

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE



**UNIVERSITE MOHAMED BOUDIAF - M'SILA**  
**FACULTE DES MATHÉMATIQUES ET**  
**DE L'INFORMATIQUE**



**DEPARTEMENT D'INFORMATIQUE**

**MEMOIRE de fin d'étude**

**Présenté pour l'obtention du diplôme de MASTER**

**Domaine : Mathématiques et Informatique**

**Filière : Informatique**

**Spécialité : Systèmes d'Informations Avancés**

**Par: CHOUITER Kheira**

**SUJET**

**Le graphe AoA minimal**

**Soutenu publiquement le : / /2016 devant le jury composé de :**

.....	Université de M'sila	<b>Président</b>
<b>Dr. N.E.MOUHOUB</b>	Université de M'sila	<b>Rapporteur</b>
.....	Université de M'sila	<b>Examinateur</b>
.....	Université de M'sila	<b>Examinateur</b>

**Promotion : 2015 /2016**

## ***Remerciement***

Je tiens tout d'abord à remercier **ALLAH**, puis j'exprimer ma profonde reconnaissance à mon Directeur de recherche, **Dr. NASSER EDDINEMOUHOUB** pour sa patience, ses idées pertinentes qui ont permis l'avancement de mon mémoire. Mes remerciements s'adressent également **Dr. HEMMEK** pour ses conseils et ses remarques fort utiles qui ont permis de bien mener ce projet.

Finalement, j'adresserai mon dernier remerciement, à ma famille qui m'ont soutenue durant toutes mes années d'études et qui m'ont toujours encouragée à aller plus loin.

## *Dédicace*

*A l'âme de mon père ;*

*A ma mère, qui m'a réellement tout donné ;*

*Toutes mes sœurs et mes frères*

*Et toute ma famille*

*Et toute mes amies*

*Je dédis ce mémoire*

# Table des matières

Remerciements.....	.i
Dédicace.....	.ii
Table des Matières.....	.iii
Liste des tableaux.....	.iv
Liste des figures.....	.v
Introduction générale .....	1

## Chapitre 1

### Les principes de base de l'ordonnement

1. Introduction .....	4
2. La recherche opérationnelle.....	4
3. Les problèmes d'optimisation.....	5
3.1. Les problèmes d'optimisation combinatoires.....	5
3.2. Les problèmes d'optimisation combinatoires et les problèmes de décision..	5
4. La théorie de la complexité.....	6
4.1. Complexité d'un algorithme.....	6
4.2. L'algorithme efficace.....	7
4.3. Classe des problèmes.....	8
4.3.1. La classe P.....	8
4.3.2. La Classe NP.....	8
4.3.3. La classe NP-Complet.....	8
4.3.4. La classe NP-Difficile.....	8
5. La fonction ordonnancement.....	9
5.1. Définition.....	9
5.2. Les tâches.....	10
5.3. Les ressources.....	10
5.4. Les contraintes.....	10
6. Caractéristiques générales des ordonnancements.....	11

7. Méthodes de résolution des problèmes d'ordonnement.....	11
7.1. Les méthodes exactes .....	12
7.2. Les méthodes approchées.....	12
7.2.1. Les méthodes Heuristiques .....	12
7.2.2. Les méthodes Métaheuristiques .....	13
8. Conclusion.....	13

## **Chapitre 2**

### **L'ordonnement dans gestion du projet**

1. Introduction.....	14
2. Qu'est-ce qu'un projet?.....	14
3. Que la gestion de projet?.....	14
3.1. Caractéristiques des projets réussis.....	15
3.2. Organisation de l'équipe projet.....	15
3.3. Cycle de vie d'un projet.....	16
3.3.1. La planification du projet.....	16
3.3.2. Que constitue un plan de projet?.....	17
3.3.3. Le calendrier d'activités .....	17
3.3.4. Création d'un calendrier d'activités.....	18
3.4. Le triangle de projet .....	19
3.5. L'ordonnement dans la gestion de projet.....	21
4. Le problème central de l'ordonnement.....	21
4.1. Modélisation du problème central.....	23
4.1.1. Le diagramme de Gantt.....	23
a. Définition .....	23
b. Objectifs du diagramme de Gantt .....	24
4.1.2. Le graphe AoN.....	25
4.1.3. Le graphe AoA.....	29
4.1.3.1. Généralités.....	29
4.1.3.2. Présentation du AoA.....	30
4.1.3.3. Utilité de la méthode AoA.....	32

4.1.3.4. Dessin du graphe AoA.....	33
4.1.3.5. La tâche fictive dans le graphe AoA.....	36
4.2. Modularité.....	38
4.3. Comparaison.....	39
5. Résolution du problème central.....	40
5.1. Dates et marges associées à une tâche.....	41
5.2. Calcul des dates et marges.....	42
5.3. Le chemin critique.....	45
6. Conclusion.....	47

### **Chapitre 3**

#### **Le graphe AoA minimal**

1. Introduction.....	48
2. Etat de l'art.....	49
3. Algorithme de F. STERBOUL et al.....	52
3.1. Notations et définitions.....	52
3.2. Graphe arc-dual d'un graphe donné.....	53
3.3. Construction.....	53
3.4. Exemple.....	56
4. Algorithme de N.E.MOUHOUB et al.....	60
4.1. Notations et définitions.....	61
4.2. Construction.....	61
4.3. Exemple.....	68
5. Conclusion.....	72

### **Chapitre 4**

#### **Présentation d'application et étude comparative**

1. Introduction.....	74
2. Présentation les deux algorithmes .....	74
2.1 Implémentation l'algorithme de Mouhoub et al.....	74
2.2 Implémentation d'algorithme de Sterboul et al.....	78
3. Les outils et l'environnement de développement.....	79

3.1. La langage de programmation « JAVA».....	79
3.2. L'environnement NetBeans.....	80
3.3 Dpo4 Base de données .....	82
3.3.1. Base de données " db4o".....	82
3.3.2. Placement de db4o dans le monde .....	82
3.3.3. Avantage de db4o .....	83
4. Présentation d'application .....	84
5. Conclusion .....	87
conclusion générale .....	88
La bibliographie .....	

# Liste des tableaux

Nom de table	N° page
<b>Table 1.1</b> Tableau récapitulatif de classification des problèmes combinatoires.....	9
<b>Table 2.1</b> Table initiale d'ordonnement T.....	25
<b>Table 2.2</b> Table d'ordonnement T1.....	27
<b>Table 2.3</b> Table initiale d'ordonnement T2.....	33
<b>Table 2.4</b> Un sous-tableau des antériorités de c et d.....	37
<b>Table 2.5</b> La table d'ordonnement T3 et l'organisation des sommets en niveaux.....	37
<b>Table 2.6</b> Une table d'ordonnement T4.....	45
<b>Table 2.7</b> dates de début au plus tot et plus tard du table 2.6.....	45
<b>Table 3.1</b> Table d'ordonnement.....	54
<b>Table 3.2</b> les prédécesseurs de sommets.....	55
<b>Table 3.3</b> les successeurs de sommets.....	55
<b>Table 3.4</b> les classes $\bar{a}_i$ .....	55
<b>Table 3.5</b> .les classes $\bar{b}_j$ .....	55
<b>Table 3.6</b> les bons arcs.....	57
<b>Table 3.7</b> Table d'ordonnement de la Figure 3.15.....	69

## Liste des figures

Nom de figure	N° page
<b>Figure 2.1</b> Cycle de vie d'un projet.....	16
<b>Figure 2.2</b> Le triangle du projet.....	19
<b>Figure 2.3</b> Le diagramme de Gantt de la table T.....	25
<b>Figure 2.4</b> La tâche u précède la tâche v dans le graphe AoN.....	27
<b>Figure 2.5</b> Le graphe AoN les sommets réorganisés en niveaux.....	28
<b>Figure 2.6</b> Représentation du réseau AoA.....	30
<b>Figure 2.7</b> Représentation d'une étape.....	31
<b>Figure 2.8</b> Représentation d'une tâche.....	31
<b>Figure 2.9</b> La tâche u précède la tâche v dans le graphe AoA.....	31
<b>Figure 2.10</b> Le graphe AoA construit par ajout successif des tâches, les durées n'étant pas inscrites.....	36
<b>Figure 2.11 (a) et (b)</b> Problème de représentation dans le graphe AoA avec introduction de la tâche fictive f dans le nouveau graphe AoA.....	36
<b>Figure 2.12</b> Le graphe AoA de la <b>Table 2.5</b> avec un minimum de 5 tâches fictives.....	38
<b>Figure 2.13</b> La modularité dans le graphe AoA.....	39
<b>Figure 2.14</b> Calcul de $do_A$ dans le cas d'un ou plusieurs arcs .....	43
<b>Figure 2.15</b> Calcul de $da_A$ dans le cas d'un ou plusieurs arcs .....	43
<b>Figure 2.16</b> Les dates de début au plus tôt et au plus tard dans le graphe AoN .....	46
<b>Figure 2.17</b> Les dates de début au plus tôt et au plus tard dans le graphe AoA .....	46
<b>Figure 3.1</b> Un arc redondant (i, j).....	51
<b>Figure 3.2</b> les sommets $a_i$ et $a_{i+1}$ forment une seule classe.....	52
<b>Figure 3.3</b> les sommets $b_j$ et $b_{j+1}$ forment une seule classe.....	53
<b>Figure 3.4</b> Le graphe des AoN.....	54
<b>Figure 3.5</b> Le graphe $H_0$ .....	54
<b>Figure 3.6</b> Le graphe $H_1$ .....	56
<b>Figure 3.7</b> Le graphe $H_2$ .....	57
<b>Figure 3.8</b> Les sommets $b_i$ ayant les mêmes successeurs contractés en un sommet unique $b_1$ dans $G_{e1}$ .....	59
<b>Figure 3.9</b> Les sommets $a_i$ ont les mêmes prédécesseurs contractés en un Sommet unique $a_1$ dans $G_{e2}$ .....	60

<b>Figure 3.10</b> Un arc transmetteur de type 1 entre deux tâches i et j et sa Contraction.....	60
<b>Figure 3.11</b> Un arc transmetteur de type 2 entre deux tâches i et j et sa contraction.....	61
<b>Figure 3.12</b> Les sommets dont les successeurs sont inclus dans l'ensemble des successeurs d'un autre sommet dans $G_{e5}$ et la réduction des tâches fictives dans $G_{e6}$ .....	62
<b>Figure 3.13</b> Les sommets dont les prédécesseurs sont inclus dans l'ensemble des prédécesseurs d'un autre sommet dans $G_{e5}$ et la réduction des tâches fictives dans $G_{e6}$ .....	63
<b>Figure 3.14</b> Un biparti complet dans $G_{e7}$ et le sommet fictive sf dans $G_{e8}$ . ....	63
<b>Figure 3.15</b> Plan pour la construction du bâtiment industriel .....	64
<b>Figure 3.16 (a)</b> Le graphe AoN $G_v$ avec 14 sommets de la table d'ordonnancement <b>3.7</b> .....	66
<b>Figure 3.16 (b)</b> Le graphe AoA $G_{e1}$ de la <b>Figure 3.16 (a)</b> . $G_{e1}$ contient 29 tâches fictives et 28 sommets.....	66
<b>Figure 3.16 (c)</b> Le graphe AoA $G_{e3}$ de la Figure 16. (a) obtenu en appliquant la règle 1 ensuite la règle 2. $G_{e3}$ contient 23 tâches fictives et 23 sommets.....	67
<b>Figure 3.16 (d)</b> Le graphe AoA $G_{e5}$ de la Figure 16 (a) obtenu en appliquant la règle 3 ensuite la règle 4, avec 16 tâches fictives et 17 sommets.....	67
<b>Figure 3.17 (e)</b> Le graphe AoA minimal $G_{e8}$ avec le sommet fictif sf0 obtenu en appliquant les règles 5, 3 ensuite la règle 7. $G_{e8}$ contient 15 tâches fictives et 18 sommets.....	67
<b>Figure 4.1</b> NetBeans 8.1.....	68
<b>Figure 4.2</b> Db4o dossier jar de base de données.....	80
<b>Figure 4.3</b> Recherche un graphe.....	83
<b>Figure 4.4</b> Modifier un graphe.....	84
<b>Figure 4.5</b> Supprimer un graphe.....	84
<b>Figure 4.6</b> Fenêtre principal d'exécution des algorithmes des Mouhoun et Sterboul .....	85
<b>Figure 4.7</b> Mise à jour de base de données.....	86
<b>Figure 4.8</b> Fenêtre de comparaisons.....	86
<b>Figure 4.9</b> Rapport d'exécution .....	87

# INTRODUCTION GENERALE

La recherche opérationnelle est la discipline des mathématiques appliquées qui traite des questions d'utilisation optimale des ressources dans l'industrie et dans le secteur public. Depuis une dizaine d'années, le champ d'application de la RO s'est élargi à des domaines soit stratégiques comme le choix d'investir ou pas, le choix d'une implantation, le dimensionnement d'une flotte de véhicules ou d'un parc immobilier... ou opérationnels comme l'ordonnancement, la gestion de stock, l'affectation de moyens (humains ou matériels) à des tâches, les prévisions de ventes... .

La gestion de projet est une composante très importante de la communauté de recherche opérationnelle. De nombreux travaux traitent de l'ordonnancement et de la gestion de projet, mais aussi de logistique, de planification, et de problèmes d'emploi du temps.

Un problème d'ordonnancement consiste à organiser dans le temps la réalisation de tâches, compte tenu de contraintes temporelles (délais, contraintes d'enchaînement) et de contraintes portant sur la disponibilité des ressources requises. Alors il est défini par le planning d'exécution des tâches (« ordre » et « calendrier ») et d'allocation des ressources et vise à satisfaire un ou plusieurs objectifs.

Pour résoudre un problème d'ordonnancement il faut passer par l'étape de modélisation. La modélisation est une étape importante dans les domaines de recherche opérationnelle ; car elle est la conception d'un modèle. Selon son objectif et les moyens utilisés, elle est dite mathématique, géométrique, informatique...

Un modèle est une traduction de la réalité pour pouvoir appliquer des outils, des techniques et des théories mathématiques, puis en traduisant les résultats obtenus en prédictions ou opérations dans le monde réel.

Donc, pour résoudre un problème il faut les modéliser, par les vastes types de modélisation on cite la modélisation graphique qui peut représenter la structure, la connexion (la relation entre ses éléments), la modélisation graphique consiste à une méthode de pensée qui permet de modéliser une grande variété de problèmes sous une forme des sommets et des arcs. L'informaticien qui peut les faire le graphe d'une manière correcte. Donc la modélisation graphique permet la compréhension de problème, mais pas la résolution.

Alors, pour résoudre un problème il faut trouver une solution dans l'espace algorithmique. Plusieurs sont les outils de théorie de graphe pour décrire les algorithmes, les étudier, exprimer leurs qualités, pouvoir les comparer entre eux.

## **I. Contexte et problématique**

Plusieurs sont les ouvrages qui étudient le problème d'ordonnement, et les problèmes de gestion de projet par exemple on cite : Ecoto [12] présente une dizaine d'arguments pour la nécessité d'apprendre les techniques d'ordonnement de projet pour les leaders d'entreprises et d'administrations. Ces techniques et méthodes répondaient à des besoins précis :

- analyser le projet en profondeur, c'est-à-dire le décomposer en tâches ;
- mettre sur pied un plan d'exécution qui permet la réalisation du projet, avec des objectifs précis, tout en respectant un certain nombre de contraintes et en rationalisant l'utilisation des ressources;
- et enfin, contrôler le bon déroulement du projet.

Par conséquent, une méthode d'ordonnement est plus que nécessaire pour mener à terme le projet. Sterboul et Mouhoub étudient le problème de construction du graphe AoA minimal après avoir vu tous ces états on sélectionne notre problématique qui devient: quel est le meilleur algorithme entre les deux proposés par Mouhoub ou Sterboul? La proposition d'un algorithme qui compare entre les deux est notre objectif.

## **II. Contributions de ce mémoire**

Nous allons présenter une étude comparative entre deux algorithmes célèbres de Mouhoub et al et de Sterboul et al.

Un des inconvénients majeurs de la méthode AoA c'est l'existence des tâches fictives ajoutées au graphe. L'introduction de ces tâches permet de résoudre certaines situations et de lever des ambiguïtés. Elles ne mettent en jeu aucun moyen matériel ou financier mais elles sont la source unique d'existence d'une infinité de graphes AoA. C'est pour cette raison que les chercheurs se sont penchés sur le problème de réduction des tâches fictives pour apporter assez de clarté au graphe AoA. Ce domaine reste toujours un sujet d'actualité puisque sa complexité est *NP-difficile*. La contribution en ce point précis ne sort pas de cette logique et propose une technique de réduction.

### **III. Plan du mémoire**

Afin de présenter les résultats de notre travail, ce manuscrit s'organise de la manière suivante :

Une introduction générale suivie avec le premier chapitre sous-titre des principes de base de l'ordonnancement, dans un premier temps on donne quelques définitions sur l'optimisation des algorithmes, et les problèmes d'optimisation combinatoires et les problèmes de décisions de leur complexité. Nous abordons ensuite, l'étude de la fonction ordonnancement avec ses éléments de base ainsi que ses caractéristiques générales et clôturer par la présentation des méthodes de résolution d'ordonnancement.

Le chapitre 2 son titre est l'ordonnancement dans gestion de projet dans ce chapitre on représente le projet et la gestion de projet, les problèmes central de l'ordonnancement ou on présente les trois méthodes de représentation : Gantt, AoN, AoA, et la résolution de ce genre de problèmes.

Le chapitre 3 est consacré au graphe AoA minimal, dans ce chapitre on traite l'état de l'art concernant la construction du graphe AoA minimal. Aussi, on présente l'algorithme de Serboul et al puis on représente l'algorithme de Mouhoub et al avec un exemple de démonstration.

Le chapitre 3 qui prend le titre le graphe AoA minimal est réservé à une étude comparative entre le nouvel algorithme de Mouhoub et al et celui de Streboul et al.

Le dernier chapitre concernant l'application propose pour résoudre le problème de cette mémoire, à l'aide de langage de programmation Netbeans; le titre de ce chapitre est Présentation d'application et étude comparative.

Enfin on termine notre mémoire avec une conclusion générale de notre travail ou nous répond à notre problématique présenté dans l'introduction générale et présente notre contribution personnelle suivie des perspectives future qui tracent le chemin des travaux futurs.

# CHAPITRE 1

## LES PRINCIPES DE BASE DE

## L'ORDONNANCEMENT

### 1. Introduction

L'informatique permet d'avoir un outil d'aide à la décision et aussi aux planifications de la production. Cette planification est d'autant plus efficace que l'algorithme **d'ordonnancement** qu'elle utilise est performant. Le but étant d'atteindre un ordonnancement optimal qui répartit au mieux la charge du travail et tient compte des diverses contraintes de production.

Les problèmes d'ordonnancement sont très variés. On peut les rencontrer dans de très nombreux domaines : les systèmes industriels de production (activités des ateliers en gestion de production et problèmes de logistique), les systèmes informatiques (les tâches sont les programmes et les ressources sont les processeurs, la mémoire...), les systèmes administratifs (Gestion du personnel, emplois du temps,...), les systèmes de transport, la construction, ... etc. C'est pour cette raison qu'ils ont fait et continuent de faire l'objet de nombreux travaux de Recherche. [20]

### 2. La recherche opérationnelle

La **recherche opérationnelle** peut être définie comme l'ensemble des méthodes et techniques rationnelles orientées vers la recherche du meilleur choix dans la façon d'opérer en vue d'aboutir au résultat visé ou au meilleur résultat possible.

Elle fait partie des « **aides à la décision** » dans la mesure où elle propose des modèles conceptuels en vue d'analyser et de maîtriser des situations complexes pour permettre aux décideurs de comprendre et d'évaluer les enjeux et d'arbitrer ou de faire les choix les plus efficaces.

Ce domaine fait largement appel au raisonnement mathématique (logique, probabilités, analyse des données) et à la modélisation des processus. Il est fortement lié à l'ingénierie des systèmes, ainsi qu'au management du système d'information. [22]

### 3. Problèmes d'optimisation

Les graphes sont un outil de modélisation puissant et très intuitif. Ils sont naturellement utilisés pour représenter des réseaux de transport, de communication, et plus généralement des flux de matières ou d'informations. Ils sont également utilisés pour modéliser des problèmes dans lesquels des objets sont en relation, les sommets du graphe représentant ces objets et les arêtes (ou arcs si une orientation est considérée), les relations entre ces objets. [5]

Le problème d'optimisation considéré est modélisé par les outils de la théorie des graphes ensuite une solution est cherchée qui est généralement un algorithme.

Parmi les problèmes modélisés par la théorie des graphes, on cite : le problème d'affectation, de flot maximum ou minimum, de transport, de sac à dos, de voyageur de commerce, de localisation et d'ordonnancement etc. Ce dernier regroupe l'ordonnancement dans la gestion de projet et l'ordonnancement dans la gestion de la production. [23]

#### 3.1. Les problèmes d'optimisation combinatoires

Un problème d'optimisation combinatoire (*POC*) est un problème de minimisation ou de maximisation auquel on associe un ensemble d'instances.

Les problèmes d'optimisation combinatoire sont présents dans plusieurs domaines d'applications industrielles, économiques et scientifiques. On les retrouve dans **l'ordonnancement** de tâches dans un système de production. Ce sont, en général, des problèmes faciles à définir, mais difficiles à résoudre. [20]

#### 3.2. Problème d'optimisation combinatoire et Problème de décision

Un problème d'optimisation combinatoire est un problème qui consiste à chercher une meilleure solution parmi un ensemble de solutions réalisables.

Un problème de décision est un problème qui consiste à apporter une réponse "oui" ou "non" à une question.

Un problème d'optimisation combinatoire est au moins aussi difficile que le problème de reconnaissance associé. De plus, on peut généralement prouver que le problème de décision n'est pas plus facile que le problème d'optimisation combinatoire. En d'autres

termes, cela signifie qu'un problème d'optimisation combinatoire est souvent du même niveau de difficulté que le problème de décision associé. [27]

#### 4. Théorie de la complexité

L'expérience montre que certains problèmes sont plus faciles que d'autres à résoudre. Une théorie de la complexité a été développée et permet mathématiquement de classer les problèmes faciles et difficiles en deux classes : les classes P et NP.

On suppose que, pour chaque problème que l'on veut résoudre, l'on dispose d'une mesure de la taille du problème. Par exemple, on utilisera comme mesure le nombre de tâches lorsqu'il s'agit d'un problème d'ordonnement de  $n$  tâches. Le nombre d'opérations élémentaires effectuées par un algorithme est donc une fonction qui dépend de  $n$ . Cette fonction s'appelle la complexité. [25]

Déterminer la complexité d'un algorithme donné n'est pas toujours si simple.

Déterminer la complexité d'un problème apparaît moins ambitieux, puisqu'il s'agit de déterminer quelle est la plus faible complexité d'un algorithme de résolution. [4]

##### 4.1. Complexité d'un algorithme

On désigne par complexité d'un algorithme, le nombre d'opérations nécessaires à celui-ci pour s'exécuter.

Bien évidemment, ce nombre peut varier en fonction de ce que l'on appelle les données d'entrées, c'est-à-dire les paramètres que l'on donne à l'algorithme. Par exemple, un algorithme de tri d'éléments dans un tableau ne s'exécutera pas avec le même nombre d'opérations s'il y a 10 éléments ou s'il y en a 100.

Ainsi, on cherchera à estimer la complexité d'un algorithme en fonction de la taille des données entrées. Par exemple, dans le cas du tableau, on exprimera la complexité en fonction de la taille du tableau. Pour une matrice, ce serait en fonction de sa largeur et de sa hauteur. De plus, la nature même des données pour une même taille peut ne pas aboutir au même nombre d'opérations à l'exécution de l'algorithme. En effet, si le tableau est déjà trié, l'exécution de l'algorithme de tri risque d'être très rapide comparée au cas d'un tableau totalement en désordre. C'est pour cela que l'on estime le nombre d'opérations dans le pire des cas.

On mesure l'efficacité d'un algorithme par une expression mathématique qui indique le nombre d'opérations indispensables à l'exécution de l'algorithme en fonction de la taille des données en entrées tout en supposant le pire des cas. [27]

#### 4.2. L'algorithme efficace

La complexité d'un algorithme 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.**

Par exemple, considérons les algorithmes A, B et C. Leur complexité est les suivantes.

- **$C_a = 80n$ .**
- **$C_b = 10n^2$ .**
- **$C_c = n!$ .**

Avec 4 éléments, il faut respectivement 320, 160 et 24 opérations aux algorithmes A, B et C pour s'exécuter. Le plus efficace pour 4 éléments est donc C. Considérons maintenant 20 éléments, il faut respectivement 1600, 4000 et 2.4 1018 opérations pour réaliser l'exécution. On s'aperçoit tout de suite que l'algorithme C n'est 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. 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.**

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... Cependant, il faut bien comprendre que l'on s'intéresse au comportement général de l'algorithme face à des problèmes de grande taille. [27]

### 4.3. Les classes de Problèmes

Tout d'abord, on fait une distinction entre les problèmes décidables et les problèmes indécidables, les problèmes décidables sont ceux pour lesquels il existe au moins un algorithme pour les résoudre, Les problèmes indécidables sont ceux pour lesquels aucun algorithme, quel qu'il soit, n'a été trouvé pour les résoudre. [27]

#### 4.3.1. La classe P :

Un Problème est dit appartenant à la classe P s'il existe un algorithme polynomial pour le résoudre. On dit que les problèmes de la classe P sont faciles.

#### 4.3.2. La classe NP :

La classe des problèmes NP (NP pour Non détermination Polynomiale) est la classe des problèmes des décisions qui peuvent être résolus par une machine de Turing non déterministe en temps polynomial. Parmi la classe de problèmes NP, on distingue deux grandes classes:

- La classe des problèmes polynomiaux (la classe P) et la classe des problèmes NP-Complets (la classe NPC).

##### 4.3.2.1. La classe NP-Complet :

Un problème de décision est dit NP-Complet si tout problème de la classe NP peut se rendre polynomialement à lui.

##### 4.3.2.2. La classe NP-difficile :

Tous les problèmes NP-difficiles ne sont pas de difficulté identique. On rencontre par exemple des problèmes qui ne peuvent pas être résolus en temps polynomial avec un codage binaire, mais qui peuvent l'être avec un codage unaire, ces problèmes sont dits NP-difficiles au sens ordinaire ou simplement NP- difficiles, les algorithmes pour cette classe de problèmes sont appelés pseudo-polynomiaux, pour d'autres problèmes, on peut ne pas trouver d'algorithme polynomial, quelque soit le codage, ceux-là sont dits NP- difficiles au sens fort. [27]

<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>			

**Table 1.1** Tableau récapitulatif de classification des problèmes combinatoires

## 5. La fonction ordonnancement

### 5.1. Définition

Selon Carlier et al [3], « ordonnancer c'est programmer l'exécution d'une réalisation en attribuant des ressources aux tâches et en fixant leurs dates d'exécution. »

Un ordonnancement donc, selon [16] constitue une solution au problème d'ordonnancement. Il décrit l'exécution des tâches et l'allocation des ressources au cours du temps et vise à satisfaire un ou plusieurs objectifs.

L'ordonnancement apparaît dans tous les domaines de l'économie: l'informatique, la construction, l'industrie et l'administration.

La théorie de l'ordonnancement est une branche de la recherche opérationnelle qui s'intéresse au calcul de dates d'exécution optimales de tâches. Pour cela, il est très souvent nécessaire d'affecter en même temps les ressources nécessaires à l'exécution de ces tâches. Un problème d'ordonnancement peut être considéré comme un sous-problème de planification dans lequel il s'agit de décider de l'exécution opérationnelle des tâches planifiées. [35]

Le mariage des deux approches n'est que bénéfique pour la résolution de ce genre de problèmes.

Les paragraphes suivants précisent ces notions de tâche, ressource, objectif,... et introduisent quelques notations. [23]

## **5.2. Les tâches**

Une tâche est une entité élémentaire de travail localisée dans le temps par une date de début et une date de fin d'exécution et qui consomme des ressources avec des quantités déterminées. Un coût (ou poids) est attribué à une tâche pour estimer sa priorité, son degré d'urgence, ou son coût d'immobilisation dans le système.

Une tâche qui peut être exécutée par morceaux est appelée tâche morcelable, le problème constitué de tâches morcelables est appelé problème préemptif. Si une tâche, une fois qu'elle démarre l'exécution, ne peut pas être interrompue, on dit que le problème est non préemptif. [26]

## **5.3. Les ressources**

La ressource est un moyen technique ou humain destiné à être utilisé pour la réalisation d'une tâche et disponible en quantité limitée.

Plusieurs types de ressources sont à distinguer :

Une ressource est renouvelable si après avoir été allouée à une ou plusieurs tâches, elle est à nouveau disponible en même quantité (les hommes, les machines, l'équipement en général), la quantité de ressource utilisable à chaque instant est limitée. [32]

## **5.4. Les contraintes**

Les contraintes expriment des restrictions sur les valeurs que peuvent prendre certaines variables. On distingue deux types de contraintes : les contraintes temporelles et les contraintes de ressources.

### **5.4.1. Les contraintes temporelles**

Les contraintes temporelles comprennent les contraintes de temps alloué, qui correspondent généralement aux impératifs liés aux tâches (délais de livraisons, disponibilité des approvisionnements) ou encore à la durée totale d'un ordonnancement.

Elles comprennent les contraintes d'antériorité ou de précédence qui correspondent à des contraintes de cohérence technologique qui positionnent les tâches les unes par rapport aux autres. Elles comprennent aussi les contraintes de calendrier qui correspondent, par exemple, aux plages horaires de travail, etc.

#### 5.4.2. Les contraintes de ressources :

Les contraintes de ressources, quant à elles, traduisent la disponibilité des ressources et le fait qu'elle soit en quantité limitée. Deux types de contraintes de ressources, liées à la nature cumulative ou disjonctive des ressources, peuvent alors être distingués.

Les ressources disjonctives ne peuvent être utilisées que par une tâche à la fois. Les ressources cumulatives, quant à elles, peuvent être utilisées par plusieurs tâches simultanément, comme dans le cas d'un ensemble de ressources. [32]

## 6. Caractéristiques générales des ordonnancements

- ✓ Ordonnement admissible : il respecte toutes les contraintes du problème, à savoir, les dates limites, précédences, limitation des ressources, ...etc.
- ✓ Ordonnement actif : dans un ordonnancement actif, aucune tâche ne peut commencer plutôt sans reporter le début d'une autre.
- ✓ Ordonnement semi-actif : dans un ordonnancement semi actif, on ne peut avancer une tâche sans modifier la séquence sur la ressource.
- ✓ Ordonnement sans retard : dans un ordonnancement sans retard, on ne doit pas retarder l'exécution d'une tâche, si celle-ci est en attente et si la ressource est disponible. [8]

## 7. Méthodes de résolution des problèmes d'ordonnement

Les méthodes d'optimisation peuvent être réparties en deux grandes classes de méthodes pour la résolution des problèmes :

- Les méthodes exactes.
- Les méthodes approchées.

### 7.1. Les méthodes exactes

Le principe des méthodes exactes consiste à rechercher, souvent de manière implicite, une solution, la meilleure solution ou l'ensemble des solutions d'un problème. L'optimisation exacte concerne toutes les méthodes permettant d'obtenir un résultat dont on sait qu'il est optimal à un problème précis. Cela va des méthodes du simplexe aux méthodes de Lagrangien en passant par la programmation dynamique. On peut classer les méthodes exactes en quatre grandes classes :

- La programmation dynamique.
- La programmation linéaire continue ou en nombres entiers.
- La programmation non linéaire avec ou sans contraintes.
- Les méthodes de recherche arborescente (Branch & Bound). [17]

### 7.2. Les méthodes approchées

Les méthodes approchées fournissent une solution approchée au problème traité. Elles sont en général conçues de manière à ce que la solution obtenue puisse être proche de la valeur optimale : de telles méthodes permettent d'obtenir des bornes inférieures ou supérieures de la valeur optimale telles que :

- Méthodes Heuristiques.
- Méthodes Méta heuristiques.

#### 7.2.1. Les méthodes Heuristiques :

Une heuristique est plutôt une méthode qui cherche (une stratégie) sans garantir le résultat. Destinée à un problème spécifique, le temps de calcul est raisonnable sans garantir la faisabilité ou l'optimalité. [1]

En recherche opérationnelle, les heuristiques sont des règles empiriques simples qui ne sont pas basées sur l'analyse scientifique (différents algorithmes). Elles sont basées sur l'expérience, les résultats déjà obtenus et sur l'analogie pour optimiser les recherches suivantes. Généralement, on n'obtient pas la solution optimale mais une solution approchée. Parmi les heuristiques, on peut citer l'algorithme glouton). [13]

### 7.2.2. Les méthodes Métaheuristiques :

Une métaheuristique est constituée d'un ensemble de concepts fondamentaux (par exemple, la liste taboue et les mécanismes d'intensification et de diversification pour la métaheuristique tabou), qui permettent d'aider à la conception de méthodes heuristiques pour un problème d'optimisation. Ainsi les métaheuristiques sont adaptables et applicables à une large classe de problèmes. [26]

Les métaheuristiques se subdivisent en deux sous-classes :

- Les méthodes de voisinage.
- Les algorithmes évolutionnaire.

## 8. Conclusion

L'ordonnement de projet consiste à placer dans le temps les tâches d'un projet en respectant ses contraintes. Les plus répandues des techniques d'ordonnement en gestion de projet sont basées sur une modélisation du problème du type AoN ou AoA. Dans ce cadre là, ces techniques visent à donner un plan pour l'exécution du projet, c'est-à-dire trouver une affectation complète des dates de début des tâches.

La majorité des problèmes d'ordonnement sont NP-complets, ça veut dire que, dans la pratique, la complexité croît exponentiellement avec le nombre de tâches et de ressources. Il n'est pas donc envisageable de résoudre, de tels problèmes avec les méthodes exactes. C'est pour cela qu'il faut développer des heuristiques dont l'objectif est de fournir des solutions aussi proches que possible de la solution exacte en un temps raisonnable.

## CHAPITRE 2

# L'ORDONNANCEMENT DANS GESTION DE PROJET

### 1. Introduction

L'ordonnancement dans un projet consiste à placer dans le temps les tâches du projet en respectant ses contraintes. Les plus répandues des techniques d'ordonnancement en gestion de projet sont basées sur une modélisation du problème du type potentiel-tâches (AoN) ou potentiel- étapes (AoA).

### 2. Qu'est-ce qu'un projet?

On parle tous de **projets** dans la vie de tous les jours : nos projets de vacances, projets de carrière, projets d'avoir des enfants... Le terme projet est donc un terme du vocabulaire courant, auquel on associe une signification relativement claire et précise : c'est un ensemble d'actions que nous souhaitons entreprendre, pour atteindre un but (devenir parent, embrasser une nouvelle carrière...). En ce sens, le projet est bien « le brouillon de l'avenir » : une ébauche, mais pas encore une réalisation. [30]

### 3. Que la gestion de projet?

C'est l'utilisation d'un savoir, d'habiletés, d'outils et de techniques dans le cadre des activités d'un projet, en vue de satisfaire ou de dépasser les exigences et les attentes des parties prenantes à l'égard de ce projet. Le gestionnaire de projet, parfois appelé coordonnateur ou chef de projet, en administre les détails, au jour le jour. Il s'agit là d'un défi constant qui demande une compréhension du contexte plus général du projet et la capacité de concilier des exigences contradictoires telles que :

- les ressources disponibles et les attentes.
- les différentes priorités des parties prenantes.
- les besoins définis et la portée du projet.
- la qualité et la quantité. [23]

### 3.1. Caractéristiques des projets réussis

Un projet réussi possède les particularités suivantes :

- **des objectifs précis** : Objectifs définis clairement dès le départ.
- **Un plan de projet bien établi** : Un plan conçu avec minutie qui permet à chaque participant de comprendre et de contribuer au projet. Il précise les responsabilités de chacun et évalue combien d'argent, de personnes, de matériel et de temps sont nécessaires à l'achèvement du projet. Ensuite, il sert d'outil de suivi et permet d'adopter des mesures correctives tôt dans le processus si les choses tournent mal.
- **La communication** : Le projet repose sur la collaboration et la communication entre toutes les personnes qui y prennent part.
- **Une envergure maîtrisée** : Tout au long du projet, il importe de porter attention aux priorités, avec une perte minimale de temps et de concentration.
- **Le soutien des intervenants** : Il importe que toutes les parties prenantes accordent leur soutien pour toute la durée du projet de façon à atteindre les objectifs. [23]

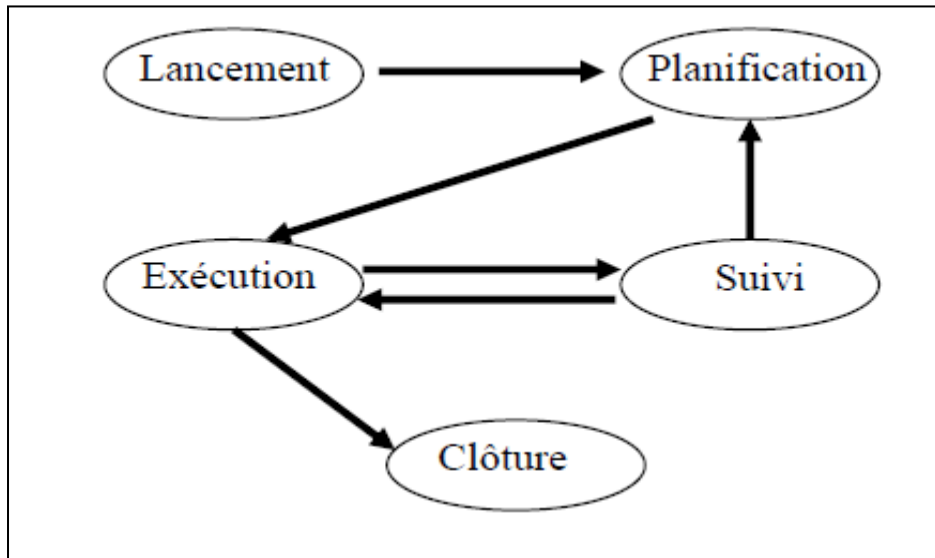
### 3.2. Organisation de l'équipe projet

Les acteurs de l'équipe projet SI sont les suivants :

- **la MOA** : C'est la maîtrise d'ouvrage. Elle doit décrire le besoin dans un document, souvent nommé CDC fonctionnel (Cahier Des Charges fonctionnel) ou spécifications fonctionnelles. Dans le cas d'un projet à consonance informatique, la MOA est aussi chargée de préparer des cas de tests fonctionnels pour vérifier que les développements/paramétrages effectués par la MOE fonctionnent.
- **la MOE** : C'est la maîtrise d'œuvre, qui prend connaissance du besoin exprimé et qui tâche d'y répondre informatiquement en rédigeant un dossier de réponse au besoin, nommé parfois CDC technique (Cahier Des Charges technique) ou dossier de paramétrage ou dossier de conception général ou encore, étude technique. La MOE se charge aussi de faire les développements/paramétrages nécessaires. [31]

### 3.3. Cycle de vie d'un projet

Les différentes phases du cycle de vie d'un projet sont : Lancement, planification, exécution, suivi et clôture comme illustré dans la figure suivante :



**Figure 2.1** Cycle de vie d'un projet [23]

#### 3.3.1. La planification du projet :

##### a. Consistance :

La planification d'un projet consiste à :

1. Découper le projet en phases.
2. Découper les phases en tâches.
3. Définir la logique d'enchaînement des tâches.
4. Analyser les résultats (délai final, chemin critique, les marges,...)
5. Optimiser le planning, en changeant certains enchaînements logiques et/ou en modifiant la durée de certaines tâches.
6. Editer le planning sous une forme temporelle claire et bien adaptée aux divers utilisateurs.

b. Objectif :

Les objectifs principaux de la planification des projets sont les suivants:

- minimiser la durée d'exécution totale du projet.
- minimiser le coût total du projet.
- gestion optimale des ressources. [19]

3.3.2. Que constitue un plan de projet?

Pour qu'il soit d'une utilité maximum, le plan de projet doit être pertinent compréhensible et tenir compte de l'importance et de la complexité du projet unique.

Le plan de projet devrait être remis au gestionnaire de projet, au promoteur, à chaque partenaire et à tous les principaux membres du personnel du projet. Il s'agit d'un outil de grande valeur qui peut permettre d'éviter la confusion quant à la portée du projet et les malentendus sur les responsabilités, les échéanciers ou la gestion des ressources.

Le plan de projet devrait inclure les éléments suivants :

- le mandat de projet.
- le calendrier d'activités.
- l'horaire de travail.
- la matrice de responsabilités.
- le budget de plan de projet.
- les étapes importantes, avec les dates cibles.
- la stratégie de gestion du risque. [23]

3.3.3 Le calendrier d'activités

Le calendrier d'activités est l'un des éléments les plus importants du coffre à outils du gestionnaire de projet. En permettant de fractionner un projet et de répartir toutes les tâches nécessaires à son achèvement, il :

- offre une vision précise de l'envergure du projet.
- permet de savoir précisément ce qui est terminé et ce qui reste à faire.
- permet de suivre de près le travail, les échéances et les coûts associés à chaque tâche.
- permet de confier la responsabilité de tâches précises aux membres de l'équipe.
- permet aux membres de l'équipe de comprendre leur rôle dans l'ensemble du tableau.

### 3.3.4. Création d'un calendrier d'activités

Le plan d'action mis au point fractionne le projet en ses activités composantes. Pour la phase initiale, la description des activités se fait sur une base mensuelle. Pour la phase pilote, elles sont présentées sur une base trimestrielle. Certaines sont suffisamment restreintes pour être gérées telles quelles, d'autres doivent être fractionnées de nouveau en éléments logiques.

Ces éléments de travail plus restreints, les tâches seront confiées à des personnes et doivent être définies avec assez de précision pour permettre le suivi et la gestion du rendement sans toutefois être restreintes au point d'obliger à consacrer un temps indu aux détails. A titre indicatif, il n'est pas utile de définir des tâches qui nécessitent moins d'une demi-journée de travail durant le cycle de vie du projet.

Chaque catégorie devrait comporter comme dernière tâche l'approbation du travail de l'étape. Lorsque tous les éléments seront terminés, le chef de projet indique au calendrier d'activités que la tâche sommaire est accomplie. Dans l'intervalle, il saura exactement quelles sont les étapes qui restent à faire. [23]

Selon [15], les tâches, suivant leur disposition dans le réseau, peuvent être:

Successives, simultanées ou convergentes. Les tâches sont *successives* lorsqu'elles se déroulent les unes après les autres. La tâche suivante ne peut démarrer que si la tâche précédente est terminée. Les tâches i et j sont *simultanées* si elles commencent en même temps. Les tâches i et j sont *convergentes* si elles précèdent la même tâche k.

#### a. Etablir la liste des tâches :

Au départ, il faut dresser la liste exhaustive des tâches à exécuter. Elle peut aller d'une dizaine à quelques centaines de tâches. Les projets importants comportent parfois plusieurs milliers de tâches.

#### b. Codifier les tâches :

Pour passer à l'étape d'examen des contraintes, il est nécessaire de repérer et de codifier les tâches. Ceci va faciliter la construction du réseau.

Au début du projet, il est recommandé d'avoir recours à des codes simples. En réalité, aucune règle ne peut être édictée, mais une codification faisant apparaître domaine et

nature de la tâche est très appropriée. Dans ce qui suit, on utilise les caractères A, B, C, etc., ou 1, 2, 3, etc. pour codifier les tâches. [23]

**c. Quel est le bon nombre de tâches à définir ?**

Ce nombre est fonction de l'ampleur du projet, mais avec un même nombre d'activités on peut planifier :

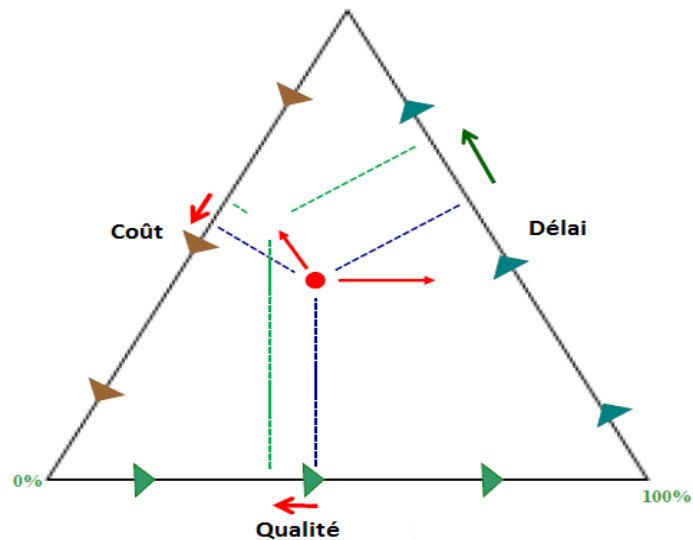
- La construction d'une usine (4 à 5 années) où changer les outils d'une machine (4 à 5 heures).
- La préparation sur 30 ans de sa retraite ou l'assemblage automatique de quelques composants en 30 secondes.

Diviser beaucoup conduit à obtenir un scénario détaillé et à repérer toutes les contraintes, mais rend difficile le tracé du graphe correspondant et le suivi de l'enchaînement par la suite. On divise peu, cela produit les effets inverses. Il faut chercher alors, un équilibre entre ces deux problèmes.

### **3.4. Le triangle du projet**

Les principes de base de la gestion de projet sont représentés par le triangle du projet, un symbole rendu populaire par Harold Kerzner. Il est caractérisé par 3 objectifs liés et antagonistes :

- Le délai : il s'agit du temps nécessaire pour achever le projet tel qu'il est décrit dans les prévisions.
- Le coût : le coût du projet est basé sur les coûts des ressources, c'est-à-dire le personnel, l'équipement et les matériels nécessaires à la réalisation des tâches.
- La qualité : il s'agit des objectifs et des tâches du projet ainsi que du travail nécessaire pour atteindre ces objectifs. [19]



**Figure 2.2** Le triangle du projet

Ces trois points sont interdépendants et doivent être pris en compte soigneusement. Un projet peut se présenter comme un point dans ce repère (point rouge), on peut améliorer une composante mais avec une détérioration des deux autres, il faut donc comprendre qu'il existe 3 possibilités:

- Rapide et pas cher  $\Rightarrow$  Mauvaise qualité : C'est ce que demandent beaucoup de clients, sans se rendre compte qu'un projet vite fait et à moindre coût aura forcément des lacunes. Cela peut être satisfaisant pour un prototype qui doit valider un concept.
- Rapide et de bonne qualité  $\Rightarrow$  Cher : Si le client peut se le permettre, c'est la solution parfaite. Un projet très important sera traité de manière prioritaire sur les autres, se verra affecter plus de moyens humains et techniques. Et donc forcément, cela a un coût important.
- Bonne qualité et pas cher  $\Rightarrow$  Lent : Un projet bien fait, mais pas cher. Il va prendre du temps à être réalisé. Pour diminuer les coûts, sa priorité est plus faible, et "on y travaille quand on a du temps". Pareil pour les ressources techniques, qui sont disponibles d'abord pour les autres projets. [19]

### **3.5. L'ordonnancement dans la gestion de projet**

L'ordonnancement se situe exactement dans la phase planification. Il réalise le suivi opérationnel du projet : gestion de ressources, suivi de l'avancement, lancement des activités.

Techniquement, ordonnancer un projet consiste à programmer dans le temps l'exécution des tâches, tout en respectant les contraintes de manière à optimiser les critères de performance retenus. C'est plus particulièrement à ce stade qu'interviennent les techniques d'ordonnancement de projet présentées dans ce qui suit. [23]

## **4. Le problème central de l'ordonnancement**

### **a. Définition :**

Les problèmes d'ordonnancement se rencontrent dans différents domaines, dès lors qu'il est nécessaire d'organiser et/ou de répartir le travail entre plusieurs entités.

Plusieurs définitions ont été proposées pour le problème d'ordonnancement, on en cite :

"Ordonnancer, c'est programmer l'exécution d'une réalisation en attribuant des ressources aux tâches et en fixant leurs dates d'exécution. Les différentes données d'un problème d'ordonnancement sont les tâches, les contraintes potentielles, les ressources et la fonction économique".

Eric Pinson, le définit comme étant "L'organisation dans le temps de l'exécution d'un projet". Pour Norman Sadeh, c'est "l'allocation dans le temps de ressources permettant l'exécution d'un ensemble de tâches".

Pour Gotha, c'est "programmer des exécutions des tâches en leur allouant les ressources requises et en fixant leur dates de début".

A partir de toutes ces définitions, on peut dire que l'ordonnancement, décrit l'exécution de tâches et l'affectation de ressources au cours du temps compte tenu de contraintes et de manière à satisfaire des objectifs.

L'objet d'un problème d'ordonnancement est de faciliter la mise en œuvre et de guider l'exécution d'un ensemble complexe de tâches (programme de recherche ou de production,

lancement d'un produit, construction d'un édifice ...). Parmi les méthodes les plus connues, on peut citer la méthode AoA qui sera détaillée ultérieurement. [21]

### **b. Contexte**

Les retards apportés aux réalisations de projets et aux livraisons ponctuelles aux clients sont dus en général à :

- la mauvaise conception du produit à fabriquer.
- la mauvaise gestion des stocks.
- la fixation arbitraire d'un calendrier de fin des travaux sans rapport ni avec l'évolution réelle des différentes tâches ni avec les moyens dont on dispose effectivement.
- le manque de coordination entre les responsables des opérations concernant l'ordre de passage des différentes tâches et leur fin, sont autant de problèmes que les entreprises doivent affronter aujourd'hui.

Afin de minimiser ces retards, l'organisation des tâches devient indispensable et fait intervenir des notions très particulières. [21]

### **c. Les objectifs**

Résoudre un problème d'ordonnancement c'est avant tout choisir ce que l'on veut optimiser. Selon le critère que l'on cherche à minimiser ou maximiser, on peut choisir entre deux grands types de stratégies, visant respectivement à l'optimalité des solutions, ou plus simplement à leur admissibilité. Comme critères, on notera particulièrement ceux :

- liés au temps
  - Le temps total d'exécution ou le temps moyen d'achèvement d'un ensemble de tâches.
  - Le stock d'en-cours de traitement.
  - Différents retards (maximum, moyen, somme, nombre, etc.) ou avances par rapport aux dates limites fixées.
- liés aux ressources
  - La quantité totale ou pondérée de ressources nécessaires pour réaliser un ensemble de tâches.
  - la charge de chaque ressource.
- liés à une énergie ou un débit.

- liés aux couts de lancement, de production, de transport, etc., mais aussi aux revenus, aux retours d'investissements. [21]

#### 4.1. Modélisation du problème central

Il existe actuellement plusieurs méthodes de modélisation du problème central parmi ces méthodes on cité: le diagramme de Gantt, la méthode AoN et la méthode AoA. Ces deux dernières utilisent comme outil de modélisation la théorie des graphes ; et plus particulièrement le réseau.

On note que dans la table initiale des données, les tâches A et B n'ont aucune antériorité (Les tirets de la dernière colonne) et que les tâches I et J n'ont aucune postériorité (I et J n'apparaissent pas en dernière colonne). [23]

Pour les besoins algorithmiques, on complète cette table pare une tâche  $\alpha$  (appelée source ou racine) de durée 0 précédant A et B et une tâche  $\omega$  (appelée puits ou anti-racine) de durée 0 succédant à I et J. On déduit la deuxième table.

Nous définissons alors *un graphe conjonctif*  $G = (X, U)$ , qui est valué ayant une racine  $\alpha$  et un puits  $\omega$  tel qu'il existe un chemin de valeur positive entre  $\alpha$  et tous les autres sommets, et un chemin de valeur positive entre tout sommet différent du puits  $\omega$  et  $\omega$ .

Une condition nécessaire et suffisante pour appliquer l'une des méthodes de modélisation des problèmes d'ordonnancement sur un graphe conjonctif est que ce graphe n'ait pas de *circuit* et les valeurs sur les arcs (les  $t_i$ ) sont de valeurs positives ou nulles. De part la nature du problème, une tâche ne peut se succéder à elle-même, donc le graphe ne contient pas de boucles. [23]

##### 4.1.1. Le diagramme de Gantt

###### a. Définition :

Le diagramme de Gantt est une technique très utilisée de construction de planning conçue par Henry L. Gantt (en 1917). Cette technique permet de modéliser les taches d'un projet sous forme de barre sur une échelle chronologique et de représenter leur indépendance. Il permet en outre d'associer les responsabilités à chacune des tâches. [33]

b. Objectifs du diagramme de Gantt

- Visualiser les tâches à réaliser et les éventuels dépassements
- Visualiser le qui fait quoi et quand (avec des possibilités de tri)
- Visualiser l'enchaînement des tâches dans le temps
- Suivre le degré d'avancement des travaux. [33]

On introduit cette technique à l'aide d'un exemple du problème de réalisation d'une construction tiré de [6]:

Code	Désignation	Durées	Antériorités immédiates
A	<b>Etablissement des plans</b>	<b>4</b>	-
B	<b>Terrassement</b>	<b>5</b>	-
C	<b>Creuser les fonds de piliers</b>	<b>6</b>	-
D	<b>Faire les fonds de piliers</b>	<b>3</b>	<b>A, B, C</b>
E	<b>Placement arrivée et évacuation assainissement</b>	<b>7</b>	<b>A, B, C</b>
F	<b>Dalle</b>	<b>5</b>	<b>A, B, C</b>
G	<b>Cloisonnage</b>	<b>8</b>	<b>D, E</b>
H	<b>Carrelage</b>	<b>6</b>	<b>D, E</b>
I	<b>Plomberie</b>	<b>5</b>	<b>F</b>
J	<b>Installation chauffage</b>	<b>3</b>	<b>F</b>
K	<b>Electricité</b>	<b>5</b>	<b>G, H, I, J</b>
L	<b>Pose faïence</b>	<b>2</b>	<b>G, H, I, J</b>
M	<b>Boiserie</b>	<b>1</b>	<b>K, L</b>
N	<b>Peinture</b>	<b>4</b>	<b>K, L</b>

**Table 2.1** Table initiale d'ordonnancement T.

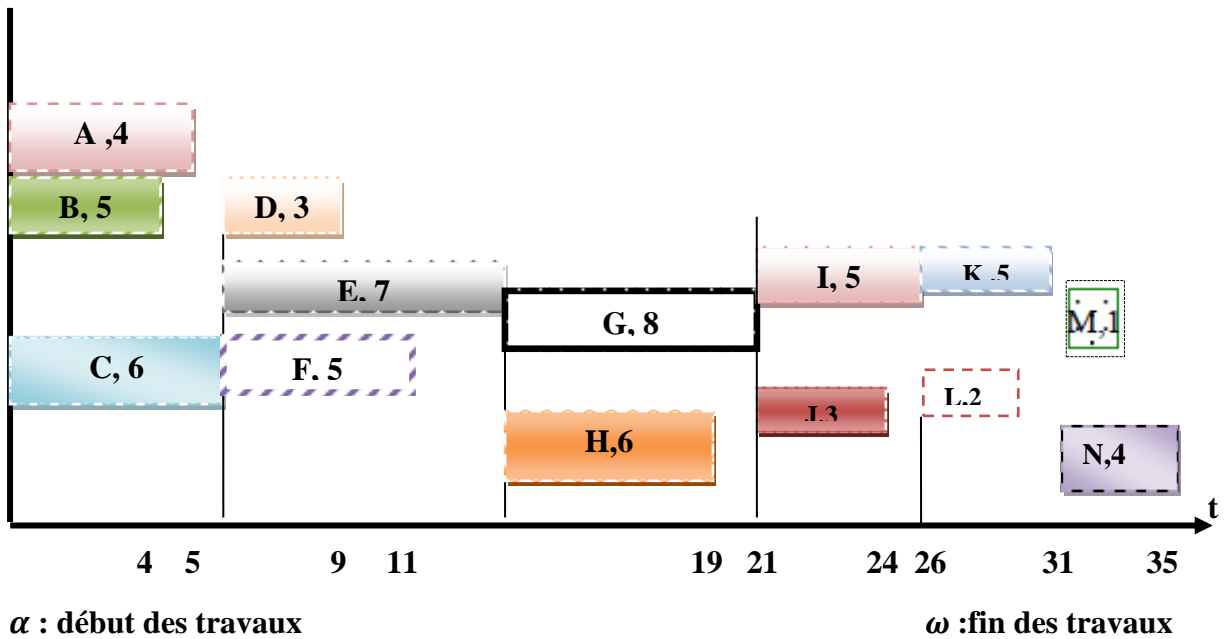


Figure 2.3 Le diagramme de Gantt de la table T

Il existe actuellement d'autres formes du diagramme de Gantt utilisant les sous forme de Tableau (**Table 2.1**), mais le principe reste toujours celui qu'on a présenté.

Le diagramme de Gantt est donc très utile pour « visualiser » le projet ; cependant si ce dernier comporte un grand nombre de tâches, les relations de succession n'apparaissent pas de manière évidente. Actuellement, ce diagramme est surtout un outil complémentaire de contrôle du déroulement du projet. [23]

#### 4.1.2. Le graphe AoN :

La Méthode AoN est une méthode d'ordonnancement basée sur la théorie des graphes, et visant à optimiser la planification des tâches d'un projet. Semblable au AoA , les principales différences entre les deux méthodes reposent essentiellement dans la construction du graphe. Elle a été développée par le chercheur français Bernard Roy, en 1958. [32]

a. Les buts de la méthode AoN :

Quelques fois préférée à la méthode AoA , La AoN est jugée beaucoup plus souple d'utilisation. Elle permet de :

- Déterminer la durée optimale nécessaire pour réaliser un projet dans les meilleurs délais.
- Définir les dates de début au plus tôt et au plus tard des tâches.
- Calculer les marges des différentes tâches (marge de manœuvre pour l'équipe de pilotage du projet).
- Identifier les tâches qui ne doivent souffrir d'aucun retard sous peine de retarder l'ensemble du projet (tâches critiques).
- Etudier les coûts de réalisation de chaque tâche et le coût global du projet.
- Effectuer le suivi du projet afin de détecter le plus tôt possible tout retard et appliquer à temps, des actions correctives. [32]

b. La construction d'un graphe AoN :

Le graph AoN se présente tel qu'il suit :

- Chaque tâche est représentée par un sommet.
- Les arcs entre les sommets traduisent uniquement les relations d'antériorité des tâches.
- la représentation des relations d'antériorité d'une tâche partageant avec une autre certains de ses antécédents ne nécessite pas, comme dans le AoA, le recours à des "tâches fictives".
- chaque tâche (ou sommet) est renseignée sur la date à laquelle elle peut commencer au plus tôt (date de début au plus tôt) et terminer au plus tard (date de fin au plus tard) pour respecter le délai optimal de réalisation du projet.
- A chaque arc est associée une valeur numérique qui représente soit une durée d'opération, soit un délai.
- La longueur des arcs n'est pas proportionnelle à cette durée.

- le graphe commence et termine sur 2 sommets, respectivement appelés « Début » et « Fin » symbolisant les début et fin des opérations. Ces deux sommets ne correspondent pas une tâche.
- Le graphe se lit de gauche à droite (du sommet "DÉBUT" à celui de "FIN"). [32]

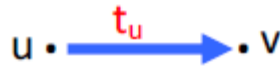
Nous détaillerons cette méthode au moyen d'un exemple (construction d'un pavillon) les travaux à réaliser, leur durée, et leurs précédents obligatoires (les autres travaux qui devront impérativement être achevés pour que la tâche, considérée puisse commencer-typiquement on ne pourra débiter la couverture qu'après en avoir terminé avec la charpente, etc.), sont résumés dans le tableau (voir **Table 2.2**) :

Tâche	Désignation	Durée	Ant.
A	<b>Maçonnerie</b>	<b>7</b>	-
B	<b>Charpente</b>	<b>3</b>	<b>A</b>
C	<b>Couverture</b>	<b>1</b>	<b>B</b>
D	<b>Plomberie électricité</b>	<b>8</b>	<b>A</b>
E	<b>Revêtements extérieurs</b>	<b>2</b>	<b>C, D</b>
F	<b>Menuiseries</b>	<b>1</b>	<b>C, D</b>
G	<b>Aménagement jardin</b>	<b>1</b>	<b>C, D</b>
H	<b>Plâtres</b>	<b>2</b>	<b>F</b>
I	<b>Peinture</b>	<b>2</b>	<b>H</b>
J	<b>Finitions</b>	<b>1</b>	<b>E, G, I</b>

**Table 2.2** Table d'ordonnement T1

Dans le graphe AoN Les tâches sont symbolisées par des sommets auxquels on donne le même code, 2 sommets u et v sont reliés par un arc de u vers v si et seulement si la tâche u précède la tâche v.

On porte en suite sur chaque arc incident extérieurement à un sommet u la durée de la tâche u correspondante, de sorte que les valeurs sur les arcs de même extrémité initiale soient égales [23].

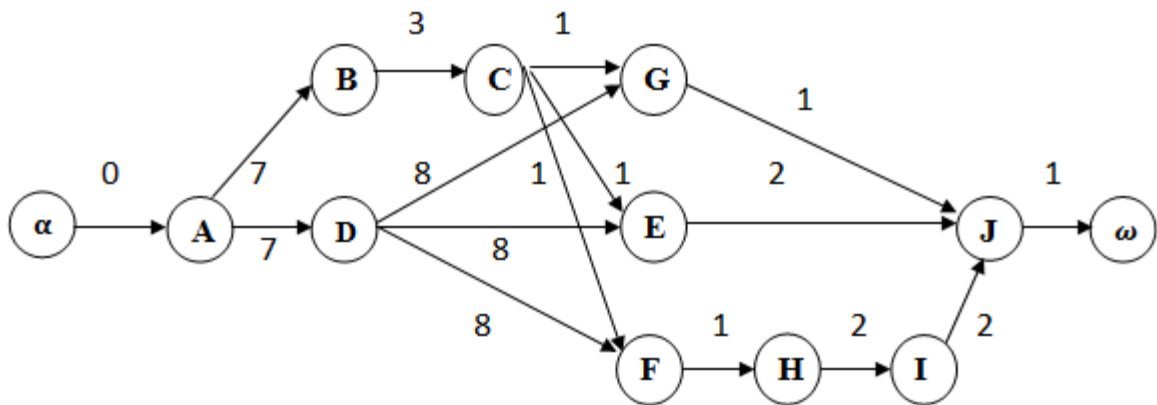


**Figure 2.4** La tâche u précède la tâche v dans le graphe AoN

On porte en suite sur chaque arc incident extérieurement à un sommet u la durée de la tâche u correspondante de sorte que les valeurs sur les arcs de même extrémité initiale soient égales. Nous verrons cependant plus loin que ces valeurs vont différer en rajoutant des contraintes sur les tâches autres que les antériorités.

Un chemin  $\alpha, A_1, A_2, \dots, A_k, A$  du graphe de  $\alpha$  à une tâche A correspond alors à une suite de tâche  $A_1, A_2, \dots, A_k$ , précède  $A_{i+1}, i=1, \dots, k-1$ , et  $A_k$  précède A. La durée d'un tel chemin est la somme des durées des tâches qui le composent. Le dessin du graphe est très simple puisqu'il suffit de disposer les sommets aléatoirement et de les relier par les arcs. Le graphe étant sans circuit et sans boucle (une tâche ne se succède pas à elle-même). [23] La réorganisation des sommets en niveaux montre clairement les antériorités. Elle se fait comme suit :

- **niveau I** : la tâche source  $\alpha$  sans antécédent.
- **niveau II** : tâches n'ayant que le prédécesseur  $\alpha$ .
- **niveau III** : tâches n'ayant que les prédécesseurs du niveau II.
- **niveau IV** : tâches n'ayant que les prédécesseurs du niveau III.
- ...etc.
- le dernier niveau est  $\omega$  qui n'a aucun successeur.



**Figure 2.5** Le graphe AoN les sommets réorganisés en niveaux [2]

Enfin, notons que le mot potentiel associé au graphe est issu de l'analogie avec la différence de potentiels entre deux nœuds d'un réseau électrique. [23]

#### 4.1.3. La méthode AoA

##### 4.1.3.1. Généralités

AoA signifie (Activité on Arc). L'outil AoA permet non seulement de visualiser un réseau d'antériorités, mais, dans le cas d'un projet, d'en déterminer dates et marges, d'en assurer contrôle et suivi. Il date de 1958 et vient des Etats-Unis où il a été développé sous l'impulsion de la marine américaine. Celle-ci a en effet créé à cette époque là une force de frappe nucléaire dont faisait partie un programme de missiles à longue portée POLARIS qui représentait : 250 fournisseurs, 9000 sous traitants, 7 ans de réalisation.

L'utilisation du AoA a permis de ramener la durée globale de réalisation du projet de 7 à 4 ans. Cette méthode s'est ensuite étendue dans l'industrie américaine, puis l'industrie occidentale. [9]

La méthode AoA est le plus souvent synonyme de gestion de projets importants et à long terme. C'est la raison pour laquelle un certain nombre d'actions préalables sont nécessaires pour réussir sa mise en œuvre.

- Tout d'abord définir de manière très précise le projet d'ordonnement.
- Définir ensuite un responsable du projet auquel on rendra compte de l'avancement du projet et qui prendra les décisions importantes.
- Analyser le projet par grands groupes de tâches, puis détailler certaines tâches.
- Définir très précisément les tâches et rechercher leur durée.
- Rechercher les coûts correspondants ce qui peut éventuellement remettre en cause certaines tâches ayant un coût trop élevé.
- Effectuer des contrôles périodiques pour vérifier que le système ne dérive pas.

Elle se distingue de la méthode des potentiels par le fait que les tâches ne sont plus associées aux nœuds mais aux arcs du réseau. [9]

L'algorithme de résolution est très semblable à celui de la méthode AoN.

La différence majeure réside donc dans la construction du graphe :

Le graphe de la méthode AoA est souvent plus difficile à construire que celui de la méthode AoN car on peut être amené à introduire des arcs fictifs qui ne correspondent à aucune tâche.

Dans la méthode AoA, chaque tâche est donc associée à un arc du graphe, la longueur de l'arc correspondant à la durée de la tâche en question. Les sommets sont utilisés pour traduire les relations de succession temporelle. Ainsi, si la tâche  $j$  doit suivre la tâche  $i$  l'extrémité terminale de l'arc représentant la tâche  $i$  coïncidera avec l'extrémité initiale de l'arc représentant la tâche  $j$ .

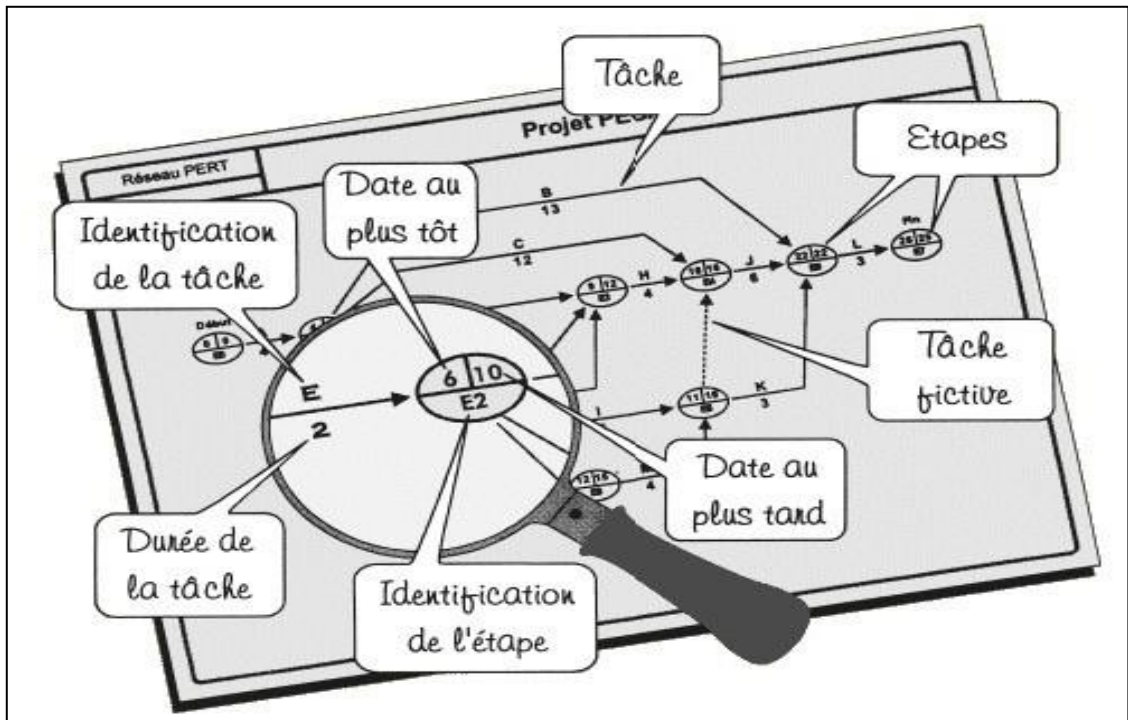
#### 4.1.3.2. Présentation du AoA :

Contrairement à celle du Gantt, la méthode AoA s'attache surtout à mettre en évidence les liaisons qui existent entre les différentes tâches d'un projet et à définir le chemin dit « critique ». Comme pour le Gantt, sa réalisation nécessite tout d'abord de définir :

- Le projet à réaliser.
- Les différentes opérations et les responsables.
- Les durées correspondantes.
- Les liens entre ces différentes opérations.

Le graphe AoA est composé d'étapes et d'opérations. On représente les étapes par des cercles, les opérations par des flèches. La longueur des flèches n'a pas de signification, il n'y a pas de proportionnalité de temps.

Seront développées par la suite, les typologies de l'outil AoA.



**Figure 2.6** Représentation du réseau AoA [9]

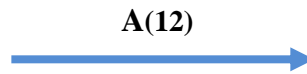
Un réseau AoA est constitué par des tâches et des étapes

- **Étape** : commencement ou fin d'une tâche. Une étape n'a pas de durée. On symbolise une étape (ou " nœud ") sur le réseau par un cercle.



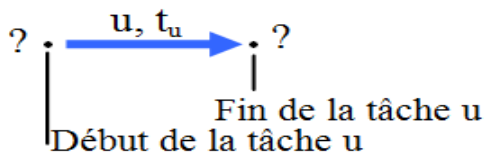
**Figure 2.7** Représentation d'une étape

- **Tâche** : Déroulement dans le temps d'une opération. Contrairement à l'étape, la tâche est pénalisante car elle demande toujours une certaine durée, des moyens (ou ressources) et coûte de l'argent. Elle est symbolisée par un vecteur (ou arc orienté, ou liaison orientée) sur lequel seront indiqués l'action à effectuer et le temps estimé de réalisation de cette tâche.



**Figure 2.8** Représentation d'une tâche

Deux arcs  $u$  et  $v$  sont dessinés de telle sorte que  $T(u) = I(v)$  (voir Figure 2.9 (b)) si et seulement si la tâche  $u$  précède la tâche  $v$ .  $T(u)$  et  $I(v)$  correspondent aux extrémités finale et initiale respectivement de l'arc  $u$ . Ces extrémités initiale et terminale d'un arc sont respectivement les événements début de tâche et fin de tâche, elles sont appelées étape (voir Figure 2.9 (a)). Les durées  $t_i$  sont portées sur les arcs correspondants.



**Figure. 9. (a)**



**Figure. 9. (b)**

**Figure 2.9** La tâche  $u$  précède la tâche  $v$  dans le graphe AoA.

Pour dessiner le graphe AoA, on balaye la table d'ordonnancement selon la colonne des codes et la tâche correspondant à la ligne de balayage en cours est rajoutée en tenant compte des antériorités. [23]

#### 4.1.3.3. Utilité de la méthode AoA

Elle permet de :

- Faciliter l'étude de l'ordonnancement des projets.
- Calculer la durée minimum du projet.
- Identifier les tâches dites "critiques" qui doivent se dérouler sans aucun retard ni délai pour respecter la durée minimum.
- Mettre en évidence les "marges" de manœuvre qui existent pour choisir les dates de déroulement des tâches non critiques.
- Simplifier le contrôle de l'enchaînement des tâches lors de l'exécution du projet.

- Rechercher d'éventuelles améliorations permettant de réduire la durée totale du projet. [9]

On va présenter la table suivant pour dessiner le graphe AoA :

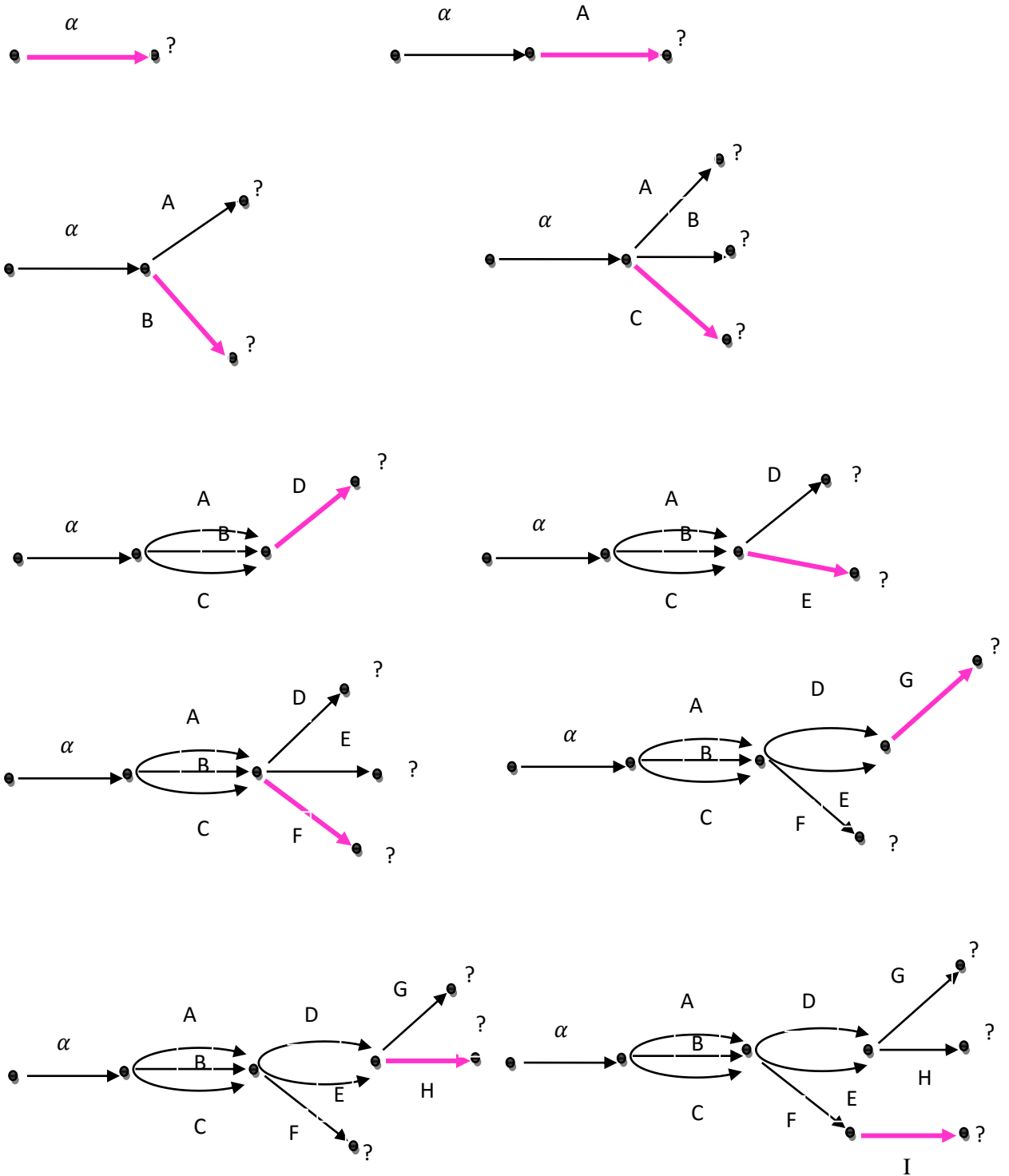
Code de tâche	Désignation	Durée	Antériorité
<b>A</b>	Etablissement des plans	4	$\alpha$
<b>B</b>	Terrassement	5	$\alpha$
<b>C</b>	Creuser les fonds de piliers	6	$\alpha$
<b>D</b>	Faire les fonds de piliers	3	A, B, C
<b>E</b>	Placement arrivée et évacuation assainissement	7	A, B, C
<b>F</b>	Dalle	5	A, B, C
<b>G</b>	Cloisonnage	8	D, E
<b>H</b>	Carrelage	6	D, E
<b>I</b>	Plomberie	5	F
<b>J</b>	Installation chauffage	3	F
<b>K</b>	Electricité	5	G, H, I, J
<b>L</b>	Pose faïence	2	G, H, I, J
<b>M</b>	Boiserie	1	K, L
<b>N</b>	Peinture	4	K, L

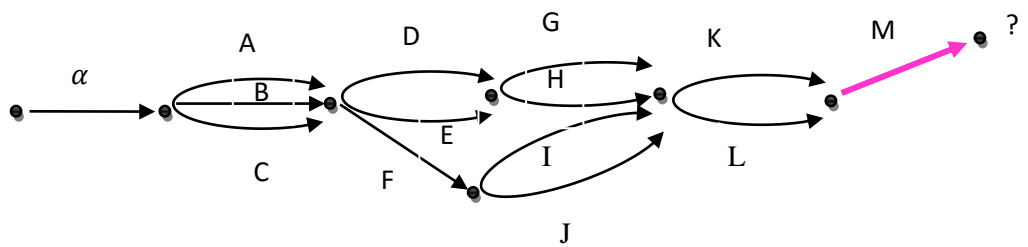
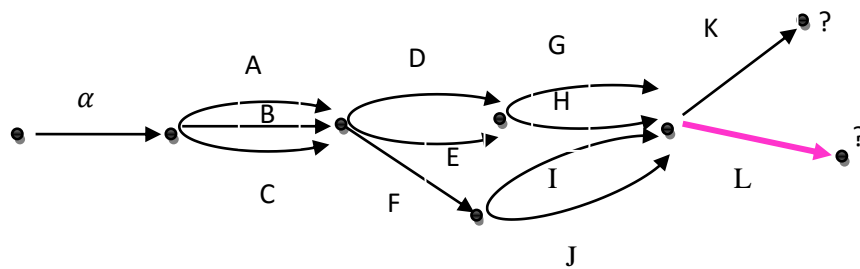
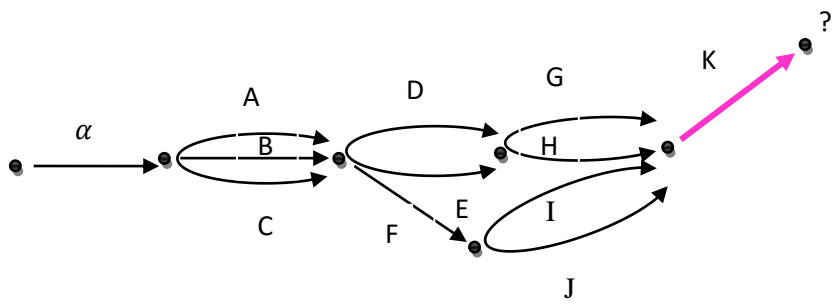
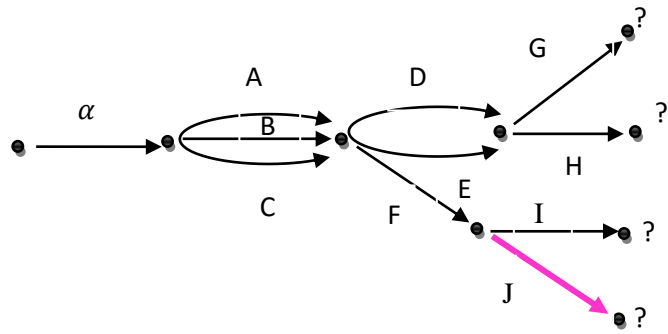
**Table 2.3** Table initiale d'ordonnement T2

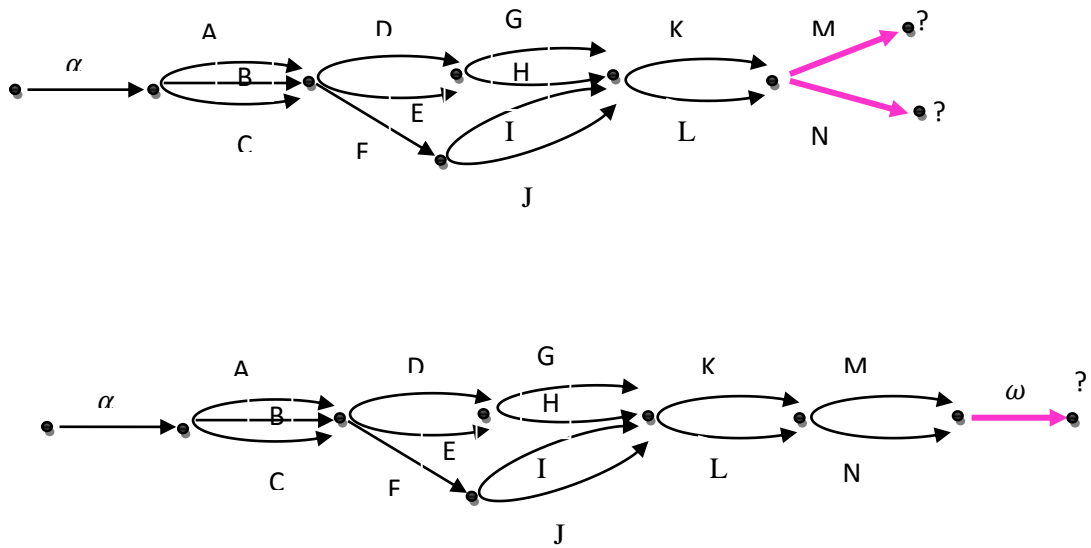
#### 4.1.3.4. Dessin du graphe AoA :

Généralement, le praticien trouve des difficultés à dessiner le graphe. La figure suivante présente une technique nouvelle de construction du graphe AoA. Elle est plutôt pédagogique. [23]

Considérons l'ordonnancement du tableau précédant (voir la **Table 2.3**)





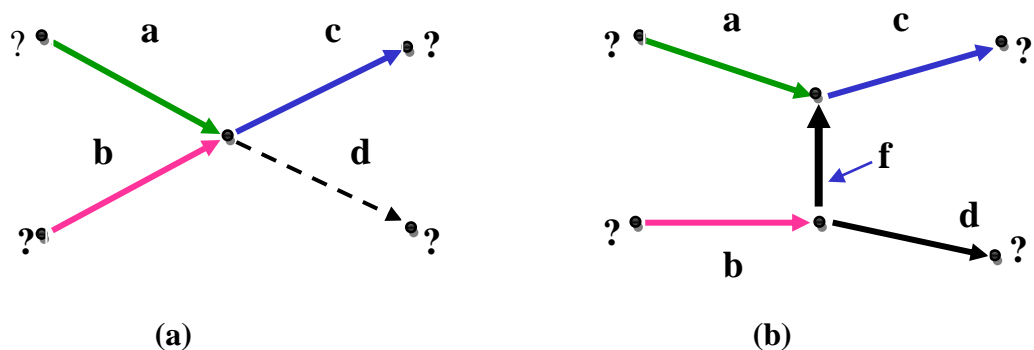


**Figure 2.10** .Le graphe AoA construit par ajout successif des tâches, les durées n'étant pas inscrites

4.1.3.5. La tâche fictive dans le graphe AoA :

Le graphe AoA (Figure 2.11 (a)) du sous tableau des antériorités (Tab 2.4) est faux, pour y remédier on introduit une tâche supplémentaire de durée 0 et qui n'influe pas sur la durée totale du projet. Cette tâche est appelée tâche fictive. On modifie alors le dessin de la Figure 2.11 (a) et le problème est résolu (voir Figure 2.11 (b)).

L'introduction des tâches fictives permet de solutionner certaines situations et de lever les ambiguïtés. Elles ne mettent en jeu aucun moyen matériel ou financier dans la gestion de projet. [13]



**Figure 2.11 (a) et (b)** Problème de représentation dans le graphe AoA avec introduction de la tâche fictive f dans le nouveau graphe AoA

Code	Ant.
<b>C</b>	<b>a, b</b>
<b>d</b>	<b>b</b>

**Table 2.4** Un sous-tableau des antériorités de c et d.

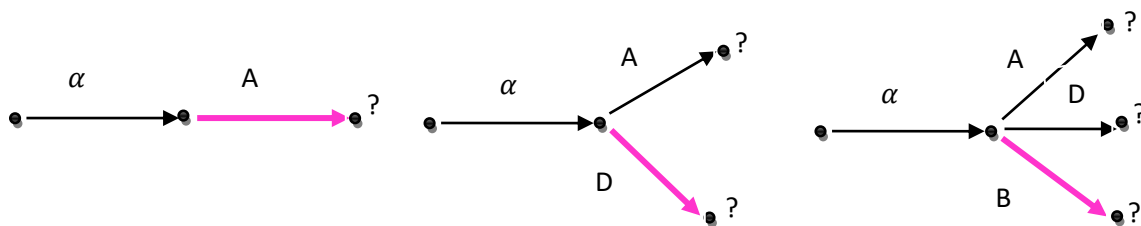
L'introduction des tâches fictives permet de solutionner certaines situations et de lever des ambiguïtés. Elles ne mettent en jeu aucun moyen matériel ou financier. Considérons l'exemple de construction simplifié suivant (tiré de [18]):

Codes	Durées	Antér.
<b><math>\alpha</math></b>	0	-
<b>A</b>	2	$\alpha$
<b>B</b>	2	$\alpha$
<b>C</b>	2	H
<b>D</b>	3	$\alpha$
<b>E</b>	4	B, G
<b>F</b>	2	C, I
<b>G</b>	3	A, D
<b>H</b>	4	B, D
<b>I</b>	5	H
<b>J</b>	3	C
<b><math>\Omega</math></b>	0	E, J, F

Niveau	Sommet
<b>Niveau I</b>	A
<b>Niveau II</b>	A, B, D
<b>Niveau III</b>	G, H
<b>Niveau IV</b>	C, E, I
<b>Niveau V</b>	F, J
<b>Niveau VI</b>	$\omega$

**Table 2.5** La table d'ordonnancement T3 et l'organisation des sommets en niveaux.

Le dessin du graphe AoA se fait exactement comme l'exemple précédent :



**Figure 2.12 (a)**

Pour dessiner la tâche G il faut que les arcs des tâches A et D coïncident dans leurs extrémités terminales, or la tâche D est antérieure à une autre tâche (H), d'où la création d'une tâche fictive f1 (voir Figure (b)).

Le même scénario se répète avec la tâche H : il faut que les extrémités de D et B coïncident, or D est antérieure à G , donc il faut créer une autre tâche fictive f2 (voir figure (c)).

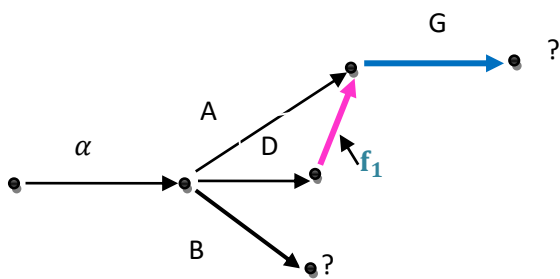


Figure 2.12 (b)

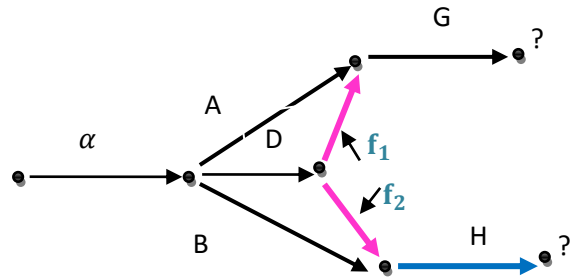


Figure 2.13 (c)

Et nous continuons le même processus jusqu'à la dernière ligne de la table (voir le **Table 2.5**) :

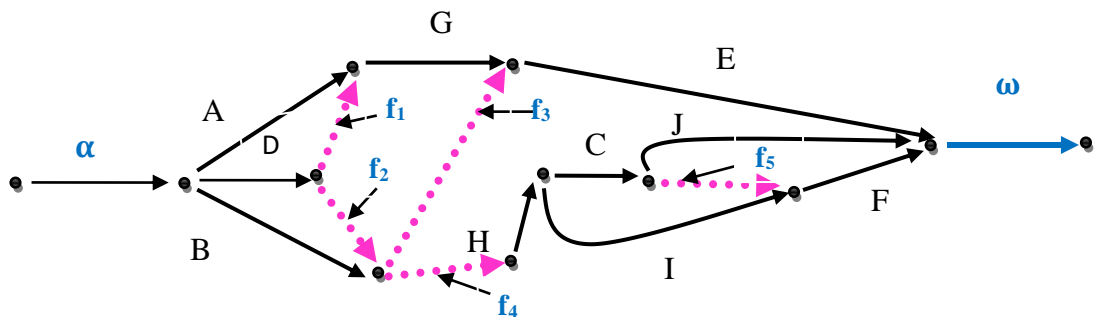
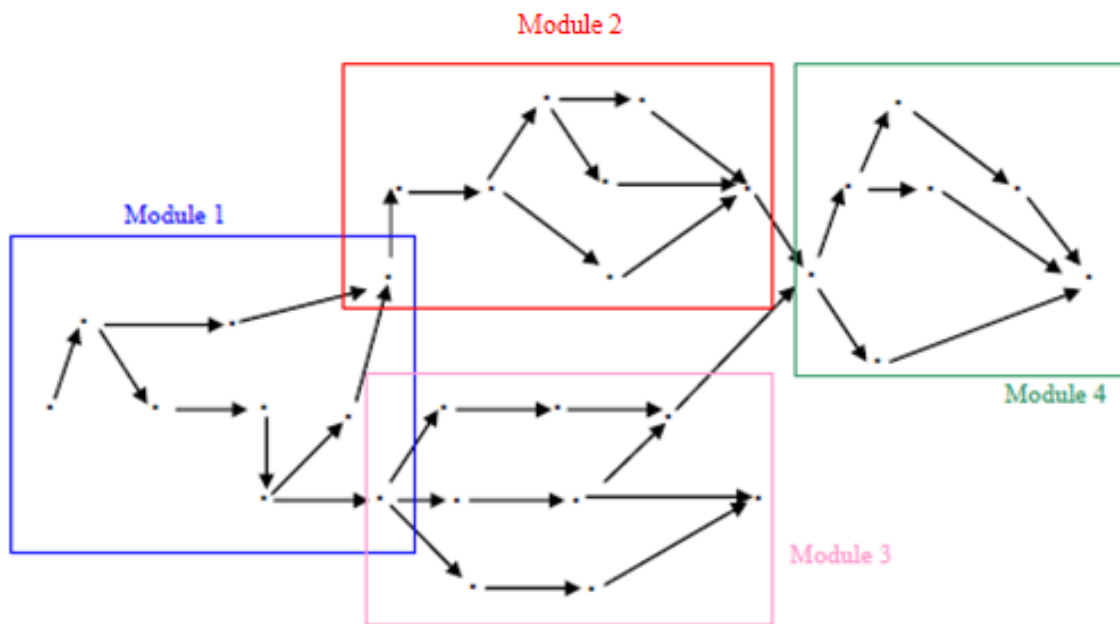


Figure 2.12 Le graphe AoA de la **Table 2.5** avec un minimum de 5 tâches fictives.

4.1.4. Modularité :

Si le graphe AoA construit est trop important, on peut le décomposer en un certain nombre de modules. La méthode AoA est alors appliquée autant de fois qu'il y a de modules.

Considérons l'exemple suivant : [23]



**Figure 2.13** La modularité dans le graphe AoA.

#### 4.1.5. Comparaison :

Nous venons de décrire la modélisation du problème central de l'ordonnancement par les graphes, et plus précisément par la méthode des potentiels et la méthode AoA.

Un inconvénient de cette dernière par rapport à la méthode AoN est de doubler initialement les sommets du graphe puisqu'on associe à chaque tâche une étape « début de tâche » et une étape « fin de tâche ».

Une autre difficulté apparaît, comme on l'a vu précédemment, il est souvent nécessaire d'introduire de nombreuses tâches fictives.

De plus, à un même problème correspondent plusieurs graphes AoA plus ou moins simples. N'oublions pas qu'il y a une infinité de graphes AoA à cause des tâches fictives qu'on peut introduire comme on veut et là où on veut puisqu'elles ne nécessitent aucun moyen matériel ou financier qui influe sur la durée ou le coût global du projet.

Parmi les avantages de cette méthode est qu'elle est plus facilement lisible par les praticiens car une tâche est représentée par un arc et un seul. Donc, le nombre de tâches représente le nombre d'arcs réels.

De plus, en disposant ces arcs horizontalement et en leur donnant une longueur proportionnelle à la durée de la tâche, on présente le graphe AoA sous forme de diagramme de Gantt. [23]

## 5. Résolution des problèmes d'ordonnancement de projet

Le graphe AoA présente une image claire des activités du projet et de leurs corrélations. Quand les durées sont imposées dans le réseau, les problèmes d'accomplissement des objectifs de l'ordonnancement deviennent évidents. Le chef de projet peut contrôler les tâches critiques et étudier l'effet du programme sur les charges de travail. Le PERT temps, dans ce contexte, est assurément un outil qui, appliqué avec vigueur et exactitude, représente une percée de gestion, de planification et de contrôle. [23]

Le graphe AoA n'est pas une étape dans le processus de calcul du temps et du coût du projet. Il demeure un outil de base pour d'autres objectifs qui découlent de ce graphe.

Pour calculer le délai le plus court pour la réalisation du projet, c'est-à-dire chercher le chemin de longueur maximale entre le début du projet ( $\alpha$ ) et sa fin ( $\omega$ ), Supposer que notre réseau de projet a  $n+1$  nœuds, l'événement initial étant 0 et le dernier événement étant  $n$ . Les tâches sont numérotées  $x_1, x_2, \dots, x_n$ , respectivement. Le début du projet à  $x_0$  sera de durée 0 et sera également le temps de début des travaux. La date de début d'une tâche doit être plus grande que la date de début de la tâche précédente. Pour une activité définie comme à partir de l'événement  $i$  et fin à l'événement  $j$ , ce rapport peut être exprimé comme contrainte d'inégalité de la forme :

$$x_j \geq x_i + D_{ij}$$

Où  $D_{ij}$  est la durée de l'activité (i, j).

Cette même expression peut être écrite pour chaque activité et doit juger vrai dans tout programme faisable. Mathématiquement, le problème d'établissement du programme du chemin critique est de réduire au minimum la période de l'accomplissement de projet ( $x_n$ ) sujet aux contraintes sus citées :

Minimiser

$$\mathbf{Z} = \mathbf{x}_n$$

Sujet à

$$\mathbf{x}_0 = \mathbf{0}$$

$$\mathbf{x}_j - \mathbf{x}_i - \mathbf{D}_{ij} \text{ pour toute activité } (i, j)$$

Plutôt que de résoudre le problème d'établissement du programme de chemin critique avec un algorithme de programmation linéaire (tel que la méthode précédente), des techniques plus efficaces sont disponibles qui tirent profit de la structure de réseau du problème. Ces méthodes de solution sont très efficaces en ce qui concerne les calculs exigés, de sorte que des réseaux très grands puissent être traités sur ordinateur. [23]

### 5.1. Dates et marges associées à une tâche

Il existe plusieurs relations utilisées dans le séquençement des tâches. Les plus utilisées sont :

- **do<sub>A</sub>** : Date de début au plus tôt

La date de début au plus tôt d'une tâche A **do<sub>A</sub>** est la date minimum à laquelle peut commencer. C'est donc la date avant laquelle un événement ne peut se réaliser.

- **fo<sub>A</sub>** : Date de fin au plus tôt

La date de fin au plus tôt d'une tâche A **fo<sub>A</sub>** est la date à laquelle A se termine en commençant à sa date de début au plus tôt.

- **da<sub>A</sub>** : Date de début au plus tard

La date de début au plus tard d'une tâche A **da<sub>A</sub>** est la date limite de sa réalisation. Après quoi le projet sera retardé.

- **fa<sub>A</sub>** : Date de fin au plus tard

La date de fin au plus tard d'une tâche A  $fa_A$  est la date minimum des dates de fin au plus tard des tâches qui lui succèdent. Elle est la date à laquelle A se termine si elle commence à sa date de début au plus tard.

- **Marge totale**

La marge totale de A, notée  $mt_A$ , est le délai qui peut être accordé à A pour son commencement sans qu'il y ait un retard dans la réalisation du projet.

- **Marge libre**

La marge libre de A, notée  $ml_A$ , est le délai accordé au commencement de A sans la modification des marges totales des tâches postérieures. Un retard supérieur à la marge libre se répercute sur les tâches suivantes en diminuant leurs marges libres. [18]

## 5.2. Calcul des dates et marges

Une tâche ne peut démarrer que lorsque les tâches de tous les chemins de  $\alpha$  à A sont exécutées, en particulier celles du plus long chemin de  $\alpha$  à A dont la longueur est la date minimum de début d'exécution de A, c'est-à-dire la date de début au plus tôt  $do_A$ .

L'exécution d'un projet ne se termine donc que lorsque toutes ses tâches sont exécutées, en particulier celles de plus long chemin de  $\alpha$  à  $\omega$ .

Les dates de début au plus tôt des tâches d'un problème d'ordonnement sont donc déterminées à partir de l'arborescence des plus longs chemins du graphe des potentiels préparé ou du graphe AoA. [18]

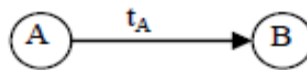
Les graphes étant sans circuits, nous appliquerons l'algorithme de Belleman pour déterminer les dates de début au plus tôt. Les durées sur les arcs de même extrémité initiale étant égales dans le graphe des potentiels préparé, on vérifie que les tâches ayant les mêmes antécédents ont les mêmes dates de début au plus tôt.

### 5.3. Calcul des dates

Pour calculer la date de début au plus tôt de la tâche A  $do_A$  on procède comme suit :

- La date de début au plus tôt de la tâche  $\alpha$  est initialisée à 0
- pour les autres tâches :

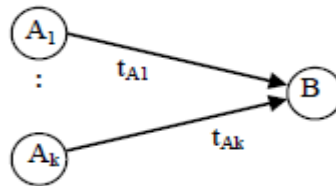
. soit il y a un seul arc entre A et B



Alors :  $do_B = do_A + t_A$

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

$$do_B = \max (do_{A1} + t_{A1}, \dots, do_{Ak} + t_{Ak})$$

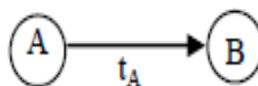


**Figure 2.14** Calcul de  $do_A$  dans le cas d'un ou plusieurs arcs.

Pour la date de début au plus tard de la tâche A  $da_A$ , la technique est la suivante :

- La date de début au plus tard de la tâche  $\omega$  est initialisée à sa date de début au plus tôt c.à.d.  $da_\omega = do_\omega$
- Pour les autres tâches :

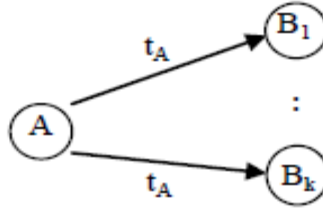
- Soit il y a un seul arc entre A et B



Alors :  $da_A = da_B - t_A$

- Soit il y a plusieurs arcs qui sortent de A alors :

$$da_A = \min (da_{B_1}, \dots, da_{B_k}) - t_A$$



**Figure 2.15** Calcul de  $da_A$  dans le cas d'un ou plusieurs arcs.

Pour le calcul de la date de fin au plus tôt de la tâche A notée  $fo_A$ , elle s'obtient aisément en ajoutant la durée de la tâche A à sa date de début au plus tôt :

$$fo_A = do_A + da_A$$

La date de fin au plus tard de la tâche A  $fa_A$  s'obtient en ajoutant sa durée à sa date de début au plus tard :

$$fa_A = da_A + t_A.$$

### ➤ Calcul des marges

Pour chaque tâche, on obtient l'intervalle de placement en faisant la différence entre la date de début au plus tard et la date de début au plus tôt de la même tâche.

$$ml_A = da_A - do_A$$

La marge libre d'une tâche A est la date de début au plus tôt minimum qui succèdent à A diminuée de sa date de fin au plus tôt.

$$ml_A = \min (do_{A_1}, do_{A_2}, \dots, do_{A_k}) - fo_A$$

Les tâches  $A_1, A_2, \dots, A_k$  sont les tâches qui succèdent à A. [18]

**Nota :** on a toujours :  $0 \leq ml_A \leq mt_A$

➤ **Le chemin critique**

Le chemin critique dans le graphe AoN ou AoA est le plus long chemin de  $\alpha a \omega$ . Sa valeur  $do_{\omega}$  est la durée minimale du projet.

Le projet n'ayant pas d'intérêt à être retardé, la fin des travaux  $\omega$  doit impérativement avoir lieu à la date  $da_{\omega} = do_{\omega}$ . Plus généralement, toute tâche A du chemin critique P est telle que :  $da_A = do_A$ . [18]

Prenons l'ordonnancement (Table 2.6 Une table d'ordonnancement T) suivant et calculons les dates et les marges :

Tache	Désignation	Durée	précédents
<b>A</b>	Maçonnerie	7	-
<b>B</b>	Charpente	3	A
<b>C</b>	Couverture	1	B
<b>D</b>	Plomberie électricité	8	A
<b>E</b>	Revêtements extérieurs	2	C, D
<b>F</b>	Menuiseries	1	C, D
<b>G</b>	Aménagement jardin	1	C, D
<b>H</b>	Plâtres	2	F
<b>I</b>	Peinture	2	H
<b>J</b>	Finitions	1	E, G, I

**Table 2.6** Une table d'ordonnancement T4

Calcul des dates de début au plus tôt	Calcul des dates de début au plus tard
Le calcul se fait dans l'ordre croissant des niveaux et les sommets d'un même niveau dans un ordre quelconque.	Le calcul se fait dans l'ordre décroissant des niveaux et les sommets d'un même niveau dans un ordre quelconque.
$do_{\alpha} = 0$ $do_A = do_{\alpha} + 0 = 0$ $do_B = do_A + 7 = 7$ $do_C = do_B + 3 = 10$ $do_D = do_A + 7 = 7$ $do_E = \text{Max}[do_C + 1, do_D + 8] = 15$ $do_F = \text{Max}[do_D + 8, do_C + 1] = 15$ $do_G = \text{Max}[do_D + 8, do_C + 1] = 15$ $do_H = do_F + 1 = 16$ $do_I = do_H + 2 = 18$ $do_J = \text{Max}[do_G + 1, do_E + 2, do_I + 2] = 20$ $do_{\omega} = do_J + 1 = 21$	$da_{\omega} = do_{\omega} = 21$ $da_J = do_{\omega} - 1 = 20$ $da_I = da_J - 2 = 18$ $da_E = da_J - 2 = 18$ $da_G = da_J - 1 = 19$ $da_H = da_I - 2 = 16$ $da_F = da_H - 1 = 15$ $da_D = \text{Min}[da_F, da_G, da_E] - 8 = 7$ $da_C = \text{Min}[da_F, da_G, da_E] - 1 = 14$ $da_B = da_C - 3 = 11$ $da_A = \text{Min}[da_D, da_B] - 7 = 0$ $da_{\alpha} = da_A, - 0 = 0$

Table 2.7 Dates de début au plus tôt et au plus tard du table 2.6

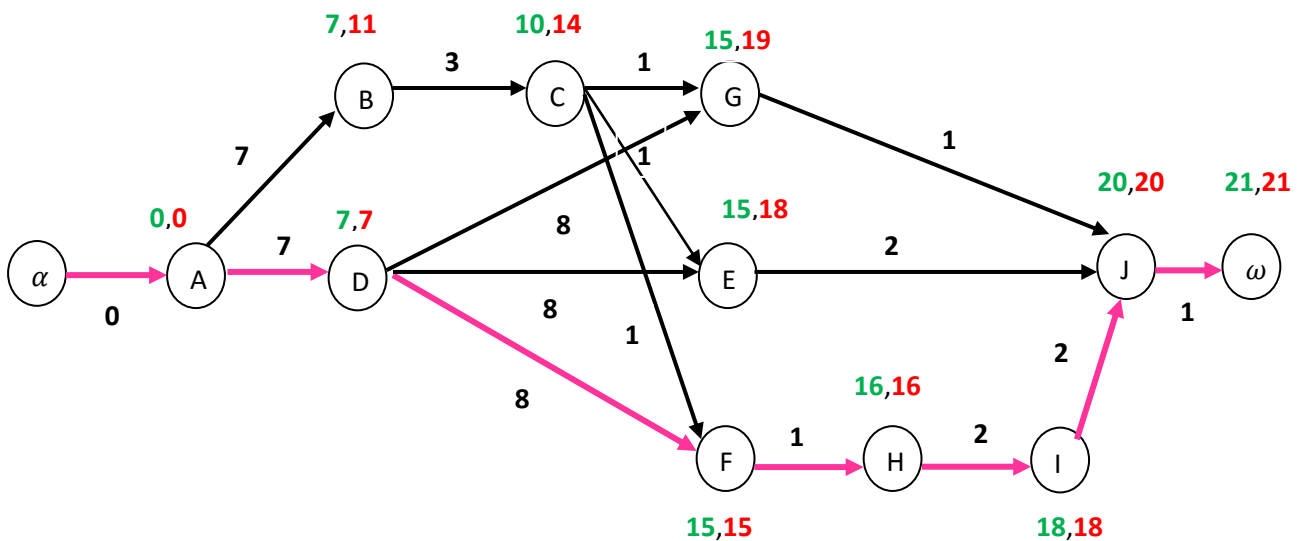
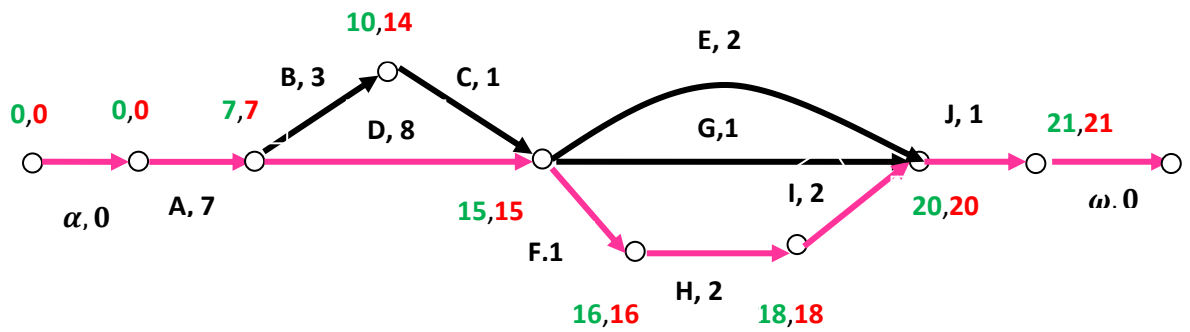


Figure 2.16 Les dates de début au plus tôt et au plus tard dans le graphe AoN. [2]



**Figure 2.17** Les dates de début au plus tôt et au plus tard dans le graphe AoA [2]

## 6. Conclusion

L'objectif de l'ordonnancement central est non seulement d'organiser les activités dans le temps, mais également de permettre aux décideurs d'entreprises d'estimer les dates finales de livrables. Dès lors, l'ordonnancement de projet revêt un intérêt capital de par les estimations qu'il propose et qui servent de base à la prise de décision et à la coordination des ressources internes, des approvisionnements, des relations de sous-traitance. [23]

## CHAPITRE 3

### LE GRAPHE AOA MINIMAL

#### 1. Introduction

Les réseaux de projet pour la planification et l'ordonnancement sont basés sur deux types de représentation: Activités sur Arcs (AoA) et activités sur les nœuds (AoN). Chaque type de représentation (AoA ou AoN) a des caractéristiques souhaitables qui sont propres à leur type. [23]

Par exemple, certains avantages des activités sur Arcs (AoA) réseaux pour la planification du réseau du projet sont : (K. Agarawal, S. E. Elmaghraby 1996) chaque arc dans le graphe AoA a son nœud d'origine et de destination, ou avec des variables une matrice d'arc de nœud. (N. H. Ahuja, S. P. Dozzi 1994) des réseaux de AoA pourraient être agencés dans une matrice totale uni-modulaire "node-arc" qui assure une solution entière pour les données de nombres entiers. (Y. Cohen 2003) AoA est adapté à certaines formulations populaires qui utilisent soustractions appariées des variables de nœud AoA. Alors que la construction de graphe AoN est relativement facile, les réseaux AoA peuvent contenir des arcs factices et leur construction est beaucoup plus problématique. Toutefois, en raison de leur importance, La différence majeure réside donc dans la construction du graphe : le graphe AoA est souvent plus difficile à construire que celui de la méthode AoN car on peut être amené à introduire des arcs fictifs qui ne correspondent à aucune tâche.

Dans la méthode AoA, chaque tâche est donc associée à un arc du graphe, la longueur de l'arc correspondant à la durée de la tâche en question. Les sommets sont utilisés pour traduire les relations de succession temporelle. Ainsi, si la tâche j doit suivre la tâche i, l'extrémité terminale de l'arc représentant la tâche i coïncidera avec l'extrémité initiale de l'arc représentant la tâche j. [23]

## 2. Etat de l'art

Les problèmes d'ordonnement sont définis par la donnée d'un certain nombre d'opérations (les tâches) et des contraintes de succession entre ces tâches (la tâche b ne peut commencer qu'après que la tâche a soit terminée), ainsi que les durées de chacune de ces tâches.

Définir un ordonnancement consiste à attribuer à chacune des tâches une date d'exécution de manière à ce que l'ensemble des tâches soit terminé au plus tôt.

Une des méthodes pour résoudre les problèmes d'ordonnement est la méthode AoA. Le premier pas de cette méthode consiste à construire un graphe (graphe AoA) dont les arcs représentent les tâches et où les relations de succession entre les tâches sont traduites ainsi : b est successeur de a si et seulement si il existe dans le graphe un chemin dont le premier arc est a et le dernier b. La construction n'est en général possible que si l'on introduit de nouvelles tâches, dites virtuelles de durée nulle. [10]

Les réseaux AoA peuvent contenir des arcs fictifs et leur construction est beaucoup plus problématique. Toutefois, en raison de leur importance, les réseaux AoA ont attiré de nombreux essais de recherche pour générer des réseaux AoA avec nombre minimal d'arcs fictifs.

Nous allons présenter, d'abord, l'état de l'art du problème du graphe AoA minimal. Nous présenterons également un des algorithmes célèbres du dit problème à savoir la méthode de Sterboul et al.

A noter cependant qu'un réseau AoN unique peut être représenté par plusieurs réseaux de graphe AoA différents avec le même indice de complexité minimale.

Nous présenterons ensuite la nouvelle méthode de Mouhoub et al utilisant un ensemble de résultats des graphes adjoints, qui nous permettra de construire le graphe AoA en minimisant le nombre de tâches fictives.

L'importance de cette minimisation dans le graphe AoA réside dans le fait que les praticiens n'aiment pas utiliser le graphe AoN, mais ils préfèrent travailler avec le graphe AoA où chaque tâche est représentée par un arc et un seul, or l'inconvénient de cette méthode est que le graphe peut avoir un nombre élevé de tâches fictives qui rendent le graphe moins

lisible et difficile à suivre lors de l'exécution du projet, d'où la nécessité de réduire au maximum ce nombre.

Le dessin du graphe AoA (Activités on Arcs) appelé également graphe PERT ou graphe américain est plus difficile à cause des tâches fictives qu'il génère. Or, les praticiens préfèrent travailler avec le graphe AoA parce qu'il est simple à lire puisque chaque tâche est représentée par un arc.

Signalons que nous travaillons, dans ce qui suit, sur des graphes non valués. On considère que le graphe AoN donné, qui est tiré de la table d'ordonnancement T, est préparé, c'est à dire que les arcs incidents extérieurement de chaque sommet sont de même valeurs. Pour les arcs qui ne le sont pas, nous appliquons alors, la technique vue précédemment. Pour avoir une source et un puits on ajoute à T une tâche  $\alpha=0$  précédant toute tâche n'ayant pas de prédécesseurs et une tâche  $\omega = n+1$  n'ayant pas de successeurs. On note que s'il existe une seule source (respectivement un seul puits), on la (resp. le) nomme  $\alpha$  (resp.  $\omega$ ).

La lecture d'un graphe se fait de gauche à droite, les tâches réelles, qui sont tirées de la table T, seront appelées les arcs bleus et les arcs fictifs seront appelés les arcs rouges.

Kelley 1961 note qu'il est avantageux pour réduire la longueur des calculs suivants de construire un graphe AoA ayant le nombre minimal possible de sommets. [23]

Le problème ainsi posé par Kelley a été abordé avec plus ou moins de succès par un certain nombre d'auteurs :

Ils en donnent un nouveau qui est ; leur article ne contient pas de preuve mathématique conformément au style du journal où ils sont publiés les résultats de leurs travaux.

Cantor et Dimsdale 1963 donnent, avec démonstration un algorithme exact. Leur méthode qui s'applique à un problème qui englobe les problèmes d'ordonnancement ,gagne en généralité mais donne lieu, pour notre cas particulier ,à des calculs inutilement longs.

Hayes 1969 donne un ensemble de recettes pour construire un graphe AoA sa méthode ne produit pas, en général, le graphe minimal.

Dimsdale (5) propose un algorithme de construction du graphe AoA minimal.

Fisher, Liebman, Nemhauser 1968 montrent que l'algorithme de Dimsdale est faux ;

Les spécialistes qui insistent sur l'utilisation du graphe AoA présentent un certain nombre d'arguments qui justifient leur choix. Selon Fink et al, le graphe AoA est plus concis.

En outre, Hendrickson et al expliquent qu'il est à proximité du célèbre diagramme de Gantt qui est utilisé jusqu'à présent, et selon Yuval et Sadeh, la structure du graphe AoA est beaucoup plus adaptée à certaines techniques d'analyse et d'optimisation des formulations.

Des exemples de minimisation dummy méthodes heuristiques peuvent être trouvés aussi tôt que dans (M. Hayes, A.C. Fisher).

Dans les années 1970 Corneil et al 1973, a affirmé avoir réalisé un algorithme optimal qui contient le minimum d'arcs fictifs, mais Syslo 1981 infirmait leur résultat.

Krishnamoorthy et Deo 1979 ont montré que la constatation du problème de minimisation Des arcs fictifs est NP-difficile. [10]

Dans les années 1980, Spinard 1980, Syslo (1981,1984) ont offert plusieurs heuristiques pour arriver au seul objectif : le graphe AoA minimal.

Syslo 1981 cherche à minimiser le nombre des arcs virtuels ce qui est un autre problème. Dans les années 1990(M. K. Agarawal, E. Demeulemeester) graphe AoA proposées méthodes réseau génératrices mais ils ne considèrent pas la redondance et d'autres problèmes soulevés par les activités fictives, et par conséquent, sont affaiblies considérablement. [10]

Dans les années 1990, Demeulemeester , Dodin et Herroelen , 1993 ont proposé des méthodes de génération de réseaux AoA, mais ils n'ont pas tenu en compte la présence d'arcs redondants, par conséquent, leurs travaux sont considérablement affaiblis. Une autre direction a été ouverte lorsque Elmaghraby et Herroelen 1980 a élaboré un indice de complexité comme un outil de mesure de la complexité dans les réseaux AoA. Kamburowski et al (1992,2000) chacun d'eux développé une méthode qui génère un AoA-index avec la complexité minimale. Noter cependant qu'à partir d'un seul réseau AoN on peut représenter plusieurs différents réseaux AoA avec le même indice minimal de complexité.

En 2007, Cohen et Sadeh ont présenté une nouvelle approche pour l'élaboration d'un graphe AoA unique et qui a d'après leurs conclusions, le nombre minimal d'arcs fictifs. [23]

Pour notre travail, on va choisir un des algorithmes célèbres qui résoud le problème de minimisation des tâches fictifs dans un AoA ,à savoir l'algorithme de Sterboul et al et un autre algorithme nouveau , celui de Mouhoub et al et terminer notre travail par une comparaison.

### 3. Algorithme de F. STERBOUL et al

#### 3.1. Notations et définitions

Soit un graphe  $G=(x, u)$  orienté

$x$  : ensemble des sommets et  $u$  : ensemble des arcs

$(i, j)$  arc reliant le sommet  $i$  au sommet  $j$ .



-u et v sont deux arcs de G,  $u \longrightarrow v$  il existe un chemin dont le premier arc est u et dont le dernier arc est v.



$\forall i \in x$  on note:

$P(i) = \{j \in x \mid (j, i) \in u\}$  les prédécesseurs directe de  $i$ .

$Q(i) = \{j \in x \mid (i, j) \in u\}$  les successeurs directe de  $i$

$\tilde{P}(i) = \{j \in x \mid \exists \text{ un chemin de } j \text{ vers } i\}$  les prédécesseurs indirects  $i$ .

$(i, j)$  est dit arc redondant s'il existe un chemin de longueur supérieure à un de  $i$  vers  $j$

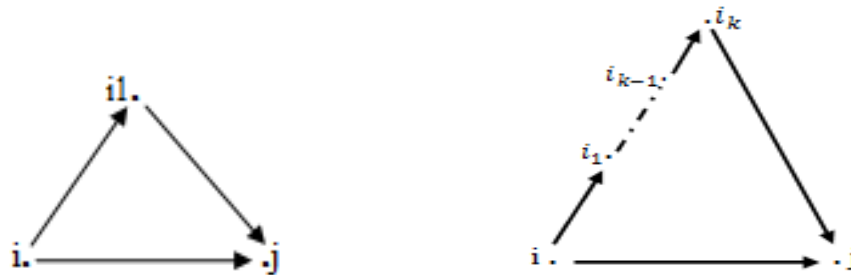


Figure 3.1 Un arc redondant  $(i, j)$

### 3.2. graphe arc dual d'un graphe

Soit  $G=(X, U)$  un graphe orienté, sans circuit et sans arc redondant on construit à partir de  $G$  un graphe  $H = (Y, V)$  appelé graphe arc dual comme suit :

- tout sommet  $i$  de  $G$  est remplacé par deux sommets  $a_i$  et  $b_i$  ( $H$  contient  $2 |X|$  sommets)
- $V$  est constitué des arcs  $(a_i, b_i)$  (par une application injective) pour tout  $i \in X$ , et des arcs  $(b_i, a_j)$  si  $i \in P(j)$  dans  $G$ . Les arcs  $(a_i, b_i)$  sont appelé les arcs réels et les arcs  $(b_i, a_j)$  sont appelé des arcs virtuels. [12]

#### Algorithme

Soit  $G=(X, U)$  un graphe orienté .Il est le graphe AoN (graphe français) ou les sommets représentent les taches et ou l'arc  $(i, j)$  appartient à  $U$  si et seulement si la tache  $i$  précède la tache  $j$ .

A partir de graphe  $G$  on construit un graphe  $H$  arc dual de  $G$  . $H$  est le graphe AoA (graphe américain) qui contient un nombre minimum de sommets. [12]

### 3.3. Construction

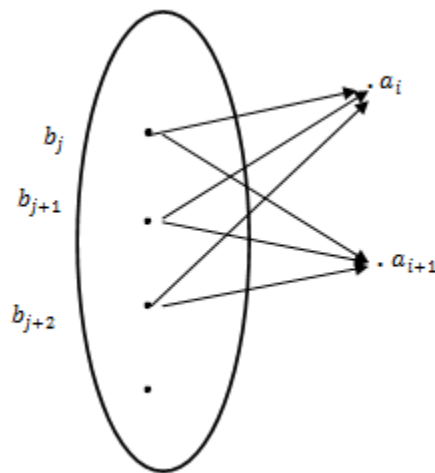
Pour cela on doit suivre les étapes suivantes:

- **Etape 1** : Construction de  $H_0$  :

A partir du graphe  $G$  on construit le graphe  $H_0 =(X_0, V_0)$  qui est le graphe arc dual de  $G$  en utilisant la définition précédant.

- **Etape 2** : construction de  $H_1$  :

dans  $H_0$  tous les sommets  $a_i$  qui ont les même prédécesseurs forment une classe  $\overline{a_1}, \overline{a_2}, , \dots, \overline{a_l}$  Sont les classes correspondantes.



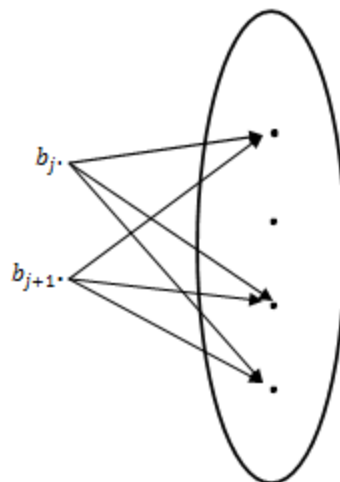
**Figure 3.2** .les sommets  $a_i$  et  $a_{i+1}$  forment une seule classe

Donc les sommets de chaque classe forment un biparti complet avec leurs prédécesseurs

Dans  $H_0$  , tous les sommets  $\bar{b}_i$  qui ont les mêmes successeurs forment une classe.

$\bar{b}_1, \bar{b}_2, \dots, \bar{b}_m$  sont les classes, correspondantes.

Donc, les sommets de chaque classe forment un biparti complet avec leurs successeurs.



**Figure 3.3** les sommets  $b_j$  et  $b_{j+1}$  forment une seule classe

On appelle contraction de type 1, l'opération qui consiste dans  $H_0$

A contracter en un sommet unique, tous les sommets d'une même classe.

Dans  $H_0$  on effectue toutes les contractions possibles de type 1 et on note le graphe obtenu

$H_1 = (Y_1, V_1)$ ,  $H_1$  Est un graphe arc dual de  $G$ .

➤ **Etape 3** : Définition d'un bon arc de  $H_1$  :

Un arc  $(\bar{b}_i, \bar{a}_j)$  est un bon arc de  $H_1$  ssi :

$$\forall \mathbf{K}, \mathbf{l} \in \mathbf{X}, \begin{cases} \mathbf{k} \in \mathbf{P}(j) \\ \mathbf{l} \in \mathbf{Q}(i) \end{cases} \Rightarrow \mathbf{k} \in \tilde{\mathbf{P}}(\mathbf{l})$$

Donc tous les sommets de la classe  $\bar{a}_j$  dans  $G$  doivent avoir chacun un arc le reliant avec tous les sommets de la classe  $\bar{b}_i$ .

On d'autre terme : les sommets de la classe  $\bar{a}_i$  dans  $G$  forme un biparti complet avec les sommets de la classe  $\bar{b}_j$ .

Si tel est le cas, l'arc  $(\bar{b}_i, \bar{a}_j)$  est dit un bon arc.

-les bons arcs de  $H_1$  n'ont deux à deux aucun sommet commun.

➤ **Etape 4** : Construction de  $H_2$  :

On appelle contraction de type 2, l'opération qui dans  $H_1$ , consiste à contracter en un sommet unique, le sommet origine et le sommet extrémité d'un bon arc.

Dans  $H_1$ , on effectue toutes les contractions de type 2 et on note le graph obtenu

$H_2 = (Y_2, V_2)$ ,  $H_2$  Est un graphe arc dual de  $G$ , et il a le nombre minimum de sommets.

[12]

**3.4. Exemple :**

Soit le graphe AoN suivant :

Code de tâche	Antériorité.
1	-
2	-
3	-
4	1, 2, 3
5	1,3
6	2,3
7	4,5
8	5
9	6

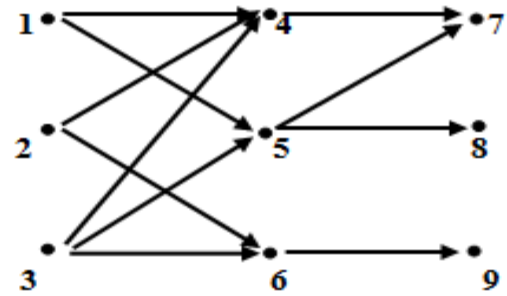


Figure 3.4 Le graphe des AoN

Table 3.1 Table d'ordonnancement

Contraction de  $H_0$  :

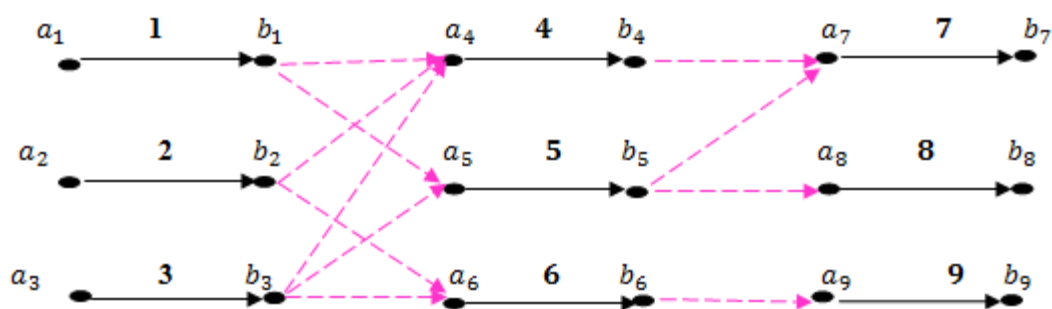


Figure 3.5 Le graphe  $H_0$

**Contraction de  $H_1$  :**

I	P(i)
1	-
2	-
3	-
4	1, 2, 3
5	1,3
6	2,3
7	4,5
8	5
9	6

**Table 3.2** les prédécesseurs de sommets

I	Q(i)
1	4,5
2	4,6
3	4, 5, 6
4	7
5	7,8
6	9
7	-
8	-
9	-

**Table 3.3** les successeurs de sommets

**Détermination des classes  $\bar{a}_i$  :** (qui ont les mêmes prédécesseurs)

Les sommets  $\{a_1, a_2, a_3\}$  forment une seule classe  $\bar{a}_1$

Le sommet  $\{a_4\}$  forment une seule classe  $\bar{a}_2$

Le sommet  $\{a_5\}$  forment une seule classe  $\bar{a}_3$

Le sommet  $\{a_6\}$  forment une seule classe  $\bar{a}_4$

Le sommet  $\{a_7\}$  forment une seule classe  $\bar{a}_5$

Le sommet  $\{a_8\}$  forment une seule classe  $\bar{a}_6$

Le sommet  $\{a_9\}$  forment une seule classe  $\bar{a}_7$

$\bar{a}_i$	I
$\bar{a}_1$	1, 2, 3
$\bar{a}_2$	4
$\bar{a}_3$	5
$\bar{a}_4$	6
$\bar{a}_5$	7
$\bar{a}_6$	8
$\bar{a}_7$	9

**Table 3.4** les classes  $\bar{a}_i$

**Détermination des classes  $\bar{b}_j$  :** (qui ont les mêmes successeurs)

Le sommet  $\{ b_1 \}$  forment une seule classe  $\bar{b}_1$

Le sommet  $\{ b_2 \}$  forment une seule classe  $\bar{b}_2$

Le sommet  $\{ b_3 \}$  forment une seule classe  $\bar{b}_3$

Le sommet  $\{ b_4 \}$  forment une seule classe  $\bar{b}_4$

Le sommet  $\{ b_5 \}$  forment une seule classe  $\bar{b}_5$

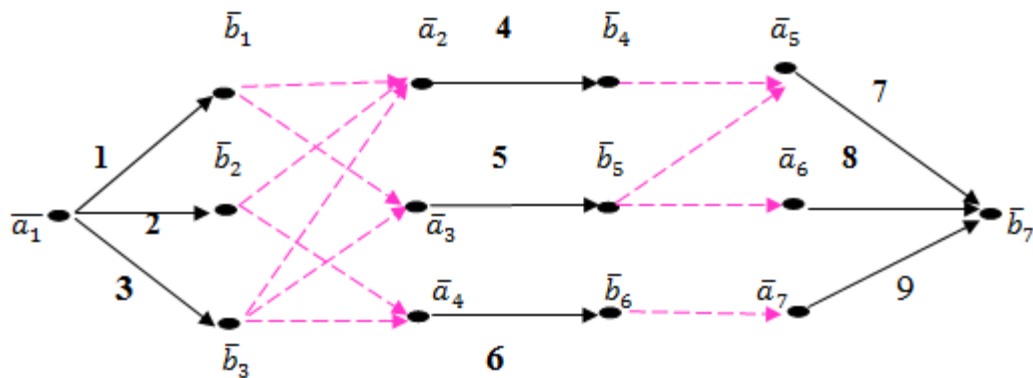
Le sommet  $\{ b_6 \}$  forment une seule classe  $\bar{b}_6$

Les sommets  $\{ b_7, b_8, b_9 \}$  forment une seule classe  $\bar{b}_7$

$\bar{b}_j$	J
$\bar{b}_1$	1
$\bar{b}_2$	2
$\bar{b}_3$	3
$\bar{b}_4$	4,5
$\bar{b}_5$	6
$\bar{b}_6$	7,8,9

**Table 3.5** .les classes  $\bar{b}_j$

On contracte en un sommet unique tous les sommets d'une même classe. [12]



**Figure 3.6** Le graphe  $H_1$

**Détermination des bons arcs :**

$(\bar{b}_1, \bar{a}_2) \begin{cases} k \in P(4) \equiv k \in \{1,2,3\} \\ l \in Q(1) \equiv l \in \{4,5\} \end{cases}$  **Donc :**  $(\bar{b}_1, \bar{a}_2)$  n'est pas un bon arc

$(\bar{b}_1, \bar{a}_3) \begin{cases} k \in P(5) \equiv k \in \{1,3\} \\ l \in Q(1) \equiv l \in \{4,5\} \end{cases}$  **Donc :**  $(\bar{b}_1, \bar{a}_3)$  est un bon arc

$(\bar{b}_2, \bar{a}_2) \begin{cases} k \in P(4) \equiv k \in \{1,2,3\} \\ l \in Q(2) \equiv l \in \{4,6\} \end{cases}$  **Donc :**  $(\bar{b}_2, \bar{a}_2)$  n'est pas un bon arc

$(\bar{b}_2, \bar{a}_4) \begin{cases} k \in P(6) \equiv k \in \{2,3\} \\ l \in Q(2) \equiv l \in \{4,6\} \end{cases}$  **Donc :**  $(\bar{b}_2, \bar{a}_4)$  est un bon arc

$(\bar{b}_3, \bar{a}_2) \begin{cases} k \in P(4) \equiv k \in \{1,2,3\} \\ l \in Q(3) \equiv l \in \{4,5,6\} \end{cases}$  **Donc :**  $(\bar{b}_3, \bar{a}_2)$  n'est pas un bon arc

$(\bar{b}_3, \bar{a}_3) \begin{cases} k \in P(4) \equiv k \in \{1,2,3\} \\ l \in Q(3) \equiv l \in \{4,5,6\} \end{cases}$  **Donc :**  $(\bar{b}_3, \bar{a}_3)$  n'est pas un bon arc

$(\bar{b}_3, \bar{a}_4) \begin{cases} k \in P(6) \equiv k \in \{2,3\} \\ l \in Q(3) \equiv l \in \{4,5,6\} \end{cases}$  **Donc :**  $(\bar{b}_3, \bar{a}_4)$  n'est pas un bon arc

$(\bar{b}_4, \bar{a}_5) \begin{cases} k \in P(7) \equiv k \in \{4,5\} \\ l \in Q(4) \equiv l \in \{7,8\} \end{cases}$  **Donc :**  $(\bar{b}_4, \bar{a}_5)$  est un bon arc

$(\bar{b}_5, \bar{a}_5) \begin{cases} k \in P(9) \equiv k \in \{4,5\} \\ l \in Q(6) \equiv l \in \{7,8\} \end{cases}$  **Donc :**  $(\bar{b}_5, \bar{a}_5)$  n'est pas un bon arc

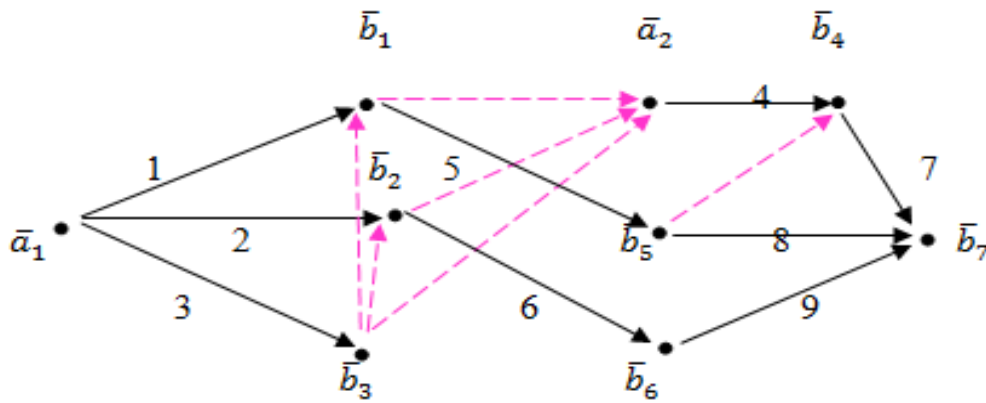
$(\bar{b}_5, \bar{a}_6) \begin{cases} k \in P(9) \equiv k \in \{5\} \\ l \in Q(6) \equiv l \in \{7,8\} \end{cases}$  **Donc :**  $(\bar{b}_5, \bar{a}_6)$  est un bon arc

$(\bar{b}_6, \bar{a}_7) \begin{cases} k \in P(9) \equiv k \in \{6\} \\ l \in Q(6) \equiv l \in \{9\} \end{cases}$  **Donc :**  $(\bar{b}_6, \bar{a}_7)$  est un bon arc

Arc	Obs.
$(\bar{b}_1, \bar{a}_3)$	<i>bon arc</i>
$(\bar{b}_2, \bar{a}_4)$	<i>bon arc</i>
$(\bar{b}_4, \bar{a}_5)$	<i>bon arc</i>
$(\bar{b}_6, \bar{a}_7)$	<i>bon arc</i>
$(\bar{b}_6, \bar{a}_7)$	<i>bon arc</i>

**Table 3.6** les bons arcs.

On contracte le sommet origine et le sommet extrémité d'un bon arc en un sommet unique



**Figure 3.7** Le graphe H2

#### 4. Algorithme de N. E. MOUHOUB et al

Dans ce qui suit, on construit un graphe AoA «  $G_e$  », avec un nombre minimal de tâches fictives, en appliquant sept règles qu'on verra ultérieurement. L'algorithme de Mouhoub et al ne tient pas compte de l'existence ou non des arcs redondants. De plus, son objectif est de minimiser le nombre de sommets et de tâches fictives à la fois. [23]

#### 4.1. Notations et définitions

Considérons  $1, 2, \dots, n$  où  $n$  est le nombre d'activités réelles dans la table d'ordonnement.

$G = (X, U)$  est un réseau.  $\Gamma^+(G, i)$  désigne les successeurs du sommet  $i$  et  $\Gamma^-(G, i)$  ses prédécesseurs dans  $G$ .

$V(G)$  and  $E(G)$  sont, respectivement, les ensembles de sommets et d'arcs de  $G$ .

$SS(G, k)$  (resp.  $SP(G, k)$ ) désigne l'ensemble maximal des sommets de  $G$  qui ont les mêmes successeurs (resp. prédécesseurs) comme  $k$ .

Nous convenons que  $k \in SS(G, k)$  et  $k \in SP(G, k)$ . Si  $P$  est un chemin de  $G$ , alors  $I(G, P)$  et  $T(G, P)$  désignent le sommet initial et le sommet terminal de  $P$ , ( $P$  peut être un arc).

Un arc  $(x, y)$  est appelé arc transitif s'il existe un chemin  $P$  de longueur supérieur ou égale à 2 avec  $I(P) = x$  and  $T(P) = y$ . [23]

#### 4.2. Construction

On construit, à partir du graphe «  $G_v$  » un graphe «  $G_{e1}$  » comme suit :

- tout sommet  $i$  de «  $G_v$  » est remplacé dans «  $G_{e1}$  » par deux sommets  $a_i$  et  $b_i$
- l'ensemble des arcs de  $G_{e1}$  est composé des arcs :

**1-**  $(a_i, b_i)$  (qui représente la tâche  $i$  dans le graphe des potentiels. (N'oublions pas que la tâche  $i$  dans le graphe AoN «  $G_v$  » est représentée par un arc dans le graphe AoA «  $G_e$  »).

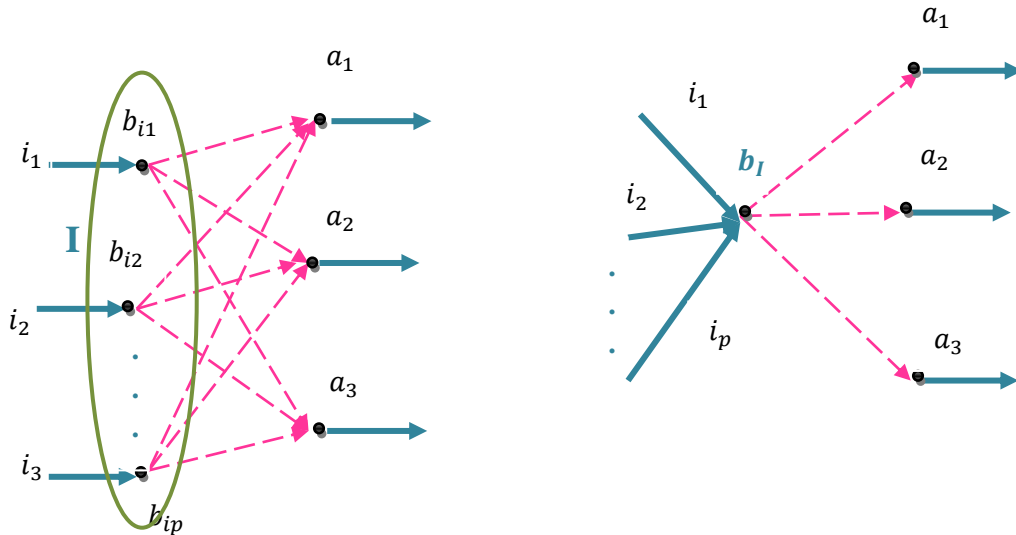
**2-**  $(b_i, a_j)$  si  $i \in P(j)$  dans «  $G_v$  ».

On appelle les arcs  $(a_i, b_i)$  les arcs réels, et les arcs  $(b_i, a_j)$  les arcs fictifs. «  $G_{e1}$  » Contient  $2|X|$  sommets et les différents chemins sont formés alternativement, d'arcs réels et fictifs.

On Remarque que le graphe «  $G_{e1}$  » n'est autre que le diagramme de Gantt en lui rajoutant les arcs d'antériorité. [23]

**Règle 1 : Contraction des  $b_i$  ayant les mêmes successeurs**

Si  $I = \{i_1, i_2, \dots, i_p\}$   $p > 1$  est un ensemble de tâches qui ont les mêmes successeurs et maximal pour cette propriété, alors contracter dans « $G_{e1}$ » les sommets  $b_{i_1}, b_{i_2}, \dots, b_{i_p}$  en un sommet unique  $b_I$  (voir Figure 3.8).



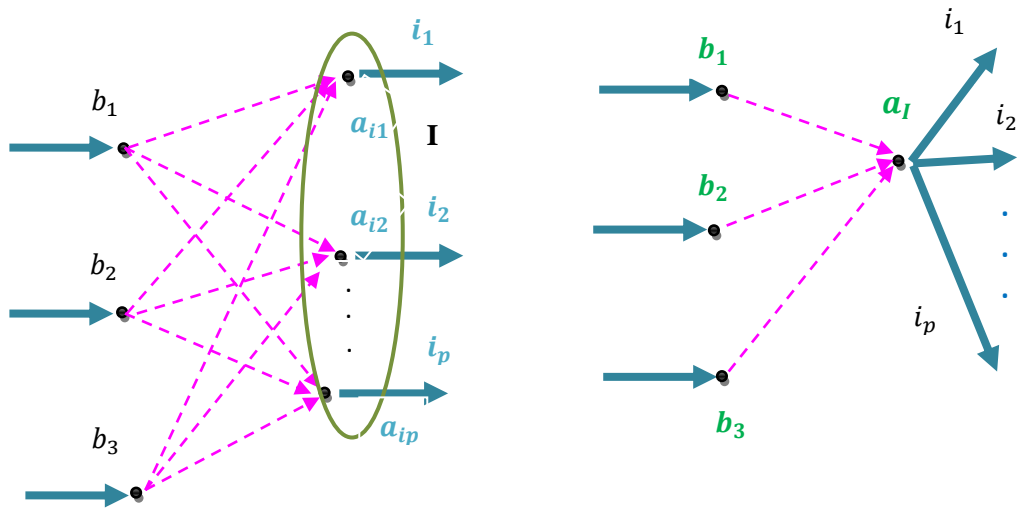
**Figure 3.8** Les sommets  $b_i$  ayant les mêmes successeurs contractés en un sommet unique  $b_I$  dans  $G_{e1}$ .

Eliminer les arcs rouges multiples et répéter cette règle autant de fois que possible pour obtenir le graphe « $G_{e2}$ ».

Noter que dans « $G_{e2}$ », il n'y a aucun arc rouge qui a  $b_i$  comme extrémité initiale.

**Règle 2 : Contraction des  $a_i$  ayant les mêmes prédécesseurs**

Si  $I = \{i_1, i_2, \dots, i_p\}$   $p > 1$  est un ensemble de tâches qui ont les mêmes prédécesseurs et maximal pour cette propriété, alors contracter dans « $G_{e2}$ » les sommets  $a_{i_1}, a_{i_2}, \dots, a_{i_p}$  en un sommet unique  $a_I$  (voir Figure 3.9).

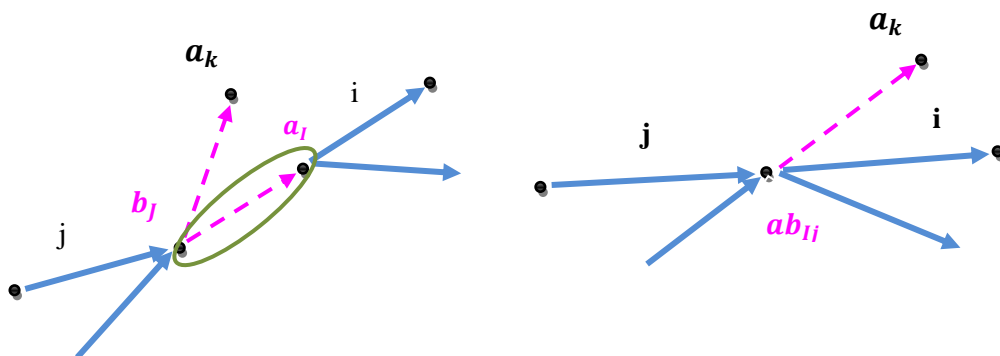


**Figure 3.9** Les sommets  $a_i$  ont les mêmes prédécesseurs contractés en un Sommet unique  $a_i$  dans  $G_{e2}$

Eliminer les arcs rouges multiples et répéter cette règle autant de fois que possible pour obtenir le graphe «  $G_{e3}$  ». «  $G_{e3}$  » N'a pas de sous ensembles de sommets avec les mêmes successeurs ou prédécesseurs. Aucun arc rouge n'a un  $a_i$  comme extrémité initiale ou  $b_i$  comme extrémité terminale. Les différents chemins de «  $G_{e3}$  » sont formés alternativement, d'arcs bleus et rouges. Il n'y a pas un chemin direct de longueur  $\geq 2$  de même couleur. [23]

**Règle 3 : Suppression des arcs transmetteurs de type 1**

On appelle arc transmetteur de type 1 tout arc fictif (rouge) qui vérifie la condition suivante : le nombre d'arcs incidents intérieurement à  $a_i = +1$  ( $d_{ai}^- = +1$ ).



**Figure 3.10** Un arc transmetteur de type 1 entre deux tâches i et j et sa Contraction

Si le sommet  $a_I$  a un seul prédécesseur  $b_J$  dans «  $G_{e3}$  » alors contracté  $a_I$  et  $b_J$  en un sommet unique  $ab_{IJ}$ . (voir Figure 3.10).

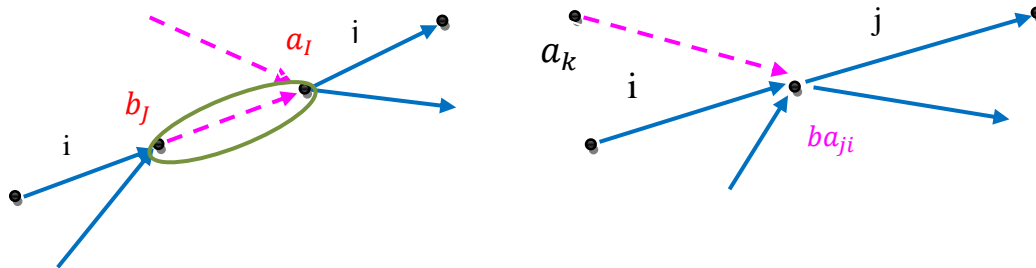
Dans le graphe «  $G_{e3}$  », on effectue toutes les contractions possibles des sommets initial et terminal de tout arc transmetteur. Le graphe obtenu est appelé «  $G_{e4}$  ».

Deux sommets distincts  $a_I$  et  $a_{I'}$ , avec un seul prédécesseur  $b_J$  et  $b_{J'}$ , respectivement ne peuvent pas avoir dans «  $G_{e4}$  » le même prédécesseur  $b_j \neq b_{j'}$ .

Noter que «  $G_{e4}$  » peut contenir un arc rouge ( $ab_{IJ}, a_K$ ) obtenu de l'arc rouge ( $b_J, a_K$ ) de «  $G_{e3}$  ».

**Règle 4 : Suppression des arcs transmetteurs de type 2**

On appelle arc transmetteur de type 2 tout arc fictif (rouge) qui vérifie la condition suivante : le nombre d'arcs incidents extérieurement à  $b_i = +1$  ( $d_{b_i}^+ = +1$ ).



**Figure 3.11** Un arc transmetteur de type 2 entre deux tâches i et j et sa contraction.

Si le sommet  $b_J$  a un seul successeur  $a_I$  dans «  $G_{e4}$  » alors contracté  $b_J$  et  $a_I$  en un sommet unique  $ba_{JI}$ . (voir Figure 3.11).

Dans le graphe «  $G_{e4}$  », on effectue toutes les contractions possibles des sommets initial et terminal de tout arc transmetteur. Le graphe obtenu est appelé «  $G_{e5}$  ».

Puisque la règle 3 est appliquée avant la règle 4,  $a_I$  a au moins deux prédécesseurs. Si  $b_k \neq b_J$  est un deuxième prédécesseur, alors ( $b_k, ba_{JI}$ ) est un arc rouge de «  $G_{e5}$  » obtenu à partir de l'arc rouge ( $b_k, a_I$ ) de «  $G_{e4}$  ».

En arrivant à «  $G_{e5}$  » nous remarquons qu'il a les propriétés suivantes :

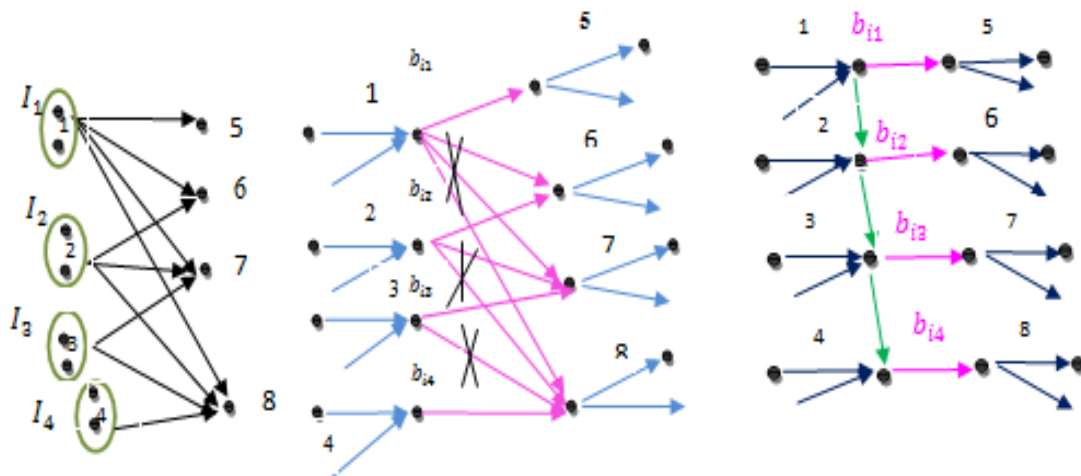
- «  $G_{e5}$  » satisfait la table d'ordonnancement T.
- Il n'existe aucun chemin rouge direct de longueur  $\geq 2$ .
- L'ensemble des sommets de «  $G_{e5}$  » sont partitionnés en quatre types :  $a_I$ ,  $b_I$ ,  $ab_{IJ}$ ,  $ba_{IJ}$ . [23]

**Règle 5 : Les sommets dont les successeurs sont inclus dans l'ensemble des successeurs d'un autre sommet**

Dans «  $G_{e5}$  » cherchons s'il existe un sommet  $b_I$  dont ses successeurs sont inclus dans les Successeurs d'un autre sommet  $b_J$ .

Tous les arcs rose qui sortent de  $b_I$  et qui sont inclus dans les successeurs d'un autre sommet  $b_J$  seront supprimés et remplacés par un arc fictif (de couleur verte) de  $b_I$  à  $b_J$ .

(Voir Figure 3.12).



**Figure 3.12** Les sommets dont les successeurs sont inclus dans l'ensemble des successeurs d'un autre sommet dans  $G_{e5}$  et la réduction des tâches fictives dans  $G_{e6}$ .

Le graphe  $G_{e7}$ , qui est le résultat de l'application de la règle 5 ou 6, satisfait les contraintes de la table d'ordonnancement. Les chemins directs de couleur rose sont de longueur  $\leq 2$ .

**Règle 6 : Les sommets dont les prédécesseurs sont inclus dans l'ensemble des prédécesseurs d'un autre sommet**

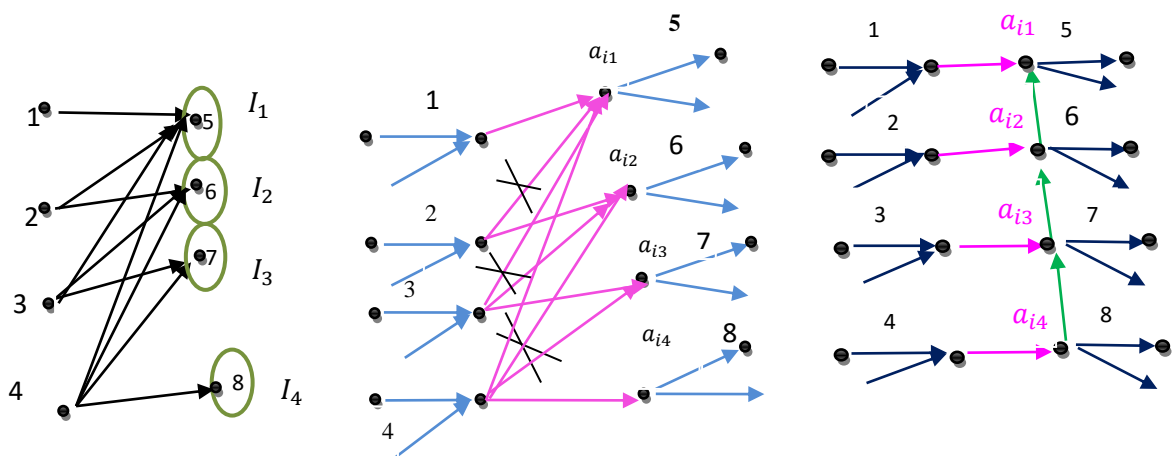
Si la règle 5 ne peut pas être appliquée, alors chercher dans  $G_{e5}$  s'il existe un sommet  $a_I$  dont ses prédécesseurs sont inclus dans les prédécesseurs d'un autre sommet  $a_J$ .

Tous les arcs rouges qui arrivent à  $a_I$  et qui sont inclus dans les prédécesseurs d'un autre sommet  $a_J$  seront supprimés et remplacés par un arc fictif (rose) de  $a_I$  à  $a_J$  (voir Figure 3.8).

Répéter cette règle autant de fois que possible pour obtenir le graphe  $G_{e6}$ .

$G_{e6}$  contient des chemins rouges directs de longueur  $\leq 2$ .

Appliquer à nouveau la règle 4 pour la suppression des arcs transmetteurs qui ont comme sommet d'extrémité initiale  $bi$ . Le graphe obtenu sera appelé  $G_{e7}$ .

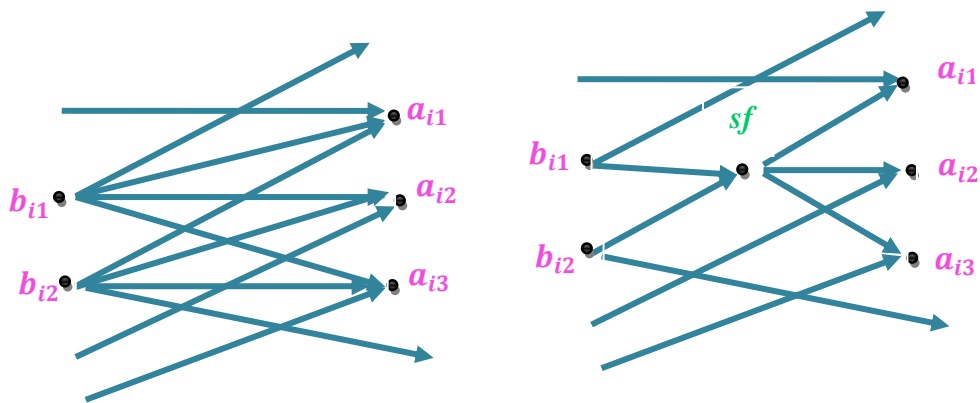


**Figure 3.13** Les sommets dont les prédécesseurs sont inclus dans l'ensemble des prédécesseurs d'un autre sommet dans  $G_{e5}$  et la réduction des tâches fictives dans  $G_{e6}$ .

**Règle 7 : Création d'un sommet fictif**

A ce stade, si la règle 5 ne peut être appliquée, ainsi que la règle 6, alors voir dans le graphe  $G_{e5}$  s'il existe un ensemble de sommets qui peut être partitionné en deux classes  $X_1$  et  $X_2$ , ( $X_1 \cup X_2$ )  $\geq 5$ , de sorte que deux sommets de la même classe ne soient jamais adjacents.  $X_1$  contient les sommets de type  $b_i$  et  $X_2$  contient les sommets de type  $a_i$ .

Les deux classes forment un biparti complet, en plus, de chaque sommet de  $X_1$  sort un ou plusieurs arcs vers un ou plusieurs sommets qui n'appartiennent pas à  $X_2$ , et à chaque sommet de  $X_2$  arrive un arc d'un ou plusieurs sommets qui n'appartiennent pas à  $X_1$ .



**Figure 3.14** Un biparti complet dans  $G_{e7}$  et le sommet fictive  $sf$  dans  $G_{e8}$

La représentation dans  $G_{e8}$  contient moins d'arcs fictifs que dans  $G_{e7}$ .

Si tel est le cas, alors créer un sommet fictif  $sf$  et supprimer les arcs multiples.

Répéter cette règle autant de fois que possible et le graphe obtenu sera appelé  $G_{e8}$ .

Le graphe  $G_{e8}$  satisfait les contraintes de la table d'ordonnancement T. [23]

4.3. Exemple de projet de construction d'un building industriel

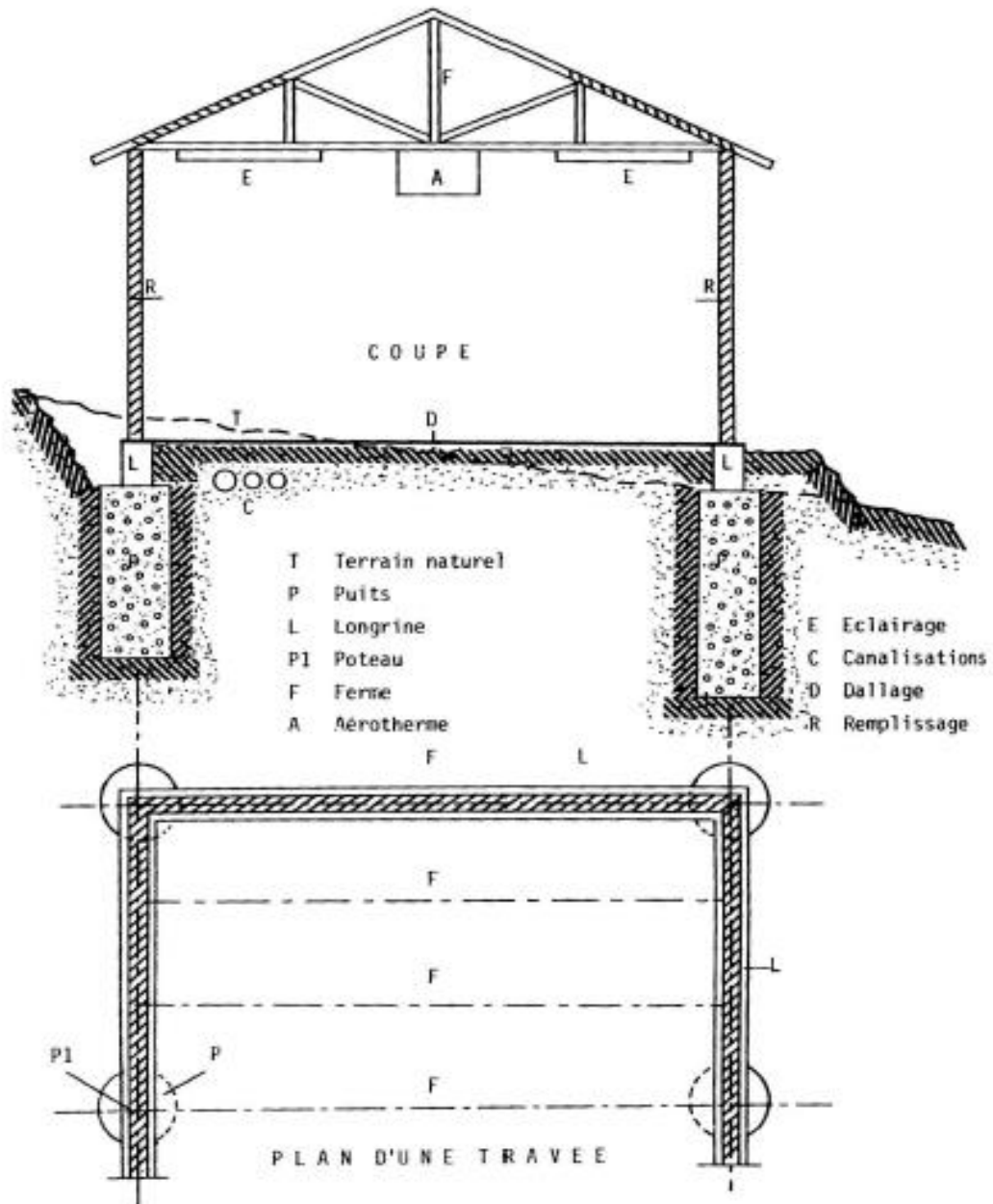


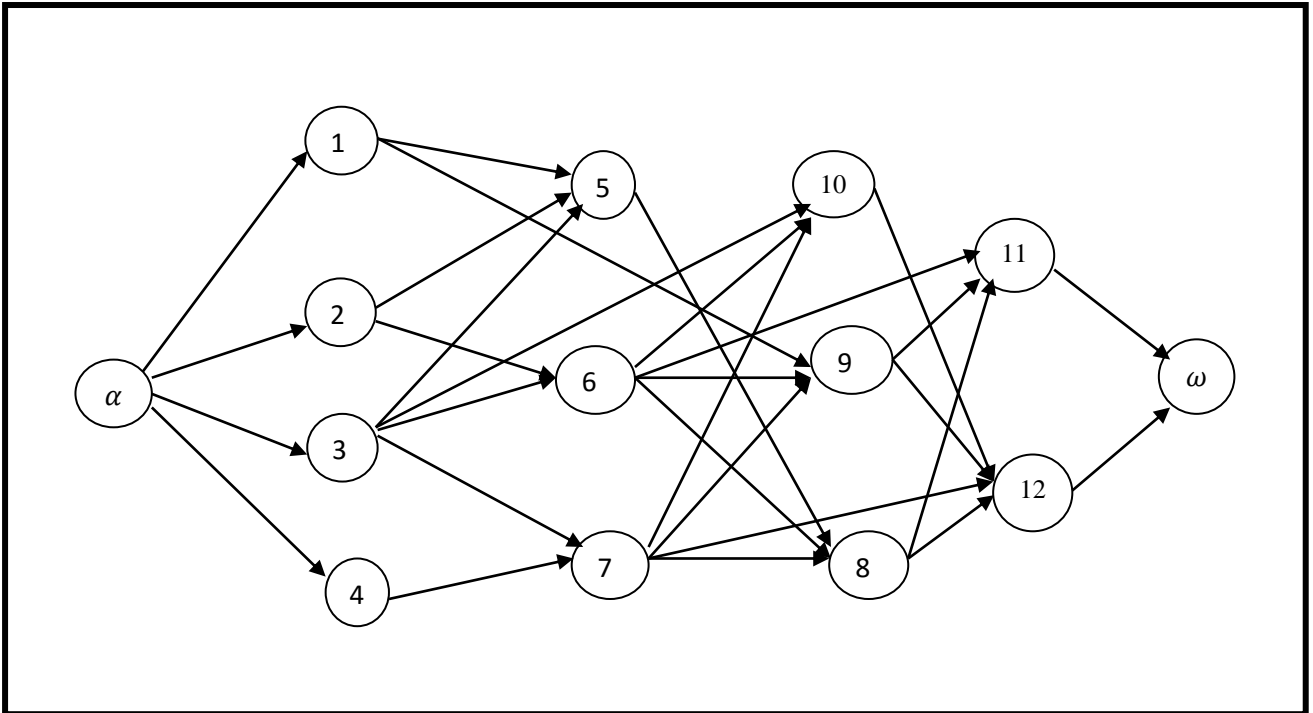
Figure 3.15 Plan pour la construction du bâtiment industriel

Le projet que nous considérons, tiré de " *Marie-Elise Ernould* " consiste à réaliser un bâtiment industriel comprenant cinq travées et un cadre métallique sur un sol irrégulier incliné (voir *Figure 3.15*).

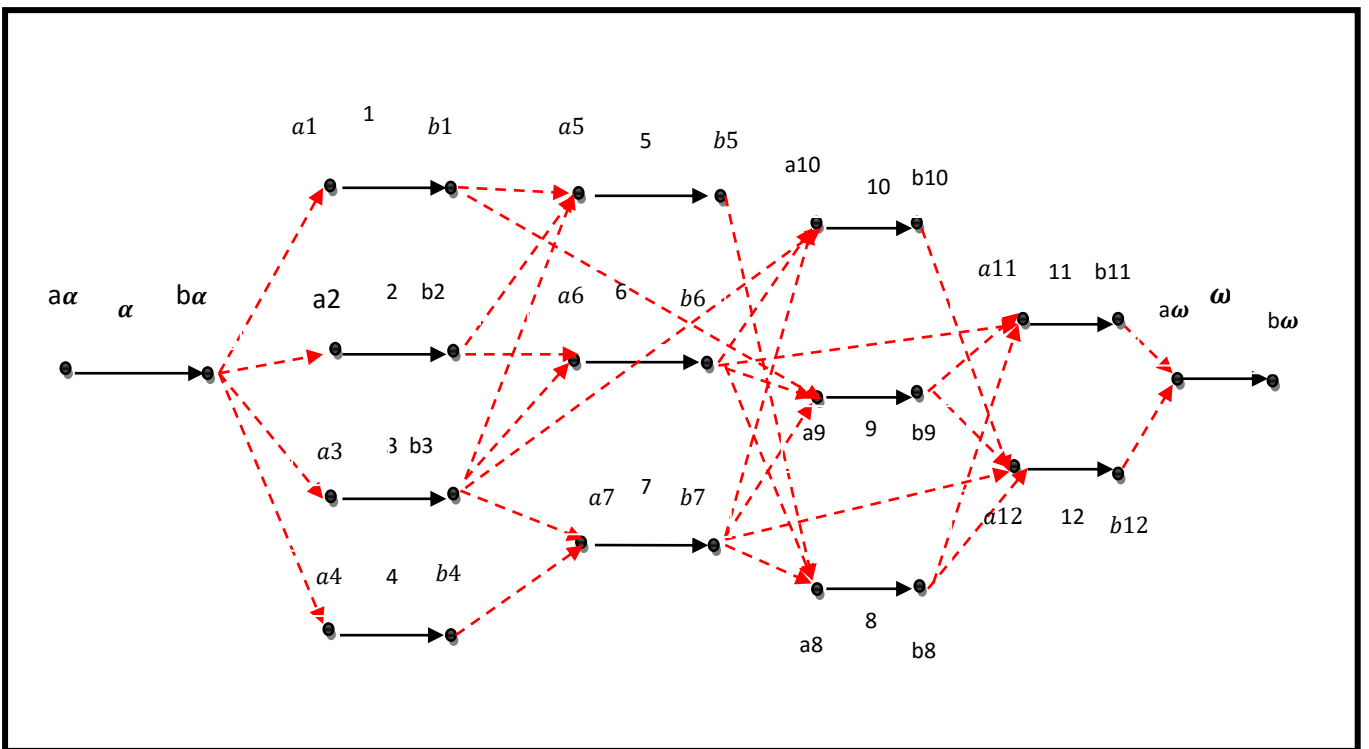
Activité	Désignation	Prédécesseurs
$\alpha$	Démarrage du projet	-
1	Étude de chauffage et de cadres	$\alpha$
2	Installations de chantiers implantation	A
3	Décapage et l'installation du chantier	A
4	Terrassement des poutres en béton	A
5	Construction de l'ossature	1, 2, 3
6	Transport des cadres	2, 3
7	Assemblage	3, 4
8	Pose charpente	1, 6, 7
9	Terrassement et pose des canalisations	5, 6, 7
10	Installation électrique	3, 6, 7
11	Peinture	6, 8, 9
12	Réalisation de la chaussée	7,8, 9, 10
$\omega$	Clôture du projet	11, 12

**Table 3.7** Table d'ordonnement de la Figure 3.15

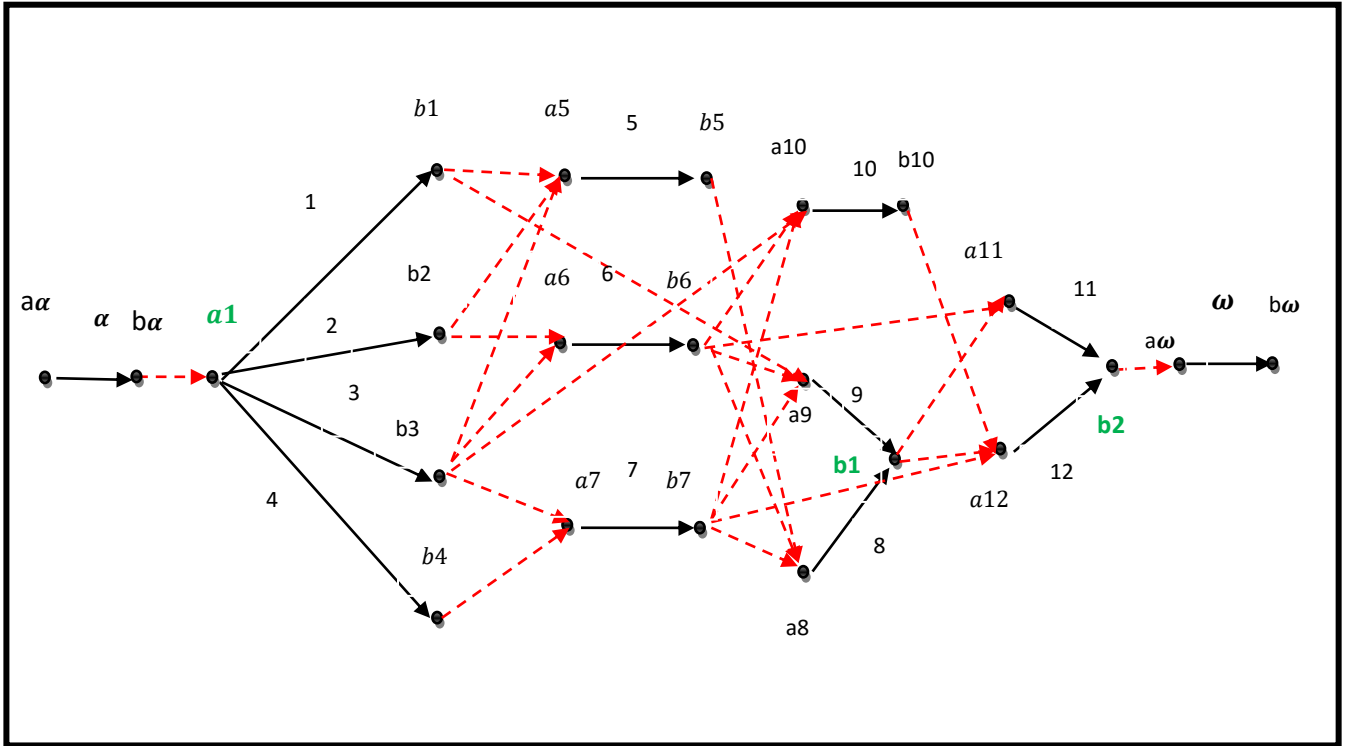
Les activités impliquées pour la construction de ce bâtiment sont définies dans le **tableau 3.7** et le graphe AoN correspondant  $G_V$  représentant les liens entre les activités du projet est montré dans la **Figure 3.16 (a)**. [23]



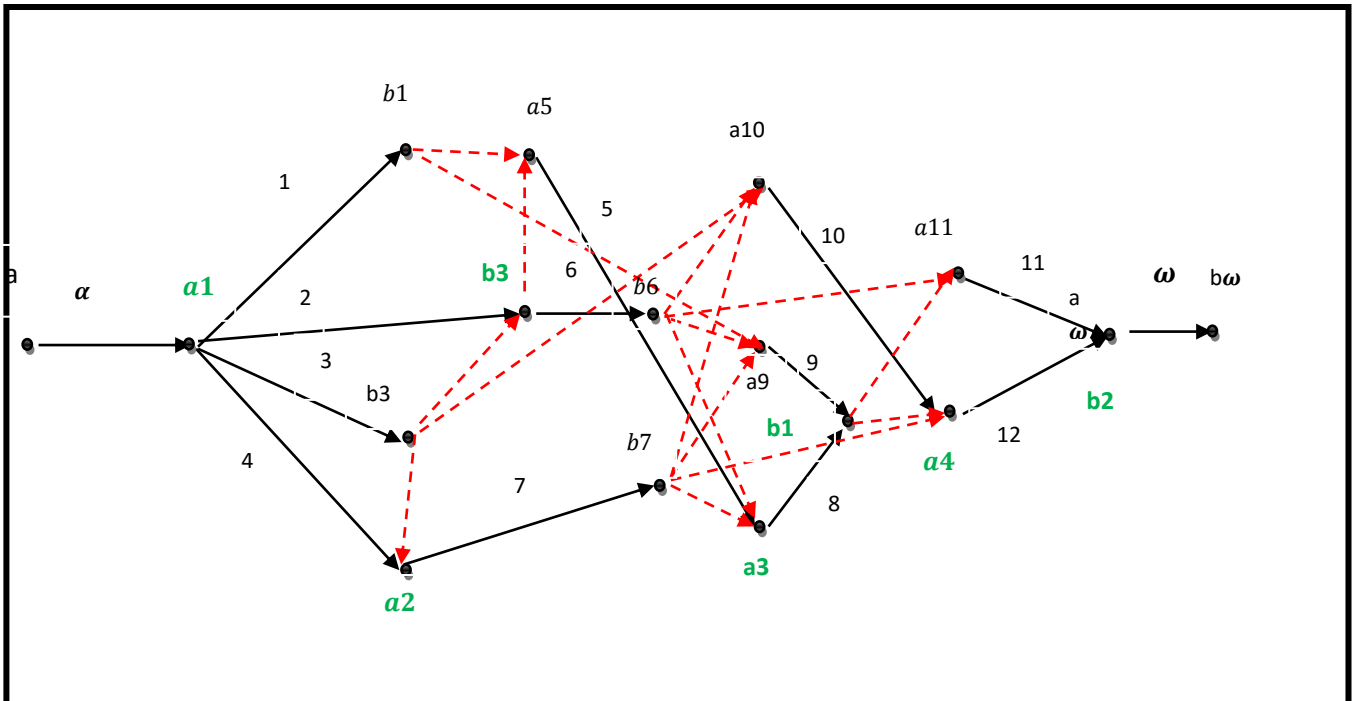
**Figure 3.16 (a)** Le graphe AoN  $G_v$  avec 14 sommets de la table d'ordonnancement 3.7



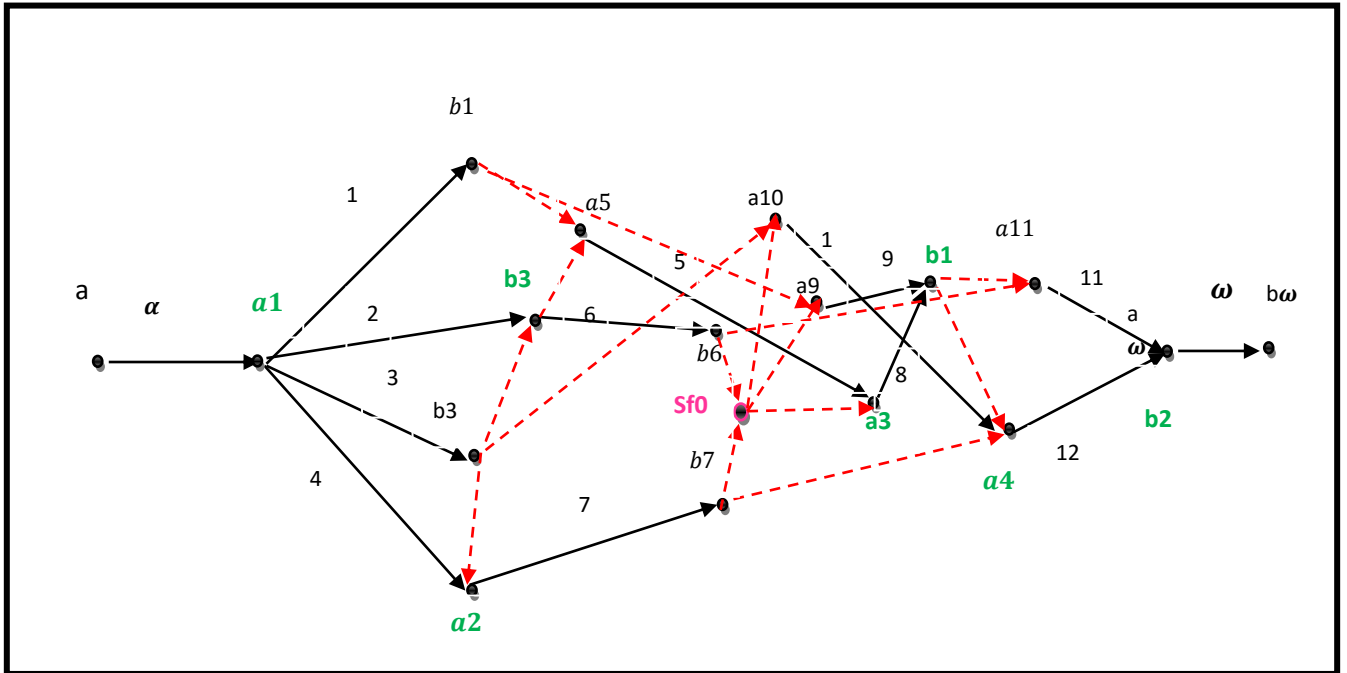
**Figure 3.16 (b)** Le graphe AoA  $G_{e1}$  de la **Figure 3.16 (a)**.  $G_{e1}$  contient 29 tâches fictives et 28 sommets.



**Figure 3.16 (c)** Le graphe AoA Ge3 de la Figure 16. (a) obtenu en appliquant la règle 1 ensuite la règle 2. Ge3 contient 23 tâches fictives et 23 sommets.



**Figure 3.16 (d)** Le graphe AoA Ge5 de la Figure 16 (a) obtenu en appliquant la règle 3 ensuite la règle 4, avec 16 tâches fictives et 17 sommets.



**Figure 3.16 (e)** Le graphe AoA minimal **Ge8** avec le sommet fictif **sf0** obtenu en appliquant les règles **5, 3** ensuite la règle **7**. **Ge8** contient **15** tâches fictives et **18** sommets.

Dans l'exemple présenté (**Figure 3.16 (a)**), *Ge1* contient  $2 | X |$  sommets et les différents chemins sont formés alternativement, d'arcs réels et fictifs (*Ge1* contient 28 sommets et 29 arcs fictifs). Pour obtenir *Ge1* il suffit d'ajouter au diagramme de Gantt les arcs de précédence (arcs rouges). Tous les réseaux *Ge1*, *Ge2* ..., *Ge8* satisfont les contraintes de la table d'ordonnement (**Tableau 3.7**). [23]

## 5. Conclusion

Ce chapitre présente une comparaison entre l'algorithme de Mouhoub et al et un autre algorithme de Sterboul et al pour générer des réseaux AoA avec un nombre minimum d'arcs fictifs. En effet, dans l'algorithme de Mouhoub et al en appliquant les sept règles R1, R2, ..., R7 dans l'ordre séquentiel, l'algorithme permet de réduire un certain nombre d'arcs fictifs jusqu'à la dernière règle qui donne le graphe AoA minimal avec le respect total des contraintes dans la table d'ordonnement cependant dans l'algorithme de Sterboul et al nous appliquant les trois étapes dans l'ordre séquentiel.

Le nouvel algorithme de Mouhoub et al est très simple à appliquer. Elle donne le AoA minimal avec un certain nombre d'arcs virtuels dans un temps très court.

Les techniques utilisées dans les sept règles de l'algorithme peuvent être exploitées dans d'autres domaines par des spécialistes de la théorie des graphes. Les résultats expérimentaux sont très positifs par rapport à l'algorithme de Sterboul et al, même lorsque les réseaux sont d'une très grande taille.

# CHAPITRE 4

## PRESENTATION D'APPLICATION ET ETUDE COMPARATIVE

### 1. Introduction

Après l'étude de partie théorique sur les principes de base sur l'ordonnement et qui est dans gestion de projet et l'étude de deux algorithmes le premier de Mouhoub et al et l'autre de Sterboul et al. Ces deux dernies faits la réduction des tâches fictives dans le graphe AoA construit à partir d'un graphe des potentiels donné ou td.et nous entamerons la phase de réalisation ou partie pratique. Nous commençons par présenter les outils et environnement de développement que nous avons utilisé, ensuite nous allons présenter les fenêtres d'application et les fonctionnalités ces fenêtres et afficher le résultat numérique de comparaison entre les deux algorithmes.

### 2. Présentation les deux algorithmes

On a présenté l'implémentation de deux algorithmes :

#### 2.1. Implémentation l'algorithme de Mouhoub et al

Pour pouvoir programmer cette méthode, il est nécessaire de la transformer en écriture algorithmique :

##### **Début**

**(Construction du graphe  $G_{e1}$  à partir du graphe  $G_v$  )**

**POUR** chaque sommet  $i$  de  $X$  **FAIRE**

-on définit deux sommets  $a_i$  et  $b_i$  ;

- on constitué des arcs  $(a_i, b_i)$  pour tout  $i$  de  $X$  ;

Ainsi que des arcs  $(b_i, a_j)$  si  $i \in P(j)$  dans  $G_v$  ;

**FIN POUR ;**

N nombre de tâches ;

**(Construction de graphe  $G_{e2}$  par la règle1 )**

I ensemble des tache qui en les même successeurs.

$I = \Phi$  ;

**POUR** allant de 1 à N **FAIRE**

**SI**  $i_p$  et  $i_{p+1}$  en les mêmes successeurs **Alors**

$I = I + i_p$  ;

$I = I + i_{p+1}$  ;

**FINSI;**

**FIN POUR;**

**POUR** chaque ensemble  $I = \{i_1, i_2, \dots, i_p\}$  **FAIRE**

Contracter dans  $G_{e1}$  les sommets  $b_{i_1}, b_{i_2}, \dots, b_{i_p}$  ayant mêmes successeurs en un seul sommet  $b_j$ ;

**FIN POUR;**

**(Construction du graphe  $G_{e3}$  par la règle 2)**

I ensemble des tâches qui ayant les mêmes successeurs.

$I = \Phi$  ;

**POUR** allant de 1 à N **FAIRE**

**SI**  $i_p$  et  $i_{p+1}$  ayant les mêmes prédécesseurs **Alors**

$I = I + i_p$ ;

$I = I + i_{p+1}$ ;

**FINSI;**

**FIN POUR;**

**POUR** chaque ensemble  $I = \{i_1, i_2, \dots, i_p\}$  **FAIRE**

Contracter dans  $G_{e2}$  les sommets  $a_{i_1}, a_{i_2}, \dots, a_{i_p}$  ayant les mêmes prédécesseurs en un seul sommet  $a_j$  ;

**FIN POUR;**

N nombre des tâches ;

**(Construction du graphe  $G_{e4}$  par la règle 3)**

**POUR** j allant 0 à N **FAIRE**

**POUR** i allant i à N **FAIRE**

**POUR** chaque arc  $(b_j, a_i)$  **FAIRE**

**SI**  $d_{ai}^- = +1$  **Alors**

Contracter  $a_i$  et  $b_j$  en un sommet unique  $ab_{ij}$  ;

**FIN SI ;**

**FIN POUR ;**

**FIN POUR ;**

**FIN POUR ;**

(Construction du graphe  $G_{e5}$  par la règle 4)

**POUR** $j$  allant 0 à N **FAIRE**

**POUR** $i$  allant i à N **FAIRE**

**POUR**chaque arc  $(b_j, a_i)$  **FAIRE**

**SI**  $d_{bj}^+ = +1$  **Alors**

Contracter  $a_i$  et  $b_j$  en un sommet unique  $ab_{ij}$  ;

**FIN SI ;**

**FIN POUR ;**

**FIN POUR ;**

**FIN POUR ;**

(Construction des graphes  $G_{e6}$  et  $G_{e7}$  par l'application d'une des règles 5, 6 ou 7)

N nombre de tâches ;

**POUR** $i$  allant 0 à N **FAIRE**

**POUR** $j$  allant i à N **FAIRE**

**Tant que** il existe un sommet  $b_l$  succes  $[b_i]$  inclus dans successeurs  $[b_j]$  **FAIRE**

- supprimer les arcs fictifs qui sortent de  $b_l$  et qui sont inclus dans les successeurs de  $b_j$ .

- remplaces par un arc fictif de  $b_i$  à  $b_j$  ;

( Par cette boucle, on obtient le graphe  $G_{e6}$  )

**Fin Taque;**

**FIN POUR;**

**FIN POUR;**

-Appliquer la règle 3 ; (pour obtenir le graphe  $G_{e7}$ )

**POUR** i allant 0 à N **FAIRE**

**POUR** j allant i à N **FAIRE**

**Tant que** il existe un sommet  $a_i$  précédé  $[a_i]$  inclus dans précédé  $[a_j]$

**FAIRE**

- supprimer les arcs fictifs qui arrivent à  $a_i$  et qui sont inclus dans les prédécesseurs de  $a_j$  ;

- remplacer par un arc fictif de  $a_i$  à  $a_j$ ; (On obtient le graphe  $G_{e6}$ )

**Fin Taque**

**FIN POUR ;**

**FIN POUR ;**

- Appliquer la règle 4 (pour obtenir le graphe  $G_{e7}$ )

$X1 = \{ b_1, b_2, \dots, b_p \}$  et  $X2 = \{ a_1, a_2, \dots, a_p \}$

**Tant que**  $X_1$  et  $X_2$  deux classes formant un biparti complet **FAIRE**

- créer un sommet fictif  $s_f$  ;

- supprimer les arcs multiples ;

(On obtient le graphe  $G_{e8}$ )

**Fin Taque**

**FIN.**

**Figure 4.1** Implémentation d'algorithme de MOUHOUB et al.[23]

## 2.2. Implémentation d'algorithme de Sterboul et al

### 2.2.1. Algorithme

1. On détermine la partition de  $\{1, \dots, N\}$  en classe  $C_1, \dots, C_L$  :  $i$  et  $j$  appartiennent à la même classe si et seulement si  $P(i) = P(j)$ .

On dresse une liste  $i_1, \dots, i_L$  constituée d'un représentant dans chaque classe.

2. On détermine la partition  $\{1, \dots, N\}$  en classe  $D_1, \dots, D_M$  :  $i$  et  $j$  appartiennent à la même classe si seulement si  $Q(i) = Q(j)$ . On dresse une liste  $j_1, \dots, j_M$

Constituée d'un représentant dans chaque la liste :

3. Pour tout  $l, 1 \leq l \leq L$ , on constitue la liste

$V(i_l) = \{j_m \mid j_m \in P(i_l), 1 \leq m \leq M\}$  et la liste :

$\tilde{V}(i_l) = \{j_m \mid j_m \in P(i_l), 1 \leq m \leq M\}$

Pour tout  $m, 1 \leq m \leq M$ , on constitue la liste :

$W(j_m) = \{i_l \mid i_l \in Q(j_m), 1 \leq l \leq L\}$

4. On pose  $m=0$ . On utilise une liste  $T$ , indexée de 1 à  $M$ , vide au départ .

**4.1** Augmenter  $m$  d'une unité ; si  $m > M$  aller en **4.5**.

Si  $W(j_m) = \emptyset$  aller en **4.1** .

Si  $|W(j_m)| = 1$ , soit  $\{i_l\} = W(j_m)$ , aller en **4.4**.

**4.2** Soit  $E = (\cap V(s)) \ll s \in W(j_m) \gg$ . S'il existe  $i_l \in W(j_m)$  tel que  $V(i_l) = E$  et  $i_l$  ne figure pas dans  $T$ , alors aller en **4.4**.

**4.3** Soit  $F = (\cap \tilde{V}(s)) \ll s \in W(j_m) \gg$ . S'il existe  $i_l \in W(j_m)$  tel que  $V(i_l) = F$  et  $i_l$  ne figure pas dans  $T$ , alors aller en **4.4** sinon aller en **4.1**.

**4.4** On pose  $T(m) = i_l$ .

Aller en **4.1**

**4.5** L'ensemble des sommets du graphe cherché  $H_2$  est l'ensemble des entiers :

$$\{2j_1, \dots, 2j_M\} \cup \{2i_l + 1 \mid i_l \text{ ne figure pas dans } T, 1 \leq l \leq L\}.$$

L'ensemble des arcs de  $H_2$  est :

Arcs réels : pour tout  $i \in \{1, \dots, N\}$ , soient  $l$  et  $m$  tels que  $i \in C_l$  et  $i \in D_m$  :

Arc  $(2i_l + 1, 2j_m)$  si  $i_l$  ne figure pas dans  $T$  ;

Arc  $(2j_k, 2j_m)$  si  $i_l = T(k)$ .

Arc virtuels : pour tout  $m \in \{1, \dots, M\}$  et tout  $i_l \in W(j_m)$  :

Arc  $(2j_m, 2i_l + 1)$  si  $i_l$  ne figure pas dans  $T$ .

Arc  $(2j_m, 2j_k)$  si  $i_l = T(k)$ ,  $k \neq m$ .

**Figure 4.2** Implémentation d'algorithme de Sterboul et al. [12]

### 3. les outils et l'environnement de développement

#### 3.1. langages de programmation "JAVA"

Java est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C. Ses caractéristiques ainsi que la richesse de son écosystème et de sa communauté lui ont permis d'être très largement utilisé pour le développement d'applications de types très disparates. Java est notamment largement utilisée pour le développement d'applications d'entreprise et mobiles.[29]

### 3.2. L'environnement NetBeans:

- Définition de NetBeans

NetBeans est un environnement open-source de développement intégré ( IDE ) pour le développement avec Java , PHP , C ++ , et d'autres langages de programmation . NetBeans est également désigné comme une plate-forme de composants modulaires utilisés pour développer des applications de bureau Java .[35]



**Figure 4.1**NetBeans 8.1

- **Les versions actuelles**

NetBeans IDE 6.0 introduced support for developing IDE modules and rich client applications based on the NetBeans platform, a Java Swing GUI builder (formerly known as "Project Matisse"), improved CVS support, WebLogic 9 and JBoss 4 support, and many editor enhancements. NetBeans 6 is available in official repositories of major Linux distributions.

NetBeans IDE 6.5, released in November 2008, extended the existing Java EE features (including Java Persistence support, EJB 3 and JAX-WS). Additionally, the NetBeans Enterprise Pack supports the development of Java EE 5 enterprise applications, including SOA visual design tools, XML schema tools, web services orchestration (for BPEL), and UML modeling. The NetBeans IDE Bundle for C/C++ supports C/C++ and FORTRAN development.[35]

NetBeans IDE 6.8 est le premier IDE pour fournir un support complet de Java EE 6 et le serveur GlassFish v3 Enterprise. Les développeurs d'hébergement de leurs projets open-source sur kenai.com bénéficient en outre de la messagerie instantanée et de suivi des problèmes d'intégration et de navigation de droite dans l'IDE, le soutien au développement d'applications Web avec PHP 5.3 et le frameworkSymfony, et le code amélioré l'achèvement, mises en page, des conseils et navigation dans des projets JavaFX.

NetBeans IDE 6.9 , sorti en Juin 2010, a ajouté le support pour OSGi , Spring Framework 3.0 , l'injection de dépendance Java EE ( JSR -299 ) , Zend Framework for PHP , et plus facile la navigation de code ( tels que " est remplacée / Mise en œuvre " annotations ) , le formatage, conseils et refactorent à travers plusieurs langues .

NetBeans IDE 7.0 est sorti en Avril 2011. Le 1er Août 2011, l'équipe de NetBeans IDE NetBeans libéré 7.0.1, qui a un support complet pour la sortie officielle de la plate-forme Java SE 7.

NetBeans IDE 7.3 est sorti en Février 2013 qui a ajouté le support de HTML5 et les technologies du web

NetBeans IDE 7.4 a été libéré le 15 Octobre 2013.

NetBeans IDE 8.0 est sorti le 18 Mars 2014.

NetBeans IDE 8.1 a été libéré le 4 Novembre, 2015.Cette dernière version qui est utilisée pour développer notre logiciel.[35]

- **Pourquoi NetBeans ?**

NetBeans ont beaucoup d'importance, nous pouvons le reprendre dans ce qui suit :

- NetBeans est simple à télécharger et à installer.
- Libre et Open Source.
- Prise en charge des normes et de plates-formes Java.
- Dynamique Langage Support.
- Avoir une JVM (java pratiquement la machine).[35]

### 3.3. Db4o base de données :

#### 3.3.1. Base de données " db4o"

db4o est un système de gestion de base de données orientée objet Open Source pour des applications Java et .Net.

db4o signifie "database for objects", c'est-à-dire. Base de données Orientée Objet. C'est un système de gestion de base de données orienté objet (OODBMS ou SGDBOO en français) écrit en Java et en .NET et de ce fait destiné à ces deux plateformes. Ce logiciel a été initialement publié en 2001 par db4Objects, Inc., et depuis lors s'est taillé une part majeure du marché de ce qu'on peut appeler les bases de données objet de deuxième génération. Il est disponible sous deux licences ("dual licensing model"), une licence OpenSource de type GPL pour l'utilisation personnelle et non-commerciale mais aussi une licence commerciale. Dans cet article, nous présenterons tout d'abord db4o. Dans le chapitre OODBMS et db4o, nous en présenterons les avantages, limites et différences d'un point de vue conceptuel mais aussi selon une approche de développeur. Puis nous ferons une présentation de la thèse d'un étudiant nommée db4o2D. Enfin nous concluons sur les perspectives du projet db4o2D.[28]

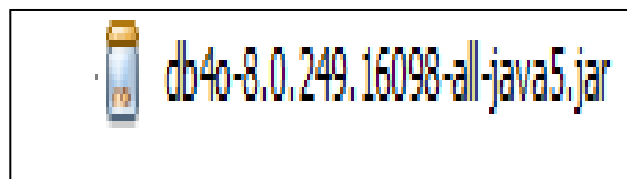
#### 3.3.2. Placement de db4o dans le monde :

db4o représente une base de données d'objets | modèle de base de données orientée objet. Un de ses principaux objectifs est de fournir une interface simple et native à la persistance de programmation orientée objet langues. Développement avec base de données db4o ne nécessite pas une création de modèle de données séparé, le modèle de classe de l'application définit la structure des données dans la base de db4o. db4o tente d'éviter la objet-relational désadaptation d'impédance | objet / relationnelle désadaptation d'impédance en éliminant la couche relationnelle d'un projet de logiciel. Les développeurs utilisant base de données relationnelle s peuvent également bénéficier de l'aide db4o, qui peut être considéré comme un outil complémentaire. L'échange de données db4o-SGBDR peut être implémentée en utilisant système de réplication db4o (DRS). DRS peut également être utilisé pour la migration entre l'objet (db4o) et relationnelle ([[SGBDR]]) technologies. .[28]

En tant que base de données embarqué db4o peut être exécuté dans le processus de l'application. Il est distribué comme une bibliothèque (JAR / dll).

### 3.3.3. Avantage de db4o :

- Pas possible de faire une recherche des objets qui ont la valeur par défaut des types primitifs.
- Des conditions ne sont pas exprimables (pas d'autres comparaison que l'égalité par exemple).
- Un constructeur qui accepte les valeurs **null** doit exister.
- Impossible de faire une recherche polymorphe sous une classe abstraite ou une interface (car pas possible de construire l'objet exemple).[7]



**Figure 4.2** Db4o dossier jar de base de données.

#### 4. Présentation les interfaces de l'application

**Recherche graph**

Saisie le numero de graph:  **Recherche**

**Retourne au Menu**

**Resultat de recherche**

Sommet	sex	pred

**Figure 4.3** Recherche un graphe

**Modifier graph**

Saisie le nom de graph :  **Afficher cette graph**

**Modifier**   **Annuler**   **Retourne au Menu**

**Resultat de recherche**

Sommet	sex	pred

**Figure 4.4** Modifier un graphe





Figure 4.7 Mise à jour de base de donnés

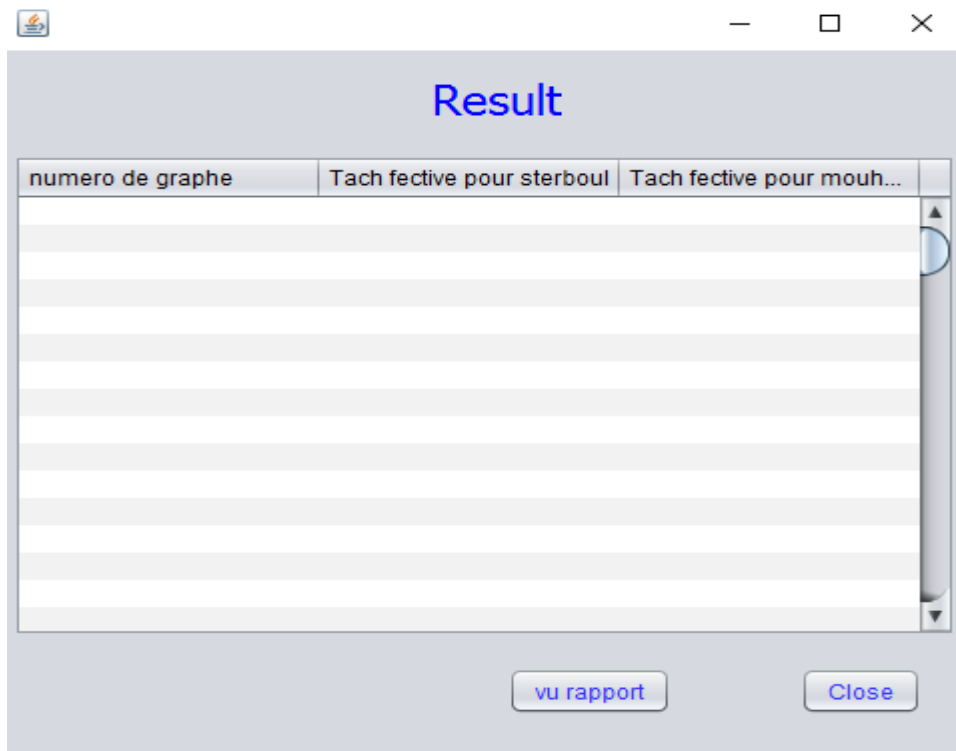


Figure 4.8 Fenêtre de comparaisons

24/05/2016

### Rapport de comparaison

Graph	nbrs_GE1	nbrs_GE8	nbrs_h0	nbrs_h2	arc_f_m	arc_f_s	meilleur
1	52	35	52	35	35	51	Ge8
2	76	23	76	23	23	11	Ge8=H2
3	66	38	66	39	34	47	Ge8
4	40	20	40	22	17	19	Ge8
5	28	17	28	17	15	17	Ge8=H2
6	54	32	54	33	31	37	Ge8
7	14	8	14	8	4	8	Ge8=H2
8	82	32	82	32	27	32	Ge8=H2
9	8	6	8	6	2	3	Ge8=H2
10	94	50	94	52	49	62	Ge8

Figure4.9 Rapport de comparaison

## 4. Conclusion

Dans ce chapitre nous avons présenté une étude comparative entre deux algorithmes Sterboul et al et Mouhoub et al. Pour comparer ces algorithmes nous avons fait une implémentation avec le langage Java et la base de données DB4O, puis on traite 100 graph, quand nous affichons le résultat dans un rapport Word on voit que l'algorithme de Mouhoub et al est très efficace dans la minimisation des tâches fictifs et des sommets fictifs.

## CONCLUSION GENERALE

Après avoir étudié les méthodes de modélisation de l'ordonnancement de projet par le biais du diagramme de Gantt, la méthode AoN et la méthode AoA,

La conclusion qu'il faut tirer est que les praticiens préfèrent utiliser la méthode AoA malgré que le graphe de la méthode AoN est unique et facile à dessiner alors que les graphes AoA sont infinis, difficiles à dessiner et la majorité des praticiens n'arrivent pas à les dessiner correctement. Cependant, la présence des activités fictives les rend encore plus difficile à être abordé surtout lorsque leur nombre est très élevé avec des réseaux de tailles très grandes.

Le problème de réduction des tâches fictives dans le graphe AoA nous a ouvert les portes pour chercher une solution radicale au problème appelé dans la littérature « le graphe AoA minimal ». Le contexte est devenu plus clair et plus général tandis que l'objectif devient plus précis mais plus complexe.

L'objectif du travail donc est de faire une étude comparative qui a été réalisée entre deux algorithmes célèbres.

Le premier est un algorithme de Mouhoub et al et le deuxième de Sterboul et al. Les deux algorithmes permettent de réduire un certain nombre d'arcs fictifs jusqu'à la dernière règle qui donne le graphe AoA minimal avec le respect total des contraintes dans la table d'ordonnancement.

Cette étude montre la supériorité de l'algorithme de Mouhoub et al dans la majorité des cas. Car le problème de réduction des tâches est vaste, il a de future perspective comme l'amélioration de ces deux algorithmes Mouhoub et Sterboul pour retourner des résultats exacte au minimiser le nombre des étapes et améliorer son efficacité.

# BIBLIOGRAPHIE

## Un ouvrage :

- [1] C.P.Vincent, " Heuristique - Création, intuition, créativité et stratégies d'innovation", BOD - Books on Demand France, 2012.
- [2] C.Yves , rocherche Opérationnelle "ordonnancement" , 26/02/2007
- [3] J.CARLIER & P. CHRETIENNE, Problèmes d'ordonnancement, modélisation, complexité, algorithmes, MASSON, ISBN 2-225-81275-6.
- [4] R.Christophe et T. Denis, Théorie de la Complexité, Notes de cours, ENSGI – INP Grenoble. 2002.
- [5] R.DIESTEL, Graph theory, Springer-Verlag, Heidelberg Graduate Texts in Mathematics, Volume 173. ISBN 978-3-642-14278-9, Springer-Verlag, 2000.

## Un article :

- [6] A. BENHOCINE, Théorie des graphes et applications, Cours de post-graduation, Université de Sétif, 2001.
- [7] A. Giorgetti et F. Sesailhan , Programmation java (java et bases de données) , laboratoire d'informatique de l'université de Franche-Comté.
- [8] A. Wahiba, Le problème conjoint de l'ordonnancement de la production et de planification de la maintenance , Boumerdes , 2014.
- [9] D. Tchamidéma , A.EL MONTASSIR A ,L.GADEDJI ,E.KEKEH Komi ,Management de projets Méthodes GANTT et PERT, 2007.
- [10] D. G. Corneil, C.C. Gotlieb, and Y.M. Lee, Minimal event-node network of project precedence relations, Communications of the SCM, 16, pp.296-298, 1973.
- [11] F. ECOTO, Initiation à la recherche opérationnelle, ELLIPES, Paris, France, 2011.

- [12] F. STERBOUL, & D. WERTHEIMER, Comment construire un graphe PERT minimal, RAIRO, France, 1980.
- [13] J. Alliot, Nicolas Durand, «Algorithmes génétiques» March 14, 2005.
- [14] N. E. MOUHOU, H. BELOUADAH & A. BOUBETRA, A new algorithm for modeling temporal constraints and constructing AoA graph, Alger. 2011
- [15] T. J. KLOPPENBORG, Contemporary Project Management, South-western CENGAGE Learning, Mason, OH, 2009.
- [16] X. KONG, J. SUN & W. XU, Permutation-based particle swarm algorithm for tasks scheduling in heterogeneous systems with communication delays. International Journal of Computational Intelligence Research, 4(1):61.70, 2008.

**Un mémoire ou une thèse :**

- [17] A. LAYEB, Utilisation des Approches d'Optimisation Combinatoire pour La Vérification des Applications Temps Réel. Thèse de Doctorat, Université Mentouri de Constantine 2010.
- [18] A. YEZZA, Concepts Les abréviations *AON* et *AOA* signifient successivement , *thèse* étendu Méthodes d'évolution de modèle produit dans les systèmes, Thèse de Ph. D, 2012.
- [19] F. AOUACHE, M. BELHARET, Mémoire de Master, Planification et gestion de projets.2012.
- [20] H. HOUARI, Planification et Ordonnancement en temps réel d'un Job shop en utilisant l'Intelligence Artificielle, Thèse de Magister, Tlemcen, 2012.
- [21] L. SMOCH, Méthodes d'Optimisation Thèse de Licence Professionnelle Logistique, Université du Littoral - Côte d'Opale, Pôle Lamartine, 2011.
- [22] L. Zaourar. Recherche opérationnelle et optimisation pour la conception testable de circuits intégrés complexes. Thèse de doctorat, Grenoble, 2010.
- [23] N. E. MOUHOU, Algorithmes de construction de graphes dans les problèmes d'ordonnancement de projet, Thèse de doctorat, Université de Sétif ,Algérie, 2011
- [24] N. TAGHEZOUT, Conception et Développement d'un système multi-agent d'Aide à la Décision pour la gestion de production dynamique. Thèse doctorat, Toulouse, 2011

[25] R. Pascal, Algorithmes Génétiques Hybrides en Optimisation Combinatoire, Thèse Ph.D, Ecole Normale Supérieure de Lyon, 1999.

[26] T. Mehenni, utilisation des métaheuristiques pour résoudre un problème d'ordonnancement sur machine à contrainte de ressource non renouvelable, Thèse de Magister, M'sila, 2006.

**Un site web :**

[27] Bruno bachelet : [www.bruno.bachelet.net/](http://www.bruno.bachelet.net/) , consulter le 25/04/2016.

[28] Developer: [www.developer.db4o.com/Resources/default.aspx](http://www.developer.db4o.com/Resources/default.aspx) , consulter le : 01/04/2015.

[29] Jmdoudoux.developpez: [www.jmdoudoux.developpez.com](http://www.jmdoudoux.developpez.com) , consulter le : 12/05/2016.

[30] Gestion-projet-informatique.vivre-aujourd'hui : [www.gestion-projet\\_informatique.vivre-aujourd'hui.fr](http://www.gestion-projet_informatique.vivre-aujourd'hui.fr) , consulté le : 17/03/2016.

[31] Gestion de projet, [www.gestiondeprojet.net](http://www.gestiondeprojet.net) , consulter le : 01/04/2016.

[32] Logistique conseil, [www.logistiqueconseil.org](http://www.logistiqueconseil.org) , consulté le : 23/25/2016.

[33] Project issimo , [www.projectissimo.com](http://www.projectissimo.com) , consulté le : 23/05/2016.

[34] Ressources.auneg, [www.ressources.auneg.fr](http://www.ressources.auneg.fr) , consulter le : 21/03/2016.

[35] Wikipedia, [www.wikipedia.org.fr](http://www.wikipedia.org.fr) , consulté le : 05/05/2016.

## ملخص

بعد دراسة نماذج جدولة المشاريع من خلال مخطط غانت و AoA و AoN نجد أن مسيري المشاريع يفضلون العمل بطريقة AoA على الرغم من صعوبة تحقيقها. في حين ان طريقة AoN سهلة و أكثر بساطة. وأعقب عرض هذه الطريقة مظاهر تثبت أنها توفر الحد الأدنى من الرسم البياني الزراعة من حيث وظائف وهمية . أن مشكلة الحد من فرص العمل الوهمية في الرسم البياني AoA فتحت الأبواب لإيجاد حل جذري لمشكلة المشار إليها في الميدان " الرسم البياني AoA للحد الأدنى " الهدف من عملنا هو المقارنة بين إثنين من الخوارزميات المشهورة الأولى ل "موهوب و آخرون " و الثانية ل" ستاربول و آخرون" و تشير النتائج في النهاية إلى أفضلية و تفوق خوارزمية "موهوب و آخرون " .

## Résumé

Après avoir étudié la modélisation de l'ordonnancement de projet par le biais du diagramme de Gantt, la méthode AoN et la méthode AoA, on conclue que les managers de projet préfèrent travailler avec le graphe AoA malgré qu'il est difficile à réaliser, alors que le graphe des potentiels offre plus de simplicité. La présentation de cette méthode a été suivie d'une démonstration prouvant qu'elle offre le graphe AoA minimal en termes de tâches fictives. Le problème de réduction des tâches fictives dans le graphe AoA nous a ouvert les portes pour chercher une solution radicale au problème appelé dans la littérature « le graphe AoA minimal ». Ce travail présente deux algorithmes de construction qui sont comparée entre, premier algorithme de Mouhoub et al et l'autre algorithme de Sterboul et al connue dans la littérature. Les résultats de cette comparaison montrent la supériorité d'algorithme de Mouhoub et al du point de vue efficacité.

## Abstract

After the study of the modification of ordonnancement of project by bias of Gantt diagramme , the AoN method and AoA ,are conclude that managers of project prefer work by AoA graph despite its difficult of relation ,so the potential of graph offer a lot of simplicity. Presentation of this method has been followed by demonstration prove offered by AoA minimal with terms of factitive task .the problem of reduction of fictitious tacks in AoA graph we have open the door for search radical solution for the problem named in the literature "minimum AoA graph " this work present two Algorithms of construction with are compared between conque in the literature ,Results of this comparison show the superiority of mouhoub algorithm in efficacy point of view.