question.*[8]

A chaque problème d'optimisation combinatoire, on peut associer un problème de reconnaissance.

Un problème d'optimisation combinatoire est au moins aussi difficile que le problème de reconnaissance associé. En d'autres termes, cela signifie qu'un problème d'optimisation combinatoire est souvent du même niveau de difficulté que le problème de reconnaissance associé. Cela justifie que la suite de ce chapitre ne concerne que les problèmes de reconnaissance.

## 2.4 Problèmes faciles et problèmes difficiles

Les problèmes indécidables sont ceux pour lesquels aucun algorithme, quel qu'il soit n'a été trouvé pour les résoudre.

A l'opposé, les problèmes décidables sont ceux pour lesquels il existe au moins un algorithme pour les résoudre .

### 2.4.1 La classe NP

La classe NP est celle des problèmes d'existence dont une proposition de solution est Oui et qui est vérifiable polynomialement. Parmi les problèmes décidables, les plus simples à résoudre sont regroupés dans la classe NP.[7]

La classe NP est également décomposée en trois catégories qui permettent d'identifier les problèmes les plus simples et les problèmes les plus compliqués de la classe.

#### **2.4.2 La classe P**

Un problème est dit polynomial s'il existe un algorithme de complexité polynomiale permettant de répondre à la question posée dans ce problème, quelle que soit la donnée de celui-ci. La classe P est l'ensemble de tous les problèmes de reconnaissance polynomiaux . 8

Pour le reste de la classe NP, on n'est pas sûr qu'il n'existe pas un algorithme polynomial pour résoudre chacun de ses problèmes. Ainsi, on sait que P est incluse dans NP mais on n'a pas pu prouver que P n'est pas NP.

#### **2.4.3 La classe NP-Complet**

La classe NP-Complet regroupe les problèmes les plus difficiles de la classe NP. Elle contient les problèmes de la classe NP tels que n'importe quel problème de la classe NP est polynomialement réductible. Entre eux, les problèmes de la classe NP-Complet sont aussi difficiles.

#### **2.4.4 La classe NP-Difficile**

La classe NP-Difficile regroupe les problèmes (pas forcément dans la classe NP) tels que n'importe quel problème de la classe NP leur est polynomialement réductible. La figure 2.2 représente un tableau récapitulatif de classification des problèmes combinatoires.

<b>Décidables</b>	<b>Classe NP</b>	<b>Classe P</b> (plus court chemin, arbre de poids min, flot maximum, flot de coût minimum,...)	<b>Classe NP-Difficile</b>
		<b>Classe NP-Complet</b> (Voyageur de commerce, Sac à dos,...)	
<b>Indécidables</b>			

FIGURE 2.2 – Tableau récapitulatif de classification des problèmes combinatoires

## Chapitre 3

# Gestion et Planification de projets

### 3.1 Quelques définitions

#### Projet

Le mot projet provient de terme latin "projectum" de "projicere" (jeter quelque chose vers l'avant). Un projet est un ensemble finalisé d'activités et d'actions entreprises dans le but de répondre à un besoin défini dans des délais fixés et dans la limite d'une enveloppe budgétaire allouée .

#### Chef de projet

Le chef de projet ou chargé de projet est la personne chargée de mener un projet et de gérer son bon déroulement.

#### Tâche

Une tâche est une entité élémentaire localisée dans le temps par une date de début et/ou de fin, dont la réalisation nécessite une durée, et qui consomme un moyen selon une certaine intensité. 11.

- $S_i$  : date de début.
- $C_i$  : date de fin.
- $p_i$  : durée d'exécution.

#### Jalon

Un jalon (milestone) est un événement particulier qui marque le début ou la fin d'une partie bien identifiée du projet. Il est en général associé à une date précise. C'est un repère prédéterminé et significatif dans le cours du projet.

#### Le Planning

Le planning correspond aux dates pour réaliser les activités, identifier les jalons et atteindre les objectifs du projet. C'est l'indispensable outil de la planification.

## Ressource

La réalisation de chaque tâche identifiée dans le projet entraîne la consommation des ressources. Celles-ci sont des facteurs clés de succès du projet. Elles doivent donc être particulièrement bien pilotées pour ne pas mettre le projet en risque.

## Livrable

Un livrable est tout résultat, document, mesurable, tangible ou vérifiable, qui résulte de l'achèvement d'une partie de projet ou du projet. Un cahier des charges et une étude de faisabilité sont des livrables.

## Triangle de la triple contrainte

Le triangle de la triple contrainte, aussi appelé triangle de la performance, est souvent utilisé pour illustrer l'interdépendance des variables d'un projet caractérisé par 3 objectifs liés :

- **Temps** : il s'agit du délai nécessaire pour achever le projet tel qu'il est décrit dans les prévisions.
- **Budget** : le budget du projet est basé sur le coût des ressources, c'est-à-dire le personnel, l'équipement et le matériel nécessaire à la réalisation des tâches.
- **Objectif** : il s'agit du résultat que l'on veut atteindre après l'achèvement du projet.

$$\text{Temps} + \text{Budget} + \text{Objectif} = \text{Qualité}$$

La figure 3.1 montre un triangle de la triple contrainte.

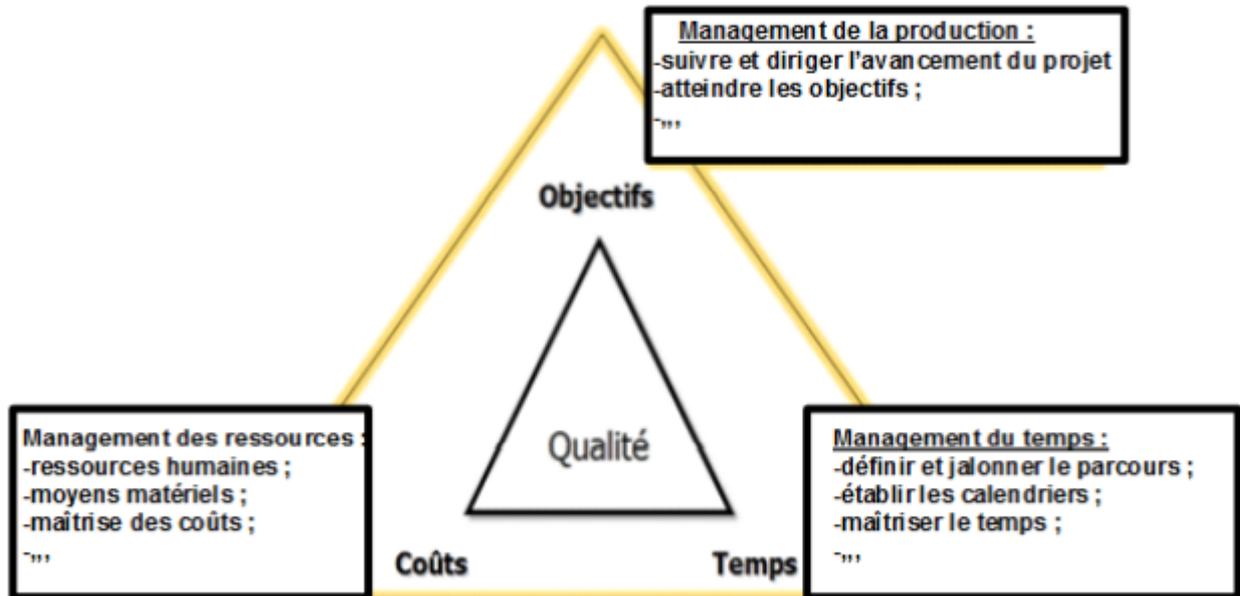


FIGURE 3.1 – Triangle de la triple contrainte

Le principe du triangle de la performance, c'est qu'il n'est pas possible de modifier le budget, la planification ou l'objectif d'un projet sans affecter au moins l'un des deux autres paramètres.

### Cahier des charges

Le terme "Cahier des charges" désigne un document qui fait office de contrat entre deux parties. Ce document décrit précisément les attentes du maître d'œuvre par le maître d'ouvrage, Il s'agit ainsi d'un contrat permettant d'une part de garantir au maître d'ouvrage que les livrables seront conformes à ce qui est écrit, d'autre part d'éviter que le maître d'ouvrage modifie son souhait au fur et à mesure du projet et demande au maître d'œuvre des nouvelles fonctionnalités non prévues initialement. Un cahier des charges doit également contenir tous les éléments permettant au maître d'œuvre de juger de la taille du projet et de sa complexité afin d'être en mesure de proposer une offre la plus adaptée possible en termes de coût, de délai, de ressources humaines et d'assurance qualité. Il s'agit à ce titre d'un outil de communication indispensable entre le vendeur et le client et entre le chef de projets et les différents prestataires ou exécutants en interne, Il permet en outre de répondre à l'obligation générale d'information du vendeur ou du prestataire vis-à-vis de son client. Éléments principaux d'un cahier des charges :

1. **Contexte** : Un cahier des charges commence généralement par une section

décrivant le contexte, c'est-à dire notamment le positionnement politique et stratégique du projet.

2. **Objectifs** : Le cahier des charges doit permettre de comprendre le but recherché, afin de permettre au maître d'oeuvre d'en saisir le sens.
3. **Périmètre** : Le périmètre du projet permet de définir le nombre de personnes ou de ressources qui seront impactées par sa mise en place.
4. **Calendrier** : Le calendrier souhaité par le maître d'ouvrage doit être très clairement explicité et faire apparaître la date à laquelle le projet devra impérativement être terminé 14.

## 3.2 Gestion de projet et planification

### 3.2.1 Le cycle de vie d'un projet

Un projet a également un cycle de vie précis : il débute, se réalise puis se termine. Les différentes phases du cycle de vie d'un projet sont :

1. Lancement
2. Planification
3. Exécution
4. Suivi
5. Clôture

Comme illustré dans la figure 3.2

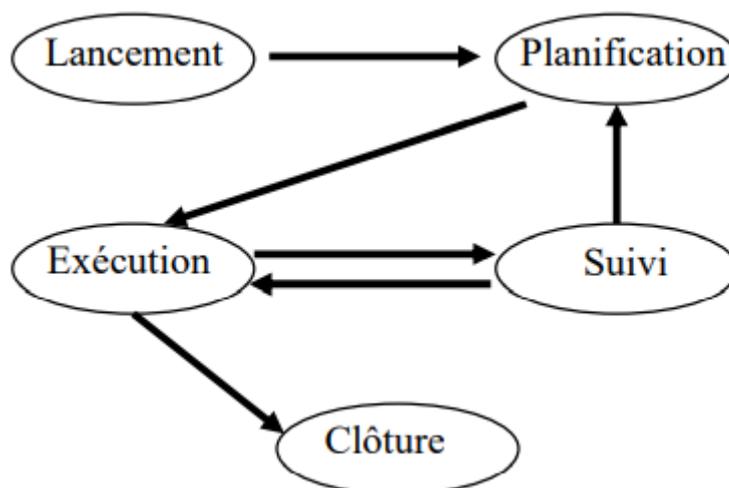


FIGURE 3.2 – Cycle de vie d'un projet

### 3.2.2 Gestion de projet

La gestion de projet ou conduite de projet est une démarche visant à structurer, assurer et optimiser le bon déroulement d'un projet. Généralement en respectant des limites de temps, de ressources ou de coûts 16.

### 3.2.3 Les étapes de la gestion de projet

1. **L'organisation** : Organisation structurelle, des flux d'informations, des acteurs et des supports de communications.
2. **La planification** : Estimation des coûts et des délais.
3. **La coordination** : Entre les différents acteurs du projet, responsables, exécutants...
4. **Le pilotage** : Organisation du déroulement du projet, découpage en activités. Suivi du déroulement, gestion des ressources.
5. **La surveillance** : Contrôle des coûts, des délais et de la qualité.

Les étapes de la gestion de projet sont représentées dans la figure 3.3

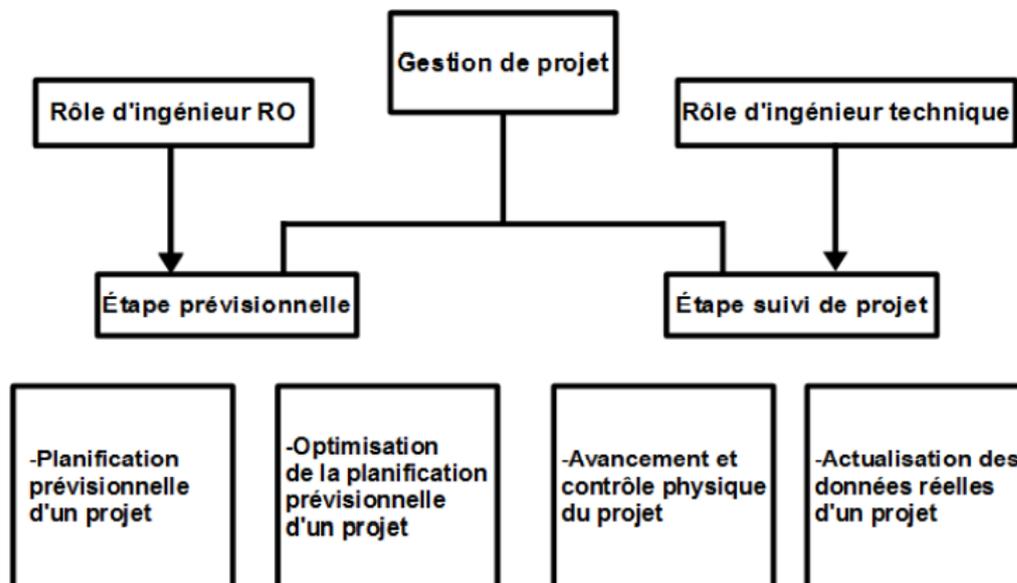


FIGURE 3.3 – Etapes de la gestion de projet

### 3.2.4 Les facteurs critiques de succès en gestion de projet

Dans la littérature, un projet est considéré comme un succès s'il répond à deux critères :

1. **Les critères internes qui peuvent être contrôlés par le gestionnaire** :  
le projet est complété à temps, selon le budget et avec un niveau de performance acceptable.
2. **Les critères externes qui sont plus ou moins contrôlables par le gestionnaire** :  
l'utilisation du projet par le client, la satisfaction des clients avec la performance du projet et l'impact du projet sur l'efficacité organisationnelle.

### 3.2.5 Planification de projet

La planification est parmi les phases d'avant-projet les plus importantes. Elle consiste à déterminer et à ordonnancer les tâches du projet et à estimer leurs charges respectives.

La planification d'un projet consiste à :

- Définir les activités (tâches) constituant le projet.
- Découper le projet en phases et les phases en tâches.
- Organiser les activités dans le temps.
- Evaluer les dépendances entre activités.
- Evaluer l'effort nécessaire pour chaque activité (durée maximale et minimale).
- Analyser les résultats (délai de réalisation du projet, chemin critique, . . . etc.)
- Optimiser le planning.
- Editer le planning sous une forme temporelle claire et bien adaptée aux divers utilisateurs.

Les objectifs principaux de la planification de projet sont les suivants :

1. Minimiser la durée d'exécution totale du projet.
2. Minimiser le coût total du projet.
3. Gestion optimale des ressources.

### 3.3 Problème d'ordonnement

L'ordonnement est l'élaboration d'un plan d'action permettant de déterminer les séquençements ou au contraire les parallélismes possibles entre l'exécution des tâches précédemment identifiées cette dernière est une étape cruciale et non négligeable dans la gestion de projet.

Pour procéder à l'ordonnement des tâches, il faut, pour chaque tâche élémentaire, lister les tâches antérieures, au vu des informations collectées sur le terrain et sélectionner les seules tâches immédiatement antérieures. Le planning doit permettre l'identification de l'ordonnement des tâches du projet.

Nous pouvons distinguer deux catégories de problèmes d'ordonnement :

- **Problème statique** : lorsque les tâches à ordonner sur une période ainsi que l'état initial de l'atelier sont connus au début de la période.
- **Problème dynamique** : lorsque les décisions sont à prendre sur la période mais toutes les tâches à réaliser sur cette période ne sont pas connues au début de la période.

On dit qu'on a affaire à un problème d'ordonnancement lorsque, en vue de la réalisation d'un objectif quelconque, il faut accomplir de multiples tâches, elle-même soumises à un ensemble de contraintes.

Les contraintes auxquelles sont soumises les diverses tâches qui concourent à la réalisation de l'objectif sont de divers type. On distingue :

### 3.3.1 Les contraintes potentielles

Elles peuvent être de deux sortes :

- Les *contraintes d'antériorité* selon laquelle une tâche  $j$  ne peut commencer avant une tâche  $i$  ne soit terminée, par exemple, la construction des piliers suit les fondations.
- Les *contraintes de localisation temporelle* impliquant qu'une tâche donnée  $i$  ne peut débuter avant une date imposée, ou qu'elle peut s'achever après une date imposée.

### 3.3.2 Les contraintes disjonctives

Une contrainte disjonctive impose la non-réalisation simultanée de deux tâches  $A$  et  $B$ . On trouve de telles contraintes dans le cas d'utilisation d'une ressource présente en un seul exemplaire (une grue, une équipe, etc.) ou pour formuler des interdictions de réalisation simultanée pour des raisons de sécurité ou des problèmes de place. Arbitrer une contrainte disjonctive consiste à décider si  $A$  sera fait avant  $B$  ou l'inverse.

### 3.3.3 Les contraintes cumulatives

On parle de contraintes cumulatives lorsque les tâches demandent une partie d'une ou plusieurs ressources présentes en quantité limitée. Le problème est beaucoup plus combinatoire que pour les contraintes disjonctives. Considérons l'exemple où nous avons cinq intervenants et cinq tâches à effectuer. Chaque tâche demande la présence d'un certain nombre de ces intervenants (voir tableau suivant de la figure 3.4)

Tâche	A	B	C	D	E
Nombre d'intervenants	4	3	2	1	1

FIGURE 3.4 – Exemple de contraintes cumulatives

Pour que l'ordonnancement soit réalisable, il faut qu'on utilise, à tout moment, au plus cinq intervenants. Cette contrainte va interdire les ordonnancements réalisant en parallèle :  $(A//B)$ ,  $(A//C)$ ,  $(A//D//E)$ ,  $(B//C//D)$ ,  $(B//C//E)$ . Ces configurations sont minimales au sens que, si  $(A//B)$  est interdit, toute configuration contenant  $(A//B)$  est aussi (par exemple :

$(A//B//D), (A//B//C//D)$  etc.

Lorsqu'une configuration minimale porte sur deux tâches (ici  $A//B$  ou  $A//C$ ), on peut la remplacer par une contrainte disjonctive entre ces deux tâches.

### 3.4 Modélisation d'un problème d'ordonnancement via les graphes

La théorie des graphes et des méthodes utiles s'avèrent être un outil très commode pour la formulation du problème de l'ordonnancement dans le langage mathématique, précisément celui de la programmation linéaire. Dans cette section, nous présentons les éléments et les notions fondamentales qui permettent une telle formulation. Il existe deux approches à cet effet :

- La méthode dite **AoN** (*Activity-on-Node*) ou en français activités sur les noeuds, dans laquelle les tâches du projet sont représentées par des noeuds (sommets) du graphe correspondant, les arcs décrivent les règles de précedence entre les différentes tâches du projet. Un projet composé des activités (tâches) a, b, c, d et e avec a précède d, a précède c, b précède e, c précède e, est représenté par la figure 3.5

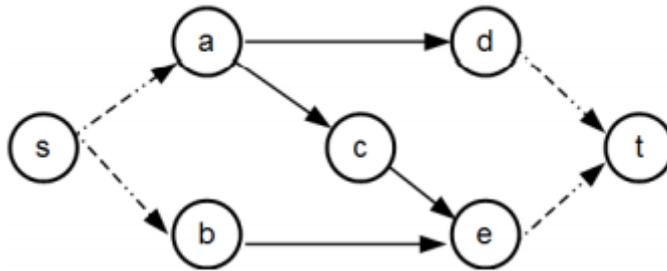


FIGURE 3.5 – Représentation AoN

Les noeuds fictifs S et T représente respectivement le début et la fin du projet. Cette représentation ne nécessite pas l'introduction d'activités fictives (noeud) autre que celles de "début" et "fin".

- La méthode dite **AoA** (*activity-On-Arow*) ou en français activités sur les arcs, Dans cette représentation chaque activité est représentée par un arc et chaque événement par un sommet. Le noeud du début de l'arc correspond au début de l'activité, le noeud d'arrivé correspond à la fin de l'activité ou de la tâche. La relation de précedence classique fin-début avec marge nulle entre deux activités a et b (notée aussi  $a \leq b$ ) signifie que l'activité b peut débuter aussi tôt que l'activité a est achevée. Graphiquement cela est traduit par la coïncidence du noeud terminal de l'arc représentant l'activité a avec le noeud initial de l'arc représentant l'activité b, (voir la figure 3.6)

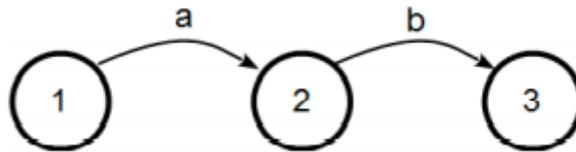


FIGURE 3.6 – représentation AoA

### 3.5 Les outils de la planification de projet

Il existe deux catégories :

1. Dans cette première catégorie, les fonctionnalités les plus communément utilisées sont : Structure de répartition de travail **WBS**, Structure Organisationnelle (**OBS**), matrice **RACI**, diagramme de **Gantt**, la technique de **PERT**, la méthode du potentiel (**MPM**) et la méthode du chemin critique.

#### 3.5.1 Structure de répartition de travail WBS

La planification commence par un recensement des tâches à réaliser. La décomposition structurée des activités (WBS : Work Breakdown Structure) permet de recenser l'ensemble des activités d'un projet et de les décomposer. La décomposition apparaît sous forme arborescente. Il s'agit d'une décomposition purement statique : elle ne tient pas compte du temps, et par conséquent ne s'attache pas à l'ordonnancement des activités. Elle permet une présentation analytique : on doit décomposer jusqu'à obtenir des activités qui soient bien définies et faciles à gérer c'est à dire dont les entrées et résultats sont parfaitement identifiés et dont la responsabilité est confiée à une ou des personnes précise(s) [17].

La figure 3.7 montre la structure de répartition de travail WBS.

#### 3.5.2 Structure Organisationnelle (OBS)

L'OBS est un schéma qui représente les responsabilités de chaque membre pour chaque tâche d'un projet, autrement dit il permet de définir les responsabilités et les actions dans les tâches.

#### 3.5.3 Matrice RACI

L'acronyme **RACI**, représente une matrice des responsabilités indiquant les rôles et les responsabilités des intervenants au sein de chaque processus et activité. Cette matrice représente l'organisation du travail en reliant dans un tableau commun la structure de découpage du projet (WBS) et la Structure organisationnelle du projet (OBS).

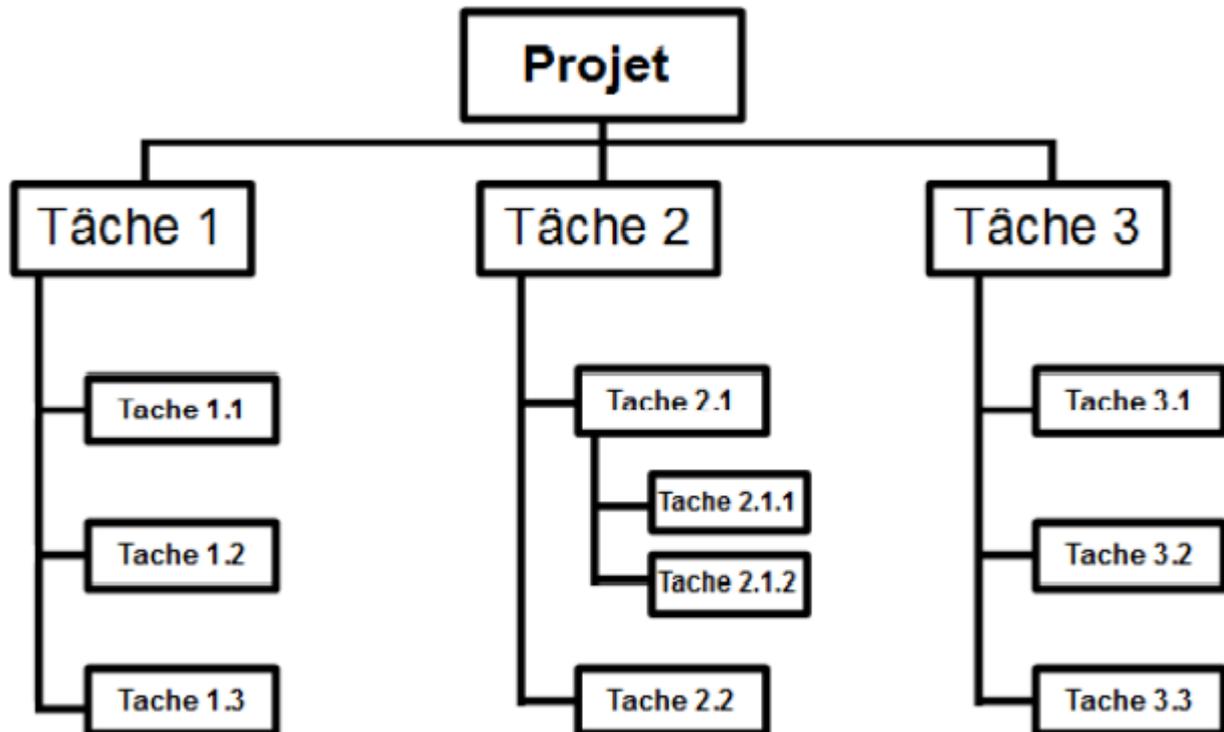


FIGURE 3.7 – Schéma WBS

Les lignes de la matrice référencent les activités identifiées, et les colonnes les rôles (personnels impliqués par métier). Dans chaque cellule [activité ; rôle] figure la lettre « R », « A », « C » ou « I », où l’acronyme anglais RACI signifie :

- R : responsable (responsable).
- A : accountable (acteur).
- C : consulted (consulté).
- I : informed (informé).

## 2. Le séquençement des activités :

Processus qui consiste à identifier et documenter les relations entre les activités du projet. Il existe des techniques sont utilisées concurremment pour réaliser le séquençement des activités d’un projet :

la méthode des potentiels(MPM) le réseau PERT et le diagramme de GANTT.

### 3.5.4 Méthode P.E.R.T

La méthode P.E.R.T (technique d’évaluation et d’examen de programmes) est une méthode de gestion de projets visant à prévoir les propriétés d’un projet en termes de temps, délais et coûts, cette méthode issue de la marine américaine

est développée aux Etats-Unis en 1958 pour la planification de la construction des sous-marins polaris (missiles balistiques). Son principe est de découper un projet en un ensemble d'actions et de les représenter sous forme graphique selon une représentation AOA, Le graphe de la méthode P.E.R.T est souvent difficile à construire du fait qu'on est souvent amené à introduire des arcs fictifs qui ne correspondent à aucune tâche. Grâce à la chronologie et l'indépendance de chacune des tâches, on structure ainsi l'ensemble et on peut alors planifier la réalisation de chacune des tâches les unes par rapport aux autres, afin de minimiser les délais, ainsi que, réduire l'impact des retards lors de l'exécution des différentes tâches.

Son objectif est de Mettre en évidence les différentes liaisons du projet, c'est-à-dire des tâches entre elles. (Nécessité de la méthode : Graphe). **Structure Bâtir :**

- Le graphe correspondant aux tâches et à leurs contraintes.
- Notion de temps rattachés à chaque tâche.
- Observer le chemin critique.

**Domaine d'application :**

La méthode Pert s'adapte à tout problème décomposable en activité elle-même soumises à des contraintes d'antériorités .De même les ressources sont supposées disponibles, les durées des activités indépendantes .c'est l'utile le plus utilisé en planification, son utilisation s'elergie à d'autre domaine construction des mobiliers, montage des fusées inter planaires, travaux administratif. Méthodologie .

### 3.5.5 Elaboration et exploitation de réseau PERT

#### 1. Définir le problème : il s'agit dans cette étape :

- Etablir la liste exhaustive des activités à exécuter ;
- De déterminer la durée de chaque activité ;
- D'établir les conditions d'antériorités.

a) **Tracer le réseau pert :** Pour construire un graphe PERT dans le cas général, on procédera de la manière suivante :

b) **Détermination des niveaux des tâches :**

- On attribuera le niveau 0 aux tâches qui n'ont pas de tâche antérieure.

- On attribuera le niveau 1 aux tâches dont les tâches antérieures sont de niveau 0.
  - On déterminera ainsi le niveau de chaque tâche : les tâches de niveau  $(k + 1)$  seront les tâches dont les tâches antérieures sont de niveau inférieur avec au moins une tâche de niveau  $(k)$  parmi elles.
  - On construira le graphe en Traçant les tâches par ordre de niveau croissant.
- c) **Déterminer les Tâches commençantes , finissantes, convergentes** : Avant de se lancer dans la construction du graphe, il sera souvent utile de détecter les tâches dites commençantes, finissantes ou convergentes.
- **Les tâches commençantes** :  
sont les tâches sans tâche antérieure, elles partent du sommet 1 du graphe (appelé entrée du graphe).
  - **Les tâches finissantes** :  
sont les tâches qui ne sont pas tâche antérieure, elles arrivent au sommet terminal du graphe (appelé sortie).
  - **Les tâches convergentes** :  
sont des tâches que l'on rencontre toujours ensemble (i.e. jamais l'une sans l'autre) dans la colonne((tâches antérieures)) , dans le graphe, elles auront le même sommet terminal.
- d) **Calculer les dates des activités et déterminer le chemin critique** :

#### Notions des dates et marges

- **Date de début au plus tôt** : La date début au plus tôt d'une tâche  $i$ , qu'on la note  $t_i$  est la date minimum à laquelle elle peut commencer. C'est donc la date avant laquelle un événement ne peut pas se réaliser.
- **Date de début au plus tard** : La date de début au plus tard d'une tâche  $i$ , notée  $T_i$ , est la date limite de sa réalisation. Après quoi le projet sera retardé.
- **Date de fin au plus tôt** La date de fin au plus tôt d'une tâche  $i$ , notée  $tf_i$  est la date à laquelle la tâche  $i$  se termine en commençant à sa date de début au plus tôt.
- **Date de fin au plus tard** La date de fin au plus tard d'une tâche  $i$ , notée  $Tf_i$  est la date minimum des dates de fin au plus tard des tâches qui succèdent. Elle est la date à laquelle la tâche  $i$  se termine si elle commence à sa date de début au plus tard.

e) **calculer les marges :**

- **Marge totale** La marge totale de la tâche  $i$ , notée  $MT_i$ , est le retard maximum que l'on peut apporter au démarrage de la tâche  $i$  sans perturber la fin du projet, mais en acceptant de perturber l'exécution d'autres tâches.
- **Marge libre** La marge libre de la tâche  $i$ , notée  $ML_i$ , le retard maximum que l'on peut apporter au démarrage de la tâche  $i$  non seulement sans allonger le délai total du projet, mais aussi sans décaler l'exécution d'aucune autre tâche du projet. [6]

**Méthode de Calcul des dates**

- a) **Calcul des dates de début au plus tôt** Pour calculer la date de début au plus tôt de la tâche  $A$   $t'_A$  on procède comme suit : La date de début au plus tôt de la tâche  $\alpha$  est initialisée à 0 donc nous avons :  $t_\alpha = 0$  Pour les autres tâches :

Soit il y a un seul arc entre les tâches  $A$  et  $B$  telle que la tâche  $A$  précède la tâche  $B$  alors : La date de début de la tâche  $B$  égale la date de début au plus tôt de la tâche  $A$  plus la durée de  $A$   $d'_A$  donc :

$$t_B = t_A + d_A$$

Soit il y a  $k$  arcs ( $k \geq 2$ ) alors

$$t_B = \max(t_{A1} + d_{A1}, \dots, t_{Ak} + d_{Ak})$$

- b) **Calcul des dates de début au plus tard** Pour la date de début au plus tard de la tâche  $A$   $T'_A$  nous avons : La date de début au plus tard de la tâche  $\omega$   $T'_\omega$  est initialisée à sa date de début au plus tôt donc :  $T_\omega = t_\omega$ . Soit il y a un seul arc entre  $A$  et  $B$  alors

La date de début au plus tard de la tâche  $A$  égale la date de début au plus tard de la tâche  $B$  moins la durée de la tâche  $A$  donc

$$T_A = t_B - d_A$$

Soit il y a  $k$  arcs qui sortent de la tâche  $A$  alors :

$$T_A = \min(T_{B1}, \dots, T_{Bk}) - d_A$$

- c) **Calcul des dates de fin au plus tôt et au plus tard** Pour le calcul de la date de fin au plus tôt de la tâche  $A$  notée  $t_{fA}$ , elle s'obtient clairement en ajoutant la durée de la tâche  $A$  à sa date de début au plus tôt :

$$t_{fA} = t_A + d_A$$

La date de fin au plus tard de la tâche A notée  $T_{fA}$ , s'obtient en ajoutant sa durée à sa date de début au plus tard :

$$T_{fA} = T_A + d_A$$

d) Calcul des marges

a **Marge totale** La marge totale d'une tâche A est égale à la différence entre la date de début au plus tard et la date de début au plus tôt (ou entre la date de fin au plus tard et la date de fin au plus tôt) d'une même tâche. C'est à dire :

$$M_{TA} = T_A - t_A \text{ (ou } M_{TA} = T_{fA} - t_{fA} \text{)}$$

b **Marge libre** La marge libre d'une tâche A est égale à la différence entre la plus petite des dates de début au plus tôt des tâches immédiatement suivantes de A et la date de fin au plus tôt de la tâche A.

C'est-à-dire :

$$M_{LA} = \min(t_{A1}, t_{A2}, \dots, t_{Ak}) - t_{fA}$$

Les tâches  $A_1, A_2, \dots, A_k$  sont les tâches qui succèdent la tâche A

c **Tâche critique** une tâche dont la marge totale est nulle, c'est en quelque sorte une tâche "urgente", une tâche sur laquelle il ne faut pas prendre de retard si l'on ne veut pas augmenter la durée totale du projet. [6]

d **Chemin critique** il est formé d'une succession de tâches, sur le chemin le plus long en terme de durées. Il est appelé chemin critique car tout retard pris sur l'une des tâches de ce chemin, entraîne du retard dans l'achèvement du projet.

### 3.5.6 Règles de construction d'un réseau P.E.R.T :

1. Tâche représentée par un arc
2. Deux tâches successives sont représentées par deux flèches successives.
3. Le graphe doit avoir une seule entrée.
4. Le graphe doit avoir une seule sortie.
5. Le graphe se comporte comme une boucle.
6. Chaque étape est référencée par un numéro.

**Remarque :** Pour les tâches fictives (arc fictif) elle sont nécessaires pour représenter les contraintes du projet, elles ont pour valeur 0 et sont souvent représentées en pointillés.

La figure 3.8 représente un tableau d'ordonnancement.

Alors nous obtenons le graphe de la figure 3.9 :

Tâche	Désignation	Antériorité	durée
A	Etudes et calculs des besoins en composants et matières	/	1
B	Délai de livraison de l'ensemble de tuyauterie	C	5
C	Appel d'offre pour l'ensemble des composants	A	10
D	Délai de livraison de la pompe	C	5
E	Délai de livraison de l'ensemble électrique	C	3
F	Installation et montage de la partie hydraulique	B et D	4
G	Branement et raccordement électrique de la pompe	E et F	2
H	Mise en route de l'ensemble	G	1

FIGURE 3.8 – Tableau d'ordonnancement

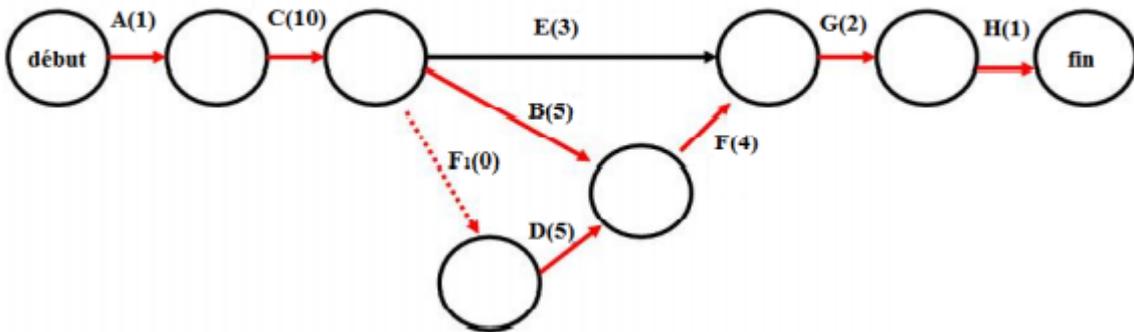


FIGURE 3.9 – Représentation graphique de la méthode PERT

### 3.5.7 Diagramme de Gantt

Le diagramme de Gantt est un outil utilisé en ordonnancement et en gestion de projet permettant de visualiser dans le temps les diverses tâches composants un projet, cette méthode a été développée au début des années 1960 par Henry Gantt, l'axe des abscisses représente le temps et l'axe des ordonnées est utilisé pour représenter les tâches, chaque tâche est représentée par une barre (segment de droite) horizontale dont sa longueur est proportionnelle à la durée de la tâche, ce diagramme est utilisé souvent en complément d'un réseau P.E.R.T. Son objectif est de mettre en évidence les durées des tâches et leurs positionnements relatifs.

#### Avantages :

- Extrêmement compréhensible par tous.
- Permet de suivre le déroulement des opérations dans le temps.
- Permet une certaine synthèse de l'ensemble du projet.

#### Inconvénients :

- Ne prend pas en compte les contraintes.

- Ne permet pas de connaître le chemin critique.
- Difficulté de remise à jour (pour les systèmes manuels).

Le diagramme associé au graphe de la figure 3.9 est montré dans la figure 3.10 .

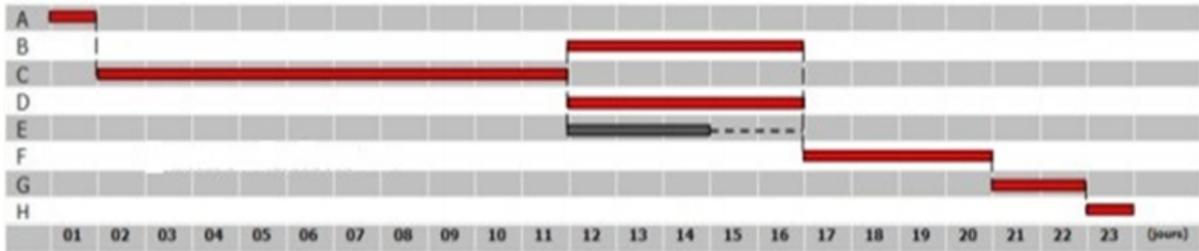


FIGURE 3.10 – Diagramme de Gantt

### 3.5.8 Méthode du Potentiel

La méthode des potentiels est une application de la théorie des graphes à l'optimisation de l'ordonnancement des tâches au sein d'un projet industriel complexe.

On associe au problème central d'ordonnancement de projet un graphe dont les sommets représentent les diverses tâches du problème, on ajoute un nœud 0 qui correspond à la date du début de projet et nœud  $f = n + 1$  qui correspond à la fin du projet. Les arcs du réseau représentent les contraintes de précédence entre les tâches.

A l'identique de la méthode P.E.R.T cette méthode permet de réduire la durée totale d'un projet. On étudie les délais sans prendre en compte les charges et les moyens disponibles. cette méthode est une représentation graphique qui permet de bâtir un « réseau ». Ce réseau est constitué par des tâches (ou étapes). Voici un tableau d'ordonnancement présenté dans la figure 3.11 :

Le graphe MPM associé au tableau ci-dessus est montré dans la figure 3.12 . La figure 3.13 représente un schéma de planification de projet.

Tâche	Durée	Précédence
1	5	-
2	4	1
3	2	2
4	2	3
5	3	4
6	5	3
7	3	2
8	3	7
9	4	6,8
10	10	5,9
11	12	10

FIGURE 3.11 – Tableau d’ordonnancement

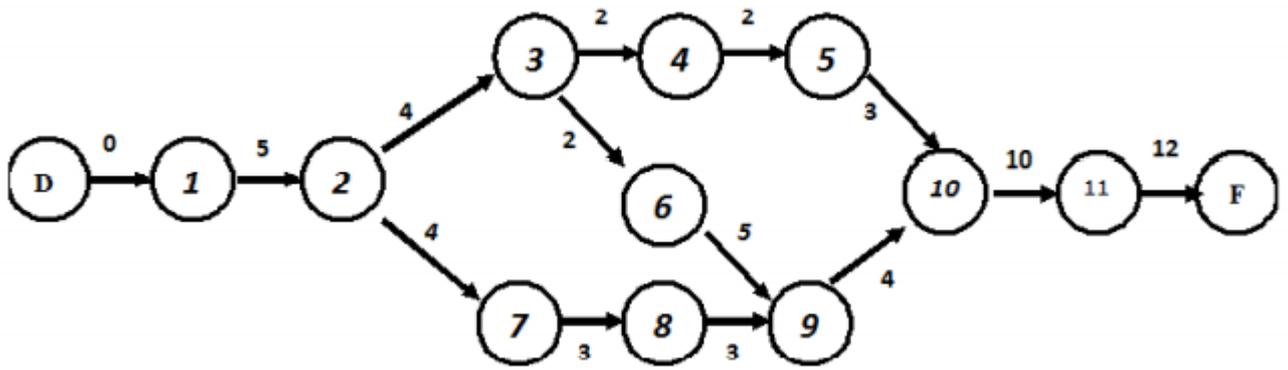


FIGURE 3.12 – Représentation graphique de la méthode du potentiel

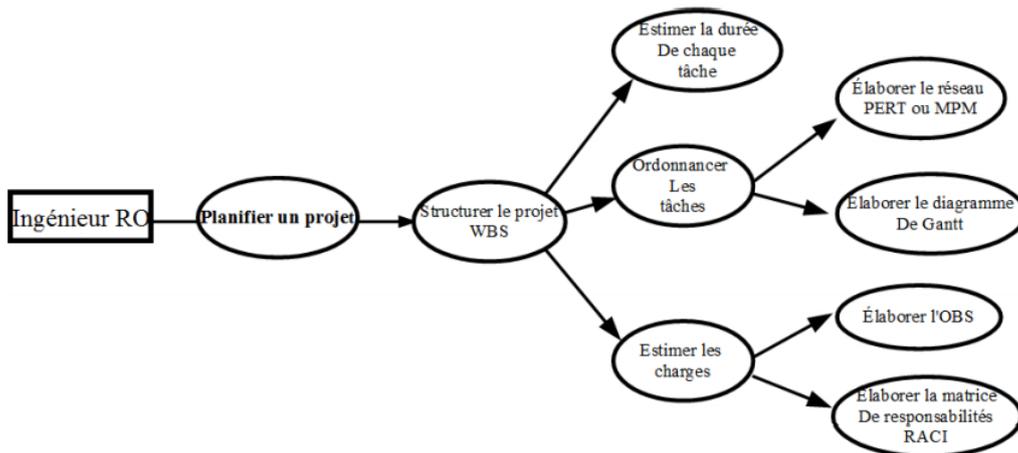


FIGURE 3.13 – Schéma planification de projet

## Chapitre 4

# Résolution de quelques problèmes concrets

Nous allons présenter quelques situations concrètes pour lesquelles nous allons modéliser sous forme de graphe afin d'optimiser certains paramètres en utilisant les méthodes citées dans le chapitre précédent.

### 4.1 Résolution d'un problème d'ordonnancement :

#### 4.1.1 Problème 1

La rénovation du séjour d'un appartement se décompose en plusieurs tâches décrites dans le tableau ci dessous. Ce dernier donne également les précédences à respecter lors de la planification des travaux ainsi qu'une estimation de la durée de chacune des tâches [9].

La figure 4.1 représente un tableau d'ordonnancement.

Rubrique	tâches	Description	Durée	tâches préc
	Début	Lancement du projet	0	-
	A	Dépose ancien carrelage	6	J
	B	Pose carrelage	4	A
	C	Joints carrelage	2	B
Murs	D	Décollage ancien papier	8	J
	E	Pose faïence	6	D
	F	Pose nouveau papier ancien papier	4	E
Plomberie	G	Dépose ancien évier	1	Début
	H	Déplacement arriv et évac	6	G
	I	Pose et raccordement evier	2	M
Mobilier	J	Dépose ancien éléments	4	G
	K	Assemb. caissons et tiroirs	8	Début
	L	Pose éléments bas	6	B,J,H,K
	M	Pose plan de travail	4	L
	N	Etanchéité plan de travail	1	E, M
	O	Pose éléments hauts	1	E, J
	Fin	Fin du projet	0	C,F,I,N,O

FIGURE 4.1 – Tableau d’ordonnement

Le diagramme de Gantt associé à ce tableau est donné dans la figure 4.2 :

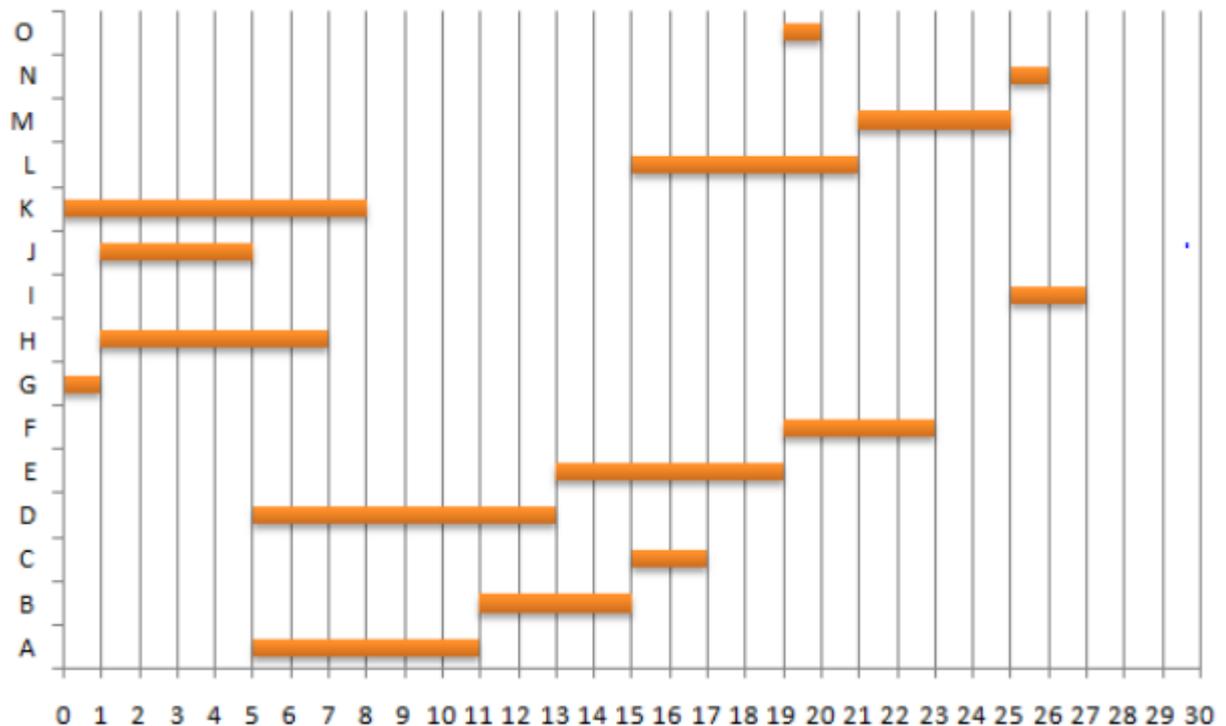


FIGURE 4.2 – Diagramme de Gantt

On peut modéliser le tableau des tâches , sous forme d’un réseau pert po-

tentiels tâches, tel que les tâches sont représentées par des sommets et les contraintes de potentiels sont représentées par des arcs. Le graphe potentiels-tâches correspondant est donné dans la figure 4.3

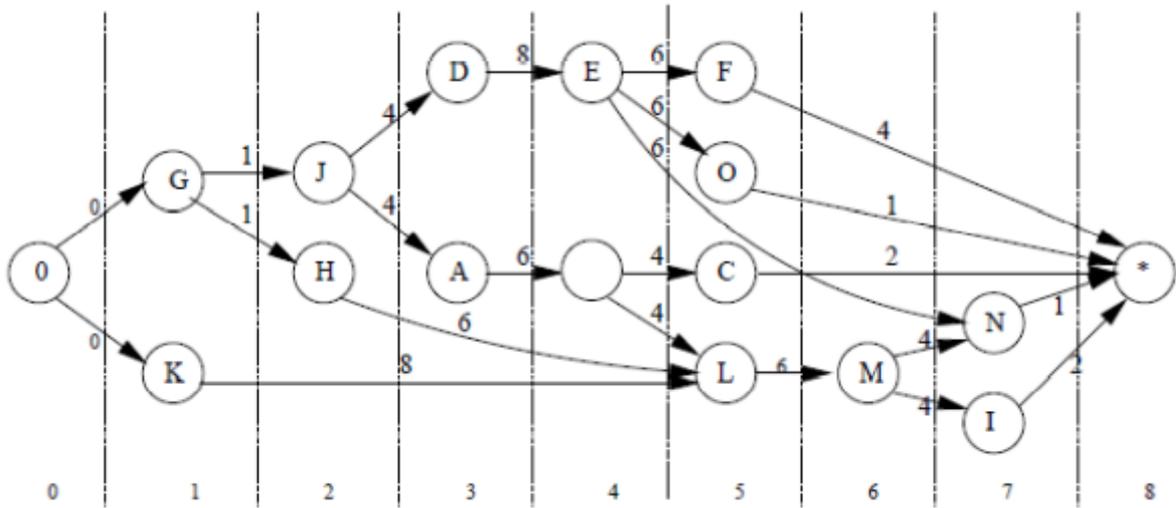


FIGURE 4.3 – Réseau Pert

#### 4.1.2 Dates de début au plus tôt :

On les calcule de la manière suivante :

$$\underline{t}_i = \max_{j \in \Gamma^-(i)} (\underline{t}_j + C_{ji})$$

$C_{ji}$  est la capacité de l'arc (j,i) (par exemple la durée de j),  $\Gamma^-(i)$  est l'ensemble des prédécesseur de i, donc :

$$\underline{t}_G = \max(t_0 + C_{0,G}) = 0 + 0 = 0$$

$$\underline{t}_K = \max(t_0 + C_{0,k}) = 0 + 0 = 0$$

$$\underline{t}_J = \max(t_G + C_{G,J}) = 0 + 1 = 1$$

$$\underline{t}_H = \max(t_G + C_{G,H}) = 0 + 1 = 1$$

$$\underline{t}_D = \max(t_J + C_{J,D}) = 1 + 4 = 5$$

$$\underline{t}_A = \max(t_J + C_{J,A}) = 1 + 4 = 5$$

$$\underline{t}_E = \max(t_D + C_{D,E}) = 5 + 8 = 13$$

$$\underline{t}_B = \max(t_A + C_{A,B}) = 5 + 6 = 11$$

$$\underline{t}_F = \max(t_E + C_{E,F}) = 13 + 6 = 19$$

$$\underline{t}_O = \max(t_E + C_{E,O}) = 13 + 6 = 19$$

$$\underline{t}_C = \max(t_B + C_{B,C}) = 11 + 4 = 15$$

$$\underline{t}_L = \max(\underline{t}_B + C_{B,L}, \underline{t}_H + C_{H,L}, \underline{t}_K + C_{K,L}) = \max(11 + 4, 1 + 6, 0 + 8) = \max(15, 7, 8) = 15$$

$$\underline{t}_M = \max(t_L + C_{L,M}) = 15 + 6 = 21$$

$$\begin{aligned} \underline{t}_N &= \max(t_E + C_{E,N}, t_M + C_{M,N}) = \max(13 + 6, 21 + 4) = \max(19, 25) = 25 \\ \underline{t}_I &= \max(t_M + C_{M,I}) = 21 + 4 = 25 \\ \underline{t}_* &= \max(\underline{t}_F + C_{F,*}, \underline{t}_O + C_{O,*}, \underline{t}_C + C_{C,*}, \underline{t}_N + C_{N,*}, \underline{t}_I + C_{I,*}) \\ &= \max(19 + 4, 19 + 1, 15 + 2, 25 + 1, 25 + 2) = \max(23, 20, 17, 26, 27) = 27 \end{aligned}$$

Les dates de début au plus tôt sont représentées dans le graphe de la figure 4.4

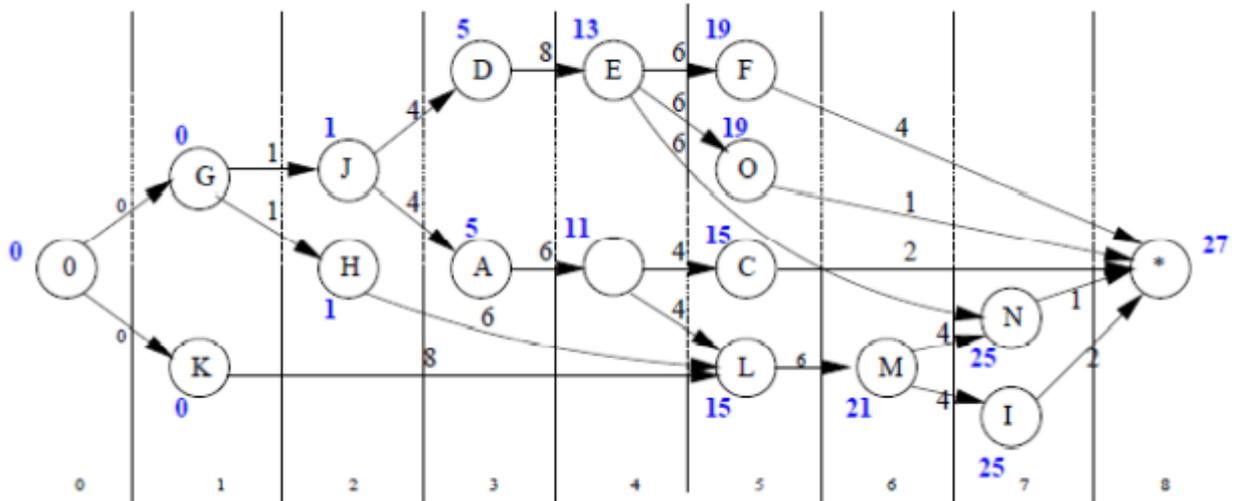


FIGURE 4.4 – Les dates de début au plus tôt

#### 4.1.3 Dates de début au plus tard

On les calcule de la manière suivante :

$$\bar{t}_i = \min_{j \in \Gamma^+(i)} (\bar{t}_j - C_{ij})$$

$C_{ji}$  est la capacité de l'arc  $(j,i)$  (par exemple la durée de  $j$ )  $\Gamma^+(i)$  est l'ensemble des successeurs de  $i$ . donc :

$$\bar{t}_I = \min(\bar{t}_* - C_{I,*}) = 27 - 2 = 25$$

$$\bar{t}_N = \min(\bar{t}_* - C_{N,*}) = 27 - 1 = 26$$

$$\bar{t}_M = \min(\bar{t}_N - C_{M,N}, \bar{t}_I - C_{M,I}) = \min(26 - 4, 25 - 4) = \min(22, 21) = 21$$

$$\bar{t}_L = \min(\bar{t}_M - C_{L,M}) = 21 - 6 = 15$$

$$\bar{t}_C = \min(\bar{t}_* - C_{C,*}) = 27 - 2 = 25$$

$$\bar{t}_O = \min(\bar{t}_* - C_{O,*}) = 27 - 1 = 26 \quad \bar{t}_F = \min(\bar{t}_* - C_{F,*}) = 27 - 4 = 23$$

$$\bar{t}_B = \min(\bar{t}_L - C_{L,B}, \bar{t}_C - C_{B,C}) = \min(15 - 4, 25 - 4) = 11$$

$$\bar{t}_E = \min(\bar{t}_N - C_{E,N}, \bar{t}_O - C_{E,O}, \bar{t}_F - C_{E,F}) = \min(26 - 6, 26 - 6, 23 - 6) = \min(20, 20, 17) = 17$$

$$\bar{t}_A = \min(\bar{t}_B - C_{A,B}) = 11 - 6 = 5$$

$$\bar{t}_D = \min(\bar{t}_E - C_{D,E}) = 17 - 8 = 9$$

$$\bar{t}_H = \min(t_L - C_{H,L}) = 15 - 6 = 9$$

$$\bar{t}_J = \min(\bar{t}_A - C_{J,A}, \bar{t}_D - C_{J,D}) = \min(5 - 4, 9 - 4) = \min(1, 5) = 1$$

$$\bar{t}_K = \min(\bar{t}_L - C_{K,L}) = 15 - 8 = 7$$

$$\bar{t}_G = \min(\bar{t}_H - C_{G,H}, \bar{t}_J - C_{G,J}) = \min(9 - 1, 1 - 1) = \min(8, 0) = 0$$

$$\bar{t}_o = \min(\bar{t}_k - C_{o,k}, \bar{t}_G - C_{o,G}) = \min(7 - 0, 0 - 0) = \min(7, 0) = 0$$

Les dates de début au plus tard sont représentées dans le graphe de la figure 4.5

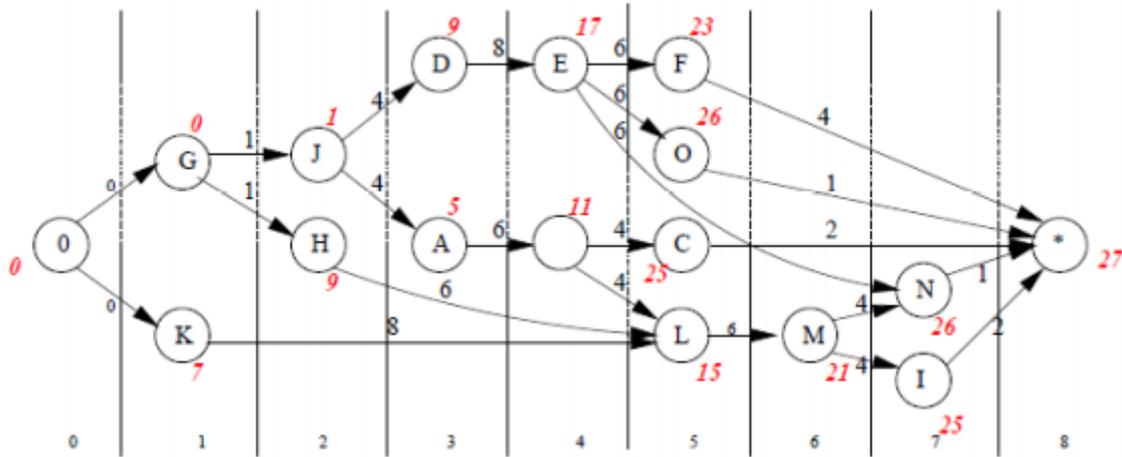


FIGURE 4.5 – Les dates de début au plus tard

#### 4.1.4 Marges totales et chemin critique

On calcule la marge totale d'une tâche  $i$  de la manière suivante :

$$MT_i = \bar{t}_i - \underline{t}_i$$

Le chemin critique est l'ensemble des tâches critiques (les tâches ayant une marge nulle) sont données dans la figure 4.6

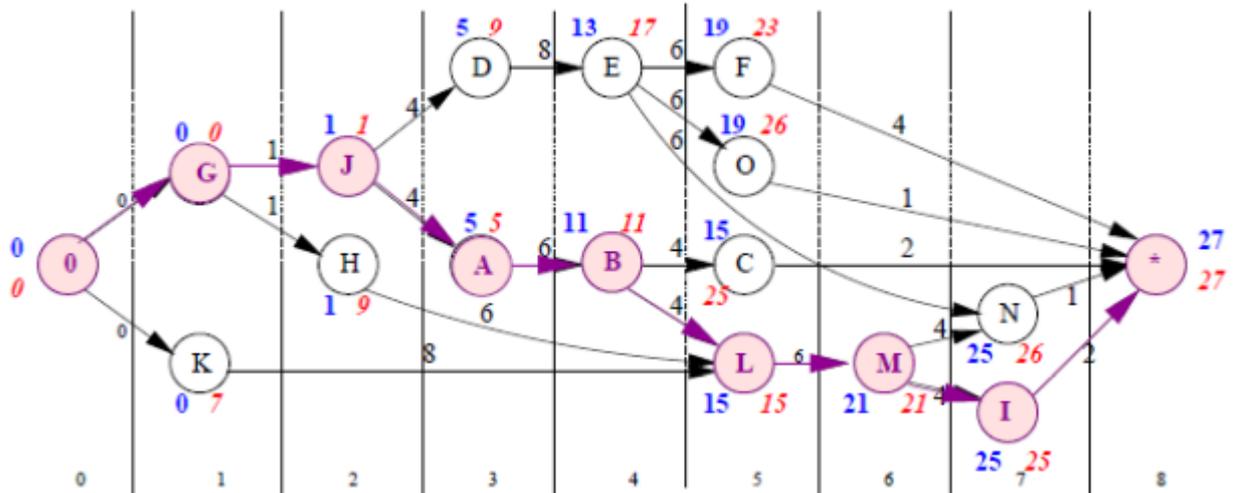


FIGURE 4.6 – Marges totales et chemin critique

Le temps minimum de réalisation de l'ensemble est lisible sur le dernier sommet (\*) =27 jours. .

## 4.2 Problème 2

### 4.2.1 Construction d'un réseau PERT

La construction d'un réseau PERT , et son exploitation , supposent d'effectuer les opérations suivantes :

- Etablir une liste des tâches
- Déterminer les tâches antérieures (ainsi que les tâches postérieures éventuellement )
- Construire le réseau PERT

### 4.2.2 Etablir une liste précise des tâches

On considère la liste de 12 tâches ,numérotés de A à L ,la durée estimée de chaque tâche est indiquée dans le tableau de la figure 4.7 . [12]

Tâche	Durée
A	3
B	1
C	5
D	6
E	4
F	2
G	9
H	5
I	8
J	2
K	3
L	7

FIGURE 4.7 – Tableau d’ordonnement

#### 4.2.3 Déterminer les tâches antérieures

L’analyse du projet et de l’ensemble des tâches le constituant nous amène à définir la relation chronologiques d’antériorités des tâches ; ces résultats sont regroupés dans le tableau de la figure 4.8 .

tâche	Durée	tâches antérieures
A	3	aucune
B	1	A
C	5	A
D	6	B
E	4	B
F	2	C,I,D
G	9	E, F
H	5	aucune
I	8	H
J	2	H
K	3	I
L	7	J ,K

FIGURE 4.8 – Tableau d’ordonnement

Par simple déduction ,on en déduit les valeurs de la colonne tâches postérieures qui sont données dans le tableau ci-dessous de la figure 4.9

Tâche e	Durée	Tâche s antérieures	Tâche s postérieures
A	3	aucune	B ,C
B	1	A	D,E
C	5	A	F
D	6	B	F
E	4	B	G
F	2	C,I,D	G
G	9	E, F	aucune
H	5	aucune	I,J
I	8	H	K
J	2	H	L
K	3	I	L
L	7	J ,K	aucune

FIGURE 4.9 – Tableau d’ordonnancement

On peut voir que l’on se retrouve ici confronté à un problème concernant la tâche I ,en effet,celle-ci précède les tâches F et K.En revanche la tâche K est précédée uniquement de la tâche I (et pas de C et D ),alors que la tâche F est précédée des tâches I,mais aussi D et C .

Si l’on faisait abstraction de ce problème ,on aurait alors une étape,à laquelle C,D et I amènent et de laquelle F et K en repartent or ce ci est incorrecte car cela signifierait alors que K précède aussi C et D en plus de I! Pour résoudre ce problème on va donc utiliser une tâche fictive.

La tâche fictive permet de spécifier que F précède I en plus de C et D et par contre que K précède I uniquement on obtient le réseau complet de la figure 4.10

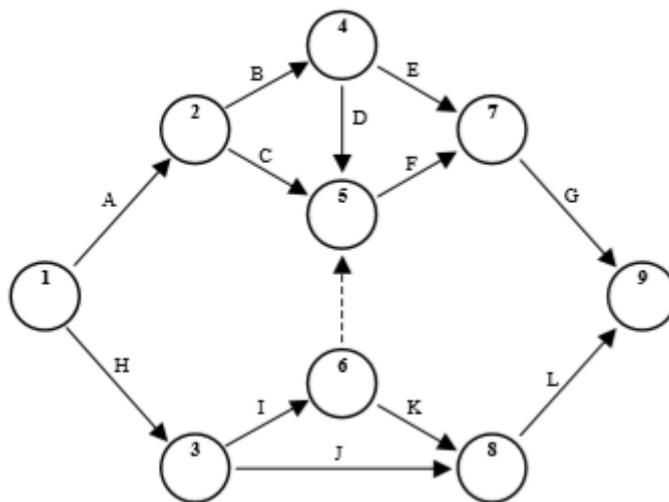


FIGURE 4.10 – réseau PERT

#### 4.2.4 Exploitation du réseau PERT

##### Calcul des dates au plus tôt

On commence tout d'abord par calculer les **dates au plus tôt** .pour une étape donnée ,cette information détermine à quelle date minimum depuis le debut de projet sera atteinte ,au plus tôt ,l'étape considérée.

Pour ce faire ,on se base sur l'estimation de la durée des tâches .On part de l'étape de debut,pour laquelle la date au plus tôt est initialisée à 0 ,et on parcourt le réseau en suivant l'agencement des tâches déterminé auparavant.

Deux méthodes de calcul existent alors selon que l'étape considérée est atteinte par 1 ou plusieurs tâches :

- 1 tâche :il n'y a qu'un seul chemin possible pour atteindre l'étape ;

La date au plus tôt vaut la date au plus tôt de la tâche antérieure à laquelle on rajoute la durée de la tâche liant les deux étapes :  $to0 = to1 + ladurée$  la figure 4.11 montre le calcul de la date au plus tôt dans le cas d'une seul tâche .

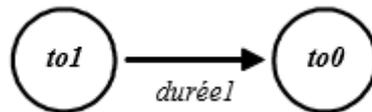


FIGURE 4.11 – calcul de la date au plus tôt dans le cas d'une seul tâche

- Plusieurs tâches :il y a plusieurs chemin pour atteindre l'étape ;

On applique le procédé décrit ci-dessus(pour 1 tâche)pour chacune des tâches antérieures ;la date au plus tôt vaut alors le maximum parmi ces resultats :

$$to0 = \max((to1 + durée1);(to2 + durée2);...)$$

La figure 4.12 représente le calcul de la date au plus tôt dans le cas de plusieurs tâches.

NB :si l'une des tâches considérée est une tâche fictive,on procède exactement de la meme manière qu' avec une tâche non-fictive,et on considère que sa durée a une valeur de 0 .

La date au plus tôt de l'étape de fin indique alors le temps minimum nécessaire à l'achevement du projet . Appliquons ce procédé à notre exemple.

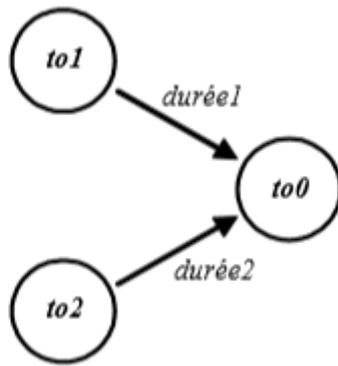


FIGURE 4.12 – calcul de la date au plus tôt dans le cas de plusieurs tâches

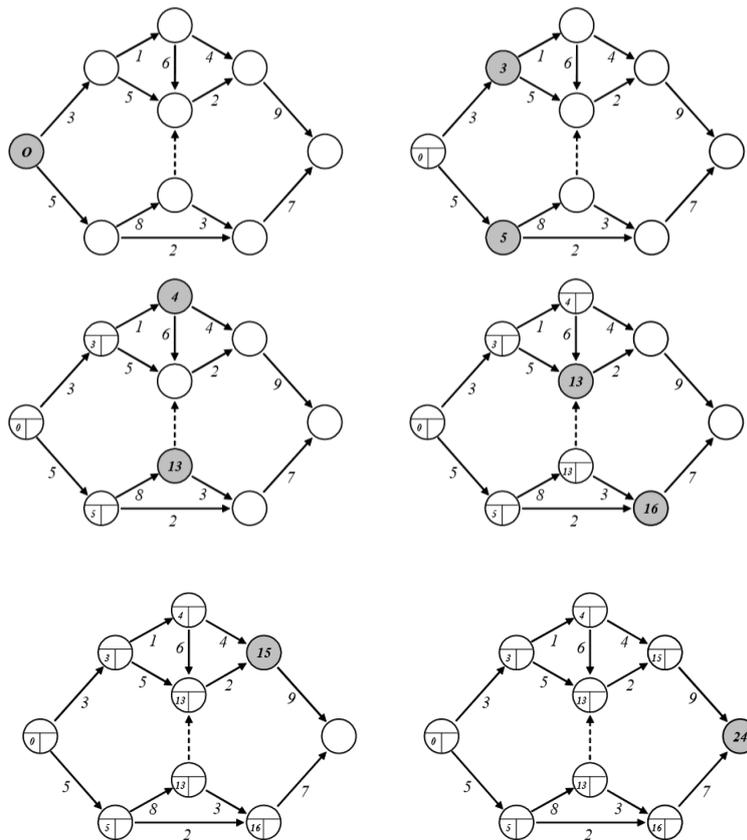


FIGURE 4.13 – calcul des dates au plus tôt

La figure 4.13 représente les dates de debut au plus tôt.

On détermine donc que le projet pourra au mieux être finalisé en 24 jours.

#### Calcul des dates au plus tard

On poursuit avec le calcul des dates au plus tard ,pour une étape donnée ,cette information détermine à quelle date maximum ,depuis le debut du projet ,doit être atteinte,au plus tard l'étape considérée ,afin que le delai de l'ensemble du projet ne soit pas modifié.

Pour ce faire ,on se base sur l'estimation de la durée des tâches

.On part de l'étape de fin ,par laquelle la date plus tard est initialisée à la même valeur que la date au plus tôt déterminée précédemment,et on parcourt le réseau en suivant l'agencement inverse des tâches.

Là encore ,il existe deux méthodes de calcul selon que 1 ou plusieurs tâches partent de l'étape considérée ;

— :il n'ya qu'un seul chemin possible pour partir de l'étape ;

La date au plus tard vaut la date au plus tard" précédente "(la tâche postérieure dans l'agencement des tâches) à laquelle on retranche la durée de la tâche liant les 2 étapes :

$$ta0 = ta1 - durée1$$

La figure 4.14 représente le calcul de la date au plus tard dans le cas d'une seule tâche.

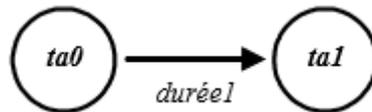


FIGURE 4.14 – calcul de la date au plus tard dans le cas d'une seule tâche

— Plusieurs tâches ,il y a plusieurs chemins possibles qui partent de l'étape. On applique le procédé décrit ci-dessus(pour 1 tâche) pour chacune des tâches "précédentes" la date au plus tard vaut le minimum parmi ces résultats :

$$ta0 = Min(ta1 - durée1); (ta2 - durée1).$$

La figure 4.15 représente le calcul des dates au plus tard dans le cas des plusieurs tâches .

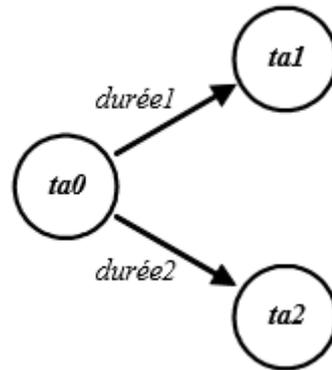


FIGURE 4.15 – calcul des dates au plus tard dans le cas des plusieurs tâches

Une fois cette étape terminée ,on regroupe les informations de dates au plus tôt et au plus tard en un seul schéma .

La figure 4.16 représente un réseau Pert avec les dates au plus tôt et au plus tard .

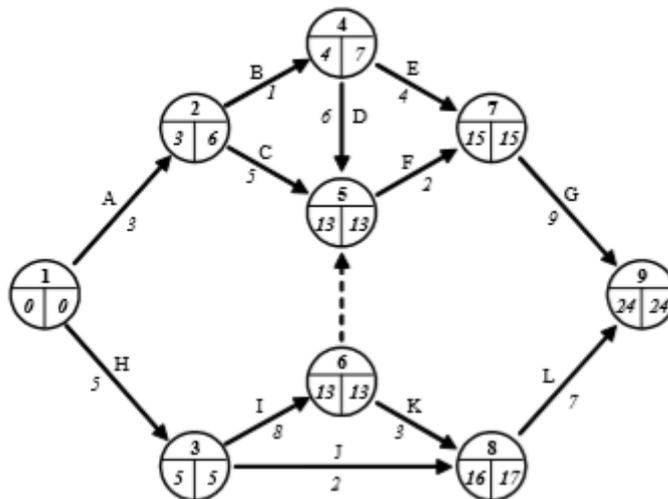


FIGURE 4.16 – Réseau PERT avec les dates au plus tôt et au plus tard

#### 4.2.5 Calcul des marges

Certaines tâches bénéficient d'une latence variable dans leur aboutissement sans pour autant remettre en cause la date d'achèvement du projet ,cette période de latence est appelée (marge) L'évaluation qualitative de ces marges permet d'optimiser la gestion du projet. En effet ,l'analyse de ces marges permet d'aménager le déroulement de certaines tâches selon des critères autre que temporels :couts,plan de charge de l'entreprise ,...

La marge relative à une tâche se détermine en considérant la valeur des dates

au plus tôt et au plus tard des étapes entourant la tâche, la marge vaut alors la date au plus tard de l'étape postérieure à laquelle on retranche la date au plus tôt de l'étape antérieure ainsi que la durée de la tâche elle-même :

$$marge = (ta2 - to1) - durée$$

La figure 4.17 montre le calcul de la marge relative à une tâche .

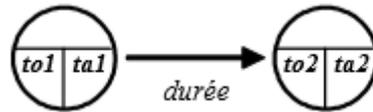


FIGURE 4.17 – calcul de la marge relative à une tâche

Dans le cadre dans notre exemple ,on a regroupé les marges calculées dans le tableau suivant :

Tâche	durée
A	6-0-3=0
B	7-3-1=0
C	13-3-5=5
D	13-4-6= 3
E	15-4-4=7
F	15-3-2=0
G	24-15-9=0
H	5-0-5=0
I	13-5-8=0
J	17-5-2=10
K	17-13-3=1
L	24-16-7=1

#### 4.2.6 Détermination du chemin critique

La figure 4.18 représente la détermination de chemin critique.

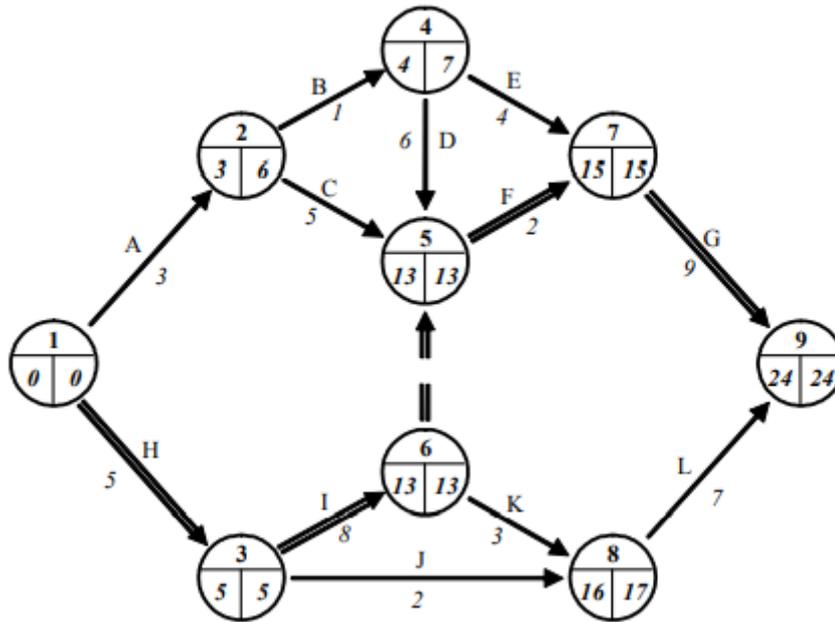


FIGURE 4.18 – détermination du chemin critique

Le chemin critique passe successivement par les tâches H,I,f et G ;soit donc les étapes 1,3,6,5,7 et 9.

### 4.3 Exemple 3

La rénovation du séjour d'un garage clothing se décompose en plusieurs tâches décrites dans le tableau ci-dessous. Ce dernier donne également les précédences à respecter lors de la planification des travaux ainsi qu'une estimation de la durée de chacune des tâches .

/	Tâches	Durée (en jours)	Antériorité
A	Enlèvement des portes	1	-
B	Ponçage et peinture des portes	3	A
C	Pose des portes	1	A
D	Tirage des fils électriques	1	A
E	Pose des prises	1	B,D
F	Ragréage des murs	3	B,C
G	Peinture du plafond	2	B
H	Peinture des cadres	1	F,G
I	Peinture du balcon	2	E,H

La figure 4.19 représente un graphe MPM .

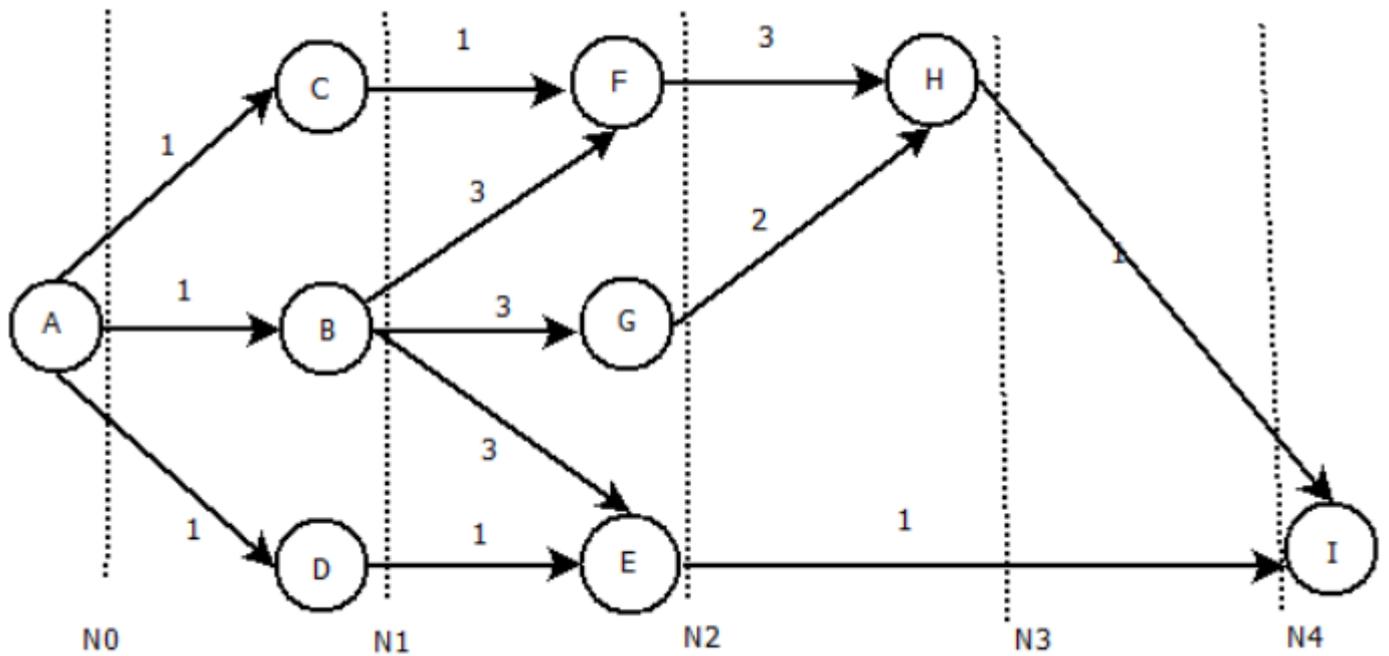


FIGURE 4.19 – Un exemple du graphe MPM

La figure 4.20 montre un graphe PERT.

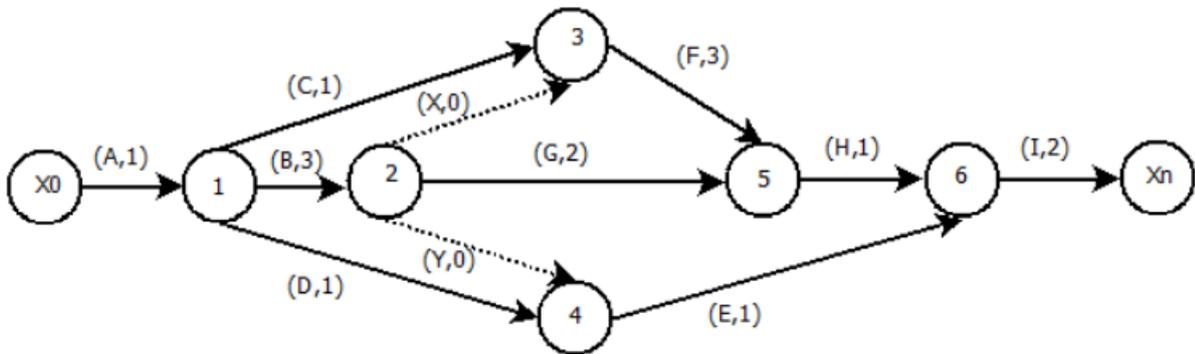


FIGURE 4.20 – Un exemple du graphe PERT

#### 4.4 Calcul des dates et des marges

La figure 4.21 montre la solution avec la méthode MPM.

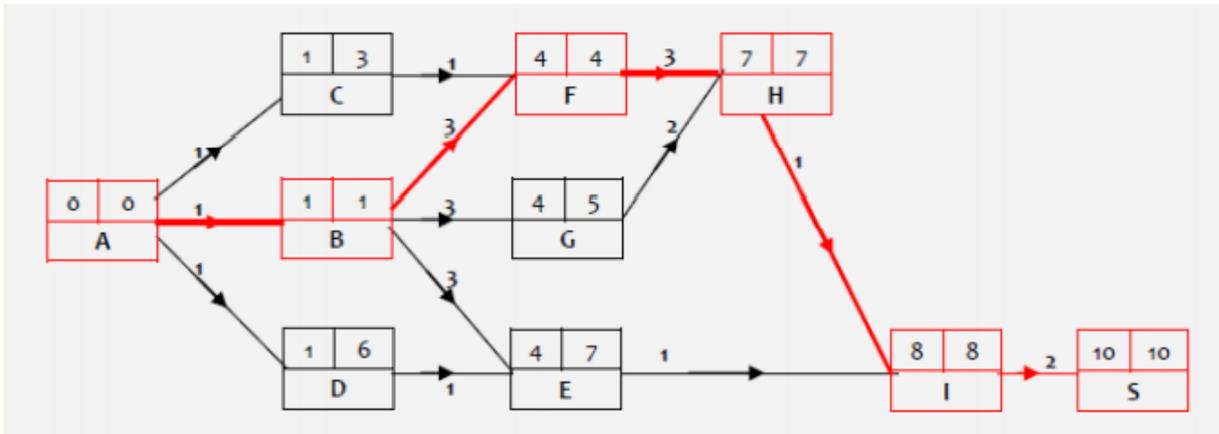


FIGURE 4.21 – Solution avec la méthode MPM

**Le chemin critique** : ABFHI et la durée minimale du projet : 10 jours.

Tâche	A	B	C	D	E	F	G	H	I
Marge Totale	0	0	2	5	3	0	1	0	0
Marge Libre	0	0	2	2	3	0	1	0	0

La figure 4.22 montre la solution avec la méthode PERT.

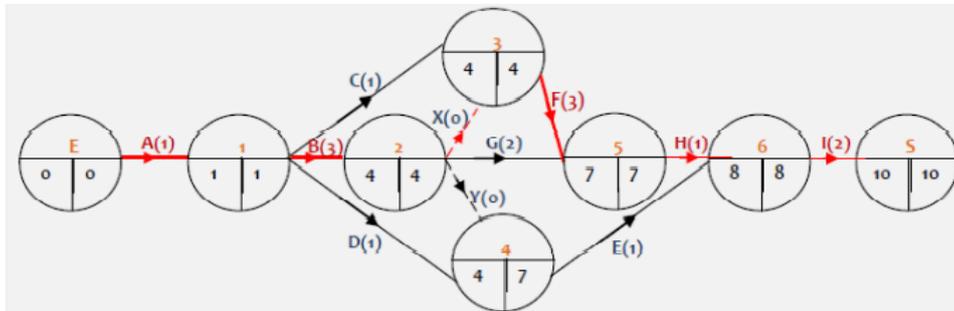


FIGURE 4.22 – Solution avec la méthode PERT

**Le chemin critique** : ABFHI et la durée minimale du projet : 10 jours .

Tâche	A	B	C	D	E	F	G	H	I
Marge Totale	0	0	2	5	3	0	1	0	0
Marge Libre	0	0	2	2	3	0	1	0	0

**X** étant une tâche fictive, elle ne peut être reprise dans le chemin critique.

## Conclusion

Dans ce chapitre on a présenté trois problèmes d'ordonnancement, dans le but de minimiser la durée de projet afin d'accomplir de multiples tâches qui demandent un certain temps d'exécution et qui doivent être exécutées dans un certain ordre.

Il est donc nécessaire d'identifier les tâches prioritaires en fonction de l'objectif à atteindre et, lors de leur exécution, il faut détecter les retards et les dépassements de moyens.

L'objectif est à organiser dans le temps la réalisation de tâches d'un projet, compte tenu de contraintes temporelles (délais, contraintes d'enchaînement) et de contraintes portant sur la disponibilité des ressources requises afin de minimiser la durée globale.

# Chapitre 5

## Application sous C++

### 5.1 Présentation de c++

C++ est un langage de programmation compilé permettant la programmation sous de multiples paradigmes (comme la programmation procédurale, orientée objet ou générique). Ses bonnes performances, et sa compatibilité avec le C en font un des langages de programmation les plus utilisés dans les applications où la performance est critique.

Le C++ n'est pas très récent : il a été développé dans les années 1980 pour apporter plus de fonctionnalités au langage C, développé dans les années 1970. La compatibilité entre les deux langages est de plus en plus limitée. Toutefois, le principal inconvénient du C++ réside dans son haut degré d'abstraction. A la différence des langages comme Python ou Ruby, qui sont relativement plus accessibles au grand public, le C++ est considéré comme un langage de bas niveau, plus proche du fonctionnement de la machine, et donc moins compréhensible et intuitif pour l'humain.

Bien que les langages Java et C soient aujourd'hui les plus complets, le C++ reste très utilisé pour :

- La plupart des applications graphiques : création de programmes en 3D, par exemple
- Le traitement de texte.
- Les jeux vidéos .
- La mise en place d'outils financiers.
- La création de certains programmes militaires.

#### 5.1.1 Planifier un projet par les activités

Dans cette partie ,nous allons gérer un projet consistant en la réalisation des travaux de construire un maison comme il est illustré dans la figure 5.1 .

Code	Designation	Durée	Successeur
1	Etude géotechnique	0	2 ,3
2	Plan de maison	4	4
3	La maçonnerie	1	5
4	La plomberie	4	6
5	La boiserie	5	7
6	La peinture	2	8
7	La décoration	3	8

FIGURE 5.1 – Tableau des tâches

### 5.1.2 Application sous C++

Le programme resoud le problème d'ordonnancement par la méthode PERT et donne le chemin critique à la fin .

Lors du lancement de notre programme, il demande :

1. D'introduire le nombre de tâches .
2. Introduire pour chaque tâche les successeurs et les durées .

La figure 5.2 représente les successeurs et la durée de chaque tâche.

```

Donner le nombre des taches: 8

Espace allouer avec succe
Donner le nombre des taches qui vient apres 1:2
Donner le numero du successeur N 1:2
Donner la duree entre 1 et 2: 0
Donner le numero du successeur N 2:3
Donner la duree entre 1 et 3: 0

Donner le nombre des taches qui vient apres 2:1
Donner le numero du successeur N 1:4
Donner la duree entre 2 et 4: 4

Donner le nombre des taches qui vient apres 3:1
Donner le numero du successeur N 1:5
Donner la duree entre 3 et 5: 1

Donner le nombre des taches qui vient apres 4:1
Donner le numero du successeur N 1:6
Donner la duree entre 4 et 6: 4

```

FIGURE 5.2 – Les successeurs et la durée de chaque tâche

la figure 5.3 représente les successeurs et la durée de chaque tâche.

```

Donner le nombre des taches qui vient apres 5:1
Donner le numero du successeur N 1:7
Donner la duree entre 5 et 7: 5

Donner le nombre des taches qui vient apres 6:1
Donner le numero du successeur N 1:8
Donner la duree entre 6 et 8: 2

Donner le nombre des taches qui vient apres 7:1
Donner le numero du successeur N 1:8
Donner la duree entre 7 et 8: 3

```

FIGURE 5.3 – Les successeues et la durée de chaque tâche

La figure 5.4 représente les dates de debut au plus tôt de tableau de la figure 5.1

```

Donner le nombre des taches qui vient apres 7:1
Donner le numero du successeur N 1:8
Donner la duree entre 7 et 8: 3

date_plutot de 1: 0
date_plutot de 2: 4
date_plutot de 3: 3
date_plutot de 4: 6
date_plutot de 5: 4
date_plutot de 6: 10
date_plutot de 7: 9
date_plutot de 8: 12
date_plutot de 9: 13
Tapper une touche .../

```

FIGURE 5.4 – Les dates de debut au plus tôt

La figure 5.5 donne les dates de debut au plus tard de tableau de la figure 5.1 .

```

date_plutard de 8: 12
date_plutard de 7: 9
date_plutard de 6: 10
date_plutard de 5: 4
date_plutard de 4: 6
date_plutard de 3: 3
date_plutard de 2: 4
date_plutard de 1: 0
date_plutard de 0: 0
Tapper une touche ...

```

FIGURE 5.5 – Les dates de debut au plutard

La figure 5.6 montre les marges et le chemin critique de tableau de la figure 5.1 .

```
La marge de 0: 0
La marge de 1: 0
La marge de 2: 0
La marge de 3: 0
La marge de 4: 0
La marge de 5: 0
La marge de 6: 0
La marge de 7: 0
La marge de 8: 0
La marge de 9: 0
Tapper une touche
==> 1 3 5 7 8
```

FIGURE 5.6 – Calcul les marges et le chemin critique

# Conclusion Générale

L'objectif de ce travail consistait à faire une étude préalable d'un projet en le modélisant sous forme d'un problème d'ordonnancement sous un contexte des ressources rares et de prendre en compte aussi bien l'aspect temps .

Tout au long de ce travail ,nous sommes passer par donner les notions de bases de la théorie des graphes ,de consacrer une partie de ce mémoire à définir la complexité algorithmique .

On a réalisé une étude bibliographique de la gestion de projet ,en rattachant les généralités sur la gestion de projet , On a étudié avec détails le problème de l'ordonnancement sous ses différents aspects : modèles, méthodes et outils de calcul.

Pour la planification des tâches on a utilisé le langage c++ qui permet de faire la résolution de la méthode PERT.

# Bibliographie

- [1] Didier. Muller, *Introduction à la théorie des graphes*, City Editions, France 2007.
- [2] Claude. Berge, *Graphes et hypergraphes*, Dunod Editions, Paris 1970.
- [3] J.A.Bondy et U.S.R.Mutty ,*Livre de la théorie des graphes* ,traduit de l'anglais par F.Havet ,2008.
- [4] D.WEST,*Introduction to graph theory*,2nd,Prentice HALL,2001 .
- [5] PH.Lacomme,M.Prins,M.Sevaux,Eyrolles,2003
- [6] J.Michel Héлары,*Algorithmes des graphes* ,2004.
- [7] Mouhoub.N.*Algorithmes de construction de graphes dans les problèmes d'ordonnancement de projet*.Thèse de doctorat.Informatique.Université Ferhat Abbas-Sétif.2011.
- [8] J.Carlier P.Chretienne , *Problèmes d'ordonnancement, modélisation, complexité, algorithmes*, MASSON, ISBN 2-225-81275-6.
- [9] H.HOUARI , *Planification et Ordonnancement en temps réel d'un Job shop en utilisant l'intelligence Artificielle* ,Mémoire de Magister en Automatique Option :Modélisation et commande des systèmes ,Université de Tlemcen , Juillet 2012.
- [10] J.RAVAY ,*Ordonnancement sous incertitude* ,Université du Québec à Rimouski ,Avril 2008.
- [11] G. Vallet ,*Techniques Analyse de Projet*, Dunod 1995.
- [12] Daniel De Wolf , **Gestion de Projets** , Université du Littoral Dunkerke ,2007.
- [13] C. OLIVIER , *Planification de Projets*, ,Ecole Supérieure de Technologie Université du Québec 2002
- [14] ISSEMLELT. H KELLAL .K .*Les méthodes PERT et MPM pour l'ordonnancement des tâches d'un projet*.Mémoire de master 2. Université M'hammed Bougara Boumerdes.2016.

- [15] Daniel De Wolf , *Gestion de Projets* , Université du Littoral Dunkerke ,2007
- [16] Boujama.W Bouras.S, *Optimisation dans les réseaux*.Mémoire de Master 2.Université de béjaia.juin 2016.
- [17] ]B.Roy et M.Dibon , **l'ordonnancement par la méthode des potentiels** , février 1966.

## Résumé

Après avoir étudié la modélisation de l'ordonnancement de projet par le biais du diagramme de Gantt, la méthode MPM et la méthode PERT, on conclue que les managers de projet préfèrent travailler avec le graphe PERT malgré qu'il est difficile à réaliser, alors que le graphe des potentiels offre plus de simplicité.

Le calcul des durées dans les problèmes d'ordonnancement est une tâche très délicate puisqu'on n'a pas les moyens nécessaires de détection des dates de début et de fin de chaque activité vu leur caractère stochastique. PERT offre une technique de calcul qui est analysée, critiquée et suivie de quelques corrections.

Enfin, ce travail aborde le problème de graphe PERT minimal, qui est un problème NP-Complet. Elle présente une méthode nouvelle de construction qui est comparée à une méthode similaire, connue dans la littérature. Les résultats de cette comparaison montrent la supériorité de notre nouvelle technique .

**Mots clés :** Projet , Tâche , Graphe

## **Abstract**

After studying the project scheduling modeling through the Gantt chart, the MPM and PERT methods, we concluded that project managers prefer to work with the PERT graph although it is difficult to achieve, while the potentials graph offers more simplicity.

The duration calculations in scheduling problems are very delicate since we did not have the necessary mean to detect the starting and ending of each activity given their stochastic character. PERT provides a computational technique which, in this work is analyzed, criticized and followed by a few corrections.

Finally, this work addresses the problem of the minimal PERT graph, which is NP-hard. It presents a new heuristic of construction which is compared to a similar method known in the literature. The results of this comparison demonstrate the superiority of our new technique .

**key words** : Project,Activity,Graph