

بسم الله الرحمن الرحيم
الحمد لله رب العالمين
والصلاة والسلام على سيدنا محمد أشرف المرسلين

A la mémoire de mon grand père

UNIVERSITE FERHAT ABBAS - SETIF
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE

THESE

Présentée pour l'obtention du diplôme de

DOCTORAT EN SCIENCES

SPECIALITE : Informatique

Par

MOUHOUB NASSER EDDINE

**Thème : Algorithmes de construction de graphes dans les problèmes
d'ordonnement de projet**

Soutenue le : / /2011

devant le jury composé de :

Président : Mr. A. Boukerram Maître de conférences, Université de Sétif

Examineurs :

MM. O. Kazar Maître de conférences, Université de Biskra

R. Maameri Maître de conférences, Université de Constantine

M. Touahria Maître de conférences, Université de Sétif

Rapporteur : Mr. H. Belouadah Professeur, Université de M'sila

Remerciements

Ce manuscrit est l'aboutissement de plusieurs années de recherche, mais c'est également le fruit de discussions, d'apprentissages et de moments de détente avec différentes personnes que j'aimerais remercier ici.

Je tiens tout d'abord à exprimer ma profonde gratitude envers Si Hocine Belouadah pour m'avoir encadré, conseillé, orienté et soutenu tout au long de ma thèse et pour m'avoir guidé et aidé lors de l'écriture de ce manuscrit.

Je suis également reconnaissant envers Si Abdelhamid Benhocine avec qui j'ai travaillé pendant plus de trois années dans le cadre de ma thèse de magistère, ensuite dans ma thèse de doctorat et auprès duquel j'ai beaucoup appris et surtout pour ses judicieux conseils.

Mes respects et ma gratitude vont également aux membres de jury qui m'ont fait l'honneur de juger ce travail.

Je remercie aussi tous les enseignants, du primaire jusqu'à l'université sans oublier ceux de la zawiyya de Bordj Bou Arreridj, qui ont contribué à ma formation.

Mes remerciements vont également à mes étudiants ingénieurs et surtout à Mlle Hajab Hayet pour l'aide qu'ils m'ont apporté pour la réalisation de ce travail.

Sommaire

Remerciements	i
Résumé de la thèse	ii
Abstract of the thesis	iii
Sommaire	5

Chapitre 1 Introduction générale	10
I Introduction	10
II Contexte et problématique	12
III Contributions de cette thèse	14
IV Plan de la thèse	15
Chapitre II Eléments fondamentaux	17
I Introduction	17
II Problèmes d'optimisation classiques	18
III Complexité	18
III.1 Complexité d'un algorithme	19
III.2 Algorithme efficace	20
III.3 Problème de reconnaissance	20
III.4 Problèmes faciles et problèmes difficiles	21
III.4.1 La classe NP	21
III.4.1.1 La classe P	21
III.4.1.2 La classe NP-Complet	22
III.4.1.5 La classe NP-Difficile	22
IV La fonction ordonnancement	22

IV.1	Définition	23
IV.2	Les tâches	23
IV.3	Les ressources	24
IV.4	Les contraintes	26
IV.4.1	Les contraintes potentielles	26
IV.4.2	Les contraintes disjonctives	26
IV.4.3	Les contraintes cumulatives	26
IV.5	La fonction économique	27
IV.6	Caractéristiques générales des ordonnancements	28
IV.6.1	Ordonnements statique et dynamique	28
IV.6.3	Ordonnements actifs et semi-actifs	28
IV.6.4	Ordonnements sans retard	29
IV.6.5	Ordonnement préemptif et non préemptif	29
V	Classification des problèmes d'ordonnement	29
VI	Notations	30
VII	Méthodes de résolution des problèmes d'ordonnement	32
IX	Conclusion	33
Chapitre III	L'ordonnement de projet	35
I	Introduction	35
II	La gestion de projet	36
II.1	Définition	36
III	Le Projet	37
III.1	Caractéristiques d'un projet	37
III.2	Acteurs du projet	38
III.3	Cycle de vie d'un projet	38
III.3.1	La planification du projet	39
III.3.2	Les éléments du plan	40
III.3.3	Le calendrier d'activités	42
III.3.3	Création d'un calendrier d'activités	43
III.4	L'ordonnement dans la gestion de projet	44
IV	Le problème central de l'ordonnement	44
IV.1	Modélisation du problème central	44
IV.1.1	Le diagramme de Gantt	45
IV.1.1.1	Avantages du diagramme de Gantt	46
IV.1.1.2	Inconvénients du diagramme de Gantt	46
IV.1.2	Le graphe des potentiels	47
IV.1.3	Le graphe PERT	49
IV.1.3.1	Dessin du graphe PERT	51
IV.1.3.2	Notion de tâche fictive	53
IV.1.4	Modularité	55
IV.1.5	Comparaison	56
V	Résolution du problème central	56
V.1	Dates et marges associées à une tâche	58

V.2 Calcul des dates et marges	59
V.3 Le chemin critique	62
VI Conclusion	65
Chapitre IV Extensions du problème central	66
I Introduction	66
II Ordonnancement avec des durées incertaines	67
II.1 Évaluation des durées des tâches	69
II.1.1 La fonction Bêta (β)	70
II.2 Discussion	73
II.2.1 Critiques de l'approche PERT probabiliste	74
II.2.2 Quelques suggestions	77
II.3. Résolution par simulation	79
III Le PERT coût	80
III.1 Diminution du coût total du projet	82
III.2 Diminution de la durée au moindre coût	82
IV Ordonnancement de projet à moyens limités	83
V Ordonnancement avec contraintes disjonctives	84
VI Ordonnancement multi-projet	84
VII Conclusion	85
Chapitre V Construction d'un graphe PERT à partir du graphe des potentiels	86
I Introduction	86
II Liens entre le graphe des potentiels et le graphe PERT	87
III Le graphe adjoint de graphe	87
II.1 Définition	88
II.2 Exemple	88
II.3 Le problème inverse	89
II.3.1 Définition de la configuration « Z » et « Δ ».	90
II.4 Quelques caractérisations des graphes adjoints	90
IV Construction du graphe PERT à partir du graphe des potentiels	93
III.1 Cas où le graphe des potentiels est un graphe adjoint	93
III.2 Cas où le graphe des potentiels n'est pas un graphe adjoint	95
III.3 Algorithme 1	96
III.3.1 Application	97
III.3.2 Discussion	99
V Liens entre le diagramme de Gantt et le graphe PERT	99

VI Liens entre le graphe des potentiels et le diagramme de Gantt	100
VII Liens entre un graphe PERT et le graphe des potentiels	101
VIII Conclusion	101
Chapitre VI Nouvelle approche optimisée dans la construction du graphe PERT	102
I Introduction	102
II Première technique de réduction des tâches fictives	102
II.1 Algorithme 2	104
II.2 Exemple	105
III Deuxième technique de réduction des tâches fictives	106
III.1 Algorithme 3	107
III.2 Exemple	108
IV Etude comparative	108
V Conclusion	111
Chapitre VII Le graphe PERT minimal	112
I Introduction	112
II Etat de l'art	113
II.1. Algorithme de F. STERBOUL et D. WERTHEIMER	114
II.1.1 Notations et définitions	114
II.1.2 Graphe arc-dual d'un graphe donné	115
II.1.3 Construction	115
II.1.4 Exemple	116
III Nouvel algorithme de construction du graphe PERT minimal	118
III. 1 Notations et définitions	118
III. 2 Construction	119
III.3 Démonstration	124
III.2.1 Lemme 1	125
III.2.2 Lemme 2	126
III.2.3 Lemme 3	126
III.2.4 Théorème	126
III.4. Exemple de projet de construction d'un building industriel	127
IV. Implémentation	130
IV. Résultats numériques	132
V Conclusion	135
Chapitre VIII Conclusions et perspectives	136

I Introduction	136
II Bilan	136
III Perspectives	138
Bibliographie	141
Annexe 1 Liste des tableaux	156
Annexe 2 Liste des figures	160
Annexe 3 Loi normale réduite	

Chapitre I

Introduction générale

Sommaire

I Introduction	10
II Contexte et problématique	12
III Contributions de cette thèse	14
IV Plan de la thèse	15

I Introduction

Depuis la deuxième guerre mondiale, une discipline –qui doit à ses origines militaires de s'appeler « **Recherche Opérationnelle** ». Elle consiste en l'application de méthodes scientifiques pour résoudre les problèmes complexes rencontrés dans la direction et la gestion de grands systèmes d'hommes, de machines, de matériaux, et d'argent dans l'Industrie, le Commerce, l'Administration et la Défense. La caractéristique de l'approche est le développement d'un modèle scientifique (incluant la mesure de facteurs tels que le hasard et le risque) avec lequel on tente de prévoir et de comparer les résultats de diverses décisions ou stratégies. Le but est d'aider le décideur à déterminer sa politique de manière scientifique.

Les domaines d'application de la Recherche Opérationnelle se situent donc autour de l'aide à la décision, l'amélioration de la productivité, la gestion de production, l'optimisation dans les choix stratégiques des organisations. Une caractéristique tout à fait particulière de la Recherche Opérationnelle réside dans le fait que la modélisation fait partie intégrante de la démarche scientifique, en opposition avec d'autres branches des mathématiques appliquées (Equations aux dérivées partielles, Economie Mathématique) où les modèles à étudier sont formulés à partir d'autres disciplines (Mécanique, Physique, Economie). La Recherche Opérationnelle a donc une vocation interdisciplinaire très marquée autour des Mathématiques

Appliquées mais aussi de l'Economie, des Sciences de Gestion, sans oublier les relations très fortes avec l'Informatique Scientifique.

La modélisation est une étape cruciale dans l'élaboration du schéma de résolution d'un problème appartenant aux domaines d'application la RO. Elle est la conception d'un modèle. Selon son objectif et les moyens utilisés, elle est dite mathématique, géométrique, informatique...

Un modèle se rapporte toujours à ce qu'on espère en déduire. En mathématiques et en informatique, un modèle est une traduction de la réalité pour pouvoir lui appliquer les outils, les techniques et les théories mathématiques, puis généralement, en sens inverse, la traduction des résultats obtenus en prédictions ou opérations dans le monde réel.

Un modèle n'est jamais parfait, ni totalement représentatif de la réalité : le choix des paramètres, des relations et outils qui les lient éclaire la finalité. Au sein d'un même modèle, le choix des valeurs des paramètres peut permettre d'appréhender divers aspects, ou encore des réalités différentes.

Sans entrer dans les détails de la théorie de la modélisation, il faut noter que cette branche a ouvert les portes pour bien comprendre les problèmes ensuite chercher les solutions.

Parmi les outils les plus puissants de la modélisation mathématique et informatique on cite **la théorie des graphes**.

Un graphe permet de représenter la structure, les connexions d'un ensemble complexe en exprimant les relations entre ses éléments : réseau de communication, réseaux routiers, interaction de diverses espèces animales, circuits électriques, . . .

Les graphes constituent donc une méthode de pensée qui permet de modéliser une grande variété de problèmes en se ramenant à l'étude de sommets et d'arcs (représentation binaire). Les travaux en théorie des graphes sont souvent effectués par des informaticiens, du fait de l'importance qu'y revêt l'aspect algorithmique.

La façon dont le graphe est représenté en tant qu'objet mathématique est très simple mais **l'algorithmique** qui en découle de cette représentation sert de support à la résolution d'un vaste échantillon de problèmes.

De nombreux outils formels ou théories ont été développés pour décrire les algorithmes, les étudier, exprimer leurs qualités, pouvoir les comparer entre eux d'où la naissance de la théorie de la complexité.

Cette association fabuleuse entre le graphe et l'algorithmique est en évolution constante tant du point de vue des recherches fondamentales que de celui des applications. Elle se donne pour objet d'étude et pour champ d'application l'optimisation des décisions en matière d'organisation, de gestion, de choix techniques ou technologiques dans l'industrie au sens large.

La **gestion de projet**, quel que soit la nature et taille d'un projet, est une composante très importante de la communauté de recherche opérationnelle et de la théorie des graphes.

Un projet est composé de plusieurs activités qui commencent par l'étude d'opportunité et de faisabilité et qui se terminent par la clôture du projet. La planification est la clé de voûte de tout projet mené à terme dans de bonnes conditions. A l'intérieur de cette phase cruciale apparaît l'ordonnancement des tâches.

Un **problème d'ordonnancement** consiste à organiser dans le temps la réalisation de tâches ou les opérations, compte tenu d'un certain nombre de contraintes afin d'atteindre un certain objectif appelé fonction économique.

L'ordonnancement joue un rôle essentiel dans de nombreux secteurs d'activités : la conception (de bâtiments, de produits, de systèmes, . . .), l'administration (gestion d'emplois du temps, gestion du personnel), l'industrie (gestion de la production), l'informatique (ordonnancement de processus, ordonnancement de réseaux). Les méthodes d'ordonnancement foisonnent dans la littérature. Elles se différencient par la nature du problème considéré (nombre de ressources, structure particulière du problème, . . .), la nature des contraintes prises en compte, les objectifs à satisfaire (minimisation des coûts, de la durée totale de mise en œuvre, . . .) et la nature de l'approche de résolution adoptée (heuristiques, méthodes exactes, métaheuristiques, approches par contraintes, . . .). [LA 05]

II Contexte et problématique

Dans son ouvrage « Initiation à la recherche opérationnelle », Ecoto [ECO 11] présente une dizaine d'arguments pour la nécessité d'apprendre les techniques d'ordonnancement 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'ordonnancement est plus que nécessaire pour mener à terme un projet.

Les méthodes d'ordonnancement existantes peuvent se regrouper en deux catégories selon leur principe de base :

- les méthodes de type diagramme de Gantt ;
- les méthodes basées sur la modélisation par les graphes à savoir la méthode des potentiels Métra (MPM) et la méthode PERT/CPM.

Jusqu'au début des années soixante, on utilisait le diagramme de Gantt ou diagramme à barres pour traiter les problèmes d'ordonnancement de projet. Ensuite, deux méthodes se sont développées parallèlement fondées sur la modélisation par les graphes : la méthode française appelée méthode des potentiels (Métra) ou méthode potentiels-tâches alors que la méthode américaine est appelée PERT/CPM ou méthode potentiels-étapes. Ces méthodes sont devenues lors des outils de gestion indispensables.

Cette thèse fait un tour d'horizon sur l'ensemble des méthodes d'ordonnancement et plus précisément celles de l'ordonnancement de projet. Elle présente les avantages et les inconvénients de chacune d'elles.

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

Malgré tout ça, selon Fink et al. [FIN 02], le réseau PERT/CPM est plus concis. En outre, Hendrickson et al. [HEN 08] explique qu'il est à proximité du célèbre diagramme de Gantt et selon Cohen Yuval et al. [CS 07], la structure du réseau PERT est beaucoup plus adaptée à certaines techniques d'analyse et d'optimisation des formulations. Donc, est-ce qu'il y a un moyen de passer d'un graphe facile, qui est le graphe de la méthode MPM, vers le graphe PERT/CPM

qui sera correct et qui respecte les contraintes de départ, à savoir les contraintes de la table d'ordonnancement? C'est un premier axe de recherche.

Deuxième problème rencontré dans cette recherche c'est la nature stochastique des durées des tâches du projet. Malgré que la méthode PERT tient en compte ce problème en proposant une solution qui se base sur la collecte d'un ensemble d'informations pessimistes, optimistes et les plus probables des durées des tâches chez les conducteurs des travaux; cette solution reste loin des valeurs réelles des durées sur le terrain; de plus, elle est entachée de plusieurs erreurs. Y a-t-il un moyen de rectifier le tir et apporter les corrections nécessaires à cette merveilleuse méthode qui a bouleversé les pronostics en faveur des Etats Unis d'Amérique en pleine guerre froide contre l'ex URSS ? Et nous avons deuxième axe de recherche.

Malgré que le graphe PERT est meilleur, du point de vue pratique, que le graphe des potentiels, le nombre important des tâches fictives dans le graphe le rend difficile à lire et à appliquer lors de la phase suivie du projet. Il y a-t-il un moyen de réduire ces tâches fictives au maximum ? Autrement dit est ce qu'on peut trouver un algorithme de recherche du graphe PERT minimal en terme de tâches fictives meilleur que ceux qui se trouvent dans la littérature et qui offre aux managers de projet une méthode facile à appliquer ? C'est un troisième axe de recherche.

III Contributions de cette thèse

La thèse présente deux techniques nouvelles et originales de dessin de graphe PERT/CPM : la première utilise une technique simple et facile de dessin, elle est plutôt pédagogique en prenant la table d'ordonnancement comme référence à tout instant de la technique. Le dessin se fait alors étape par étape en dessinant à chaque fois un seul arc représentant une ligne de la table d'ordonnancement. La seconde est plus complexe, mais facile à appliquer et à implémenter. Elle fait intervenir les notions de graphes adjoints de graphes et à partir du graphe MPM, qui est unique, et s'il ne contient pas un sous graphe formant un 'Z', la méthode le décompose en graphes bipartis complets, dans le cas où il contient la configuration 'Z', l'algorithme crée des arcs fictifs pour le ramener au premier cas. Après une série de transformations, le graphe PERT est construit. A la fin de l'application de l'algorithme, il propose une vérification en s'assurant que le graphe résultant respecte les contraintes de la table d'ordonnancement. Cette technique ne s'arrête pas ici, elle est optimisée à deux fois : une par [AKR 07] et l'autre par nous-mêmes. Les deux techniques d'optimisations font l'objet d'une étude comparative qui conclue que notre solution est plus efficace.

La contribution suivante dans cette thèse réside dans un ensemble de propositions et de suggestions pour l'amélioration du PERT stochastique après avoir analysé et critiqué cette méthode pour la recherche des valeurs des durées qui ne sont qu'approximativement.

L'un des inconvénients majeurs de la méthode CPM/PERT c'est l'existence des tâches fictives ajoutées au graphe. L'introduction ces tâches permet de solutionner 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 PERT. 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 PERT. Ce domaine reste toujours un sujet d'actualité puisque sa complexité est *NP-difficile*. Notre contribution en ce point précis ne sort pas de cette logique et propose une technique de réduction.

Elle consiste en la recherche d'une approche nouvelle pour la construction de graphe PERT à partir d'un graphe des potentiels avec le minimum d'arcs virtuels et la comparer avec un algorithme bien connu dans la littérature.

IV Plan de la thèse

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

Dans le chapitre II, dans un premier temps on donne quelques définitions sur les problèmes combinatoires classiques, des algorithmes et de leur complexité. Nous abordons ensuite l'étude de la fonction ordonnancement avec ses éléments de base ainsi que ses caractéristiques générales. Le chapitre dresse une classification des problèmes d'ordonnancement avec les notations utilisées et il est clôturé par la présentation des méthodes de résolution.

Le chapitre III traite de l'état de l'art concernant l'ordonnancement de projet. Nous présentons tout d'abord quelques définitions sur la gestion de projet et où se situe exactement la phase ordonnancement. Par la suite nous étudierons le problème central de l'ordonnancement et on présente les méthodes courantes appliquées pour le résoudre (notamment celles du chemin critique PERT, MPM, etc.). Le chapitre sera clôturé par l'étude du PERT temps ainsi que les avantages qu'apporte cette technique pour réaliser le projet en un minimum de temps.

Le chapitre IV traite le problème du PERT stochastique. Il s'agit de chercher la détermination des durées des tâches du projet qui ne sont pas connues d'une manière exacte. On présentera d'abord, la méthode avec ses trois paramètres : optimistes, pessimistes et les plus probables, ensuite nous présenterons un certain nombre de propositions qui apporteront les corrections nécessaires à la dite méthode. On verra également les extensions de la méthode PERT à savoir le PERT coût qui offre une double technique de recherche du coût minimum d'un projet.

Nous avons réservé le chapitre V pour la présentation d'une nouvelle approche de construction du graphe PERT à partir d'un graphe des potentiels donné en utilisant un certain nombre de concepts et définitions des graphes adjoints de graphes. Le chapitre est couronné par la présentation de l'algorithme résultant.

Le chapitre VI se focalise sur l'optimisation de l'algorithme présenté dans le chapitre précédent où une première amélioration de réduction des tâches fictives a été dressée suivie d'un deuxième algorithme d'amélioration et finalisé par une étude comparative entre les deux approches d'optimisation.

Le chapitre VII sera consacré à l'étude du graphe PERT minimal. On présentera d'abord l'état de l'art du problème de construction du réseau PERT minimal qui sera suivi d'une étude détaillée d'un algorithme connu dans la littérature, à savoir l'algorithme de Sterboul et Wertheimer. Ensuite nous présenterons un nouvel algorithme permettant de construire, à partir d'un graphe des potentiels donné, le graphe PERT minimal tout en réduisant au maximum le nombre de tâches fictives. La dernière section du chapitre est réservée à une étude comparative entre le nouvel algorithme et celui de Sterboul qui sera suivie par un conclusion focalisée sur les avantages qu'apporte notre nouvelle technique de construction du graphe PERT minimal.

On terminera cette thèse par une conclusion générale de notre travail suivie des perspectives qui tracent le chemin des travaux futurs.

Chapitre II

Eléments fondamentaux

Sommaire

I Introduction	17
II Problèmes d'optimisation classiques	18
III Complexité	18
III.1 Complexité d'un algorithme	19
III.2 Algorithme efficace	20
III.3 Problème de reconnaissance	20
III.4 Problèmes faciles et problèmes difficiles	21
III.4.1 La classe NP	21
III.4.1.1 La classe P	21
III.4.1.2 La classe NP-Complet	22
III.4.1.5 La classe NP-Difficile	22
IV La fonction ordonnancement	22
IV.1 Définition	23
IV.2 Les tâches	23
IV.3 Les ressources	24
IV.4 Les contraintes	26
IV.4.1 Les contraintes potentielles	26
IV.4.2 Les contraintes disjonctives	26
IV.4.3 Les contraintes cumulatives	26
IV.5 La fonction économique	27
IV.6 Caractéristiques générales des ordonnancements	28
IV.6.1 Ordonnements statique et dynamique	28
IV.6.3 Ordonnements actifs et semi-actifs	28
IV.6.4 Ordonnements sans retard	29
IV.6.5 Ordonnement préemptif et non préemptif	29
V Classification des problèmes d'ordonnement	29
VI Notations	30
VII Méthodes de résolution des problèmes d'ordonnement	32
IX Conclusion	33

I Introduction

Les problèmes d'optimisation combinatoire abordés dans la littérature sont la plupart du temps issus de situations concrètes de l'industrie ou de service. Ils représentent une vue simplifiée d'une situation, et leur résolution aide à la prise de décision. [FIN 02]

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

Le plus souvent, un problème d'optimisation se présente comme un objectif à poursuivre (maximisation d'un profit, minimisation d'une distance, ou d'un coût, etc.). Cet objectif peut être monocritère ou multicritère. Dans ce qui suit on ne s'intéresse qu'aux problèmes monocritères, des éléments sur l'optimisation multicritères pouvant être consultés dans [ROY 93].

Les solutions possibles à ces problèmes doivent respecter un ensemble de contraintes, liées à la situation étudiée. Parmi les techniques de formulation de ces problèmes on cite la programmation linéaire, dynamique,... ainsi que les problèmes d'optimisation combinatoire en théorie des graphes. Cette thèse se penche sur l'étude d'un des problèmes combinatoires, à savoir les problèmes d'ordonnement.

II Problèmes d'optimisation classiques

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. [DIE 00]

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'ordonnement etc. Ce dernier regroupe l'ordonnement dans la gestion de projet et l'ordonnement dans la gestion de la production. Nous donnerons plus tard les principales caractéristiques ainsi que leur classification avec plus de détails.

III Complexité

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

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

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

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

III.1 Complexité d'un algorithme

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

Dans un modèle de calcul, si un programme destiné à une résolution d'un problème s'arrête pour toutes les données d'entrée de ce problème, alors on dit qu'il est un algorithme [XUO 91].

La complexité d'un algorithme A est une fonction $C(A_N)$, donnant le nombre d'instructions caractéristiques exécutées par A dans le pire des cas, pour une donnée de taille N [LCM 03].

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

Prenons l'exemple d'un algorithme de tri d'un tableau à une dimension. Il est bien clair que le temps nécessaire pour exécuter cet algorithme avec 5 éléments n'est pas le même avec 500 éléments.

[BEN 07] présente un autre facteur déterminant dans le temps d'exécution d'un algorithme est la taille des données en entrée. Un tableau à deux dimensions prend plus de temps qu'un tableau à une seule dimension.

Enfin, un dernier facteur considéré dans la complexité est celui de la situation de la structure de données. Par exemple, l'exécution d'un tableau trié au départ n'est pas la même que celui du même tableau désordonné. Donc, la meilleure solution pour ce cas est de prendre le nombre d'opérations dans le pire des cas.

Le critère de rapidité n'est pas toujours celui qui nous intéresse. On peut aussi vouloir estimer la place utilisée par un algorithme dans la mémoire de l'ordinateur. Dans ce cas, on parle de complexité spatiale alors que jusqu'à présent on considérait la complexité temporelle. La complexité spatiale peut se définir d'une manière semblable à la complexité temporelle [BAC 01].

III.2 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.

Considérons les algorithmes A, B et C. Leurs complexités (C) sont les suivantes :

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

Tableau II.1 Exemple de 3 algorithmes avec leurs complexités.

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

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

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

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

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

III.3 Problème de reconnaissance

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

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

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

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

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

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

III.4 Problèmes faciles et problèmes difficiles

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

A l'opposé, les problèmes décidables sont ceux pour lesquels il existe au moins un algorithme pour les résoudre [SAK 84].

III.4.1 La classe NP

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

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

III.4.1.1 La classe P

Un problème est dit polynomial s'il existe un algorithme de complexité polynomiale permettant de répondre à la question posée dans ce problème, quelle que soit la donnée de celui-ci. La classe P est l'ensemble de tous les problèmes de reconnaissance polynomiaux [CHA 1996].

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

III.4.1.2 La classe NP-Complet

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

III.4.1.3 La classe NP-Difficile

La classe NP-Difficile regroupe les problèmes (pas forcément dans la classe NP) tels que n'importe quel problème de la classe NP leur est polynomialement réductible.

On peut résumer ce que nous avons vu dans le tableau récapitulatif suivant :

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

Tableau II.2. Tableau récapitulatif de classification des problèmes combinatoires tiré de [BAC 01].

IV La fonction ordonnancement

Lors de tout projet, un problème crucial se pose. C'est celui du calendrier d'exécution des tâches. Comment déterminer l'ordre dans lequel doivent s'enchaîner les diverses opérations de façon à optimiser un certain objectif.

La conception et la gestion d'un projet qui est composé de multiples travaux élémentaires pose également divers problèmes:

- . de planification (prévision de déroulement de tâches; établissement d'un calendrier)
- . de contrôle d'exécution de projet (coordination, réajustement des prévisions initiales, etc.).

IV.1 Définition

Selon Carlier et al. [CC 88], « *ordonnancer c'est programmer l'exécution d'une réalisation en attribuant des ressources aux tâches et en fixant leurs dates d'exécution.* »

Présentons une autre définition qui est plus explicite : « *Un ordonnancement 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. Plus précisément, on parle de problème d'ordonnancement lorsqu'on doit déterminer les dates de début et les dates de fin des tâches, alors qu'on réserve le terme de problème de séquençement au cas où l'on cherche seulement à fixer un ordre relatif entre les tâches qui peuvent être en conflit pour l'utilisation des ressources. Un ordonnancement induit nécessairement un ensemble unique de relations de séquençement.* » [LOP 03]

Les techniques d'ordonnancement ont pour objet d'aider à la résolution de ce type de problèmes.

Un ordonnancement donc, selon [KSX 08] 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 née en mathématiques où on traite des modèles théoriques, mais elle traite également des modèles du monde réel qui sont fort complexes. 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.

Pour une description plus détaillée de la problématique de l'ordonnancement, le lecteur est prié de se référer aux ouvrages [CFF 76], [BEPSW 96] ou [PIN 95].

IV.2 Les tâches

La définition d'une tâche n'est ni immédiate ni triviale [CC 88]. Prenons l'exemple de réalisation d'un immeuble où les tâches dépendent de la finesse de découpage: on pourra considérer la boiserie comme une seule tâche, ou bien comme plusieurs tâches (pose d'un placard, pose des portes, etc.).

Une tâche i est donc, un travail (opération ou ensemble d'opérations) concrètement identifiable, concerné par la réalisation, et dont l'exécution se trouve complètement décrite par les trois caractéristiques suivantes:

- caractéristiques d'époques: une tâche doit avoir des limites chronologiques bien définies.

d_i époque de début de la tâche

f_i époque de fin de la tâche

- caractéristique de durée:

$t_i = f_i - d_i$ durée du travail

- caractéristique de moyens: Il s'agit de divers moyens (matériels, personnel, fournitures, monnaie, etc.) qui sont nécessaires à la réalisation. [LL 08] On détaillera plus tard cette caractéristique.

Lorsqu'on connaît à l'avance l'ensemble des tâches à exécuter et à partir de quelle date, on dit qu'on est devant un *problème statique*, et lorsque l'ensemble des tâches évolue avec le temps de façon indéterministe on dit qu'on a un *problème dynamique*. Dans ce qui suit, les tâches ont une durée fixe et connue.

Selon les problèmes, les tâches peuvent être exécutées par morceaux, ou doivent être exécutées sans interruption ; on parle alors respectivement de *problèmes préemptifs* et *non préemptifs* [EL 99].

Quand les tâches ne sont pas liées entre elles par des contraintes qu'on détaillera plus tard, on dit qu'elles sont indépendantes.

En gestion de projet (voir chapitre III), nous conserverons le terme de *tâches* pour désigner les activités qui constituent un projet et nous donnerons plus de détails sur ce point. Pour la gestion de la production (paragraphe IV.3), nous préférons parler d'opérations.

IV.3 Les ressources

Pour l'exécution des tâches, ces dernières requièrent certaines ressources telles que des machines, la main d'œuvre, les moyens financiers, etc.

Une ressource k est donc, un moyen humain ou technique qui est utilisé dans la réalisation d'une tâche. Elle est disponible en quantité limitée.

La disponibilité est généralement exprimée par une capacité propre à chaque ressource k notée Q_k ($Q_k \geq 1$).

On distingue deux types de ressources: les *ressources renouvelables* et les *ressources consommables*.

Une ressource est consommable si, après avoir été allouée à une tâche, elle n'est plus disponible pour les tâches suivantes. Le cas pour l'argent, la matière première, etc.

Une ressource est renouvelable si, après avoir été allouée à une tâche, elle redevient disponible après la fin de cette tâche pour les tâches suivantes. C'est le cas pour les machines, les processeurs, les fichiers, le personnel, etc.

On distingue par ailleurs, principalement dans le cas de ressources renouvelables, les ressources disjonctives (ou non partageables) qui ne peuvent exécuter qu'une tâche à la fois (machine, robot, etc.) et les ressources cumulatives (ou partageables) qui peuvent être utilisées par plusieurs tâches en même temps (équipe d'ouvriers, poste de travail, etc.).

Les problèmes d'ordonnancement à ressources disjonctives couvrent une classe importante d'applications qu'on appelle les *problèmes d'atelier* ou *de machines*.

Les problèmes d'ordonnancement d'atelier et sous les contraintes cumulatives feront l'objet d'une étude à vol d'oiseau ultérieurement dans ce chapitre, alors que le problème central fera l'objet d'un développement détaillé à partir du chapitre III.

La nature des ressources prises en considération permet de dresser une typologie des problèmes d'ordonnancement (*Figure II.1*).

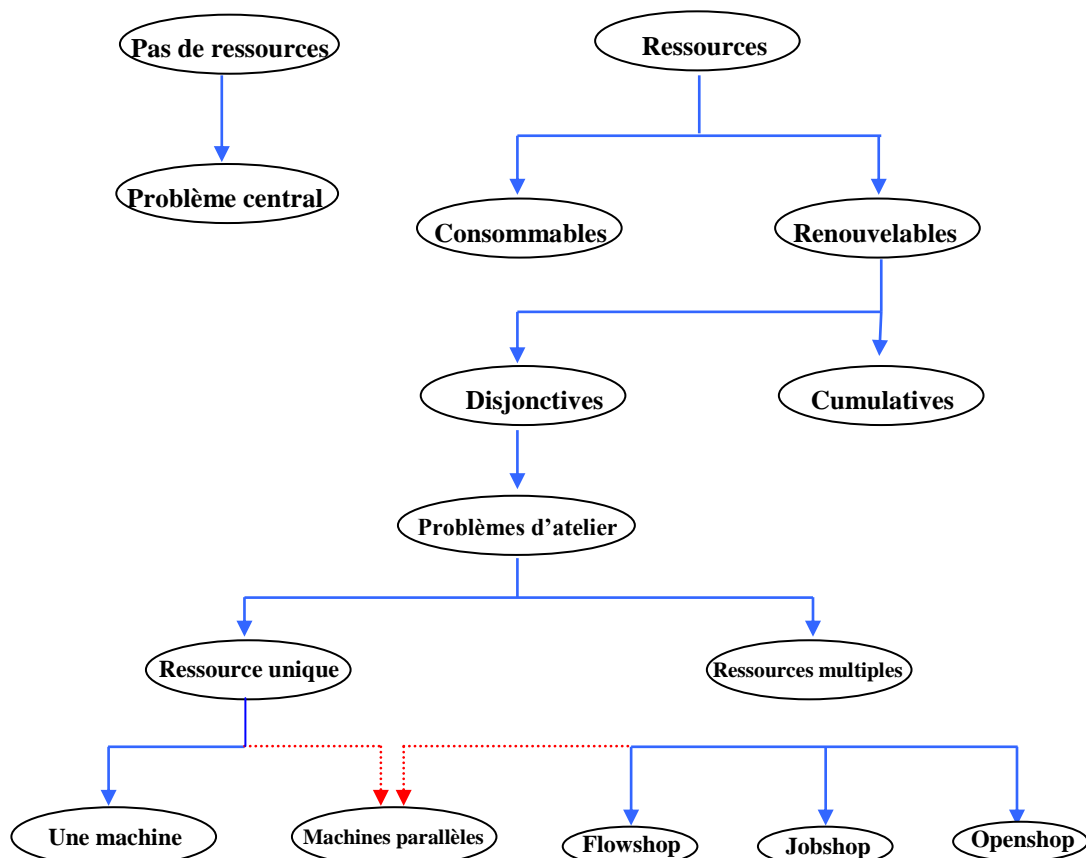


Figure II.1. Typologie des problèmes d'ordonnancement par les ressources

IV.4 Les contraintes

Une contrainte est une restriction sur les valeurs que peuvent prendre une ou plusieurs variables de décision sur le temps (variable d'ordonnancement) ou bien sur les ressources (variables d'affectation) [SDM 08].

Selon [CDB 10], les contraintes auxquelles sont soumises les diverses tâches qui concourent à la réalisation du projet sont de divers types. On distingue :

IV.4.1 Les contraintes potentielles

Elles peuvent être de deux sortes:

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

Nous reviendrons à ce sujet, avec plus de détails, dans le chapitre consacré à l'ordonnancement de projet.

IV.4.2 Les contraintes disjonctives

Une contrainte disjonctive impose la non-réalisation simultanée de deux tâches A et B. On trouve de telles contraintes dans le cas d'utilisation d'une ressource présente en un seul exemplaire (une grue, une équipe, etc.) ou pour formuler des interdictions de réalisation simultanée pour des raisons de sécurité ou des problèmes de place. Arbitrer une contrainte disjonctive consiste à décider si A sera fait avant B ou l'inverse. [FIN 02]

IV.4.3 Les contraintes cumulatives

Selon [HABT 07], on parle de contraintes cumulatives lorsque les tâches demandent une partie d'une ou plusieurs ressources présentes en quantité limitée. Le problème est beaucoup plus combinatoire que pour les contraintes disjonctives. Considérons l'exemple où nous avons cinq intervenants et cinq tâches à effectuer. Chaque tâche demande la présence d'un certain nombre de ces intervenants (voir tableau)

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

Tableau II.3. Exemple de contraintes cumulatives

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

Lorsqu'une configuration minimale porte sur deux tâches (ici A//B ou A//C), on peut la remplacer par une contrainte disjonctive entre ces deux tâches. [FIN 02]

IV.5 La fonction économique

Lorsqu'on cherche la résolution d'un problème d'ordonnancement, on est amené à décider entre deux familles de principes. Le premier vise à chercher l'optimalité de la solution par rapport à un critère ou même à plusieurs critères, et le second cherche l'admissibilité de cette solution par rapport aux contraintes.

L'approche d'optimisation suppose que les solutions proposées sont classées dans un certain ordre par rapport à un ou plusieurs critères d'évaluation. Ce classement permet de mesurer la qualité des solutions.

Donc, selon le domaine d'application, la fonction ordonnancement peut avoir d'autres objectifs que celui de veiller au simple respect des contraintes de temps et de ressource. Il peut s'agir de satisfaire des objectifs économiques, de respecter une législation du travail en vigueur dans une entreprise ou de gérer au mieux le risque en présence d'incertitudes.

Lorsque ces objectifs sont quantifiables et exprimables en fonction des variables d'ordonnancement, ils sont injectés dans le problème d'ordonnancement, soit par ajout de nouvelles contraintes, soit par ajout d'un ou plusieurs critères d'optimisation.[TRU 05]

La fonction coût peut tenir compte de différents critères, parmi lesquels :

- flexibilité, tolérance aux pannes, vitesse de réordonnancement [FRC 93] [SF 96] [AS 94] ;
- délai de fabrication, avances, retard [BRU 93] [CT 94] [CS 89] [HH 97] [IHI 96] [GOT 93] [ST 94] [CHU 95c] [BDF 97] [WSL 95] [CHU 95b] ;
- durée globale du projet [YN 96] [IHI 96] [GOT 93] [ST 94] [GS 78] [GJ 79] [CHU 95c] ;

- charge des machines, équilibrage de charge [BUM 91] [SF 96] [NA 94] [HD 90] [IHI 96] [AS 94] [WSL 95] ;
- nombre de machines actives [CT 94] [ÖZD 96] ;
- temps global de stockage (attente devant les machines) [WSL 95] ;
- taille des stocks, dimensionnement des lots [LSS 93] [ÖZD 96] [GOT 93] ;
- nombre de changement d'outils [HW 96] [ÖZD 96] ;

Il existe bien d'autres critères : par exemple, l'objectif peut consister à minimiser les transports (de matières premières ou de produits intermédiaires...). Deux familles de critères peuvent être distinguées : les critères réguliers et les critères non réguliers.

Un critère est dit régulier s'il est une fonction décroissante des dates de fin d'exécution des opérations. Par exemple, la durée globale de fabrication d'un produit est un critère régulier [DUV 00].

IV.6 Caractéristiques générales des ordonnancements

IV.6.1 Ordonnements statique et dynamique

Si l'ensemble des informations nécessaires à la résolution d'un ordonnancement est fixé au départ (ensembles des tâches, des contraintes, des ressources, etc.), on est alors devant un problème d'ordonnement *statique*.

Il y a une nuance entre la solution proposée qui est, généralement accompagnée d'un plan prévisionnel d'exécution des tâches, et l'exécution réelle de ces tâches. Si le plan n'est pas respecté et les objectifs sont modifiés, on est devant un problème d'ordonnement *dynamique* qui nécessite une résolution d'une série de problèmes statiques et chaque étape doit débiter par une prise d'informations permettant d'actualiser le modèle à résoudre.

IV.6.2 Ordonnements admissibles

Un ordonnancement est dit admissible s'il respecte toutes les contraintes du problème (dates de début, dates de fin, précédence, contraintes de ressources, etc.).

IV.6.3 Ordonnements actifs et semi-actifs

Dans un ordonnancement actif, aucune tâche ne peut commencer au plus tôt et qui entraîne l'ordre relatif entre au moins deux tâches.

Dans l'ordonnancement semi actif, on ne peut pas avancer une tâche sans modifier la séquence sur la ressource.

IV.6.4 Ordonnements sans retard

Dans un ordonnancement sans retard, on doit exécuter la tâche qui est en attente à condition que la ressource soit disponible.

IV.6.5 Ordonnement préemptif et non préemptif

Selon les problèmes, les tâches peuvent être exécutées sans interruption, c'est-à-dire si on commence l'exécution d'une tâche elle n'est pas interrompu jusqu'à son achèvement. On parle alors d'un ordonnancement préemptif. Par contre, si les tâches sont exécutées par morceaux, l'ordonnancement est appelé non préemptif.

V Classification des problèmes d'ordonnement

Selon la nature des variables mises en jeu, la nature des contraintes, ou encore la structure particulière du graphe non-conjonctif associé à un problème, plusieurs classifications des problèmes d'ordonnement sont proposées dans la littérature.

Dans le domaine de l'ordonnement d'atelier, la notion structurante de gamme opératoire est souvent utilisée comme critère de classification. En effet, une gamme opératoire impose un ordre de passage des produits sur les machines, et donc un ordre des opérations associées à chaque travail. De ce fait, le graphe potentiels-tâches non conjonctif associé au problème possède parfois une structure particulière, qu'il est possible d'exploiter pour élaborer des méthodes d'ordonnement efficaces. Dans [ML 93], sept types de problèmes d'ordonnement d'atelier sont ainsi distingués :

- le problème jobshop où chaque travail a sa gamme opératoire propre.
- le problème flowshop où chaque travail a une gamme identique ; le problème open shop où l'ordre de passage sur les machines est libre pour chaque travail ;
- le problème openshop où l'ordre de passage sur les machines est libre pour chaque travail ;

- le problème flowshop de permutation où chaque travail à une gamme identique et où l'ordre de passage des travaux est le même pour chaque machine ;
- le problème à une machine où chaque travail est assimilé à une opération unique, exécutée par une seule et même machine m de capacité $Q_m = 1$;
- le problème à machines parallèles où chaque travail est assimilé à une opération unique, exécutée par une machine à sélectionner dans un ensemble ;
- le problème jobshop à machines dupliquées où chaque travail a sa gamme opératoire propre et où chaque opération est réalisée par une machine à sélectionner dans un ensemble.

VI Notations

La notation $\alpha/\beta/\gamma$ proposée dans [Graham et al. 79], puis reprise dans [BEPSW 96], s'est rapidement imposée comme faisant référence. Elle permet en effet de caractériser un problème d'ordonnancement de manière précise. Le champ α décrit la structure du problème et se décompose généralement en deux sous champs α_1 et α_2 , le premier indiquant la nature du problème (job shop, flow shop, etc.), le second précisant le nombre de machines ou de pools. Le champ β décrit les types de contraintes prises en compte. Enfin, le champ γ indique la fonction objectif considérée ou la description des critères.

A titre indicatif, quelques valeurs classiques de α_1 , β et γ sont décrites dans les tableaux II.4, II.5 et II.6.

Valeur	Description
\emptyset	Machine unique
P	Machines parallèles identiques
Q	Machines parallèles proportionnelles
R	Machines parallèles non reliées
F	Flow shop
J	Job shop
O	Open shop
FH	Flow shop hybride
JG	Job shop généralisé
OG	Open shop généralisé
X AG avec $X \in \{P, Q, R, F, J, O\}$	Problème X avec affectation générale

Tableau II.4. Quelques valeurs du champ α_1

Valeur	Description
$prec$	il existe des contraintes de précédence générale entre les opérations
r_i	une date de début au plus tôt r_i est associée à chaque travail i
d_i	une date d'échéance préférentielle d_i est associée à chaque travail i
\tilde{d}_i	une date d'échéance stricte \tilde{d}_i est associée à chaque travail i
$p_i = 1$	les durées opératoires sont unitaires
$pmtn$	la préemption des opérations est autorisée
no_wait	les opérations de chaque travail doivent se succéder sans attente
$Snsd(Rsnd)$	les ressources doivent être préparées avant et/ou après chaque exécution indépendamment de la séquence des travaux

Tableau II.5. Quelques valeurs du champ β

Valeur	Fonction objectif à minimiser
C_{max}	la durée totale (makespan)
L_{max}	le plus grand retard algébrique
T_{max}	le plus grand retard vrai
ΣU_i	le nombre de travaux en retard
$\Sigma [w_i]C_i$	la durée moyenne ou pondérée des travaux
$\Sigma [w_i]U_i$	le nombre moyen ou pondéré de travaux en retard
$\Sigma [w_i]T_i$	le retard moyen ou pondéré

Tableau II.6. Quelques valeurs du champ γ .

En ordonnancement de projet, la notion de gamme opératoire est généralement absente. Toutefois une notation de type $\alpha/\beta/\gamma$ a également été proposée dans [BDMNP 99], qui se veut proche de celle définie en ordonnancement d'atelier. La principale différence concerne les valeurs du champ α qui dépend, non plus de l'existence de gamme opératoire, mais de la nature des variables de décision (ordonnancement et/ou affectation) et de la nature des ressources (renouvelable, non renouvelable). Le tableau II.7 indique quelques valeurs possibles du champ α . [HDD 99].

Valeur	Description
PS	Ordonnancement de projet.
MPS	ordonnancement de projet avec plusieurs modes d'exécution par activités.
PSm, σ, ρ	Problème PS avec m ressources, de capacité σ , et chaque activité requiert au plus ρ unités de ressource.
$MPSm, \sigma, \rho, \mu, \tau, \omega$	Problème MPS avec m ressources renouvelables, de capacité σ , consommables en au plus ρ unités et avec μ ressources non renouvelables, de capacité τ , consommables en au plus ω unités.

Tableau II.7. Valeurs du champ α en gestion de projet.

Remarquons enfin que plusieurs extensions ont été proposées pour les champs β et γ (voir [BEPSW 96], [TB 02]) afin de prendre en compte des catégories de problème particulières (stochastiques, répétitifs, cycliques, dynamiques, multicritères, etc.) et qu'il est probable que dans l'avenir, d'autres extensions soient encore créées [HOA 05].

VII Méthodes de résolution des problèmes d'ordonnement

Comme nous l'avons signalé précédemment, les problèmes appartenant à la classe P ont des algorithmes efficaces et sont de complexité polynomiale tel que la méthode de chemin critique, la méthode PERT dans l'ordonnement de projet... Pour les autres problèmes, il est peu réaliste de trouver des algorithmes de ce type. Alors, on fait appel à une famille de méthodes qui sont *exactes* ou *approchées*.

Plusieurs auteurs [ML 93], [ESS 03] partitionnent les méthodes de résolution en trois classes : les méthodes optimales efficaces, les méthodes énumératives et les méthodes heuristiques.

Selon [WOE 08], les méthodes optimales efficaces garantissent, pour un problème et un critère donnés, la détermination d'une solution optimale en un temps de calcul polynomial, i.e. le temps maximal consommé pour trouver une solution optimale est une fonction polynomiale des données du problème (nombre de ressources, de tâches). De telles méthodes ne sont évidemment disponibles que pour des classes réduites de problèmes d'ordonnement. Parmi les plus connues, citons :

- la règle SPT (Shortest Processing Time) pour le problème $1/\Sigma C_j$;
- la règle WSPT (Weighted Shortest Processing Time) pour le problème $1/\Sigma w_j C_j$;
- la règle EDD (Earliest Due Date) pour le problème $1/d_j / L_{\max}$;
- l'algorithme de Moore et Hodgson pour le problème $1/d_j / \Sigma U_j$;
- l'algorithme de Johnson pour le problème $F2/prmu/C_{\max}$.

Selon [KUM 07], les méthodes optimales énumératives procèdent quant à elles à une énumération partielle de l'espace de recherche. Leur complexité temporelle théorique est généralement exponentielle, mais elles fournissent en pratique, sur des problèmes de taille moyenne, des solutions optimales en un temps raisonnable. Dans cette classe, on peut distinguer :

- les Procédures par Séparation et Évaluation (PSE) qui énumèrent par une recherche arborescente un ensemble de solutions, en éliminant les branches

de l'arbre de recherche montrées non-optimales (utilisation de bornes inférieure et supérieure du critère) ;

- les méthodes de Programmation Linéaire (PL), modélisant les critères et les contraintes comme des fonctions linéaires de variables mixtes (réelles, entières) ;
- les méthodes basées sur la Programmation Dynamique (PD) qui procèdent à une décomposition en sous-problèmes, que l'on résout optimalement à rebours, en tenant compte à chaque étape des informations issues de la résolution du sous-problème précédent.

Les méthodes heuristiques sont souvent utilisées pour traiter des problèmes que les méthodes optimales sont incapables de résoudre en un temps acceptable. Elles produisent généralement une solution faisable de bonne qualité en un temps relativement court. La qualité d'une heuristique doit être évaluée sur plusieurs jeux d'instance de taille variable afin de mesurer d'une part, la déviation moyenne du critère par rapport à sa valeur optimale, et d'autre part, l'évolution du temps de calcul en fonction de la taille ou de la structure du problème. Ces heuristiques sont classiquement classifiées en trois grands types :

- les algorithmes gloutons dans lesquels les décisions d'ordonnement sont prises progressivement, à temps croissant, au fur et à mesure que les ressources se libèrent, grâce à des règles de priorité simples de type SPT, EDD, etc. ; et pour lesquels on ne remet jamais en cause une décision qui a été prise.
- les méthodes de recherche locale (tabou, recuit simulé, algorithmes génétiques, etc.) qui, partant d'une solution initiale, définissent un voisinage, qui est ensuite exploré pour trouver des solutions meilleures ; (en s'autorisant parfois à dégrader la solution courante pour augmenter les chances d'obtenir une solution meilleure).
- les méthodes de recherche arborescente tronquée, proches des PSE, excepté que l'arbre de recherche est volontairement restreint, quitte à perdre des solutions optimales, afin de gagner en temps de calcul [TRU 05].

IX Conclusion

L'ordonnement est généralement décrit comme une fonction particulière de décision au sein d'un système de gestion du travail concernant la production de bien, d'ouvrages ou de services.

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 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.

Il faut encore souligner que les problèmes d'ordonnement sont extrêmement divers, et nous n'avons vu qu'une partie. De plus, il n'existe pas de théorie générale dans ce domaine.

Chapitre III

L'ordonnancement de projet

Sommaire

I Introduction	35
II La gestion de projet	36
II.1 Définition	36
III Le Projet	37
III.1 Caractéristiques d'un projet	37
III.2 Acteurs du projet	38
III.3 Cycle de vie d'un projet	38
III.3.1 La planification du projet	39
III.3.2 Les éléments du plan	40
III.3.3 Le calendrier d'activités	42
III.3.3 Création d'un calendrier d'activités	43
III.4 L'ordonnancement dans la gestion de projet	44
IV Le problème central de l'ordonnancement	44
IV.1 Modélisation du problème central	44
IV.1.1 Le diagramme de Gantt	45
IV.1.1.1 Avantages du diagramme de Gantt	46
IV.1.1.2 Inconvénients du diagramme de Gantt	46
IV.1.2 Le graphe des potentiels	47
IV.1.3 Le graphe PERT	49
IV.1.3.1 Dessin du graphe PERT	51
IV.1.3.2 Notion de tâche fictive	53
IV.1.4 Modularité	55
IV.1.5 Comparaison	56
V Résolution du problème central	56
V.1 Dates et marges associées à une tâche	58
V.2 Calcul des dates et marges	59
V.3 Le chemin critique	62
VI Conclusion	65

I Introduction

Dans ce chapitre, on s'intéresse à l'ordonnancement de projet qui est une partie prenante dans la gestion de projet.

La modélisation des problèmes d'ordonnancement de projet nécessite la connaissance des concepts de base de la théorie des graphes.

Comme nous l'avons fait dans le chapitre précédent et pour bien nous situer, nous allons faire un tour d'horizon sur la gestion de projet, ensuite nous allons aborder, en détail, l'ordonnancement de projet.

Pour présenter les concepts et méthodes de l'ordonnancement de projet, il est bien utile de définir *le problème central*, où on ne tient pas compte des contraintes de ressources, c'est-à-dire on considère que les ressources sont à quantité illimitée. L'étude de ce cas permet d'analyser l'enchaînement des tâches au cours du temps et introduire les méthodes de résolution qui seront utilisées dans l'étude des problèmes d'ordonnancement en tenant compte des contraintes de ressources.

II La gestion de projet

On peut presque considérer la gestion de projet comme une discipline universelle puisqu'elle englobe un large éventail d'activités : de l'élaboration de nouveaux produits et services, à l'organisation de conférences et d'ateliers, en passant par la modernisation de bureaux d'une entreprise. Si on dispose d'un budget illimité et de tout temps nécessaire, la gestion de projet serait plutôt facile. En réalité, le temps et l'argent sont des facteurs déterminants et, par conséquent, la gestion de projet devient une compétence importante à maîtriser.

Selon [CGR 08], dans la vie d'aujourd'hui où on compte une infinité de projets qui se réalisent, on remarque que ces projets n'atteignent pas souvent leurs objectifs à cause du dépassement de délais (surcoûts importants, ou bien la qualité technique du produit est insuffisante). Notons également que les projets se déroulent dans un milieu complexe (acteurs divers dans une entreprise : étude, production, marketing, environnement extérieur non maîtrisable : marché, social, politique, concurrence, etc.). C'est pour ces raisons qu'il faut avoir une démarche claire et rigoureuse pour la gestion d'un projet.

II.1 Définition

La gestion de projet 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 d'un projet [DP 01].

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 priorités différentes des parties prenantes;
- les besoins définis et à la portée du projet;
- la qualité et la quantité.

III Le Projet

Un projet est défini comme une démarche spécifique permet de structurer méthodiquement et progressivement une réalité à venir [GIA 91].

Selon [LIZ 2005], un projet est un effort complexe pour atteindre un objectif spécifique de qualité, devant respecter un échéancier et un budget. Un projet est défini et mis en œuvre pour élaborer une réponse au besoin d'un utilisateur, d'un client ou d'une clientèle et il implique un objectif et des actions à entreprendre avec des ressources données. Il est réalisé une seule fois, doté d'un début et d'une fin déterminée et qui vise à créer un produit ou un savoir unique. Il peut nécessiter la participation d'une seule ou de milliers de personnes. Sa durée peut être de quelques jours ou de plusieurs années. Il peut être entrepris par une seule organisation ou par un groupe d'organismes intéressés. Il peut s'agir de quelque chose d'aussi simple que l'organisation d'un événement d'une journée ou d'aussi complexe que la construction d'un barrage sur une rivière.

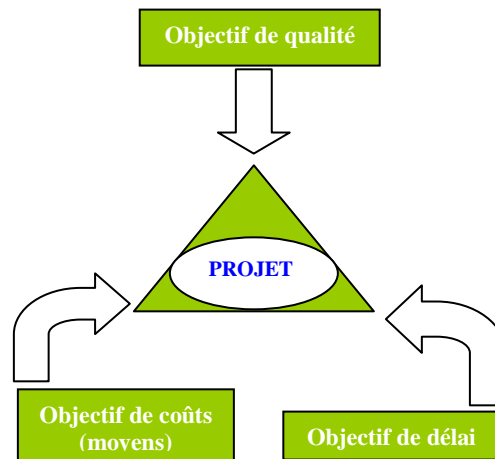


Figure III.1. Les objectifs du projet.

III.1 Caractéristiques d'un projet

Un projet réussi doit contenir les particularités suivantes :

- **Des objectifs précis** Les projets les plus réussis ont des objectifs définis clairement dès le départ.

- **Un plan de projet bien établi** Un plan conçu avec minutie est utile pour deux raisons. D'abord, il 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 entre toutes les personnes qui y prennent part. Une communication réelle et continue doit s'établir entre les parties, si elles veulent oeuvrer ensemble à la réussite du projet [LIZ 2005].

- **Une envergure maîtrisée** Tout au long du projet, le chef fait face à plusieurs situations qui ne contribuent pas toutes objectifs tracés. Il importe que le chef de projet porte attention à ses priorités, avec une perte minimale de temps et de concentration.

- **Le soutien des intervenants** D'ordinaire, les projets sont le fait de plusieurs parties prenantes. Il importe que celles-ci accordent leur soutien pour toute la durée du projet de façon à ce que l'équipe atteigne ses objectifs.

III.2 Acteurs du projet

- **Maître d'ouvrage (MOA)** personne physique ou morale propriétaire de l'ouvrage. Il détermine les objectifs, le budget et les délais de réalisation.
- **Maître d'oeuvre (MOE)** personne physique ou morale qui reçoit mission du maître d'ouvrage pour assurer la conception et la réalisation de l'ouvrage.

III.3 Cycle de vie d'un projet

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

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

La figure suivante (Figure III.2) illustre le déroulement du processus de gestion d'un projet.

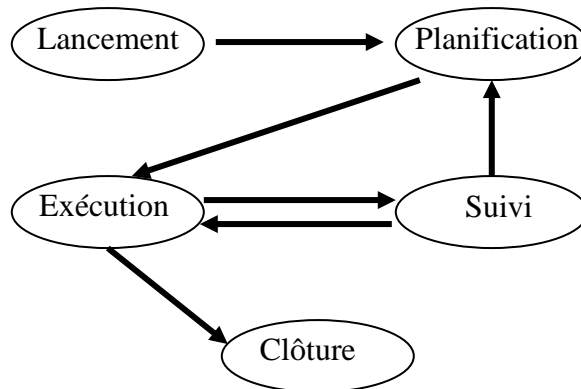


Figure III.2. Cycle de vie d'un projet.

III.3.1 La planification du projet

Plusieurs problèmes rencontrés lors de certains projets auraient pu être évités ou minimisés si, dès le départ, un plan de projet détaillé avait été mis au point.

Le plan de projet établit les règles de base et les énonce de façon claire. C'est une étape particulièrement importante puisque les projets mettent habituellement en jeu plusieurs partenaires et intervenants ayant des intérêts et des points de vue différents [CMGL 08].

La question pertinente qui se pose dans la gestion de projet est la suivante : pourquoi la planification? Quatre éléments de réponses confirment l'utilité et la nécessité d'une planification dans la réalisation d'un projet.

En premier lieu, selon [GM 96], il est essentiel que tous les partenaires comprennent et acceptent les règles fondamentales qui régiront le projet. Il faut veiller à ce que les objectifs soient formulés clairement de façon à éviter tout désaccord ultérieur.

En second lieu, le plan de mise en oeuvre d'un projet contribue à la maîtrise et à la mesure de son progrès. Une fois la constitution de l'équipe est faite et les possibilités de financement sont déterminées, le chef de projet devra passer en revue le plan et y ajouter les détails particuliers qui faciliteront une gestion réussie.

Troisièmement, le plan de projet aidera à faire face aux changements susceptibles de survenir ou, plutôt, qui surviendront inévitablement. Que se passera-t-il, par exemple, si l'un des intervenants ajoute un nouvel objectif au projet? Un plan de projet détaillé aide donc, à affronter ce type de situation et à tenir aux objectifs généraux du projet [LIZ 05].

Enfin, le plan d'exécution d'un projet contribue à consolider le soutien des parties prenantes au cours des mois ou des années que durera le projet. Encore une fois, ce fait est

particulièrement important parce que le chef de projet aura besoin du soutien de personnes appartenant à divers organismes et sur lesquelles il ne pourra exercer aucune autorité directe.

III.3.2 Les éléments du plan

Puisqu'il n'y a pas deux projets pareils, il n'existe pas non plus deux plans de projet qui soient identiques. Pour qu'il soit d'une utilité maximum, le plan de projet doit être pertinent, compréhensible et doit tenir compte de l'importance et de la sa complexité. Il devrait inclure les éléments suivants [POR 91] :

1. le mandat de projet;
2. le calendrier d'activités;
3. l'horaire de travail;
4. la matrice de responsabilités;
5. le budget de plan de projet;
6. les étapes importantes, avec les dates cibles;
7. la stratégie de gestion du risque.

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 mandat de projet est un document qui témoigne de l'appui des cadres supérieurs, qui autorise le gestionnaire du projet à mener le projet et à attribuer les ressources nécessaires. La création d'un mandat de projet est chose facile. Il suffit d'indiquer le nom et le but du projet ainsi que le nom du gestionnaire de projet et une déclaration de soutien de la part des cadres supérieurs. Elle est signée par la haute direction de l'organisme en charge et les organismes participants.

Le calendrier d'activités est l'un des éléments les plus importants du coffre à outils du gestionnaire de projet. Le paragraphe suivant sera consacré au calendrier puisque c'est la où se situe notre recherche.

L'horaire de travail précise le lien logique entre les activités du projet, assure que le personnel est disponible au moment approprié et l'aide à gérer le temps de façon efficace afin de terminer le projet au moment prévu. Au moment d'établir l'horaire de travail, le chef de projet examine toutes les tâches et l'ordre dans lequel elles doivent être réalisées.

Le projet nécessitera la collaboration d'un grand nombre de personnes et d'organismes qui travaillent à un but commun. La gestion d'une équipe variée, souvent dispersée à plusieurs endroits, peut présenter des défis particuliers.

La matrice de responsabilités est un outil précieux de gestion de projet destiné à aider à relever ces défis. Une matrice de responsabilités attribue à quelqu'un la responsabilité de chaque activité principale du projet, évitant ainsi que certains éléments « échappent à la surveillance ». Il n'est pas nécessaire qu'elle soit complexe et peut être facilement réalisée en se rapportant à l'horaire de travail du projet [CFT 09].

Il est important d'avoir l'estimation la plus détaillée et la plus précise possible des principaux coûts du projet (habituellement les salaires, les matériaux et les fournitures ainsi que les frais généraux), dès le début du projet. Avec ces renseignements, la production du budget du plan de projet est simple.

L'art de l'estimation est très difficile à maîtriser. Celle-ci est basée principalement sur des données historiques, l'expérience et le jugement. Hormis le secteur de la construction, où il existe des estimateurs professionnels, il est très compliqué d'estimer de façon précise la plupart des projets, par exemple, le développement d'un logiciel informatique.

Le chef de projet doit estimer :

- Les immobilisations corporelles (terrains, bâtiment, équipement, ...)
- Les immobilisations incorporelles (frais d'études, frais de démarrage, frais financiers,...)
- Les dépenses d'exploitation (frais liés à l'activité, frais financiers,...)

Il existe actuellement plusieurs techniques et méthodes d'estimation. On peut citer entre autres :

- globale (expert/oracle)
- par analogie
- analytique (comme la méthode PERT qu'on verra plus tard)
- Paramétrique.

Les étapes importantes marquent un événement significatif dans le cours du projet, habituellement l'achèvement d'un produit livrable important. Dans le cadre du plan d'action du projet, le chef de projet a défini les étapes importantes et a fixé des dates cibles.

La réalisation des étapes dans les délais prévus évite le gaspillage des ressources, maintient au projet son élan et instaure une crédibilité par rapport aux partenaires potentiels.

La gestion d'un projet nécessite la conciliation constante des ressources et des priorités. Ces contraintes, auxquelles s'ajoutent les effets inattendus de circonstances qui échappent au contrôle peuvent compromettre la réalisation des étapes et le respect des dates d'échéance. Le

calendrier d'activités, l'horaire de travail et la matrice de responsabilités sont des outils privilégiés qui aident à maintenir le cap [LIZ 05].

La première étape dans la création d'un plan de réponse consiste à définir les risques qui pourraient nuire au projet. Le gestionnaire de projet, les principaux membres du personnel et les partenaires du projet devraient lancer des idées en s'inspirant de l'arrêté de projet, du calendrier d'activités, de l'échéancier et du budget afin de mettre en relief les risques possibles. Les intervenants peuvent souvent définir les risques en se fiant à leur expérience. En outre, il existe des ressources documentaires qui précisent les risques dans plusieurs domaines d'application.

Les initiatives d'apprentissage communautaires comportent souvent les sources de risques suivantes :

- les risques techniques tels que la technologie non éprouvée;
- les risques de gestion de projet tels que les lacunes relatives à la répartition du temps et des ressources;
- les risques organisationnels tels que les conflits de ressources par rapport à d'autres activités;
- les risques externes tels que la modification des priorités chez les partenaires.

III.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, le calendrier des activités :

- 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.

La production d'un calendrier d'activités demande un effort considérable, et on pense des fois que notre projet n'a pas l'ampleur voulue pour justifier cette démarche. Toutefois, grâce au plan d'action mis au point au départ, on a déjà en main l'information nécessaire pour commencer le projet.

III.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 confiés à des personnes et doivent être définis 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.

Selon [KLO 09], Les tâches, suivant leur disposition dans le réseau, peuvent être: successives, simultanées ou convergentes. Les tâches sont *successives* lorsqu'elles se déroulent les une 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.

- **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.

- **Codifier les tâches**

Pour passer à l'étape d'examen des contraintes, il est nécessaire de repérer, 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.

- **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) ou 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.

III.4 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 et 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.

IV Le problème central de l'ordonnancement

Pour le problème central de l'ordonnancement, il s'agit d'ordonnancer en une durée minimale un ensemble $I = \{1, 2, \dots, n\}$ de tâches soumises à des contraintes temporelles de type inégalité de potentiel.

Donc, dans le problème central de l'ordonnancement, on ne tient compte que des contraintes de potentiels qui sont d'antériorité et de durée. Les contraintes disjonctives et cumulatives sont ignorées. Les ressources sont supposées à quantité illimitée.

IV.1 Modélisation du problème central

Il existe actuellement trois méthodes de modélisation du problème central : *le diagramme de Gantt*, *la méthode des potentiels* et *la méthode PERT/CPM*. 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).

Pour les besoins algorithmiques, on complète cette table par une tâche α (appelée source ou racine) de durée 0 précédant A et B et une tâche ω (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 α et un puits ω tel qu'il existe un chemin de valeur positive entre α et tous les autres sommets, et un chemin de valeur positive entre tout sommet différent du puits ω et ω .

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.

IV.1.1 Le diagramme de Gantt

Au début du siècle dernier, le gouvernement américain déclara la guerre à l'Allemagne, Gantt développa alors, une représentation graphique de déroulement des projets, qui prit plus tard le nom de diagramme de Gantt du nom de son inventeur.

Donc, pour faire le planning de l'exécution du projet et pour en contrôler le déroulement on utilise le diagramme à barres (Bar Chart) ou diagramme de Gantt.

Chaque tâche est symbolisée par un rectangle dans lequel sont inscrits le code et la durée de la tâche. Commenant par représenter les tâches avec antériorité α , une tâche est portée dans le diagramme lorsque toutes ses antériorités y sont déjà portées. Ceci est toujours possible, car une tâche ne se succède pas à elle même, les rectangles sont ainsi toujours placés à droite (c'est le même principe que celui des niveaux d'un graphe sans circuit). Cette représentation indique bien le déroulement des travaux mais moins bien les antériorités.

On introduit cette technique à l'aide d'un exemple du problème de réalisation d'une construction tiré de [BEN 01] :

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

Tableau III.1. Table initiale d'ordonnancement T.

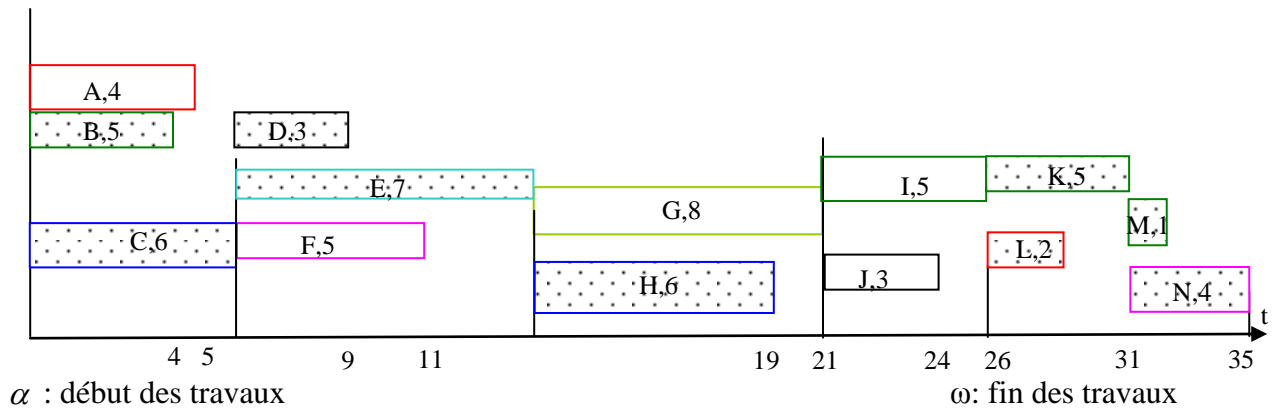


Figure III.3. Le diagramme de Gantt de la table III.1.

Il existe actuellement d'autres formes du diagramme de Gantt utilisant les sous forme de tableau, mais le principe reste toujours celui qu'on a présenté.

IV.1.1.1 Avantages du diagramme de Gantt

Selon [ECO 11], le diagramme de Gantt a plusieurs avantages :

- facilement compréhensible par les exécutants, de par sa clarté et sa simplicité ;
- peut servir de base à des plans d'action intermédiaires plus détaillés ;
- permet de suivre le déroulement des opérations dans le temps ;
- résume assez bien l'analyse du projet établie par les responsables respectifs.

IV.1.1.2 Inconvénients du diagramme de Gantt

- cache les erreurs de forme et de fond commises au niveau de l'analyse du projet ;
- ne met pas en évidence les tâches critiques au niveau desquelles tout retard apporté au niveau de l'exécution, entraîne un retard équivalent quant à la réalisation de l'ensemble du projet ;
- impossibilité de rectifier ponctuellement la durée d'une tâche précise, sans avoir à décaler d'autant les suivantes et à redresser sinon complètement, du moins partiellement l'édifice. Si un des maillons de la chaîne change, c'est donc tout l'édifice qui s'écroule.
- Insuffisance également dans la mise en évidence des liaisons existant entre les différentes tâches.

Jusqu'à l'année 1958 on utilisait le diagramme de Gantt pour traiter les problèmes d'ordonnancement. Cette année là, se sont développées en parallèle deux méthodes fondées sur la théorie des graphes pour ordonnancer les travaux. La méthode des potentiels de Bernard ROY appelée méthode française, et la méthode PERT/CPM appelée méthode américaine.

Algorithmiquement, les deux méthodes sont équivalentes. Elles se distinguent seulement par la manière d'écrire le graphe de réseau. Nous les présenterons dans ce qui suit.

IV.1.2 Le graphe des potentiels

La méthode des potentiels a été développée vers la fin des années 50 parallèlement à la méthode PERT. Elle est appelée également la méthode MPM (méthode des potentiels Metra) ou encore méthode des potentiels –tâches.

Considérons l'exemple de réalisation précédent :

Codes	Durées	Antériorités
α	0	-
A	4	α
B	5	α
C	6	α
D	3	A, B, C
E	7	A, B, C
F	5	A, B, C
G	8	D, E
H	6	D, E
I	5	F
J	3	F
K	5	G, H, I, J
L	2	G, H, I, J
M	1	K, L
N	4	K, L
ω	0	M, N

Tableau III.2. La table initiale d'ordonnancement T pour les traitements algorithmiques, désignations non reprises.

Le tableau des données (Tableau III.2) n'est autre que la matrice des prédécesseurs d'un graphe valué n'admettant qu'une source α et un puits ω : c'est le graphe des potentiels.

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 .

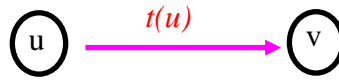


Figure III.4. La tâche u , de durée $t(u)$, précède la tâche v

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. 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 α 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).

La réorganisation des sommets en niveaux montre clairement les antériorités. Elle se fait comme suit :

- **niveau I** : la tâche source α sans antécédent ;
- **niveau II** : tâches n'ayant que le prédécesseur α ;
- **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 ω qui n'a aucun successeur.

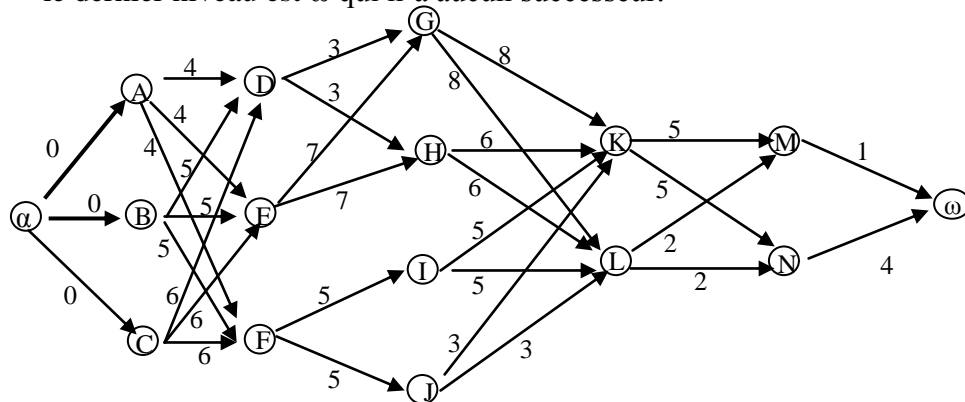


Figure III.5. Le graphe des potentiels avec les sommets réorganisés en niveaux.

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

IV.1.3 Le graphe PERT

PERT et le CPM ont été indépendamment développés vers la fin des années 50 depuis, ils ont été parmi les plus largement appliqués.

Les versions originales de PERT et du CPM ont eu quelques différences importantes. Cependant, elles également ont eu beaucoup de commun, et les deux techniques ont été graduellement fusionnées au cours des années. En conséquence, les praticiens emploient maintenant généralement les deux noms l'un pour l'autre, ou les combinent dans l'acronyme simple PERT/CPM.

La méthode PERT (Program Evaluation and Review Technique – technique d'ordonnancement et de contrôle des programmes) a été mise au point par la marine américaine qui devait conduire à la réalisation des missiles à ogive nucléaire Polaris. Cette technique a permis de coordonner les travaux de près de 6000 constructeurs, 250 fournisseurs et 9000 sous-traitants dans un délai de réalisation de 7 ans.

L'utilisation de la méthode PERT a permis de ramener la durée globale du projet de 7 à 4 ans.

Selon [IKA 04] PERT et CPM se différencient par la nature de l'information requise, par la capacité de prévoir la date de fin du projet et d'estimer les chances de minimiser le risque de retard dans l'exécution du projet.

Dans CPM, l'estimation des durées respectives se fait de façon déterministe - c'est le cas par exemple des projets de construction qui sont en général tributaires de ressources et s'appuient sur des estimations précises - alors que PERT considère que les durées des activités sont des variables aléatoires et qu'en conséquence leur estimation se fait de façon probabiliste [CHA 97], [MP 96], [ROB 97]. CPM se prête plus aisément à l'analyse du compromis coût/durée alors que l'un des avantages majeurs du PERT est de prendre en compte l'incertitude et le risque qui entourent les durées respectives du projet (et donc le temps de réalisation du projet) et de calculer la probabilité de compléter le projet à temps [IZU 90]. Évidemment PERT est employé dans la gestion des délais et des échéanciers des projets de R&D ou de tout autre projet complexe ou de grande taille où les durées des activités sont sujettes à une forte variabilité et donc à un degré élevé d'incertitude sur leur temps de réalisation.

Bien avant la méthode PERT, les problèmes d'ordonnancement ont été abordés par le diagramme de Gantt. Comme on l'a constaté précédemment s'il constituait un moyen simple et commode pour transmettre l'information relative à un ordonnancement, c'est un moyen médiocre pour concevoir ce même ordonnancement malgré qu'il puisse exprimer d'une manière stricte l'exigence de postériorité.

La méthode PERT est venue alors pour éliminer les défauts du diagramme de Gantt. En effet, si on a par exemple le diagramme suivant :

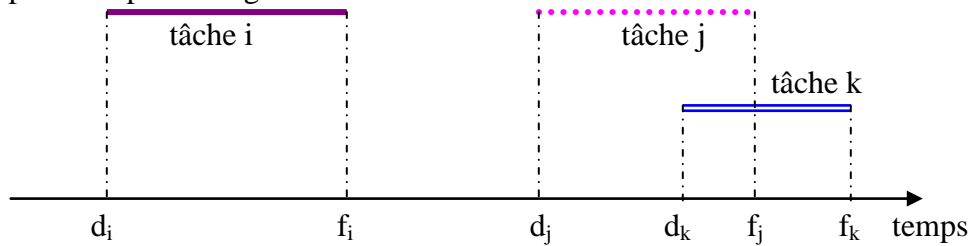


Figure III.6. (a)

Remarquons que les barres sont horizontales et le diagramme de ce fait ne tire pas réellement parti des deux dimensions du plan dans lequel il se situe.

Essayons d'abandonner cette convention, et dessinons les barres en prenant en compte la postériorité des deux tâches j et k par rapport à i [ROY 70] :

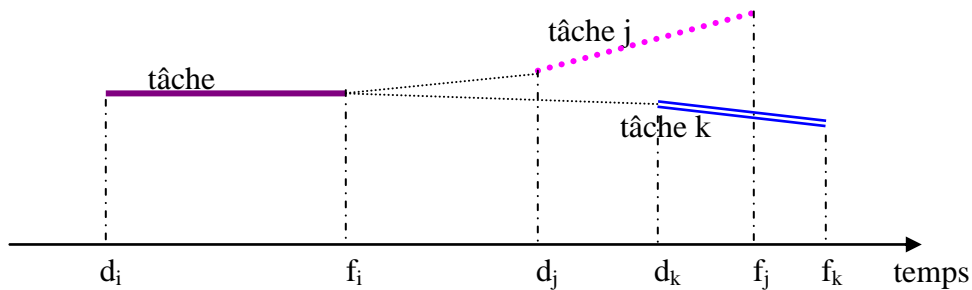


Figure III.6. (b)

Ce nouveau diagramme conduit tout naturellement au graphe suivant :

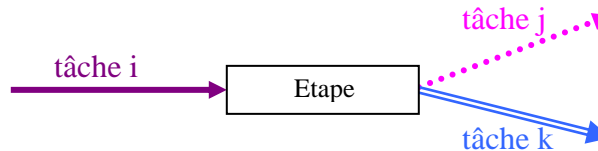


Figure III.6. (c)

Figure III.6. Le passage du diagramme de Gantt au graphe PERT.

Selon Roy [ROY 70], le passage du diagramme au graphe s'accompagne de la disparition de l'axe des temps et de la métrique associée.

Signalons que ce passage n'est pas encore définitif. On verra plus tard l'introduction des tâches fictives et sa représentation.

Dans le graphe PERT alors, appelé graphe potentiels-étapes, une tâche est représentée par un arc auquel on donne le même code, deux arcs u et v tels que $T(u) = I(v)$ si et seulement si la tâche u précède la tâche v . Les extrémités initiale et terminale d'un arc sont respectivement les événements début de tâche et fin de tâche, elles sont appelées étape. Les durées sont portées sur les arcs correspondants.

Dans la figure suivante :

- le graphe se construit de gauche à droite ;
- les points d'interrogation indiquent que les tâches antérieures à u et postérieures à v ne sont pas encore traitées.
- chaque flèche représente une tâche à effectuer ;
- les points représentent les étapes ;
- les tâches (arcs) qui partent d'un point (étape) ne peuvent commencer que lorsque les tâches qui arrivent au point sont tous terminées ;
- la longueur des flèches n'a aucun lien avec les durées inscrites sur l'arc.

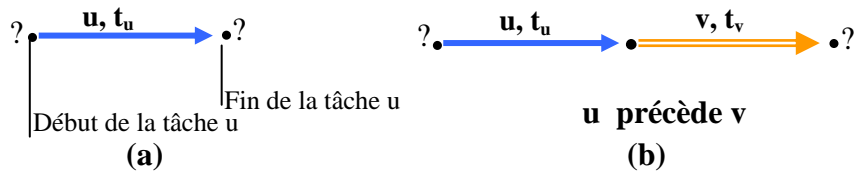
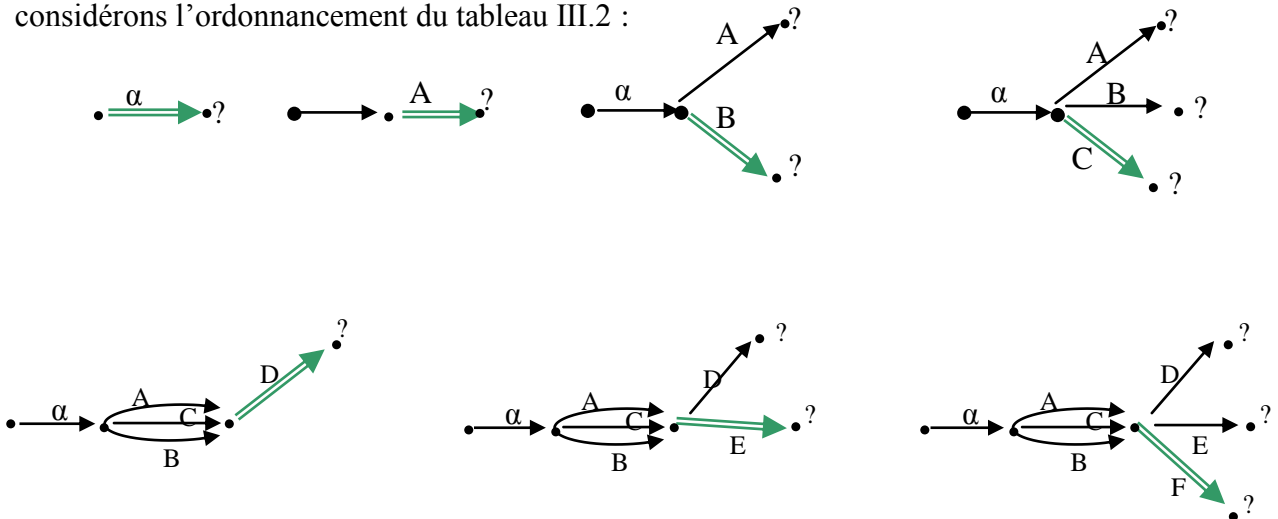


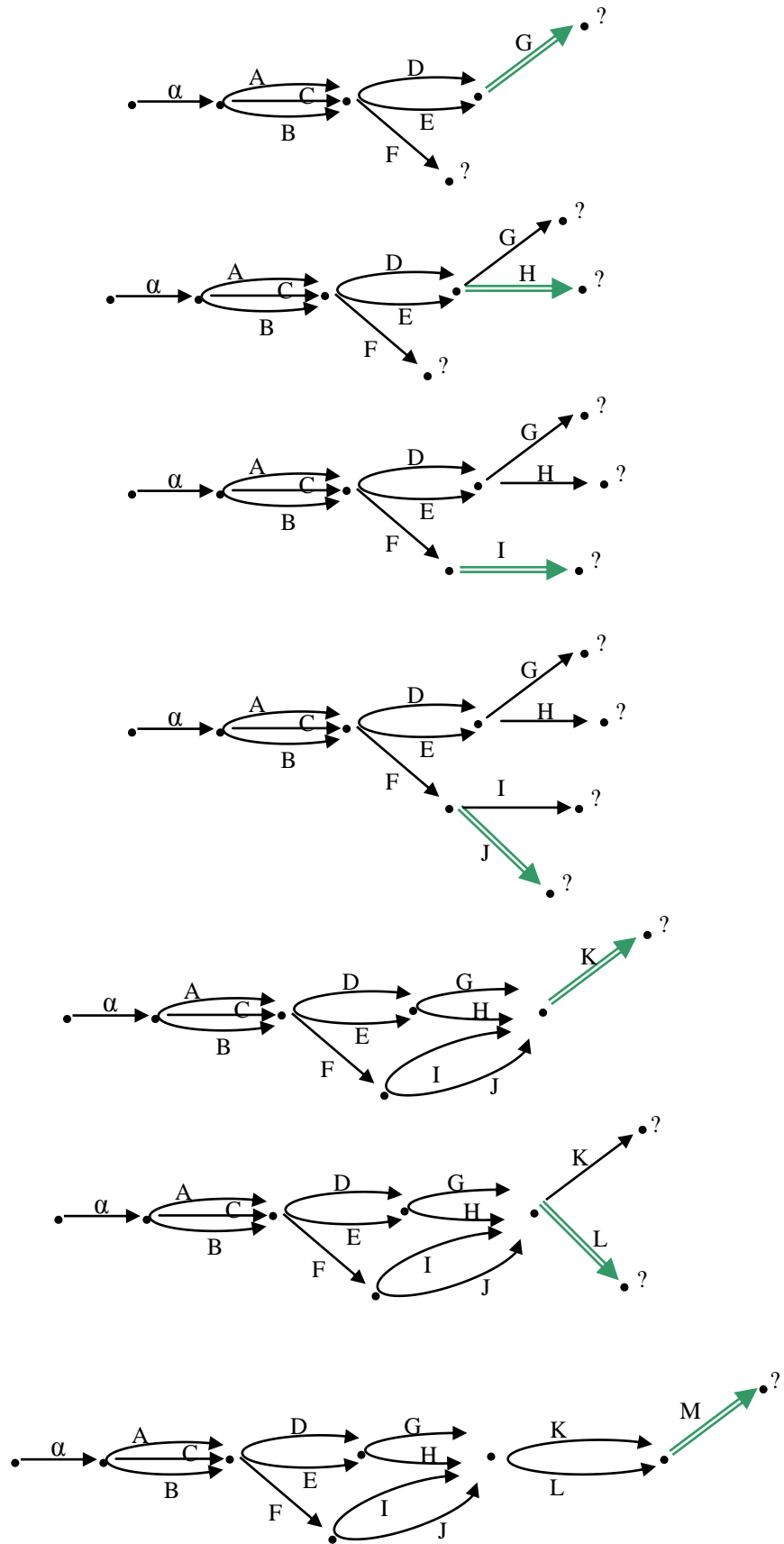
Figure III.7. La tâche u précède la tâche v dans le graphe PERT.

IV.1.3.1 Dessin du graphe PERT

Généralement, le praticien trouve des difficultés à dessiner le graphe. La figure suivante (Figure III.8.) présente une technique nouvelle de construction du graphe PERT. Elle est plutôt pédagogique.

Notre méthode donne les étapes successives de la construction du graphe PERT : on prépare d'abord la table par un tri topologique vu précédemment, ensuite on balaye la table selon la colonne des codes et la tâche correspondant à la ligne de balayage en cours est rajoutée en tenant compte des antériorités. On verra dans le chapitre V une autre technique de construction du graphe PERT en utilisant les résultats sur les graphes adjoints. Pour illustrer cette technique, considérons l'ordonnancement du tableau III.2 :





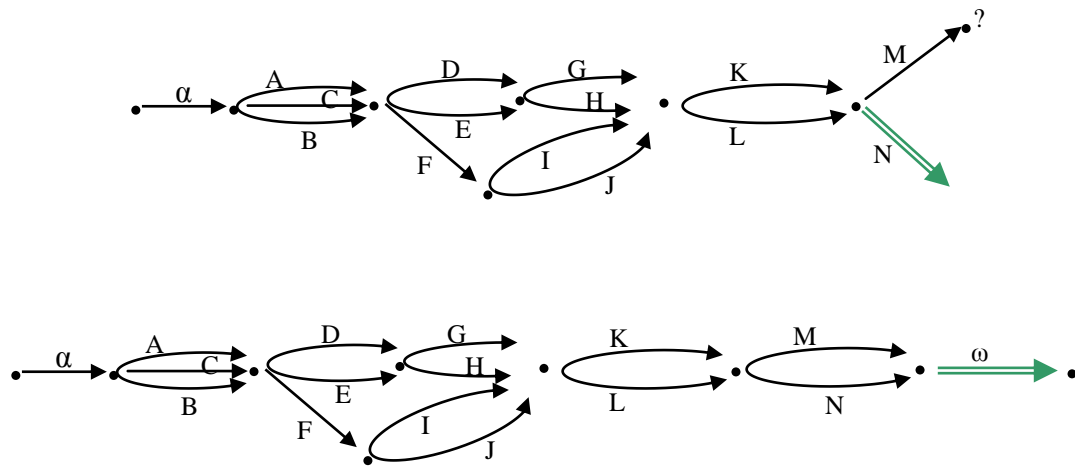


Figure III.8. Le graphe PERT construit par ajout successif des tâches, les durées n'étant pas inscrites.

IV.1.3.2 Notion de tâche fictive

Si le graphe de la méthode des potentiels et celui de la méthode PERT sont très proches, ce n'est pas toujours le cas. La construction du graphe PERT pose des problèmes qui amènent à ajouter des arcs fictifs (virtuels ou artificiels) qui ne correspondent à aucune tâche.

Supposons maintenant que le problème d'ordonnancement comporte 4 tâches a, b, c et d telles que a précède c sans précéder d et b précède c et d.

Dans le graphe PERT, la représentation n'est pas possible sans une légère modification ; en effet, a et b précèdent c, les extrémités terminales de a et b devraient coïncider avec l'extrémité initiale de c. Or, l'extrémité initiale de d devrait coïncider aussi avec l'extrémité terminale de b donc de a mais a ne précède pas d. Ce problème est résolu par l'introduction d'une tâche fictive f de durée 0 dont le principe est de séparer les extrémités terminales de a et b en éclatant le sommet s, tout en gardant les contraintes de succession.

L'introduction de la tâche fictive modifie le tableau des antériorités lequel donne la correspondance dans le graphe des potentiels.

Code (s)	Anté.
c	a, b
d	b

Tableau III.3. (a) Un sous-tableau des antériorités de c et d.

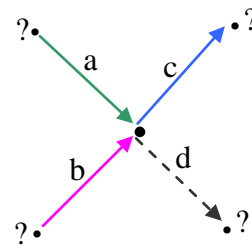


Figure III.9. (a) Problème de représentation

Code (s)	Anté.
c	a, f
d	b
f	b

Tableau III.3. (b) Nouveau sous-tableau des antériorités c, d et f.

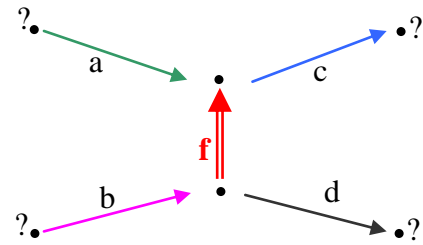


Figure III.9. (b) Introduction de la tâche fictive f et représentation dans le graphe PERT.

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 [FIN 02]):

Codes	Durées	Antér.
α	0	-
A	2	α
B	2	α
C	2	H
D	3	α
E	4	B,G
F	2	C,I
G	3	A,D
H	4	B,D
I	5	H
J	3	C
ω	0	E, J, F

Niveau	Sommet
Niveau I	α
Niveau II	A, B, D
Niveau III	G, H
Niveau IV	C, E, I
Niveau V	F, J
Niveau VI	ω

Tableau III.4. La table d'ordonnement T et l'organisation des sommets en niveaux.

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

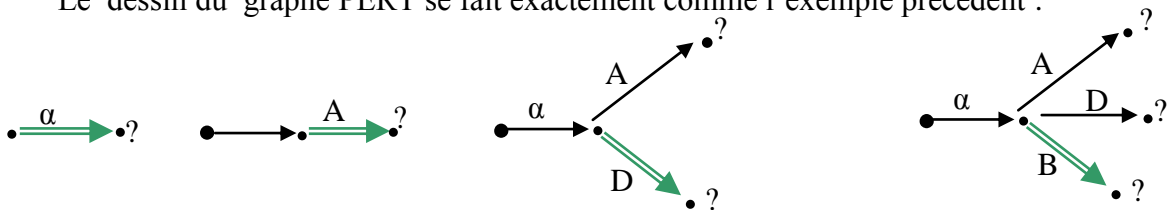


Figure III.10 (a).

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 f_1 (voir *Figure III.10 (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 f_2 (voir figure III. 10 (c)).

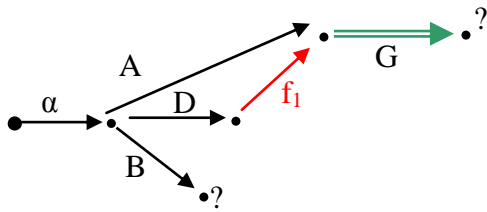


Figure III.10. (b)

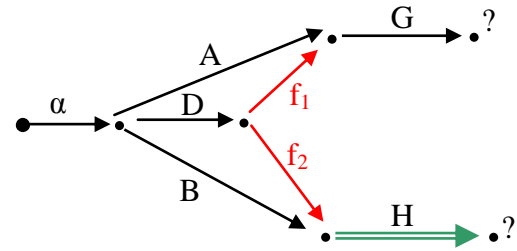


Figure III.10. (c)

et nous continuons le même processus jusqu'à la dernière ligne de la table :

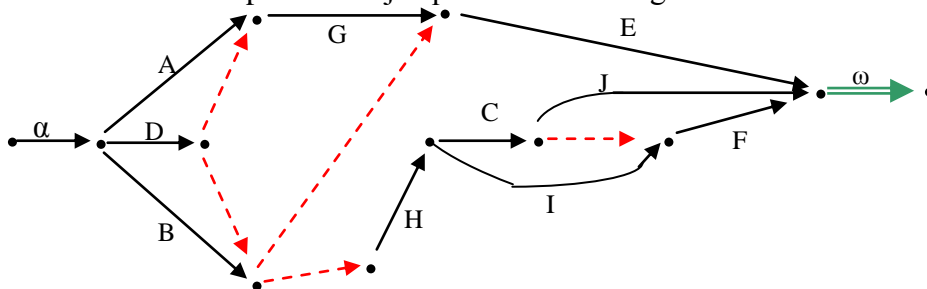


Figure III.10. Le graphe PERT de la table III.4 avec un minimum de 5 tâches fictives.

IV.1.4 Modularité

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

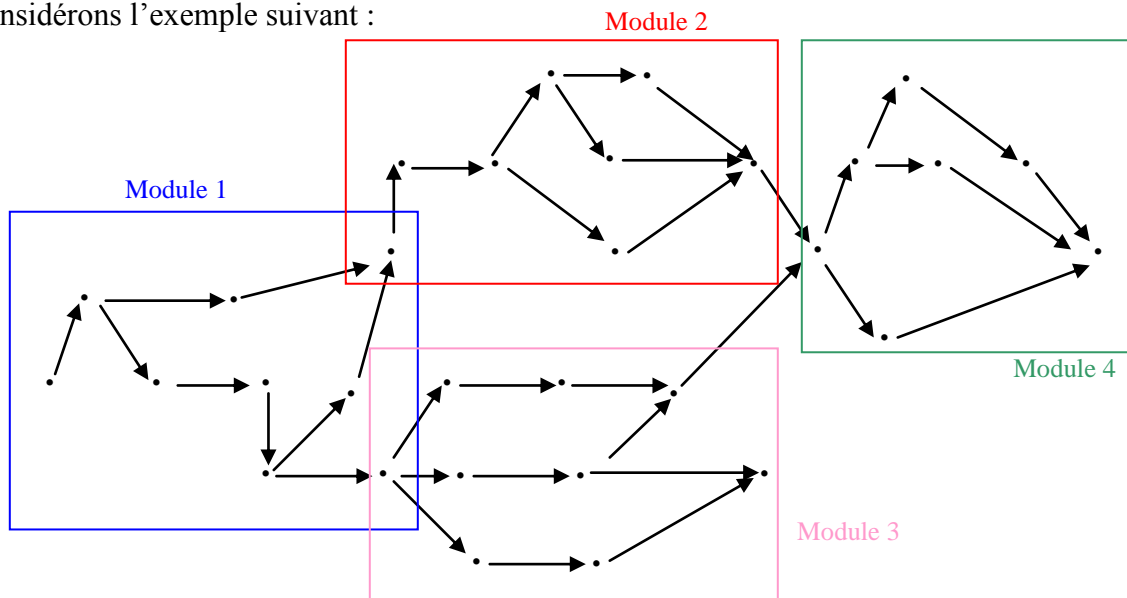


Figure III.11. La modularité dans le graphe PERT.

IV.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 PERT.

Un inconvénient de cette dernière par rapport à la méthode des potentiels 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 PERT plus ou moins simples. N'oublions pas qu'il y a une infinité de graphes PERT 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 PERT sous forme de diagramme de Gantt [DUV 00].

V. Résolution du problème central

La croissance de la gestion de systèmes dans le domaine militaire-industriel est le facteur le plus important qui a permis de développer le PERT-temps. L'ordonnancement des grands projets nécessite un dispositif de planification et de gestion obligatoire. Ce dispositif n'est que le PERT-temps.

Selon [HIL 04] le PERT-temps a prouvé qu'il est un outil puissant dans le kit des directeurs ou chefs de projets pour la planification, coordonnant et intégrant les grands projets multidimensionnels.

Le graphe PERT 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.

Le graphe PERT 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 (α) et sa fin (ω), [HEN 08] propose une méthode de résolution par la programmation linéaire.

Supposer que notre réseau de projet a $n+1$ noeuds, 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 :

$$x_j \geq x_i + D_{ij} \quad \text{où } D_{ij} \text{ est la durée de l'activité } (i,j).$$

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

$$Z = x_n$$

Sujet à

$$x_0 = 0$$

$$x_j - x_i - D_{ij} \text{ pour toute activité } (i, j)$$

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.

[PAR 05] propose une démarche en deux phases qu'il les a nommé propagation avant et propagation arrière :

- Propagation avant : Cette première phase permet de déterminer les dates de début au plus tôt des tâches en parcourant le réseau dans le sens des contraintes de précédence. A partir de la date de début au plus tôt de la première tâche du graphe, un algorithme de plus long chemin entre cette tâche et toutes les autres est appliqué afin de déterminer la date la date de début au plus tôt de toutes les

tâches. La différence entre la date de fin de la dernière tâche et la date de début de la première permet de déterminer la durée minimale du projet.

- Propagation arrière : Cette seconde phase permet de déterminer la date de début ou de fin au plus tard des tâches. L'objectif est de minimiser la durée du projet. La date de fin au plus tard de la dernière tâche est donc posée égale à sa date de fin au plus tôt. Comme lors de la première phase, le chemin le plus long entre la dernière tâche et chacune des autres permet d'obtenir la date de fin au plus tard de chacune des tâches.

On détermine ainsi le (les) chemin (s) critique (s) correspondant à une suite de tâches dont les dates de début au plus tôt et au plus tard sont égales.

V.1 Dates et marges associées à une tâche

a) Date de début au plus tôt :

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

b) Date de début au plus tard

La date de début au plus tard d'une tâche A da_A est la date limite de sa réalisation. Après quoi le projet sera retardé.

c) Date de fin au plus tôt

La date de fin au plus tôt d'une tâche A fo_A est la date à laquelle A se termine en commençant à sa date de début au plus tôt.

d) 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.

e) Intervalle de placement

L'intervalle de placement (ou de flottement) ip_A est la date de fin au plus tard de A diminuée de sa date de début au plus tôt. A ne peut être exécutée que dans cet intervalle qui sera symbolisé dans les figures par des crochets [].

f) 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.

g) 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. ml_A est symbolisée dans les figures suivantes par •

h) Marge liée

La marge liée est la différence entre la marge totale et la marge libre. Elle caractérise le couplage entre l'ordonnancement d'une tâche et celui des tâches qu'elle précède.

i) Marge indépendante

La marge indépendante ou certaine d'une tâche est l'écart positif (s'il existe) qui sépare sa date de début au plus tôt de sa date de début au plus tard.

i) Date de marge libre

La date de marge libre d'une tâche A notée dml_A , est la date la plus tardive pour que A commence afin de ne pas diminuer les marges totales des tâches suivantes.

V.2 Calcul des dates et marges

Une tâche ne peut démarrer que lorsque les tâches de tous les chemins de α à A sont exécutées, en particulier celles du plus long chemin de α 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 α à ω .

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

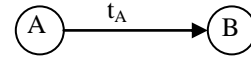
Les graphes étant sans circuits, nous appliquerons l'algorithme de Belleman pour déterminer les dates de début au plus tôt. Les durées sur les arcs de même extrémité initiale étant

égales dans le graphe des potentiels préparé, on vérifie que les tâches ayant les mêmes antériorités ont les mêmes dates de début au plus tôt.

a) Calcul des dates

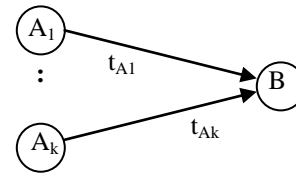
Pour calculer la date de début au plus tôt de la tâche A do_A on procède comme suit :

- La date de début au plus tôt de la tâche α 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 :

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

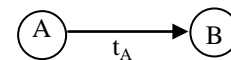
Figure III.12 Calcul de do_A dans le cas d'un ou plusieurs arcs.

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 ω est initialisée à sa date de début au plus tôt c.à.d.

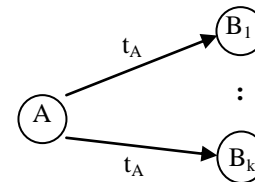
$$da_\omega = do_\omega$$

- 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 :

$$da_A = \min (da_{B1}, \dots, da_{Bk}) - t_A$$

Figure III.13. 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 :

$$fo_A = do_A + d_A$$

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 : $fa_A = da_A + t_A$

b) 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.

$$ml_A = \min (do_{A1}, do_{A2}, \dots, do_{Ak}) - fo_A$$

Les tâches A_1, A_2, \dots, A_k sont les tâches qui succèdent à A.

La date de marge libre de A est la date de début au plus tôt de A augmentée de sa marge libre : $dml_A = do_A + ml_A$

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

Ces notations sont très simples à représenter dans un diagramme de Gantt. Cependant, les calculs se feront soit dans le graphe des potentiels préparé, soit dans le graphe PERT. Nous utilisons dans ce qui suit le graphe des potentiels.

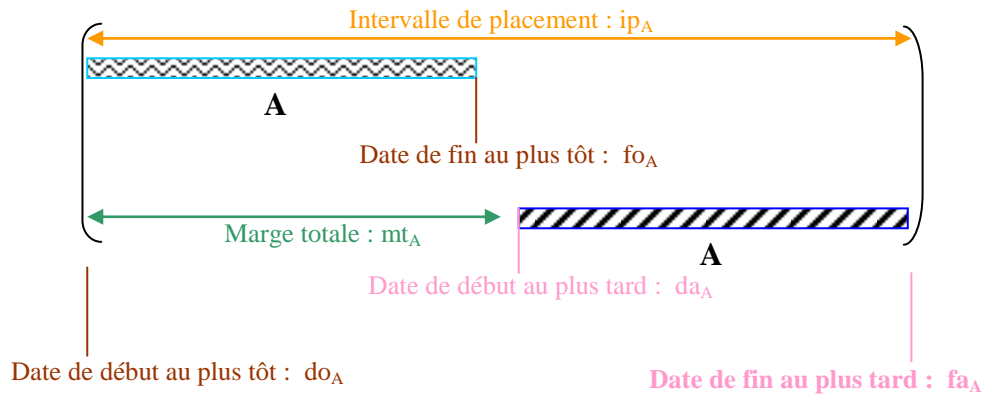
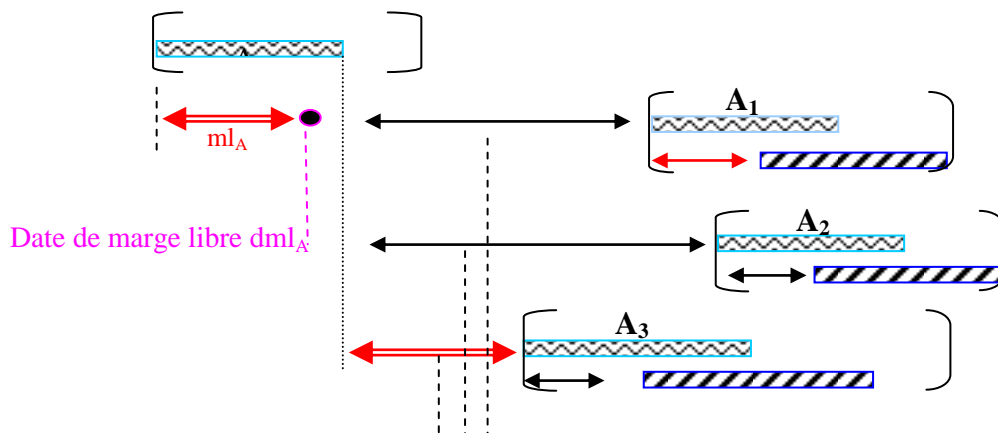


Figure 14. (a).



$$\text{Marge libre de A : } ml_A = \min [do_{A1}, do_{A2}, do_{A3}] - fo_A$$

Les marges totales de A_1, A_2, A_3 doivent rester inchangées

Figure 14. (b).

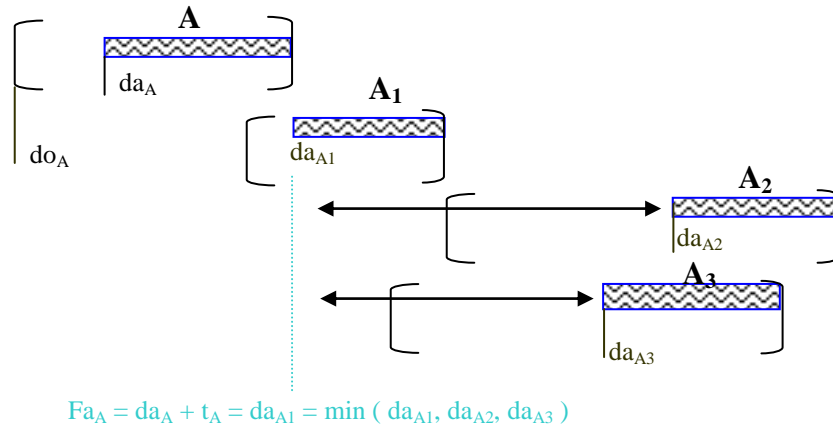


Figure 14. (c).

Figure III.14. L'intervalle de placement d'une tâche A, les dates et les marges associées à A représentées par le diagramme de Gantt. A₁, A₂ et A₃ sont les tâches qui succèdent à A.

V.3 Le chemin critique

Le chemin critique dans le graphe des potentiels ou PERT est le plus long chemin de α à ω . Sa valeur do_ω est la durée minimale du projet.

On appellera sous-chemin tout sous-chemin critique. Si A₁, A₂, ..., A_k sont les postériorités de A, on sait que : $fa_A = da_A + t_A \leq da_{Ai}$, $i = 1, \dots, k$ et $da_A = \min_{i=1, \dots, k} da_{Ai} - t_A$

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

$$da_A = do_A$$

Prenons l'ordonnancement suivant (Tableau III.7) et calculons les dates et les marges :

Code	Durée	Antériorités
α	0	-
A	5	α
B	7	α
C	8	α
D	17	A,B,C
E	11	A,B,C
F	9	D,E
G	12	D,E
H	13	D,E
I	8	F
J	10	G,H
K	11	G,H
ω	0	I,J,K

Tableau III.7. Une table d'ordonnancement T.

Calcul des dates de début au plus tôt
Le calcul se fait dans l'ordre croissant des niveaux et les sommets d'un même niveau dans un ordre quelconque.
$do_{\alpha} = 0$ $do_A = do_{\alpha} + 0 = 0$ $do_B = do_{\alpha} + 0 = 0$ $do_C = do_{\alpha} + 0 = 0$ $do_D = \text{Max}[do_A + 5, do_B + 7, do_C + 8] = 8$ $do_E = \text{Max}[do_A + 5, do_B + 7, do_C + 8] = 8$ $do_F = \text{Max}[do_D + 17, do_E + 11] = 25$ $do_G = \text{Max}[do_D + 17, do_E + 11] = 25$ $do_H = \text{Max}[do_D + 17, do_E + 11] = 25$ $do_I = 24 + 9 = 34$ $do_J = \text{Max}[do_G + 12, do_H + 13] = 38$ $do_K = \text{Max}[do_G + 12, do_H + 13] = 38$ $do_{\omega} = \text{Max}[do_I + 8, do_J + 10, do_K + 11] = 50$

Calcul des dates de début au plus tard
Le calcul se fait dans l'ordre décroissant des niveaux et les sommets d'un même niveau dans un ordre quelconque.
$da_{\omega} = do_{\omega} = 50$ $da_I = do_{\omega} - 8 = 42$ $da_J = do_{\omega} - 10 = 40$ $da_K = do_{\omega} - 12 = 38$ $da_F = da_I - 9 = 33$ $da_G = \text{Min}[da_J, da_K] - 12 = 26$ $da_H = \text{Min}[da_J, da_K] - 13 = 25$ $da_D = \text{Min}[da_F, da_G] - 17 = 8$ $da_E = \text{Min}[da_F, da_G] - 11 = 14$ $da_A = \text{Min}[da_D, da_E] - 5 = 3$ $da_B = \text{Min}[da_D, da_E] - 7 = 1$ $da_C = \text{Min}[da_D, da_E] - 0 = 0$ $da_{\alpha} = \text{Min}[da_A, da_B, da_C] - 0 = 0$

Tableau III.8. (a) Dates de début au plus tôt et au plus tard du tableau de la figure III.4.

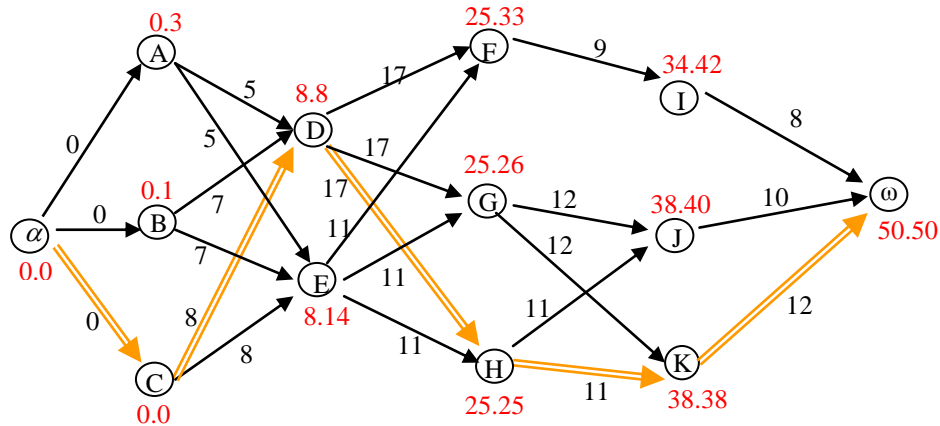


Figure III.15. Les dates de début au plus tôt et au plus tard dans le graphe des potentiels.

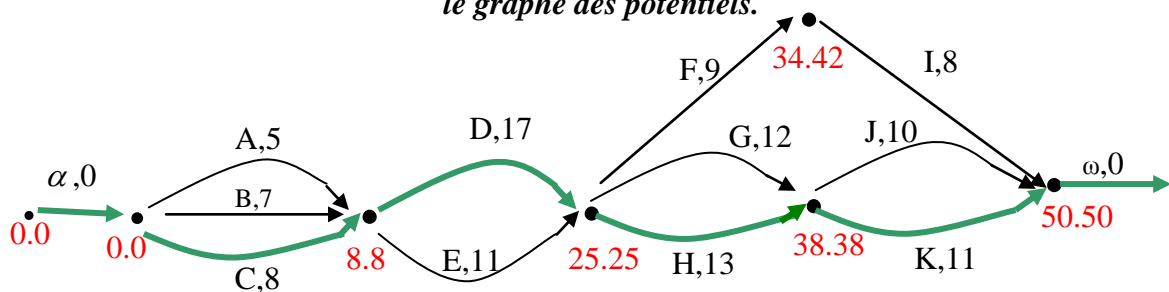


Figure III.16. Les dates de début au plus tôt et au plus tard dans le graphe PERT

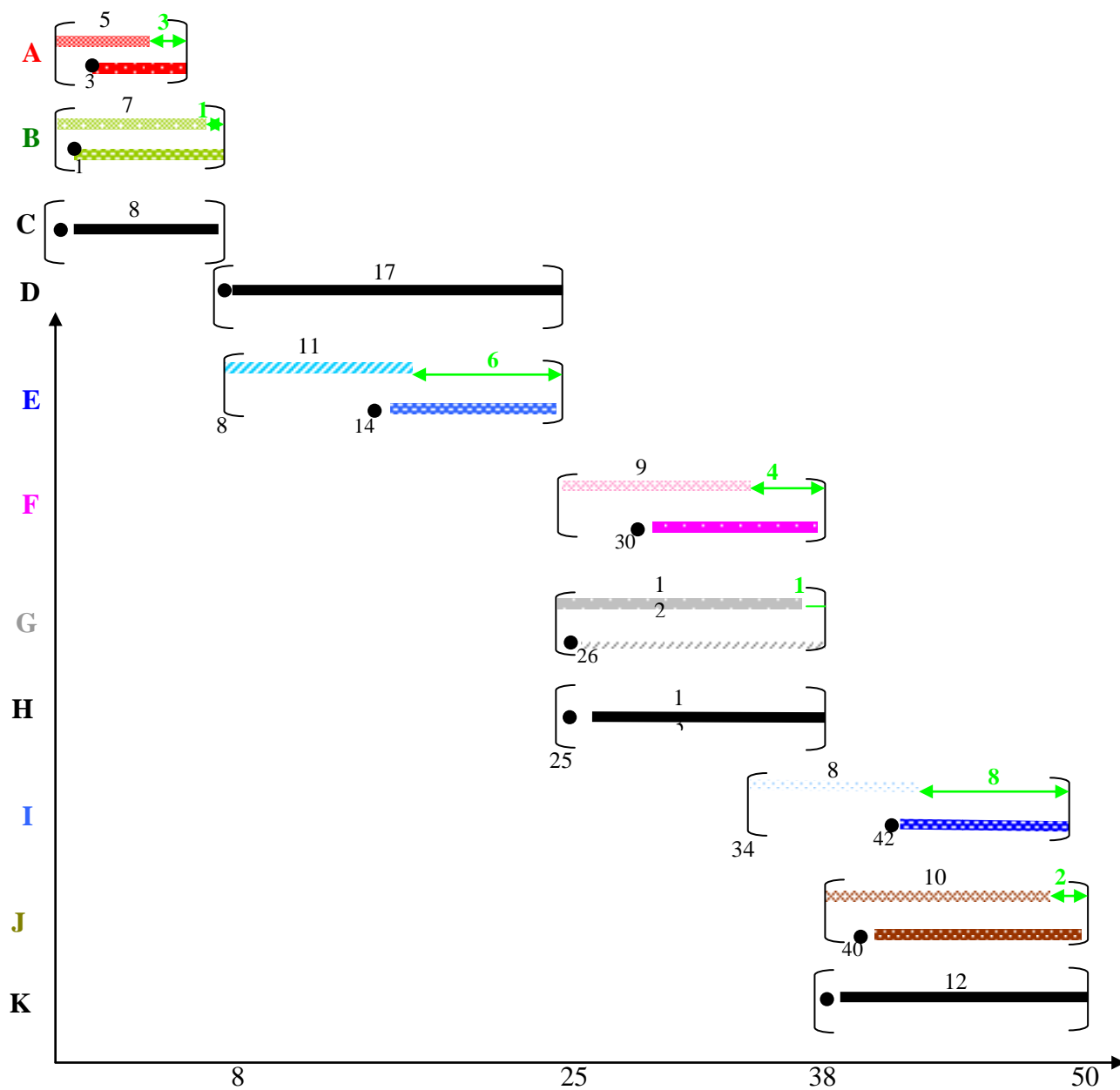


Figure III.17. Le diagramme de Gantt.

Toutes les données concernant cet ordonnancement peuvent être regroupées dans le tableau suivant :

Tâche	Antériorité	Durée	Début au Plus tôt	Fin au Plus tôt	Début au plus tard	Fin au plus tard	Intervalle de placement	Marge totale	Marge libre	Date de marge libre
α	-	0	0	0	0	0	0	0	0	0
A	α	5	0	5	3	8	8	3	3	3
B	α	7	0	7	1	8	8	1	1	1
C	α	8	0	8	0	8	8	0	0	0
D	A,B,C	17	8	25	8	25	17	0	0	8
E	A,B,C	11	8	19	14	25	17	6	6	14
F	D,E	9	25	34	33	42	17	8	0	25
G	D,E	12	25	37	26	38	13	1	1	26
H	D,E	13	25	38	25	38	13	0	0	25
I	F	8	34	42	42	50	16	8	8	42
J	G,H	10	38	48	40	50	12	2	2	40
K	I,J,K	12	38	50	38	50	12	0	0	38
ω	I,J,K	0	50	50	50	50	0	0	0	50
(1)	(2)	(3)	(4)	(5)= (3)+ (4)	(6)	(7)= (3)+(6)	(8)= (7)-(4)	(9)= (6)-(4)	(10)	(11)= (4)+ (10)

Tableau III.9. Tableau des données et des résultats.

VI Conclusion

Nous venons de voir dans ce chapitre la gestion de projet avec ses différentes étapes. Ensuite nous avons situé la phase ordonnancement dans le cycle de vie d'un projet.

Nous avons étudié les différentes modélisations des problèmes d'ordonnancement de projet dans la littérature ainsi que sa résolution. Cette étude est accompagnée de plusieurs points nouveaux tels que le dessin du graphe PERT. On verra plus tard une autre technique nouvelle de dessin du graphe PERT par les bipartis.

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 et du suivi du projet [PAR 05].

Chapitre IV

Extensions du problème central d'ordonnancement

Sommaire

I Introduction	66
II Ordonnancement avec des durées incertaines	67
II.1 Évaluation des durées des tâches	69
II.1.1 La fonction Bêta (β)	70
II.2 Discussion	73
II.2.1 Critiques de l'approche PERT probabiliste	74
II.2.2 Quelques suggestions	77
II.3. Résolution par simulation	79
III Le PERT coût	80
III.1 Diminution du coût total du projet	82
III.2 Diminution de la durée au moindre coût	82
IV Ordonnancement de projet à moyens limités	83
V Ordonnancement avec contraintes disjonctives	84
VI Ordonnancement multi-projet	84
VII Conclusion	85

I Introduction

L'établissement de l'ordonnancement du projet est une matière qui a reçu la recherche étendue ces dernières décennies. Le chapitre III a décrit les techniques fondamentales d'ordonnancement qui sont largement appliquées et soutenues par de nombreux systèmes d'ordonnancement commerciaux. Une variété de techniques spéciales a été également développée pour résoudre des circonstances ou des problèmes spécifiques. Avec la disponibilité des ordinateurs et des logiciels plus puissants, l'utilisation des techniques avancées devient d'une plus grande importance pour la pratique. En ce chapitre, nous examinons une extension du PERT, à savoir le cas probabiliste qui peut être utilisé à cet égard.

La technique PERT de trois évaluations est une approche traditionnelle et bien connue chez les experts de l'évaluation. Les estimateurs proposent des durées optimistes, pessimistes et les plus probables. Ensuite, une distribution de probabilité bêta est établie pour adapter les données [PES 07].

Cette technique sera présentée dans ce chapitre, qui sera suivie d'une analyse, critiques et de nouvelles propositions sont présentées pour apporter des corrections à cette méthode.

Le chapitre survole l'ensemble des techniques basées sur les principes de la simulation.

Nous présenterons également dans ce chapitre, l'extension du problème central en étudiant la minimisation des coûts.

Nous consacrerons une partie de ce chapitre à la prise en compte des contraintes disjonctives et nous le terminerons par l'étude de l'ordonnancement de projet à moyens limités où l'objectif est soit : minimiser la durée totale de réalisation ou bien l'optimisation de l'utilisation des ressources qui est liée à un coût.

II Ordonnancement avec des durées incertaines

La section V.3 du chapitre III a décrit la recherche *du chemin critique* pour la situation dans laquelle les durées des tâches sont fixes et connues. Malheureusement, les durées des tâches sont des évaluations du temps réel requis, or il y a une quantité significative d'incertitudes liées aux durées réelles.

Pendant les étapes de planification préliminaires pour un projet, l'incertitude dans les durées des tâches est particulièrement grande mais dans le problème central la portée et les obstacles au projet sont éliminés.

Remarquons que les tâches qui sont en dehors de la commande du gestionnaire sont susceptibles d'être plus incertaines. Par exemple, le temps requis pour gagner l'approbation de normalisation pour des projets peut changer énormément. D'autres événements externes tels que le temps défavorable, les effondrements de fossé, ou les grèves de main-d'oeuvre rendent les évaluations de durée particulièrement incertaines. Tout cela mène à revoir l'approche déterministe du problème central.

[ARM 01] présente deux approches simples pour traiter l'incertitude dans les durées des tâches justifient une certaine discussion avant de présenter les algorithmes plus formels pour traiter l'incertitude.

D'abord, l'incertitude dans les durées des tâches peut simplement être ignorée et l'établissement de l'ordonnancement se fait en utilisant la durée prévue ou la plus susceptible pour chaque tâche. Puisque seulement une évaluation de durée doit être faite pour chaque tâche, cette approche réduit le travail exigé en installant le programme original. Les méthodes formelles

en présence d'incertitude dans le processus d'établissement d'ordonnancement exigent plus de travail et de prétentions. Tandis que cette approche simple pourrait être défendue selon [LIT 06], elle a deux inconvénients : D'abord, l'utilisation des durées prévues des tâches a typiquement comme conséquence des programmes excessivement optimistes pour l'accomplissement du projet. En second lieu, l'utilisation des durées simples des tâches produit souvent un ensemble rigide et inflexible de la part des planificateurs. Comme les conducteurs des travaux apprécient l'utilisation des durées des tâches variables, ils peuvent desserrer la confiance au réalisme d'un programme basé sur le principe de contrôle continu, de surveillance et de mise à jour du programme à la lumière de l'impératif réel d'expérience. Autrement dit, le programme de projet est rapidement périmé.

Une deuxième approche simple qui contourne l'incertitude mérite également la mention. Beaucoup de directeurs identifient que l'utilisation des durées prévues peut avoir comme conséquence des programmes excessivement optimistes, ainsi selon [HEN 03] ils incluent une allocation d'éventualité dans leur évaluation des durées des tâches. Par exemple, une tâche avec une durée prévue de deux jours pourrait être programmée pendant une période de 2.2 jours. L'application systématique de cette éventualité aurait comme conséquence une augmentation de dix pour cent du temps prévu pour l'accomplissement du projet. L'objectif de l'utilisation de cette *heuristique* d'éventualité est d'avoir des programmes plus précis, il est probable que les méthodes d'établissement du programme formelles qui incorporent l'incertitude plus formellement soient utiles pour obtenir une plus grande exactitude ou en comprenant les effets du retard des tâches.

L'approche formelle la plus commune pour incorporer l'incertitude dans le processus d'établissement du programme d'ordonnancement est de l'appliquer tel qu'il est décrit dans la section V.3 du chapitre III, ensuite analyser les résultats d'une perspective probabiliste. Ce processus est désigné habituellement sous le nom de méthode PERT-Temps aléatoire (The PERT Three-Estimate Approach) [MOJ 05].

L'évaluation est une activité fondamentale dans la planification du projet, et l'effort considérable a été dépensé pour concevoir les modèles formels d'évaluation. Toujours, dans la plupart des cas, l'évaluation est effectuée officieusement, en utilisant l'expérience, intuition, probablement quelques données historiques. Cette approche est connue en tant qu'évaluation experte [JOR 04a], signifiant qu'un ou plusieurs experts fournissent une évaluation basée sur leur expérience.

Bien que largement adoptée, l'évaluation des experts est loin de la perfection. Une issue importante avec l'expert est que l'évaluation manque du réalisme [JOR 04c], [JTM 04b]: les experts sont souvent trop optimistes (des temps estimés sont souvent excédés) et présomptueux (ils fournissent les gammes étroites pour leur évaluation, même devant l'incertitude élevée).

II.1 Évaluation des durées des tâches

La durée d'une tâche est le temps écoulé requis pour la réalisation de cette tâche. Selon [RUL 00], estimer la durée est probablement l'un des dispositifs les plus critiques de CPM. Plusieurs raisons poussent le manager de projet à écarter la réalisation d'un calendrier rigide.

Pour cette raison, PERT propose une technique d'évaluation des tâches basée sur trois estimations de la durée d'une tâche donnée.

[DH 02] explique que chacune des trois évaluations de temps assume un niveau statique d'utilisation de ressource. Les évaluations devraient être aussi bonnes que possible parce que les résultats de PERT dépendent directement d'elles. Il n'est pas facile d'obtenir des évaluations précises. Elle exigera la recherche, la collaboration avec des membres d'équipe de planification, et de réalisation.

Ces trois évaluations devraient répondre aux critères suivants :

- 1) *le temps optimiste* : la période de temps minimum pour que la tâche soit accomplie, c.-à-d., le temps qu'elle prendrait pour son exécution si tout allait mieux que prévu (cette estimation est notée a)
- 2) *temps le plus susceptible* : la meilleure évaluation de la période de temps où la tâche peut être accomplie, c.-à-d., la plus probable (cette estimation est notée m)
- 3) *temps pessimiste* : la période de temps maximum que prendrait la tâche pour qu'elle soit accomplie, c.-à-d., la plus pessimiste (cette estimation est notée b).

Ces nombres (a , b et m) peuvent être obtenus, par exemple, en questionnant les chefs d'ateliers, de chantiers, de laboratoires, etc. qui exécutent les tâches ou bien en utilisant les formules de probabilités qu'on va citer dans ce qui suit.

Il est acceptable d'énoncer ces évaluations en jours, semaines, ou mois aussi longtemps que la mesure est employée uniformément. Une fois que faites, les évaluations de temps de la tâche sont connexes et ne devraient pas être changées. Les rapports suivants de temps doivent être respectés : $a \leq m \leq b$

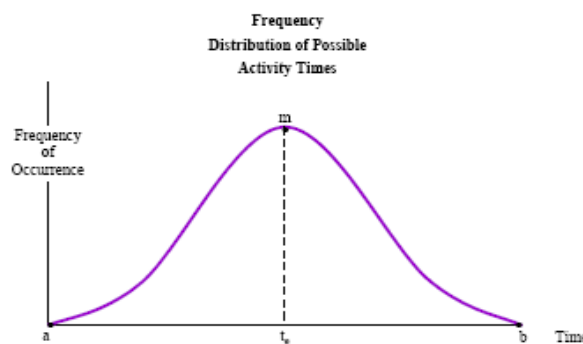


Figure IV.1. Distribution du temps des activités tiré de [KIM 01].

Les valeurs relatives (positions) de a , de m , et de b sur chaque distribution, naturellement, dépend du jugement et des calculs de l'estimateur. Une fois qu'établies, leurs positions relatives sur la distribution affectent la valeur ou la position du t_e .

Pour l'usage du procédé de PERT, une évaluation de t_e (temps moyen de la tâche) se calcule comme suit :

$$t_e = \frac{(a + 4m + b)}{6}$$

Si on connaît les distributions des durées des tâches, on peut calculer la durée moyenne et la variance pour chaque tâche. Mais lorsqu'on ne possède aucune statistique à l'avance, on peut faire appel alors à une distribution de fréquence $f(x)$ qui a les propriétés suivantes :

- La probabilité est nulle lorsque la tâche demande un temps supérieur à une durée pessimiste b et inférieur à une durée optimiste a .
- Le temps pessimiste b est la durée la plus longue pour une tâche, autrement dit le temps probable plus toutes les majorations dues aux événements redoutés.

Le temps optimiste a est la durée la plus courte pour une tâche, autrement dit le temps probable moins toutes les diminutions dues à des circonstances favorables et sans événements défavorables.

- Il existe un temps réaliste qui est la durée la plus probable pour une tâche donnée et on le note m .

- L'écart entre la durée maximale b et la durée minimale a d'une tâche est censé représenter six écarts-types de la distribution.

II.1.1 La fonction Bêta (β)

La fonction $\beta(p, q)$ est définie par :

$$\beta(p, q) = \int_0^1 x^{p-1} (1-x)^{q-1} dx = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)}, p, q > 0$$

Un changement de variable de la forme :

$$u = \frac{t-a}{b-a} \text{ donne } \beta(p, q) = \frac{1}{(b-a)^{p+q-1}} \int_a^b (x-a)^{p-1} (b-x)^{q-1} dx$$

De sorte que sa densité de probabilité soit :

$$f(x) = \begin{cases} \frac{(x-a)^{p-1} (b-x)^{q-1}}{(b-a)^{p+q-1} \beta(p, q)} & \text{si } x \in [a, b] \\ 0 & \text{sinon} \end{cases}$$

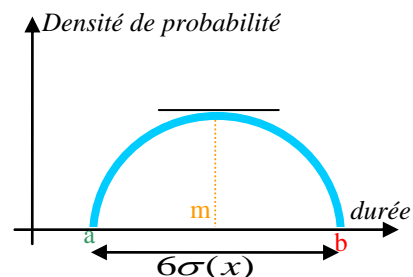


Figure IV.2. La loi Bêta.

La maximum de $f(x)$ est atteint en $m = \frac{(q-1)a + (p-1)b}{p+q-2}$,

on vérifie que $E(x) = a \cdot \frac{(b-a)p}{p+q}$ et $V(x) = \sigma^2(x) = \frac{(b-a)^2 pq}{(p+q)^2(p+q+1)}$.

En écrivant que b-a égale 6 écarts-types, ($b-a = 6\sigma(x)$), il s'ensuit le choix : $p = 3 + \sqrt{2}$ et

$p = 3 - \sqrt{2}$, alors $E(x) = \frac{a+4m+b}{6}$ et $V(x) = \sigma^2(x) = \frac{(b-a)^2}{36}$

Pour chaque tâche A, on détermine les estimations des quantités a, b, puis m et on déduit des formules précédentes, les estimations de sa durée moyenne E (A) et de sa variance $\sigma^2(A)$.

La durée moyenne de tout le projet E (P) et la variance $\sigma^2(P)$ sont alors respectivement la somme des durées moyennes et la somme des variances des tâches critiques.

Sachant que :

si

. X_1, X_2, \dots, X_n sont des variables aléatoires indépendantes de même lois de probabilité,

. la somme $Y = \sum_{i=1,n} X_i$ est une variable aléatoire d'espérance E(Y), et d'écart-type $\sigma(Y)$,

alors

la variable aléatoire centrée réduite $T = (Y-E(Y))/\sigma(Y)$ tend vers une loi normale lorsque n tend vers l'infini,

et supposant que le chemin critique comporte un grand nombre de tâches, on peut déduire de la table de Laplace-Gauss, la probabilité de réalisation du projet en x unités de temps en passant à la variable centrée réduite :

$$t = \frac{(x6E(P))}{\sigma(P)}$$

Mettons à l'application toutes ces connaissances et considérons l'exemple suivant ; les valeurs moyennes des durées des tâches sont calculées en utilisant la formule :

$$E(x) = \frac{a+b+4m}{6} = t_e$$

Et les variances par la formule : $\sigma^2(x) = \left(\frac{b-a}{6}\right)^2$

a, b, m étant connus pour chaque tâche, et que l'on ait obtenu les résultats suivants de l'exemple tiré de [MOU 02]:

Tâche	a	b	m	Durée	Anté.	
α	0	0	0	0	-	0
A	7	19	13	13	α	4
B	3	15	9	09	α	4
C	4	40	16	18	α	36
D	6	24	12	13	B	9
E	4	28	7	10	B	16
F	1	7	04	04	A,E	1
G	3	27	9	11	C,E	16
H	2	20	5	07	C	9
I	4	28	7	10	D	16
J	7	25	10	12	H	9
K	7	37	4	10	F	25
ω	0	0	0	0	I,J,K	0
				E(x)		V(x)

Table IV.1. Table des antériorités et les durées calculées à partir de a, b et m.

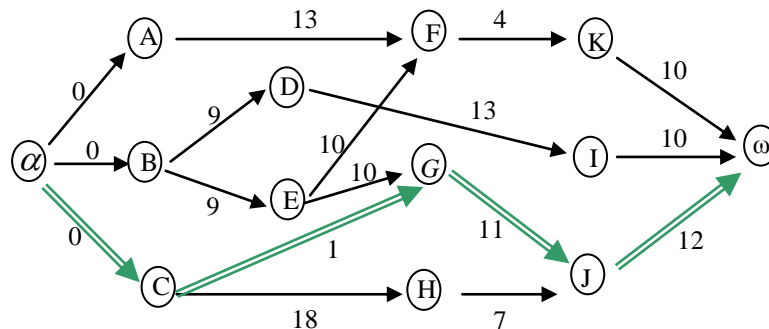


Figure IV.3. Graphe des potentiels correspondant au tableau IV.1.

En utilisant ces durées moyennes les tâches critiques sont α , C, G, J et ω .

La durée moyenne de réalisation du projet est égale à la somme des durées moyennes de α , C, G, J et ω , donc : $E(x) = 0+18+11+12 = 41$.

La variance est égale à la somme des variances de α , C, G, J et ω donc : $\sigma^2(x) = 0+36+16+9+0 = 61$. L'écart type $\sigma = \sqrt{\sigma^2(x)} = 7,81$.

On calcule la valeur centrée réduite $t = \frac{x - E(x)}{\sigma(x)}$,

et on peut trouver dans une table de Laplace-Gauss (voir annexe) la probabilité pour que le projet soit exécuté en un temps donné, par exemple :

Durée du projet	Valeur centrée réduite	Probabilité
35	$t = (35-41)/7,81 = -0,76$	$P(+t)=1-P(-t)$ $P(-t)=1-P(+t)$ $= 22,36 \%$
37	$t = (37-41)/7,81 = -0,51$	$P(-0,51) = 30,50 \%$
39	$t = (39-41)/7,81 = -0,25$	$P(-0,25) = 40,13 \%$
41	$t = (41-41)/7,81 = 0$	$P(0) = 50 \%$
43	$t = (43-41)/7,81 = 0,25$	$P(0,25) = 59,87 \%$
45	$t = (45-41)/7,81 = 0,51$	$P(0,51) = 69,50 \%$
47	$t = (47-41)/7,81 = 0,76$	$P(0,76) = 77,64 \%$
etc.		

Tableau IV.2. Récapitulation de la probabilité correspondante à chaque durée totale choisie pour le projet.

La probabilité de réalisation du projet avant la date 35 n'est que 11,51%. Il y a près de 90 chances sur 100 que le projet ne soit pas terminé. Par contre il y a près de 89% de chances que le projet soit réalisé avant la date 47.

Ces résultats, servent, pour la personne chargée de la réalisation, à prendre les décisions adéquates. Si le contrat est assorti de pénalités de retard très lourdes, il faut chercher alors à savoir si l'on a intérêt à accélérer les travaux malgré l'augmentation de coût qui peut en découler.

II.2 Discussion

Contrôler le projet est d'une importance cruciale pour les organisations et est une préoccupation de premier ordre pour le gestionnaire de projet confronté à une économie mondiale où faibles coûts et qualité sont les mots- clés de la performance [CGP 1998]. Étant donné que la qualité est difficilement mesurable, l'essentiel des recherches en contrôle de projet portent sur l'évolution des délais et des coûts.

Sans pour autant chercher à faire l'inventaire de tous les projets ayant échoué, on peut retenir qu'en gestion de projet le dépassement des délais ou des coûts est fréquent et le respect des contraintes budgétaires et temporelles, plutôt exceptionnel. En effet, les projets doublent en général leur temps prévu de réalisation et dépassent leur coût de 40 % [MH 87].

Il est toutefois intéressant de jeter un regard critique sur la méthode PERT à trois estimations pour relever les insuffisances qui compromettent la validité de prédiction de performance de l'outil dans un contexte d'interdépendance des chemins du graphe PERT et son efficacité dans la prise de décision en contexte d'incertitude sur le projet.

Dans le cas particulier des projets, la performance va se mesurer via le triangle vertueux «*coût – délai – qualité*» que les professionnels appellent encore «*Sainte Trinité*» [KD 00]. La performance d'un projet est donc mesurée en termes de temps de réalisation du projet, du coût du projet et de sa qualité.

II.2.1 Critiques de l'approche PERT probabiliste

PERT repose sur l'estimation de trois durées pour déterminer les durées espérées de chaque activité et la durée totale espérée du projet ainsi que la variance de délai : les durées de réalisation optimiste (a), la plus probable (m) et pessimiste (b). On comprend aisément que si des erreurs entachent les données de base du PERT, elles remettent en cause la probabilité plus juste de compléter le projet à temps, compte tenu du biais systématique qu'elles induisent dans le calcul de la durée totale espérée du projet et de sa variance [IKA 04].

Nous dressons dans ce qui suit un ensemble de points qui nous semblent fondamentales et qui remettent en cause l'approche probabiliste du PERT.

- *Erreurs potentielles liées à l'inexactitude des estimations (a), (b), et (m)*

L'estimation des données de base (a), (m) et (b) du PERT est en général un défi de taille pour le gestionnaire de projet. Il est fréquent de constater que les trios (a), (m) et (b) sont présentés systématiquement sous une forme régulière et symétrique, du type 20, 30, 40. C'est pour résoudre ce problème que certains auteurs ont proposé de remplacer les valeurs certaines (a) et (b) en pour cent d'ordres 5% et 95% respectivement ou encore 10% et 90%. D'où une source d'erreur [PES 07].

En considérant la possibilité d'avoir des estimations incorrectes, il semble peu sûr d'obtenir les paramètres exacts de la présumée distribution bêta de la durée des activités.

Évidemment, l'estimation dépend de l'expérience de la personne en charge. Les estimations peuvent être invraisemblablement optimistes ou bien pessimistes compte tenu de l'élément humain ou de la subjectivité qui les entoure. Et la corrélation entre l'expérience acquise dans la gestion de projets similaires d'une part et la qualité des estimations de durée des activités d'un projet novateur d'autre part n'est peut-être pas forte. À tout ceci, il faut ajouter le risque de gonflage des durées par le gestionnaire qui veut se donner une marge de sécurité assez large. .

Prendre en considération les erreurs potentielles qui entachent la détermination des durées espérées et des variances de la durée des activités prises individuellement ne suffit pas. Il faut encore tenir compte des erreurs liées à la configuration particulière des réseaux dans PERT.

- *Le déterminisme dans la détermination du chemin critique*

La détermination du chemin critique dans PERT porte sur les durées espérées des activités de sorte que l'élément stochastique, soit la variance des durées des activités, n'est pris en compte qu'en fin d'analyse dans le calcul de la probabilité de compléter le projet à temps [ROB 97]. En conséquence, l'analyse du chemin critique est réduite à une forme déterministe comme dans CPM : un déterminisme qui handicape le probabilisme de PERT [RUL 00].

Il en résulte que la durée totale du projet dans PERT est en général plus petite et jamais plus grande que la durée réelle du projet au temps t . En effet, «les retards provoqués par l'imprévu et les vents adverses sont souvent plus longs que les avances dues à la bonne chance ou à un Éole collaborateur» [NOP 99]. La durée totale espérée du projet dans PERT subit en général un biais vers la gauche de la distribution soit vers le temps optimiste. C'est ce qui fait dire à Soroush [SOR 94] que l'approche classique dans le PERT conventionnel ignore le fait que le probabilisme des durées des activités donne de fortes chances à une pluralité de chemins de devenir potentiellement critiques et de ce fait, conduit à une estimation plutôt excessivement optimiste de la probabilité de compléter le projet à temps. C'est là une autre source d'erreur.

Archibald et al. [AV 67] suggèrent notamment de remplacer le concept de chemin critique par le concept plus valide d'activité critique dans la mesure où dans un modèle stochastique, chaque chemin à un moment t donné, a une probabilité spécifique de devenir le chemin le plus long. Le chemin critique déterminé dans PERT n'étant pas nécessairement le plus long, donc pas nécessairement critique. En d'autres termes, le chemin critique déterminé dans PERT ne comprend pas nécessairement les activités critiques, celles-là mêmes qui sont les seules censées retenir particulièrement l'attention du gestionnaire de projet s'il veut éviter tout retard dans l'exécution du projet. C'est là une autre source d'erreur.

- *L'interdépendance des chemins ou la corrélation des chemins*

Deux ou plusieurs chemins peuvent partager des activités en commun. Ils sont donc interdépendants ou corrélés. Les chemins du réseau ne sont donc pas toujours indépendants du point de vue de leur durée, comme le présuppose PERT dans la détermination du chemin critique.

Par conséquent, si un chemin a une très grande durée, les autres chemins qui ont avec lui des activités en commun pourraient peut-être avoir une longue durée aussi [JOR 04c].

- *Erreurs potentielles liées à l'hypothèse de la distribution bêta*

L'une des hypothèses fondamentales du PERT est la distribution bêta de la durée des activités (*Figure IV.4*).

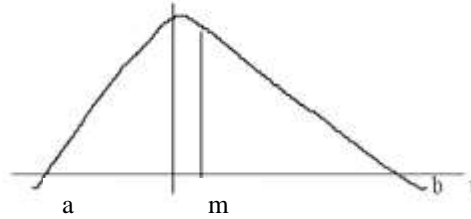


Figure IV.4. Courbe de la distribution Bêta tiré de [IKA 04].

Archibald et al. [AV 67] rappellent qu'étant donné que la forme réelle de la distribution des durées des activités reste probablement inconnue et compte tenu des propriétés présumées de la distribution d'une activité supposer a priori que *la distribution de la durée d'une activité suit une loi bêta est potentiellement source d'erreur*.

La moyenne et la variance de la distribution bêta ne suffisent pas à elles seules pour caractériser une seule et même distribution bêta de la durée des activités, l'exigence d'asymétrie excluant le modèle normal où la moyenne et la variance déterminent une seule distribution (voir *Figure IV. 5*).

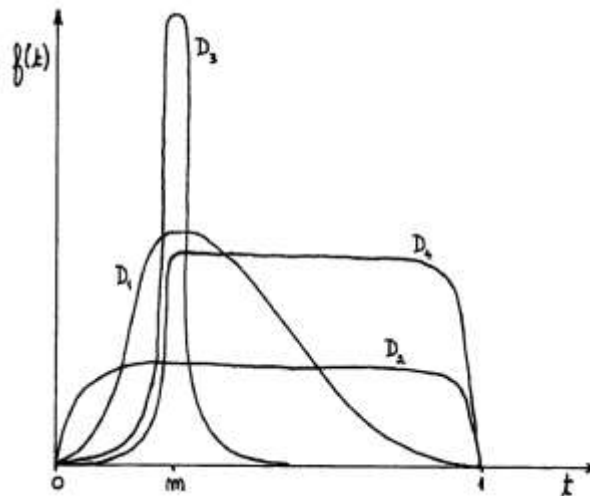


Figure IV.5. Exemple de distributions possibles.

Pourtant pour des raisons pratiques, il s'avérait inévitable de caractériser la durée d'une activité par sa moyenne et sa variance uniquement. D'où une *autre source d'erreur* [JOR 04a].

Troisième source d'erreur : des distributions comme la distribution triangulaire qui satisfont autant sinon parfois mieux aux propriétés susvisées ont des moyennes et des variances tout aussi différentes de celles de la loi bêta.

Toutefois, l'hypothèse de la distribution bêta ne porte pas trop à conséquence étant entendu qu'elle répond à l'exigence d'asymétrie de la fonction de densité et justifie l'expression de la durée espérée en fonction du mode (m) et des valeurs extrêmes (a) et (b). De toute façon, les erreurs liées à la distribution bêta sont jugées négligeables par rapport aux erreurs qui suivent.

- *Erreurs potentielles liées à σ et à t_e*

En prenant la durée minimale de l'activité si tout se passait exceptionnellement bien (a), la durée de l'activité si l'on rencontrait des difficultés en cours de réalisation (b) et la durée la plus vraisemblable, la plus attendue par le gestionnaire de projet (m), PERT détermine la durée espérée (t_e) de chaque activité et sa variance. Il est en effet plus intéressant de se fier à la valeur du temps espéré des activités que de s'en tenir aux valeurs extrêmes de la distribution des durées des activités, la moyenne étant généralisatrice et compensatrice.

En considérant a priori que la distribution de la durée d'une activité suit une loi bêta et sachant que l'expression de la fonction de densité de cette loi est bien connue, on s'expose à deux sources d'erreurs :

Première source d'erreur, l'écart - type de la durée d'une activité a été supposé égal à $(b - a)/6$ comme expliqué précédemment. Ce n'est certainement pas la vraie valeur de l'écart - type d'une distribution bêta!

Deuxième source d'erreur, l'approximation linéaire $t_e = (a + b + 4m)/6$ faite pour la vraie valeur du temps espéré qui est plutôt une racine d'une équation cubique. D'où l'incompatibilité des formules du temps espéré et de l'écart- type dans la pratique [PES 07].

- *Erreurs liées à l'hypothèse d'indépendance des durées des activités*

L'hypothèse d'indépendance des durées des tâches critiques est peu vraisemblable. Les rigueurs du climat par exemple, peuvent affecter de la même manière plusieurs tâches et amener les responsables à accélérer le projet. L'effet net des facteurs inflationnistes et déflationnistes sur la durée totale du projet serait négligé si des durées de tâches étaient supposées indépendantes [JOR 04b].

II.2.2 Quelques suggestions

Pour remédier aux insuffisances qui entachent la méthode PERT stochastique, [MBBO 10] proposent un ensemble de suggestions qui sont les suivantes :

- *Une technique du PERT avec deux estimations*

Étant donné les sérieuses difficultés éprouvées par les gestionnaires de projet pour obtenir l'estimation du temps le plus probable et attendu que dans la pratique, «le temps le plus probable est généralement le temps le plus improbable», ne serait-il pas indiqué d'envisager une technique du PERT avec deux estimations : la durée optimiste (a) et la durée pessimiste (b) ? En considérant une distribution bêta de la durée des activités, Golenko- Ginzburg [GOL 88] observe que l'estimation de la durée la plus probable (m) d'un ensemble de réseaux de projets est «pratiquement inutile» et que leur valeur (m) est souvent proche de $(2a + b)/3$, et qu'en plus, il n'y a aucune différence statistique significative à utiliser les valeurs de (m) plutôt que les valeurs calculées. Il en déduit les formules suivantes des durées espérées et des variances :

$$t_e = 0.2(3a + 2b) \quad \text{et} \quad \sigma^2 = 0.04(b - a)^2$$

Le PERT avec deux estimations a l'avantage de simplifier l'analyse sans pour autant affecter les paramètres du projet et offre l'opportunité d'une approche combinée PERT/CPM de gestion des délais/coûts du projet, quel que soit son environnement.

- *L'approche combinée PERT/CPM et l'analyse du compromis entre allocation de ressources et risque de retard du projet*

PERT et CPM, comme on l'a vu, sont deux techniques concourantes et synergiques. Pourquoi ne pas tenter de les combiner pour en exploiter les forces et corriger les faiblesses? MacLeod et Petersen [MP 96] se sont penchés sur la question.

En remarquant que PERT et CPM reposent au total sur une seule estimation de temps soit le temps espéré (t_e) dans PERT et le temps normal (n) dans CPM et que PERT et CPM ont aussi une limite inférieure de durée pour les activités du projet soit le temps minimal de réalisation de l'activité (c) dans CPM et le temps optimiste (a) dans PERT, on peut valablement accepter avec MacLeod et Petersen [MP 96], l'hypothèse que $t_e = n$ et $a = c$. En un mot, les temps espérés calculés dans PERT sont pour PERT ce que les temps normaux sont pour CPM et la durée minimale d'une activité est pour CPM ce que la durée optimiste est pour PERT.

Avec une telle hypothèse, il est désormais possible de calculer les effets de l'allocation de ressources supplémentaires sur la probabilité de compléter le projet à temps. L'approche combinée PERT/CPM permet alors de bénéficier de l'analyse du compromis coût/durée que seul CPM permet et d'évaluer les chances de compléter le projet à temps (ce qui n'est possible qu'avec PERT), le tout en utilisant un système basé sur le compromis entre allocation de ressources et risque de retard du projet.

- *La prise en compte de l'hypothèse de dépendance des durées des chemins dans la détermination de la durée totale du projet et de sa variance*

L'hypothèse d'indépendance des durées des chemins est sujette à des erreurs dans l'appréciation du risque de retard du projet. C'est ce qui justifie l'opportunité de la prise en compte de l'hypothèse inverse de dépendance des durées des chemins. Évidemment en considérant l'hypothèse plus plausible de corrélation des durées des chemins du réseau, on améliore les chances d'obtenir une probabilité plus juste de compléter le projet à temps. Seulement prendre en compte l'interdépendance ou la corrélation des durées des chemins, c'est accepter de faire face à un calcul fastidieux et compliqué des chances de compléter le projet en dedans du temps t donné compte tenu de la nature multi-activités et multi-chemins du réseau.

Plusieurs chercheurs ont étudié le problème et ont suggéré de recourir si l'envergure du projet l'autorise, à la simulation de Monte Carlo, ou à une approximation par des heuristiques ou encore de borner la probabilité de compléter le projet à une date t donnée dans un intervalle. Cette dernière méthode (l'algorithme du Probabilistic Network Evaluation Technique) est stochastique et les bornes inférieure et supérieure sont respectivement les probabilités de compléter le projet dans les hypothèses d'indépendance et de dépendance statistique des chemins [AAC 75].

- *La construction d'un intervalle de confiance autour du PMB (Performance Measurement Baseline) dans les conditions d'une distribution bêta de la durée des activités, de l'interdépendance des durées des chemins, du PERT à deux estimés et du théorème central limite.*

Le *PMB* est le coût planifié (P), abstraction faite des provisions et des frais généraux. En tant que tel, le *PMB* est directement lié à la valeur acquise ou gagnée du projet soit V . D'un point de vue pratique, la valeur n'est gagnée que si les lots de travail ont été livrés [PES 2007].

II.3 Résolution par simulation

[FSKM 03] présente une approche qui consiste à simuler un grand nombre de réalisations potentielles du projet étudié et à en tirer un certain nombre d'enseignements.

Pour simplifier, supposons que la durée estimée de la tâche A soit 8, 9 ou 10 jours avec des probabilités respectives de 20%, 50% et 30%. Si l'on fait 100 réalisations potentielles, dans 20 des réalisations, la durée de A devra être de 8 jours, dans 50 de 9 jours et dans 30 de 10 jours. Les fonctions aléatoires habituelles (random, alea) permettent de générer un nombre aléatoire entre 0 et 1. Pour fixer la durée de A, on établira la correspondance suivante :

- si le nombre aléatoire est inférieur à 0,2, la durée sera 8 jours ;

- si le nombre aléatoire est entre 0,2 et 0,7, la durée sera 9 jours ;
- si le nombre aléatoire est supérieur à 0,7, la durée sera 10 jours.

Pour simuler une réalisation potentielle, on va ainsi tirer la durée de chacune des tâches, en respectant sa loi de probabilité et calculer les dates induites par ces durées. Mais il est impossible d'affirmer quoi que ce soit au vu d'une seule simulation. Pour vérifier qu'un dé n'est pas pipé, un seul lancer ne suffit pas. On doit effectuer un grand nombre de lancer et vérifier que les faces 1, 2, 3, 4, 5 et 6 sortent sensiblement avec la même fréquence. On doit de même effectuer un grand nombre de simulation au moins 100). A partir de ces valeurs, on peut calculer la moyenne et l'écart type des dates au plus tôt de chaque tâche et de la durée du projet. Comptabiliser le nombre de fois où une tâche est critique permet de déterminer la probabilité de criticité d'une tâche. Selon la manière dont les durées vont interagir, les chemins critiques peuvent être différents.

Les renseignements obtenus par cette approche sont plus riches que ceux apportés par l'approche analytique. Les calculs d'un PERT sont très rapides (ils sont proportionnels au nombre de liaisons entre tâches). Il nous semble qu'ici la simulation est à préférer à la méthode analytique [FIN 02].

III Le PERT coût

La plupart des chefs de projet sont confrontés à un travail double. Chacun d'eux est en grande partie responsable de commander financièrement les ressources de son entreprise, et il doit fournir un produit ou un service au coût minimum.

Le système d'analyse du coût de l'ordonnancement avec un réseau de PERT-temps doit satisfaire ces deux besoins [LIN 03].

Dans un projet, les coûts sont associés à chaque tâche. Ceux-ci représentent les dépenses engagées pour le travail effectué et pour les matières premières utilisées. Il sera souvent possible de raccourcir les durées d'exécution des tâches en prévoyant une plus grande quantité de main d'œuvre ou en utilisant des matières premières plus onéreuses [PRI 74].

En règle générale, dans le cas de l'ordonnancement relatif au coût deux problèmes surgissent :

- comment diminuer le coût global du projet ?,
- et comment accélérer le projet au moindre coût ?

En fait, les réalisateurs du PERT-coût (département de la défense et de la NASA) se sont penchés sur l'étude de ce problème dès les années soixante du vingtième siècle.

Le principe de la réponse aux questions précédentes est d'étudier la relation qui existe entre la durée d'une tâche et son coût. On associe alors, à chaque tâche un coût γ , fonction de sa durée t .

La courbe des coûts d'une tâche en fonction de sa durée a en général l'allure de la figure IV.6:

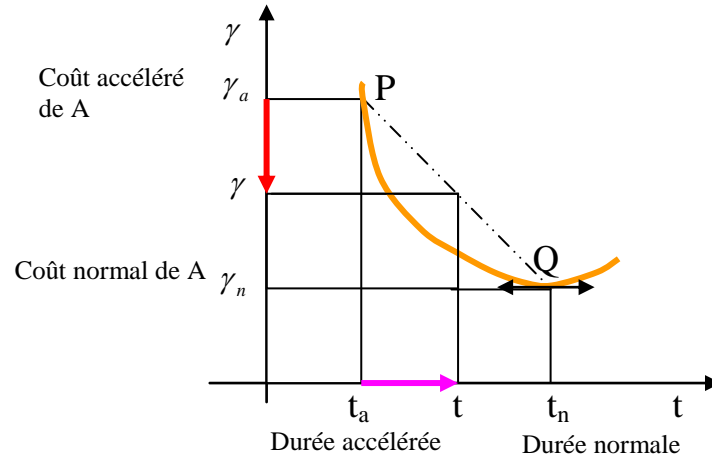


Figure IV.6. Courbe des coûts γ d'une tâche A en fonction de sa durée.

Le coût diminue avec l'augmentation de la durée jusqu'à un certain seuil, après lequel le coût a tendance à augmenter. Dans les calculs qui vont suivre, cette courbe est approximée par le segment de droite PQ qui a comme équation :

$$\gamma = \frac{\gamma_n - \gamma_a}{t_n - t_a} t + \frac{\gamma_a t_n - \gamma_n t_a}{t_n - t_a}, t \in [t_a, t_n] \text{ et } \gamma \in [\gamma_n, \gamma_a]$$

t_a est la durée accélérée de la tâche et γ_a son coût accéléré, t_n est la durée normale de la tâche et γ_n son coût normal.

L'approximation de la courbe par le segment de droite PQ fournit des résultats approchés, or elle rend de grands services pour la résolution des problèmes cités avant.

Le gain unitaire pour une tâche i est alors, la pente de la droite :

$$G.U_i = \frac{\text{coût accéléré} - \text{coût normal}}{\text{temps accéléré} - \text{temps normal}}$$

Dans le problème d'ordonnancement, on peut ainsi modifier les durées des tâches selon le contexte économique : Si la tâche est rallongée de $t - t_a$ unités de temps, le coût d'exécution du projet diminuera de $\gamma_a - \gamma$.

Revenons maintenant aux deux questions précédentes. Nous allons d'abord dans la première phase diminuer le coût total du projet en prenant le programme accéléré, puisque les durées des tâches sont moins que celles du programme normal. Ensuite nous allons prendre uniquement dans la deuxième phase les tâches critiques du programme normal et on diminue autant que possible la durée de chaque tâche et on calcule le coût correspondant.

Il s'agit donc, de choisir le projet le moins cher parmi les plus rapides ; soit de manière plus générale, d'arbitrer entre la durée globale et le coût global du projet, sans gaspillage, en sélectionnant le projet le moins cher pour une durée donnée (ou le plus court pour un coût donné).

III.1 Diminution du coût total du projet

Pour diminuer le coût global d'un projet, on doit avoir le programme normal et un programme accéléré où les durées des tâches sont moins que les durées du programme normal alors que le coût de chaque tâche est plus élevé.

On prend le programme accéléré uniquement, on peut diminuer son coût en allongeant la durée des opérations non critiques : l'algorithme consiste à annuler les marges du projet, c. à. d.:

1. Annulation des marges libres.
2. Obtenir un nouveau programme.
3. Annulation des marges totales.

III.2 Diminution de la durée au moindre coût

La deuxième technique, qui aboutit au même résultat que la première consiste à prendre l'ordonnancement normal et savoir pour chaque durée accélérée possible du projet, le coût minimum correspondant. La diminution consiste uniquement les durées des tâches critiques. Cette technique comporte les étapes suivantes :

1. Déterminer l'ordonnancement normal du projet.
2. Déterminer l'opération critique dont le coût unitaire de raccourcissement est minimum.
3. Diminuer le plus possible la durée de cette tâche tant qu'aucune autre tâche ne devient critique.

Déterminer à nouveau l'ordonnancement et aller à 2 jusqu'à ce qu'il n'y aura plus de tâche à diminuer sa durée.

IV. Ordonnement de projet à moyens limités

Le problème d'ordonnement de projet à moyens limités ou RCPSP pour Resource Constrained Project Scheduling consiste en l'ordonnement d'un ensemble donné de tâches sur une ou plusieurs ressources dont la capacité est fixée et connue. Les tâches sont liées entre elles par des contraintes de précédence, et chaque tâche nécessite une quantité donnée de chacune des ressources tout au long de son exécution. L'objectif de la fonction économique est de minimiser la durée totale du projet.

Une instance de RCPSP peut se formuler comme suit :

Un projet est la donnée d'un ensemble $A = \{1, \dots, n\}$ de tâches et d'un ensemble $R = \{1, \dots, m\}$ de ressources. Une quantité constante R_k d'unités de ressource $k \in R$ est disponible à chaque instant. Toute tâche $i \in A$ doit être exécutée une et une seule fois et sans interruption, en p_i unités de temps. Durant ce temps, i monopolise alors une quantité positive ou nulle r_{ik} d'unités de chaque ressource k . Enfin, une relation de précédence entre les tâches est modélisée par un graphe orienté acyclique $(A; E)$.

Le problème d'ordonnement consiste alors à caractériser le n -uplet des dates de début d'exécution des tâches $S = (S_1, S_2, \dots, S_n)$ de sorte que :

- à tout moment et pour toute ressource k , la quantité d'unités de ressource k utilisée par toutes les tâches en cours d'exécution est inférieure ou égale à la capacité de k ;
- pour tout arc $(i; j) \in E$, l'exécution de la tâche j ne peut commencer qu'après la complétion de i , à savoir $S_j \geq S_i + p_i$;
- la date d'achèvement $C_{\max} = \max \{S_i + p_i \mid i \in A\}$ du projet est minimale [DEM 01].

Le RCPSP est un problème d'optimisation NP-difficile en raison de sa nature fortement combinatoire [GJ 99]. De ce fait, les méthodes arborescentes telles que la méthode de séparation et évaluation, sont des outils privilégiés pour sa résolution de manière exacte.

[NSZZ 03] propose un état de l'art sur l'ordonnement de projet avec créneaux de ressources (time windows) et ressources critiques (scare resources). De nombreuses méthodes exactes et heuristiques y sont proposées pour différents critères d'optimisation qui ont été identifiés important pour les applications.

Un grand nombre de méthodes ont été développées pour tenter de trouver le meilleur ordonnancement possible. Ces méthodes peuvent être regroupées en deux classes :

- les méthodes exactes qui trouvent une solution optimale et prouvent leur optimalité (approches analytiques, combinatoires ou optimales)
- les méthodes approchées qui fournissent une bonne solution sans garantie d'optimalité (heuristiques et méta-heuristiques) [PAR 05].

V Ordonnement avec contraintes disjonctives

Les problèmes d'ordonnement en présence des ressources sont généralement NP-complets, y compris lorsque toutes les ressources existent en quantité unitaire (problèmes disjonctifs traités dans cette section). Il n'existe donc pas d'algorithmes pour trouver la solution optimale, quel que soit le problème, avec un temps de calcul raisonnable.

Une contrainte disjonctive entre A et B interdit l'exécution simultanée de A et B. On trouve de telles contraintes lorsque les ressources considérées sont en quantité unitaires.

[FIN 02] présente le problème en le modélisant par un programme linéaire avec des variables 0, 1. Après avoir organisé les tâches en niveaux il dresse deux types de variables de décision :

- F_i la date de fin au plus tôt de la tâche i .
- x_{ij} variables bivalentes définissant l'ordre de passage des tâches i et j :
 - $x_{ij} = 1$ si i précède j ,
 - $x_{ij} = 0$ sinon.

L'objectif est de minimiser la date de fin de la tâche finale $\{\text{Min } F_n\}$.

Nous devons introduire des contraintes entre les dates d'achèvement de deux tâches i et j qui se suivent (contrainte de type 1). Pour chaque relation de précédence :

Si i précède j on a $F_j \geq F_i + t_j$.

Pour chaque disjonction entre i et j ($i < j$) il faut de plus traduire les conditions suivantes (contraintes de type 2) :

- si i précède j ($x_{ij} = 1$) alors $F_j \geq F_i + t_j$;
- si j précède i ($x_{ij} = 0$) alors $F_i \geq F_j + t_i$.

La résolution de ce genre de problèmes se fait par la méthode de séparation et évaluation et les temps de calculs deviennent prohibitifs lorsque le nombre de variables entières en $\{0, 1\}$ croît.

VI Ordonnement multi-projet

Un environnement multi-projet comprend plusieurs projets, chacun étant défini par un ensemble d'activités déterminées et consommant des ressources pour leur réalisation respective. Le problème d'ordonnement consiste à affecter les ressources entre les projets concomitants selon l'avancement et la priorité accordée à chacun d'eux sans excéder le niveau de disponibilité des ressources. L'avancement d'un projet particulier n'est donc pas indépendant de l'avancement des autres projets. Le nombre et le contenu des projets sont rarement connus avec certitude au-delà d'un horizon de planification à court terme ce qui rend l'environnement dynamique [VGP

00]. La mise en œuvre de l'ordonnancement dans un environnement multi-projet se situe au niveau des arbitrages entre les projets et concernent notamment :

- l'affectation des ressources sur différents projets (coordination),
- le phasage des projets et les priorités de réalisation à donner entre les projets,
- les synergies, synchronisations et interactions entre les projets,

Il est bien clair que ce type de problèmes, qui est le plus répandu dans l'industrie, est peu exposé dans la littérature [DUM 92]. Les recherches en planification de projet se sont surtout concentrées sur l'ordonnancement de projet unique et statique multi-ressource ainsi que les modèles d'ordonnancement d'ateliers. En raison de sa plus grande complexité et de la difficulté toujours présente de résoudre les problèmes plus simples de façon optimale, il est nécessaire de recourir en pratique à une approche heuristique pour la résolution du problème d'ordonnancement dynamique multi-projet [DUM 92] [MPP 94].

L'environnement aléatoire des projets complique le processus de planification et comme il est nécessaire d'imiter le caractère stochastique d'un tel environnement, il convient bien d'utiliser la simulation qui offre suffisamment de flexibilité pour aborder une telle problématique. Le lecteur pourra trouver plus de détails dans [BBK 98] [DUM 92] [OUE 88] [MPP 94].

VII. Conclusion

L'approche formelle la plus commune pour incorporer l'incertitude dans le processus d'établissement du planning est d'appliquer le processus d'établissement de la méthode du chemin critique puis analyser les résultats d'une perspective probabiliste. Ce processus désigné habituellement sous le nom PERT avec trois évaluations. Cette méthode était présentée en détail et nous avons ensuite abordé les critiques qui touchent cette fameuse méthode. Nous avons terminé cette étude par un ensemble de suggestions que nous avons vu nécessaires pour la correction des résultats fournis par la technique PERT. Une étude d'autres techniques basées sur les principes de la simulation a été présentée dans ce chapitre.

Nous avons présenté rapidement, parmi les extensions du problème central de l'ordonnancement, la méthode du PERT-coût avec ses deux techniques de diminution du coût global du projet. La prise en compte des ressources avec des moyens limités appelé RCPSA a été abordée. Nous avons également étudié le cas des contraintes disjonctives et l'ordonnancement multi-projet qui sont des problèmes NP-complet.

Rappelons que la gestion de projet ne se limite pas seulement aux seuls aspects techniques, mais elle est aussi un problème de management.

Chapitre V

Construction d'un graphe PERT à partir du graphe des potentiels

Sommaire

I Introduction	86
II Liens entre le graphe des potentiels et le graphe PERT	87
III Le graphe adjoint de graphe	87
II.1 Définition	88
II.2 Exemple	88
II.3 Le problème inverse	89
II.3.1 Définition de la configuration « Z » et « Δ ».	90
II.4 Quelques caractérisations des graphes adjoints	90
IV Construction du graphe PERT à partir du graphe des potentiels	93
III.1 Cas où le graphe des potentiels est un graphe adjoint	93
III.2 Cas où le graphe des potentiels n'est pas un graphe adjoint	95
III.3 Algorithme 1	96
III.3.1 Application	97
III.3.2 Discussion	99
V Liens entre le diagramme de Gantt et le graphe PERT	99
VI Liens entre le graphe des potentiels et le diagramme de Gantt	100
VII Liens entre un graphe PERT et le graphe des potentiels	101
VIII Conclusion	101

I. Introduction

Les trois représentations (le diagramme de Gantt, le graphe des potentiels et le graphe PERT) sont utilisées pour résoudre le problème d'ordonnement.

Chacune d'elles comporte des avantages et des inconvénients qui font qu'elles soient en général utilisées toutes les trois en même temps. En outre, selon la méthode de construction utilisée, le problème d'ordonnement peut être représenté par plusieurs graphes PERT.

Le passage d'une représentation à l'autre est très utile pour en profiter des avantages de chacune d'elles et éviter ses inconvénients. Dans ce chapitre on s'intéresse aux différents passages entre ces représentations.

Dans la section suivante, nous introduisons quelques concepts préliminaires sur les graphes adjoints qui sont nécessaires pour l'exposé de notre approche.

II Liens entre le graphe des potentiels et le graphe PERT

Les praticiens de la gestion de projet préfèrent utiliser la méthode PERT/CPM lors de la phase planification malgré que le graphe de la méthode des potentiels est unique et facile à dessiner alors que les graphes PERT/CPM sont infinis, difficiles à dessiner et la majorité des praticiens n'arrivent pas à les dessiner correctement. Cependant, la présence des tâches fictives leur rend encore le problème plus difficile à être abordé surtout lorsque le nombre de ces tâches fictives est très élevé avec des réseaux de tailles très grandes.

Malgré ces contraintes, les spécialistes insistent sur l'utilisation du graphe PERT et ils présentent un certain nombre d'arguments qui justifient leur choix. Selon Fink et al. [FIN 02], le graphe PERT/CPM est plus concis. En outre, Hendrickson et al. [HEN 08] explique qu'il est à proximité du célèbre diagramme de Gantt qui est encore utilisé à nos jours, et selon Yuval et Sadeh [CS 07], la structure du graphe PERT est beaucoup plus adaptée à certaines techniques d'analyse et d'optimisation des formulations.

Donc, notre travail, dans ce chapitre sera axé sur la recherche d'un moyen de passer d'un graphe facile, qui est le graphe de la méthode MPM, vers le graphe PERT/CPM qui sera correct et qui respecte les contraintes de départ, à savoir les contraintes de la table d'ordonnancement, mais avant de donner plus de détails sur un éventuel algorithme qui résout le problème, étudions d'abord les graphes adjoints de graphes.

III Le graphe adjoint de graphe

Dans l'étude des graphes, le concept de distance d'un point à un autre joue un rôle très important [BOR 07]. De plus, dans certains problèmes il ne faut pas seulement savoir comment les sommets sont reliés mais aussi comment les directions sont faites. Cette observation donne une association entre un graphe adjoint $L(G)$ et un graphe arbitraire G donné. Dans cette section on se réfère à [BEN 01], [MBBO 06] et [MBBE 11].

III.1 Définition

Soit $G = (X, U)$ un graphe orienté simple ou multiple. On construit à partir de G un graphe noté $L(G)$, appelé graphe adjoint ou line graphe (graphe représentatif des arcs) de G comme suit :

- Les sommets de $L(G)$ sont en correspondance biunivoque avec les arcs de G . Pour des raisons de simplicité, on donne le même nom aux arcs de G et aux sommets correspondants de $L(G)$.
- 2 sommets u et v de $L(G)$ sont reliés par un arc de u vers v si et seulement si les arcs u et v de G sont tels que l'extrémité terminale de u coïncide avec l'extrémité initiale de v i.e. $T(u) = I(v)$.

III.2 Exemple

Soit le graphe G suivant:

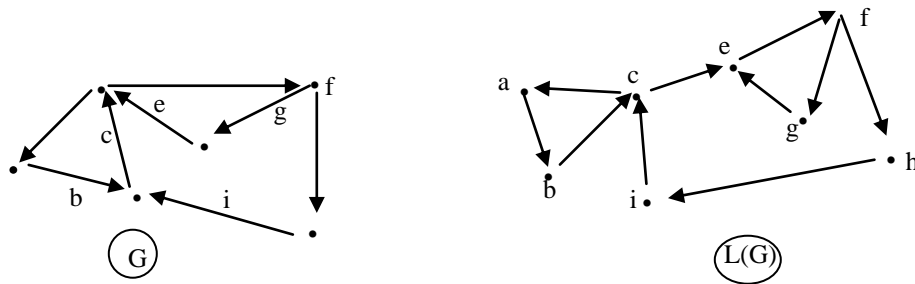


Figure V.1. Un graphe G et son graphe adjoint $L(G)$.

Selon R. L. HEMMINGER et al. [HB 83] Si $G=(X, U)$ un graphe direct de n sommets: $x = \{x_1, x_2, \dots, x_n\}$ et m arcs.

Le graphe adjoint $L(G)$ admet n' sommets et m' arcs et nous avons immédiatement :

$$n' = m ; \quad m' = \sum_{i=1}^n d^-(x_i) \cdot d^+(x_i). \quad \text{De plus si } x' \text{ est un sommet de } L(G) (x' = (x_i, x_j)) \text{ alors :}$$

$$d^-(x') = d^-(x_i), \text{ et } \quad d^+(x') = d^+(x_j).$$

Parmi les applications des graphes adjoints on cite la coloration des arêtes. Elle consiste à affecter à toutes les arêtes du graphe une couleur de telle sorte que deux arêtes adjacentes ne portent pas la même couleur. L'indice chromatique du graphe G est le plus petit entier k pour lequel il existe une coloration des arêtes; on le note $\chi(G)$. Sur le graphe ci-dessous, on a eu besoin de trois couleurs pour colorer les arêtes de telle sorte que deux arêtes adjacentes ont des couleurs différentes. Pour colorer les arêtes d'un graphe, on peut se ramener au problème de la

coloration des sommets ensuite les arêtes. Il suffit pour cela de travailler non pas sur le graphe lui-même, mais sur le graphe adjoint.

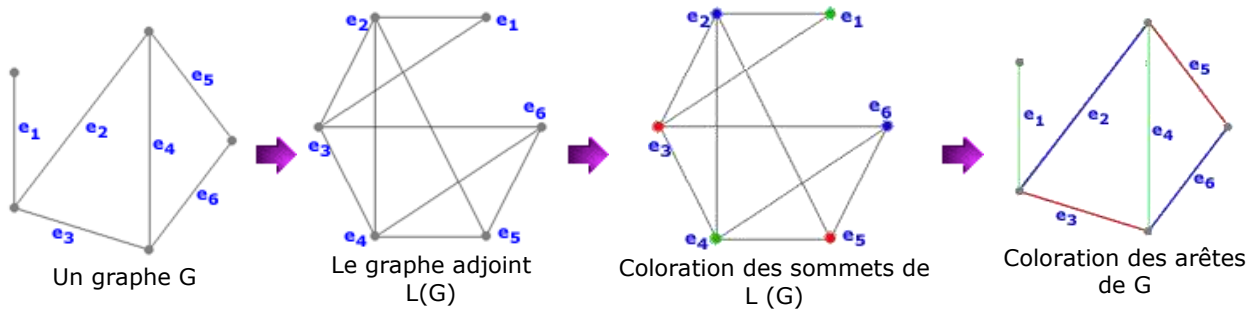


Figure V.2. Exemple d'utilisation des graphes adjoints dans la coloration d'arêtes.

Les graphes adjoints ont été très étudiés mais nous ne donnons dans ce chapitre que les résultats qui nous intéressent.

Par la définition, tout graphe G admet un graphe adjoint $L(G)$ unique. Par contre, deux graphes non isomorphes peuvent avoir le même graphe adjoint.

Les graphes G_1 , G_2 et G_3 suivant ont le même graphe adjoint H :
 $H=L(G_1) = L(G_2) = L(G_3)$

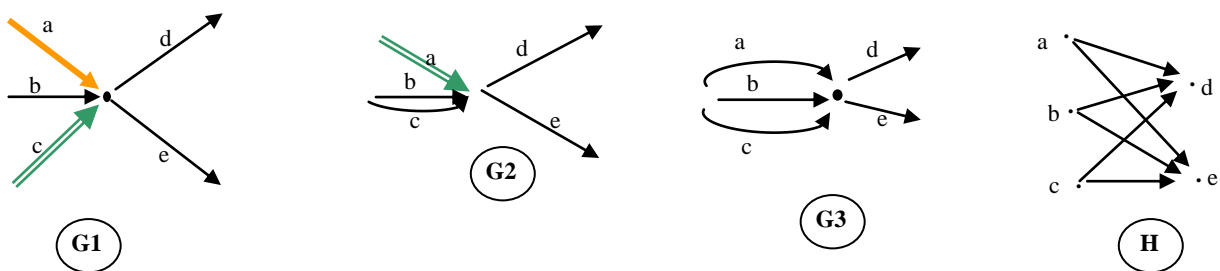


Figure V.3. Les graphes G_1 , G_2 et G_3 du graphe adjoint H .

III.3 Le problème inverse

On pose le problème inverse suivant:

Etant donné un graphe H , est-il le graphe adjoint d'un graphe? Autrement dit, existe-t-il un graphe G tel que $L(G)$ soit isomorphe à H , où $H = L(G)$?

Avant de répondre à cette question, donnons la définition d'une configuration « Z » et « Δ ».

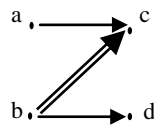
III.3.1 Définition de la configuration « Z » et « Δ »

G admet une configuration “Z” si G contient 4 sommets a,b,c et d tels que si (a,c), (b,c) et (b,d) sont des arcs de G, alors (a,d) n’est pas un arc de G.

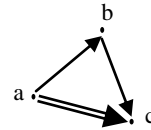
Dans le seul but de simplicité, on donnera le nom de *barre du “Z”* l’arc (b,c).

La configuration “Z” apparaît lorsque 2 sommets ont des successeurs communs et des successeurs non communs ou par symétrie lorsque 2 sommets ont des prédécesseurs communs et des prédécesseurs non communs.

G admet une configuration « Δ » si G contient des arcs (a, b), (b, c) et (a, c).



(a) La configuration “Z”



(b) La configuration « Δ »

Figure V.4. Les configurations « Z » et « Δ ».

III.4 Quelques caractérisations des graphes adjoints

Les graphes adjoints ont été largement étudiés, mais nous ne donnons dans cette section que les résultats qui nous intéressent et qui sont tirés de [BEN 01] et [MOU 02].

1. H est le graphe adjoint d’un graphe si et seulement si H ne contient aucune configuration ‘Z’.
2. H est le graphe adjoint d’un graphe G si et seulement si les arcs de H peuvent être partitionnés en bipartis complets $B_i = (X_i, Y_i)$, $i=1, \dots, m$, tels que $X_i \cap X_j = \emptyset$ et $Y_i \cap Y_j = \emptyset$, $\forall i \neq j$.

Les bipartis B_i de H sont alors en bijection avec les sommets notés aussi B_i qui ne sont ni sources ni puits, deux sommets B_i et B_j de G étant reliés par un arc de B_i vers B_j si et seulement si les bipartis complets B_i et B_j de H sont tels que $Y_i \cap X_j = \emptyset$.

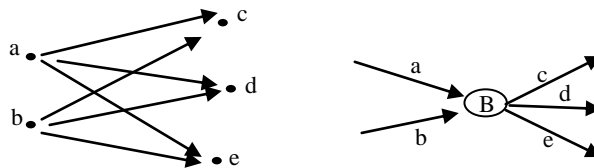


Figure V.5. Un biparti complet B de H l’étoile de G associée à B

Exemple :

Soit le graphe H suivant
(L'exemple est tiré de [MBBO 06]) :

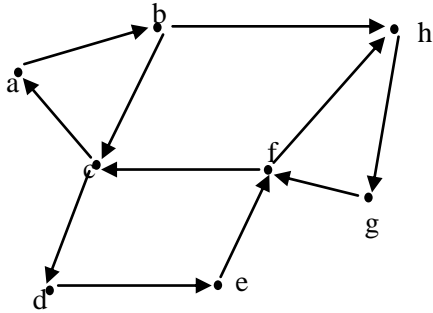


Figure V.6. (a) un graphe H.

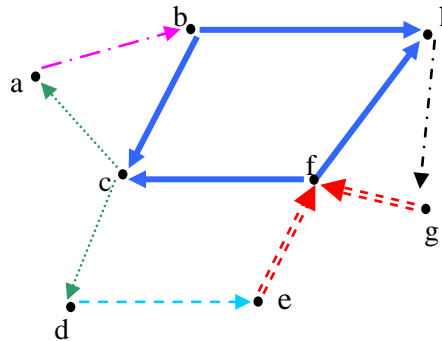


Figure V.6. (b) La partition des arcs de H en bipartis complets. Les arcs de chaque style de trait représentent un biparti complet.

Supposons que le graphe H est le graphe adjoint d'un graphe G qu'on doit chercher. Pour cela, partitionnons les arcs de H en bipartis complets :

- | | |
|----------------------------|-------------------------|
| $B_1 = \{(a), (b)\}$ | $B_2 = \{(c), (a,d)\}$ |
| $B_3 = \{(d), (e)\}$ | $B_4 = \{(e, g), (f)\}$ |
| $B_5 = \{(b, f), (c, h)\}$ | $B_6 = \{(h), (g)\}$ |

Le graphe G résultant tel que $H=L(G)$ est:

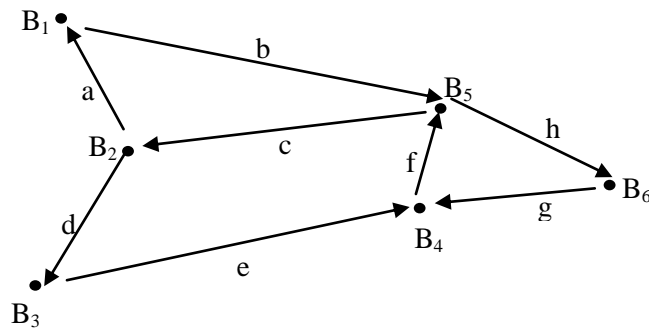


Figure V.6. (c) Le graphe G construit à partir de son graphe adjoint H de la figure V.6. (a).

3. H est le graphe adjoint d'un graphe sans boucles si et seulement si H ne contient aucune configuration « Z » ou « \triangle ».
4. H est le graphe adjoint d'un graphe si et seulement si toute paire de sommets ayant des successeurs communs ont tous leurs successeurs communs.
5. H est le graphe adjoint d'un graphe si et seulement si toute paire de sommets ayant des prédécesseurs communs ont tous leurs prédécesseurs communs.

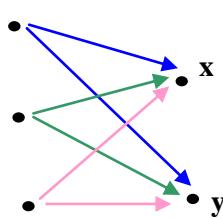
6. Si G est fortement connexe alors $L(G)$ l'est également et vice versa.

7. $L(G)$ contient un circuit hamiltonien si G contient un circuit eulérien.

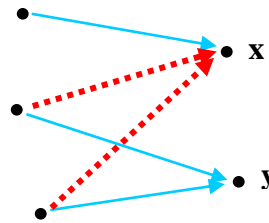
Ainsi H n'est le graphe adjoint d'aucun graphe si est seulement s'il existe une paire de sommets ayant des successeurs communs et des successeurs non communs ou des prédécesseurs communs et des prédécesseurs non communs.

On se pose alors le problème de savoir comment transformer H pour en faire un nouveau graphe qui est le graphe adjoint d'un graphe.

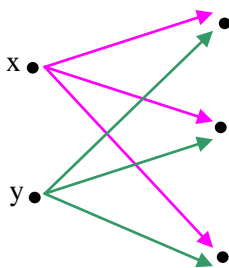
Dans le cadre des problèmes d'ordonnancement, les éventuelles transformations doivent naturellement préserver les contraintes de succession.



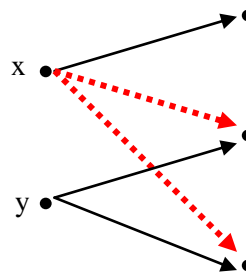
(a) *x et y ont les mêmes prédécesseurs dans un graphe H.*



(b) *x et y n'ont pas les mêmes prédécesseurs, d'où la présence des « Z ».*



(c) *x et y ont les mêmes successeurs.*



(d) *x et y n'ont pas les mêmes successeurs, d'où présence des « Z ».*

Figure V.7. Sous graphes contenant de la configuration « Z ».

IV Algorithme de construction du graphe PERT à partir du graphe des potentiels

Dans les problèmes d'ordonnement de projet, le suivi opérationnel des tâches est très important. Le chef de projet dresse le planning en utilisant la modélisation par les graphes.

Le dessin du graphe des potentiels est facile à cause de son unicité malgré le nombre important d'arcs qu'il génère. Par contre le graphe PERT est plus difficile à cause des tâches fictives qu'il génère lui aussi. Mais les praticiens préfèrent travailler avec le graphe PERT parce qu'il est simple à lire (chaque tâche est représenté par un arc).

Nous avons présenté une technique nouvelle de dessin du graphe PERT dans le chapitre III, et nous présenterons dans ce chapitre une deuxième technique de construction du graphe PERT en utilisant quelques résultats sur les graphes adjoints.

Le problème qui se pose est le suivant : est ce qu'il y a une technique ou une méthode qui nous permet de passer du graphe des potentiels (facile) au graphe des PERT (difficile) et qui aide le praticien à le faire ?

De plus, est ce qu'on peut trouver un algorithme réalisant ce travail et qui peut être programmé sur ordinateur ?

Le problème de passage du graphe des potentiels au graphe PERT est un sujet nouveau. Il sera le cœur de notre recherche dans cette thèse.

On se pose alors le problème de savoir comment transformer H (qui est le graphe des potentiels) pour en faire un nouveau graphe qui est le graphe adjoint (graphe PERT).

Le problème qui se pose, est ce que H contient des configurations Z ou non ? S'il ne contient pas des Z il est alors adjoint et la transformation est immédiate. Mais s'il contient des Z on est amené à éliminer la barre de chaque Z préservant naturellement les contraintes de succession. Etudions chaque cas à part :

IV.1 Cas où le graphe des potentiels est un graphe adjoint

Construisons le graphe PERT à partir du graphe des potentiels dans le cas où celui-ci est un graphe adjoint.

En vertu des résultats du paragraphe II.3., on procède comme suit :

On partitionne les arcs du graphe des potentiels en bipartis complets $B_i = (X_i, Y_i)$.

Dans le graphe PERT que l'on veut construire, chaque B_i est représenté par un sommet encore noté B_i et sera le centre de l'étoile.

Considérons la table d'ordonnement suivante [AKR 07] :

Code	Durée	Ant.
α	0	-
a	6	α
b	4	α
c	3	a, b
d	7	a, b
e	9	c
f	1	c
g	5	d
h	8	e, f, g
i	2	e, f, g
j	3	e, f, g
ω	0	h, i, j

Table V.1. Une table d'ordonnement

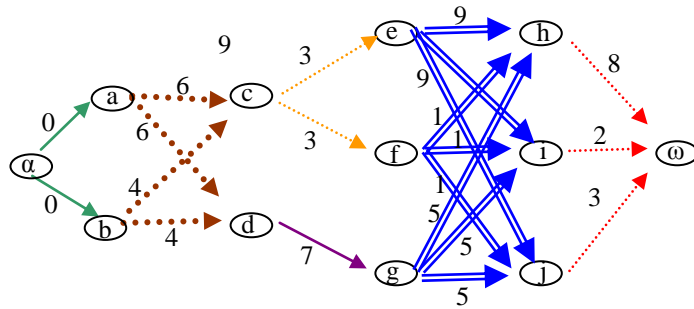
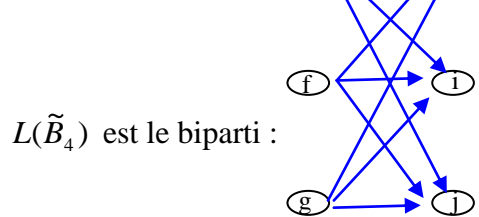
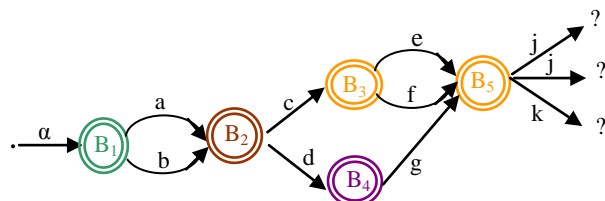
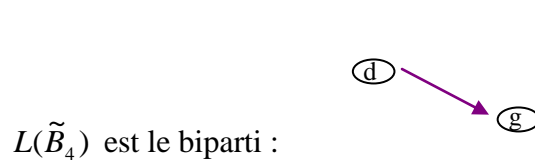
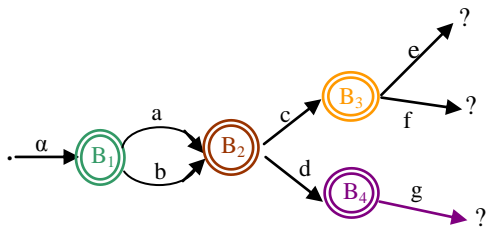
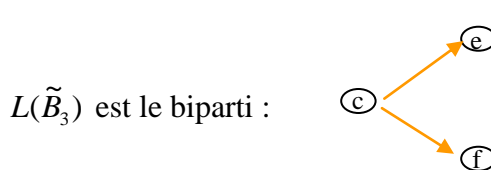
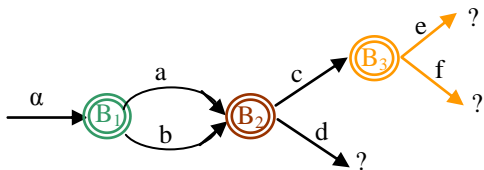
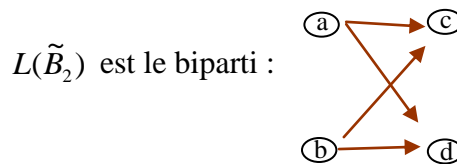
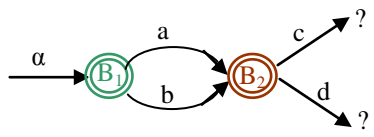
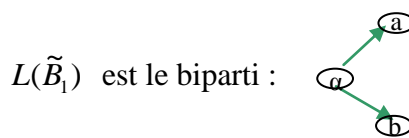
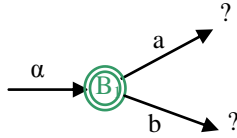


Figure V.8. Le graphe des potentiels de la Table VII.1. dont les arcs sont partitionnés en bipartis complets.

Chaque sommet B_i qui est le centre de l'étoile dans le graphe PERT représente un biparti dans le graphe des potentiels.

- $B_1 = (\{\alpha\}, \{a, b\}),$ $B_2 = (\{a, b\}, \{c, d\}),$
- $B_3 = (\{c\}, \{d, e\}),$ $B_4 = (\{d\}, \{g\}),$
- $B_5 = (\{e, f, g\}, \{h, i, j\}),$
- $B_6 = (\{h, i, j\}, \{\omega\}),$



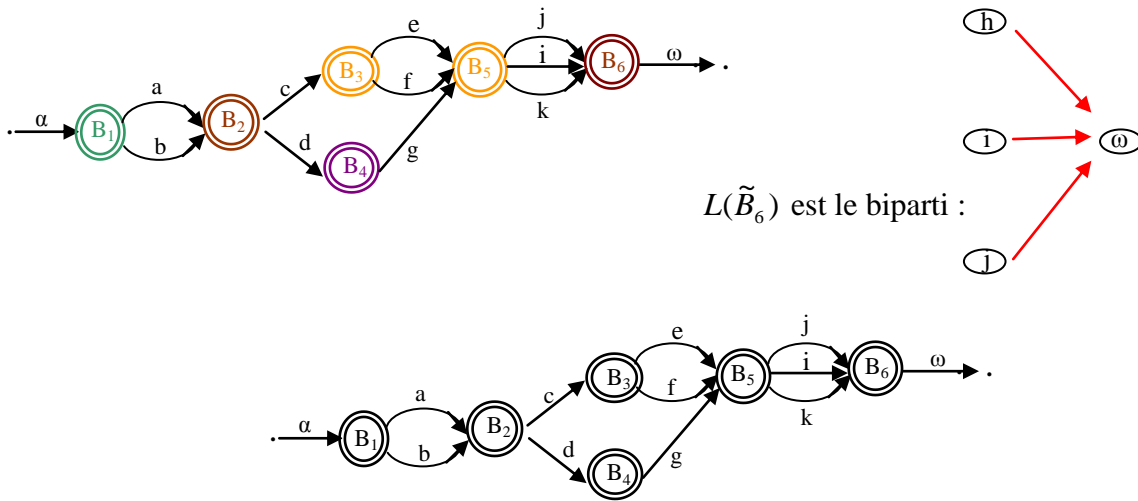


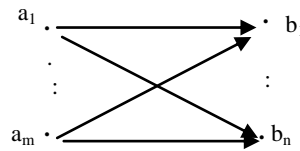
Figure V.9. Le graphe PERT construit à partir des bipartis complets du graphe des potentiels.

IV.2 Cas où le graphe des potentiels n'est pas un graphe adjoint

La construction du graphe PERT est cependant plus complexe dans le cas général où le graphe des potentiels *n'est pas un graphe adjoint* : il n'admet pas de partition des arcs en bipartis complets. C'est dans ce cas qu'on doit le modifier afin de le transformer en graphe adjoint en préservant les contraintes d'antériorités.

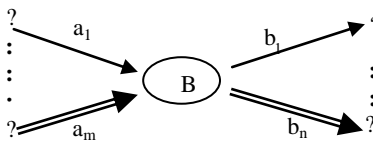
Supposons que les tâches a_1, \dots, a_m précèdent les tâches b_1, \dots, b_n . Dans le graphe des potentiels, ces contraintes d'antériorité sont représentées par un biparti complet. Dans le graphe PERT, elles sont représentées par une étoile.

Code	Antériorités
b_1	a_1, \dots, a_m
\vdots	\vdots
b_n	a_1, \dots, a_m



Le sous-tableau des antériorités de b_1, \dots, b_n

le biparti complet $B = (\{a_1, \dots, a_m\}, \{b_1, \dots, b_n\})$ dans le graphe des potentiels



Le sommet B du graphe PERT correspondant au biparti B.

Figure V.10. Correspondance entre un biparti complet du graphe des potentiels et l'étoile dans le graphe PERT.

Revenons au problème de tâche fictive dans le graphe PERT. Si on a par exemple 4 tâches a, b, c et d avec les contraintes d'antériorité suivantes : a et b précèdent c, mais d est précédée par b uniquement.

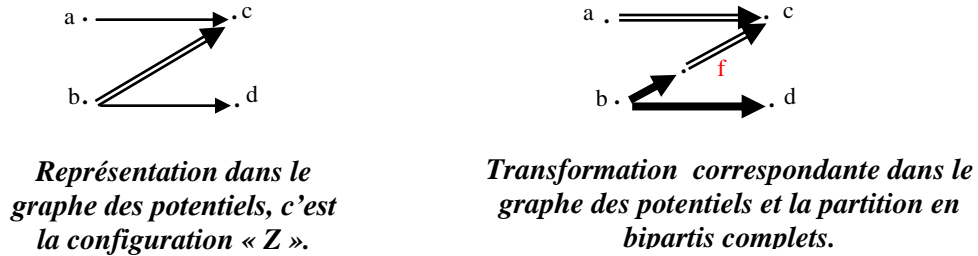


Figure V.11. La configuration « Z » et sa transformation dans le graphe des potentiels.

Dans le graphe des potentiels, il n'y a aucun problème pour la représentation de ces tâches. Elle est faite comme la montre la figure VI.11. Or, pour le passage du graphe des potentiels (qui est considéré comme le graphe adjoint H), on est obligé à éliminer toutes les configurations « Z ». On introduit alors, dans le graphe des potentiels une tâche fictive f dans tout Z :

L'introduction des tâches fictives vise donc à *éliminer* toutes les configurations « Z » du graphe des potentiels, les contraintes restant inchangées. Il faut rappeler que les tâches fictives ne sont nullement nécessaires dans le graphe des potentiels mais ne sont introduites que pour construire le graphe PERT.

On pose alors le problème de la recherche des « Z » dans le graphe des potentiels, c'est-à-dire des sommets ayant des successeurs communs et des successeurs non communs ou des sommets ayant des prédécesseurs communs et des prédécesseurs non communs [BEN 00].

IV.3 Algorithme 1

Soit G_v un graphe des potentiels. G_v doit être orienté, valué, connexe et sans circuit et qui a une source et un puits. G_v étant un graphe conjonctif, organisé en niveaux. On veut construire le graphe PERT correspondant qui est appelé G_e .

Selon [MBBO 06], une première technique d'élimination des Z dans G_v consiste à remplacer la barre (b, c) de tout « Z » par deux arcs (b, f) et (f, c), selon la figure 11, c'est la plus simple mais la plus mauvaise, le nombre des « Z » pouvant être arbitrairement grand. On verra plus tard (chap. VI) une approche optimisée de cette technique :

Début

Si G_v contient des configurations Z **alors**

Repérer les Z_i ($i = 1, 2, \dots, m$)

Pour i allant de 1 à m **faire**

- Créer le sommet f_i dans G_v
- Remplacer la barre (b_i, c_i) de Z_i par $(b_i, f_i), (f_i, c_i)$

Fpour

FSi

- Repérer les bipartis dans G_v ;
- représenter chaque biparti B_i dans G_v par un sommet B_i dans G_e ;
- représenter les arcs de façon que :
 - un arc est dessiné entre 2 sommets B_i et B_j dans G_e ssi les 2 bipartis B_i et B_j dans G_v sont tels que $Y_i \cap Y_j = \emptyset$.
- Vérification si la table d'ordonnancement est respectée.

Fin

Figure V.12. Algorithme (1) de passage du graphe des potentiels au graphe PERT.

IV.3.1 Exemple

Considérons le tableau des contraintes suivant et le graphe des potentiels associé, les durées n'étant pas représentées:

Code	Anté.
α	-
A	α
B	α
C	A, B
D	A, B
E	A
F	D
G	D
H	D, E
I	C, F
J	C, F, G
K	J, K
ω	I, L

Table V. 2. La table d'ordonnancement, désignations non prises.

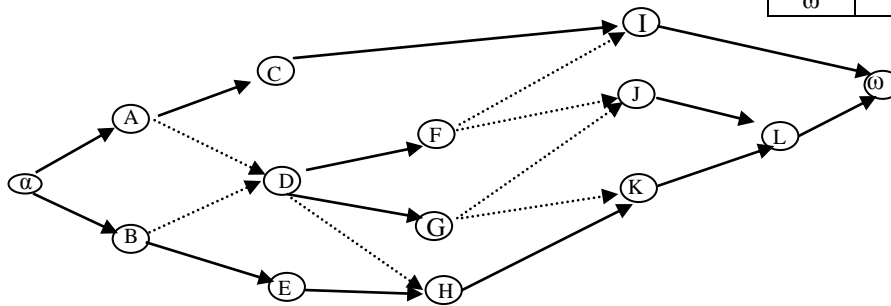


Figure V.13. (a). Le graphe des potentiels correspondant à la table V.2. avec les barres de « Z » en gras.

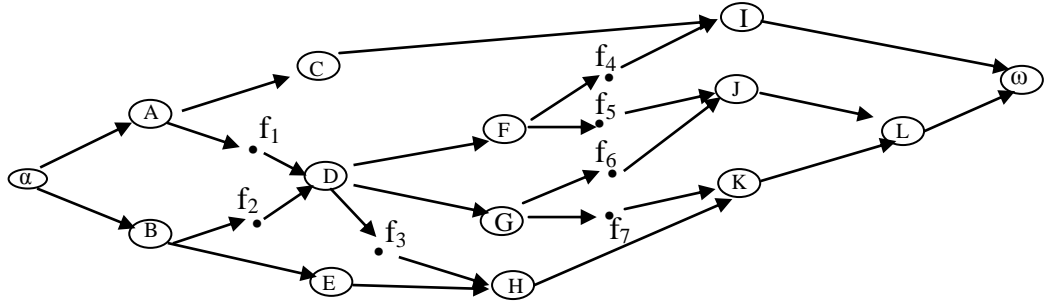


Figure V.13. (b). Le graphe des potentiels modifié par l'introduction des tâches fictives f_i .

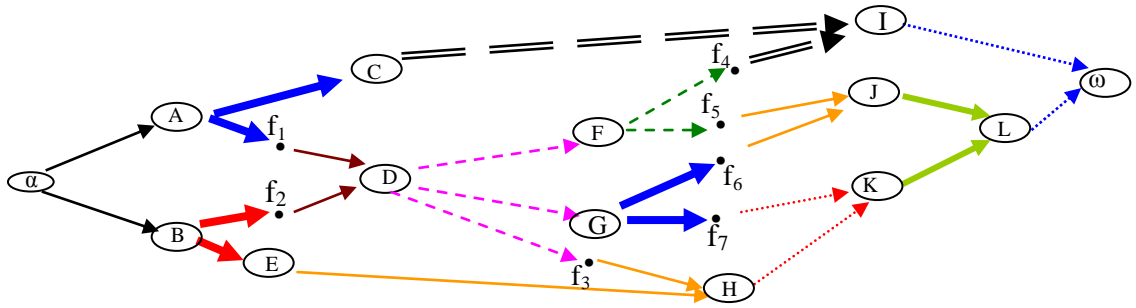


Figure V.13. (c). Le graphe des potentiels modifié par l'introduction des tâches fictives f_i avec réorganisation des tâches en niveaux et partition des arcs en bipartis complets.

Les bipartis complets du graphe des potentiels sont:

- | | | |
|------------------------------------|---------------------------------|--------------------------------|
| $B_1 = (\{\alpha\}, \{A, B\}),$ | $B_2 = (\{A\}, \{C, f_1\}),$ | $B_3 = (\{B\}, \{f_2, E\}),$ |
| $B_4 = (\{f_1, f_2\}, \{D\})$ | $B_5 = (\{D\}, \{F, G, f_3\}),$ | $B_6 = (\{F\}, \{f_4, f_5\}),$ |
| $B_7 = (\{G\}, \{f_6, f_7\}),$ | $B_8 = (\{E, f_3\}, \{H\}),$ | $B_9 = (\{C, f_4\}, \{I\}),$ |
| $B_{10} = (\{f_5, f_6\}, \{J\}),$ | $B_{11} = (\{f_7, H\}, \{K\}),$ | $B_{12} = (\{J, K\}, \{L\}),$ |
| $B_{13} = (\{I, L\}, \{\omega\}),$ | | |

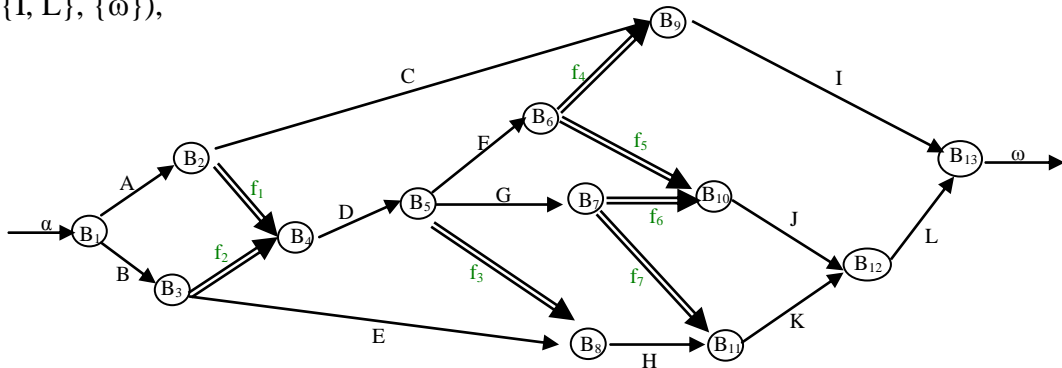


Figure V.13. (d). Construction du graphe PERT à partir du graphe des potentiels.

IV.3.2 Discussion

Nous venons de voir une technique nouvelle qui a pu introduire les graphes adjoints dans les problèmes d'ordonnancement de projet. L'algorithme résultant est de complexité polynomiale ($O(n^3)$). L'algorithme se termine puisqu'il est constitué d'instructions séquentielles sauf dans le cas d'existence des « Z » où on doit créer, dans le graphe des potentiels les tâches fictives f_i .

V Liens entre le diagramme de Gantt et le graphe PERT

Ce paragraphe n'a en fait d'autres buts que de comprendre l'introduction des contraintes d'antériorité dans le diagramme de Gantt.

Il peut aisément être modifié pour construire un graphe PERT sans ou avec un nombre élevé de tâches fictives: la contrainte 'A précède B' est représentée par l'un des schémas de la figure IV.14., f désignant une tâche fictive de durée 0.

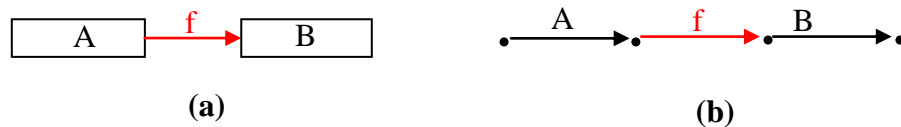


Figure V.14. Introduction de la tâche fictive dans le diagramme de Gantt et modifié en un graphe PERT

Soit le diagramme de Gantt suivant :

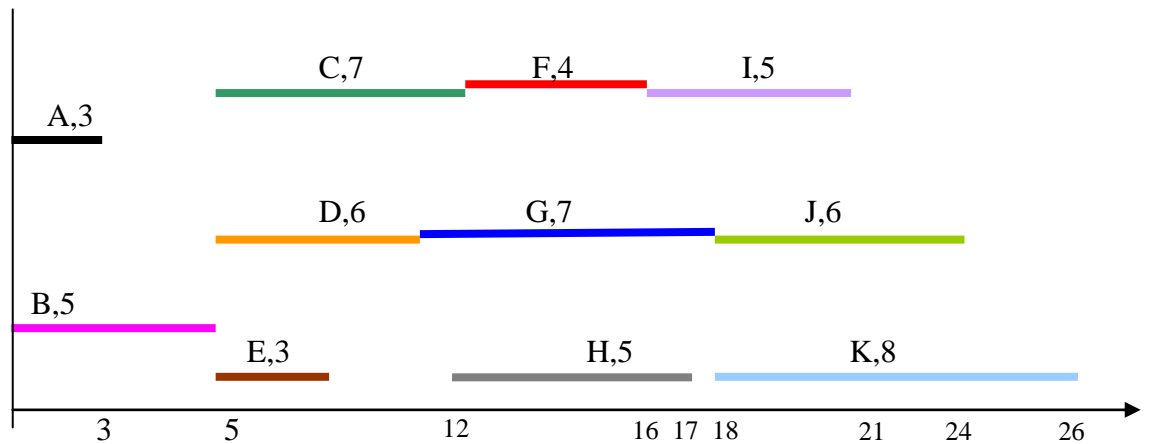


Figure V.15. (a)

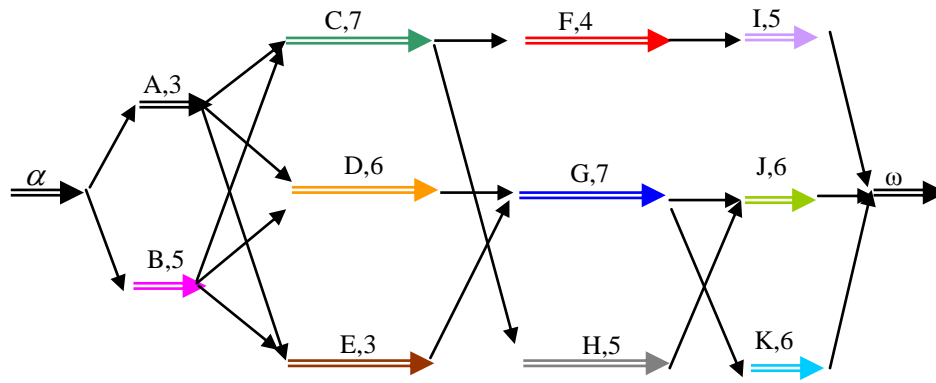


Figure V.15. (b)

Figure V. 15. Le diagramme de Gantt de la figure IV.15.(a)

VI Liens entre le graphe des potentiels et le diagramme de Gantt

En raison du lien étroit qui existe entre le graphe potentiels-étapes et le diagramme de Gantt, le graphe PERT se prête mieux que l'autre à la mise en évidence des responsabilités, aux repérage de l'état d'avancement des travaux, à la transmission des ordres d'exécution, etc.

On peut également avoir les mêmes avantages avec le graphe des potentiels, à condition de travailler, pour ces discussions et échanges d'informations, sur un graphe appelé « graphe planning » (cf. [ROY 70]).

A partir du graphe des potentiels on fait apparaître les barres du diagramme de Gantt par étirement de sommets. On donne à chacune d'elles une longueur proportionnelle à la durée inscrite sur l'arc et en fonction de la succession des tâches, chaque barre est placée de façon à ce que son extrémité gauche coïncide avec la tâche suivante.

Prenons à titre d'exemple, considérons le graphe des potentiels suivant :

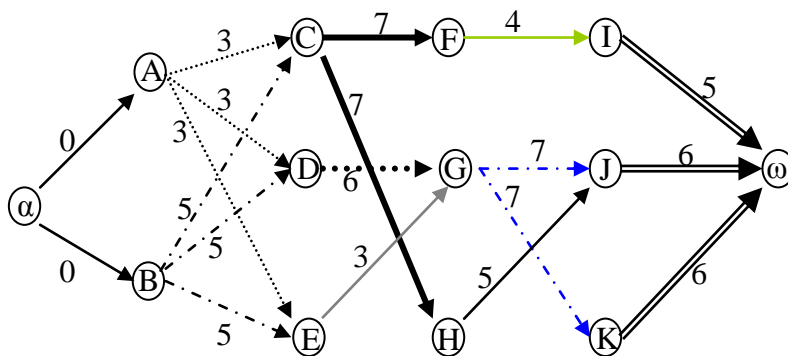


Figure V. 16. (a)

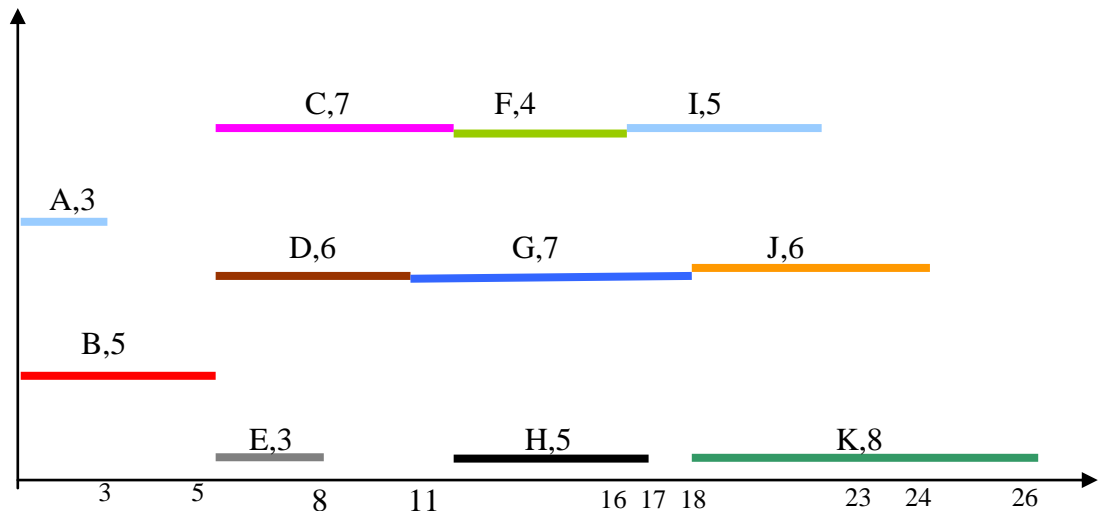


Figure V. 16. (b)

Figure V.16. Le graphe des potentiels et le diagramme de Gantt résultant.

VII Liens entre un graphe PERT et le graphe des potentiels

Puisque le graphe des potentiels est unique alors que le graphe PERT a une infinité de formes, le passage d'un graphe PERT qui appartient à une famille de graphe de l'ordre de l'infini, vers le graphe des potentiels, est facile.

N'oublions pas que dans le graphe des potentiels, une tâche est représentée par un sommet, et dans le graphe PERT elle représentée par un arc.

Il suffit donc, de supprimer les sommets du graphe PERT, ensuite, mettre sur chaque arc (sauf les arcs fictifs) un sommet, ensuite dessiner les arcs reliant ces sommets, en respectant bien sûr les antériorités et en tenant compte des tâches fictives. Notons enfin, que les sommets dessinés doivent être organisés en niveaux. Le graphe résultant est le graphe des potentiels.

VIII Conclusion

Rappelons, que nous avons vu dans le chapitre IV une technique nouvelle de dessin du graphe PERT à partir de la table d'ordonnancement.

On vient de voir , dans ce chapitre, une idée originale de construction d'un graphe PERT par les bipartis, à partir d'un graphe des potentiels donné, en utilisant les notions de graphe adjoint du graphe. Cette idée est transformée en algorithme de complexité polynomiale.

On verra, dans le chapitre suivant, une optimisation de cette technique en vue de réduire les tâches fictives dans le graphe PERT.

Chapitre VI

Nouvelle approche optimisée dans la construction du graphe PERT

Sommaire

I Introduction	102
II Première technique de réduction des tâches fictives	102
II.1 Algorithme 2	104
II.2 Exemple	105
III Deuxième technique de réduction des tâches fictives	106
III.1 Algorithme 3	107
III.2 Exemple	108
IV Etude comparative	108
V Conclusion	111

I Introduction

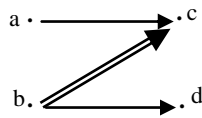
Nous avons vu précédemment que le graphe des potentiels est unique alors que nous avons une infinité de graphes PERT qui correspondent au graphe des potentiels à cause de l'introduction des tâches fictives dans certaines situations. Cependant, et selon [EL 99], le graphe PERT devient vite illisible dans le cas général où on répertorie dans les applications réelles plus de 40 % de tâches fictives supplémentaires.

On pose alors, dans ce chapitre, le problème de réduction des tâches fictives dans le graphe PERT construit dans le chapitre précédent à partir d'un graphe des potentiels donné.

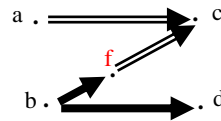
On verra une première technique d'optimisation de l'algorithme qui se base sur les sous graphes bipartis complets, suivie d'une deuxième technique, à mon avis, qui est plus efficace, en tenant compte des résultats de comparaison réalisée entre les deux techniques. Le chapitre sera clôturé par une conclusion qui fait une synthèse sur ce qu'a été fait jusqu'à ce point.

II Première technique de réduction des tâches fictives

Rappelons que la première technique présentée jusque là consiste à éliminer des Z dans le graphe des potentiels G_v en remplaçant la barre (b, c) de tout « Z » par deux arcs (b, f) et (f, c), selon la figure VI.1.



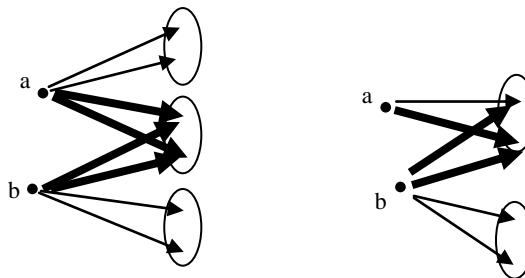
Représentation dans le graphe des potentiels, c'est la configuration « Z ».



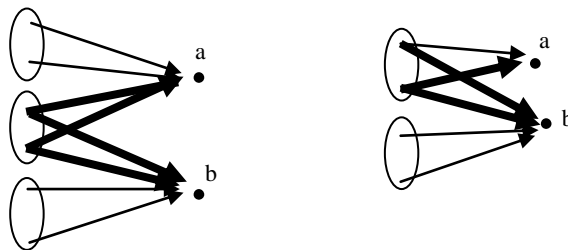
Transformation correspondante dans le graphe des potentiels et la partition en bipartis complets.

Figure VI. 1. La configuration « Z » et sa transformation dans le graphe des potentiels.

Selon [AKR 07], l'algorithme 1 étudié dans le chapitre précédent peut être amélioré grâce à une technique de réduction des tâches fictives car leur présence en nombre encombre le graphe et devient illisible. Elle consiste à éliminer, dans G_v , plusieurs « Z » à la fois par regroupement des barres ayant même extrémité initiale ou même extrémité terminale et qui appartiennent au même biparti complet (Figure VI.2. et VI.3.).



(b) a et b ont des successeurs communs et des successeurs non communs dans les deux



(a) a et b ont des prédécesseurs communs et des prédécesseurs non communs dans les deux

Figure VI. 2. Les sommets a et b ayant des successeurs communs et des successeurs non communs, des prédécesseurs communs et des prédécesseurs non communs d'où la présence des Z.

Les « Z » dans G_v sont détruits, avant toute transformation, comme suit :

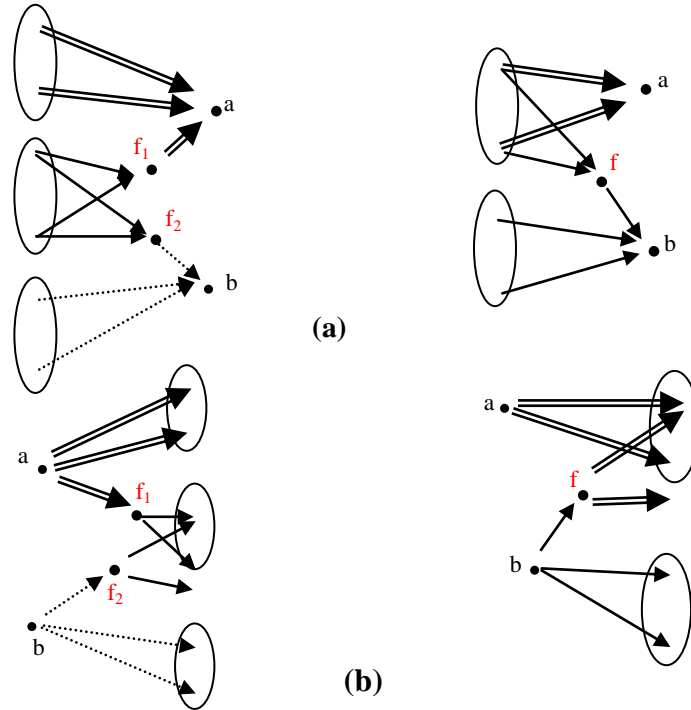


Figure VI. 3. Regroupement des barres de "Z", tâches fictives et partition des arcs en bipartis complets.

II.1 Algorithme 2

Début

Si G_v contient des configurations Z alors

Repérer les Z_i ($i = 1, 2, \dots, m$)

Pour tout biparti contenant 2 et plus de barre de Z faire

- Créer des sommets fictifs au nombre de barres de Z
- Regrouper les barres ayant même extrémité initiale ou même extrémité terminale appartenant au même biparti complet

Fpour

Fsi

- Repérer les bipartis dans G_v
- Représenter chaque biparti B_i dans G_v par un sommet B_i dans G_e de façon que :
Un arc est dessiné entre 2 sommets B_i et B_j dans G_e ssi les 2 bipartis B_i et B_j dans G_v sont tels que $Y_i \cap Y_j = \emptyset$
- Vérification si la table d'ordonnancement est respectée.

Fin

Figure VI. 4. Algorithme (2) de passage du graphe des potentiels au graphe PERT.

II.2 Exemple

Considérons le tableau des contraintes suivant et le graphe des potentiels associé, les durées n'étant pas représentées:

Code	Anté.
0	-
1	0
2	0
3	1, 2
4	1, 2
5	2
6	3, 4
7	4
8	4, 5
9	6
10	3, 6, 7
11	7, 8
12	10, 11
13	9, 12

Table VI.1. La table d'ordonnancement T avec les non prises.

désignations

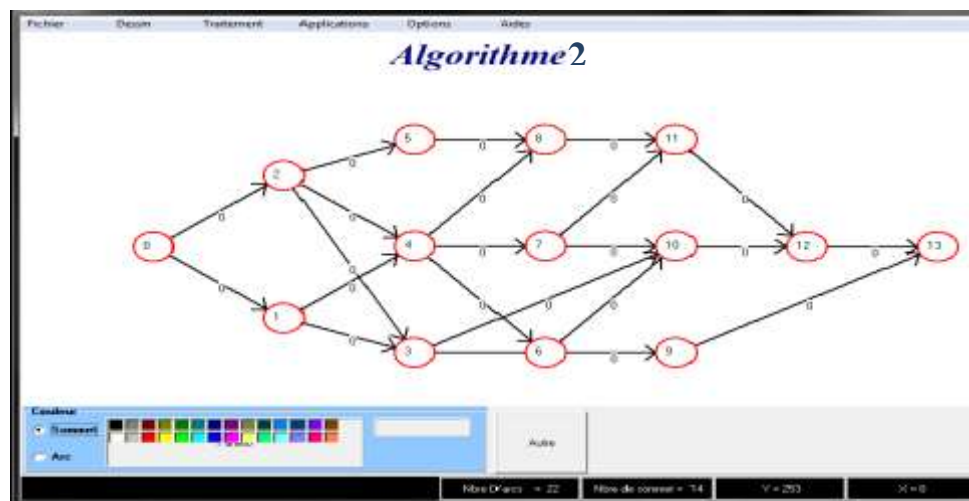


Figure VI.5. (a) Le graphe des potentiels correspondant à la table V.1.

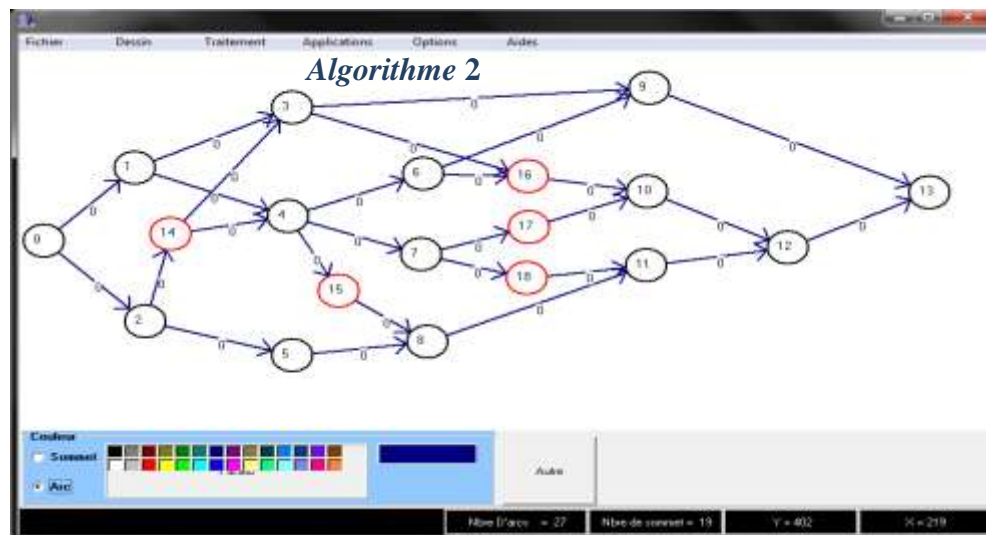


Figure VI.5. (b) Le graphe des potentiels avec l'introduction des tâches fictives (sommets 14,..., 18) avec regroupement des barres de Z appartenant au même biparti et organisation des sommets en niveaux.

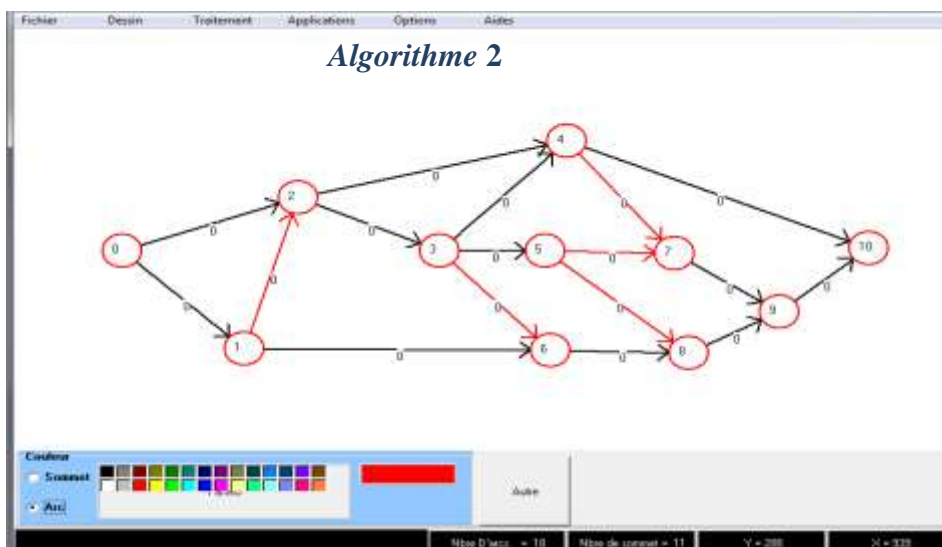


Figure VI.5. (c) Construction du graphe PERT à partir du graphe des potentiels par l'application de l'algorithme 2 avec 5 tâches fictives (1, 2), (3, 6), (4, 7), (5, 7), (5, 8).

L'algorithme se termine puisque la boucle *Pour n*'est exécutée que lorsqu'il y a des Z, et le nombre de Z dans G_v est fini. La suite de l'algorithme est une succession d'instructions simples. La complexité de l'algorithme est polynomiale elle est de l'ordre $O(n^3)$.

III Deuxième technique de réduction des tâches fictives

Notre nouvelle technique agit sur le graphe PERT après sa construction en réduisant le nombre des tâches fictives et non pas sur le graphe des potentiels comme il est déjà fait dans la technique précédente (voir section II).

Donc, après la construction du graphe PERT en utilisant l'algorithme n° 1 (algorithme étudié au chap. V), on applique deux astuces d'élimination de tâches fictives sur le graphe PERT résultant nommé G_e :

- Recherche des sommets (initiales des arcs fictifs) ayant les mêmes successeurs ou les mêmes prédécesseurs (terminaux des arcs fictifs) et les contracter en un seul sommet.

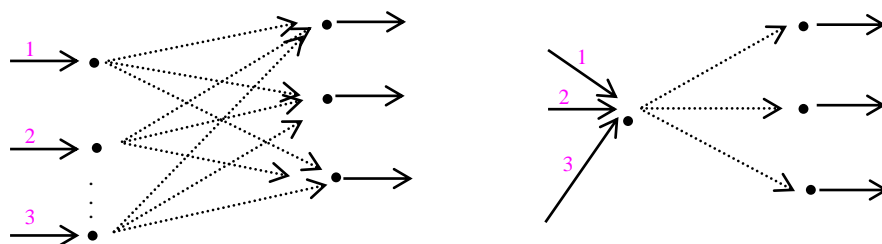


Figure VI. 6. Les sommets ayant les mêmes successeurs contractés en un sommet unique dans G_e .

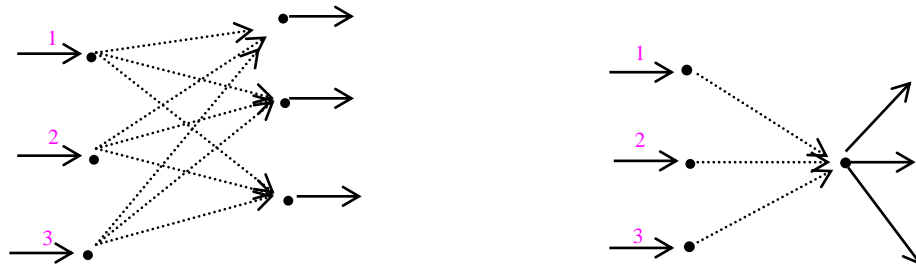


Figure VI. 7. Les sommets ont les mêmes prédécesseurs contractés en un sommet unique dans G_e .

- Contraction des sommets initial et terminal de tout arc fictif u débutant seul d'un ou plusieurs arcs réels ou arrivant seul à un ou plusieurs arcs réels.

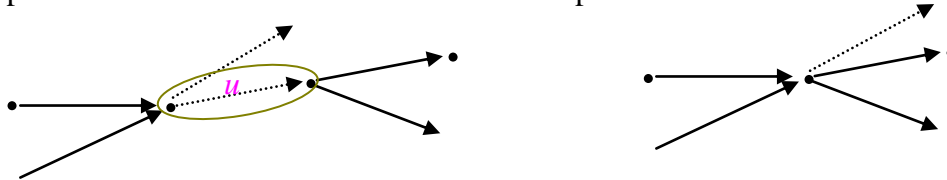


Figure VI. 8. Contraction des sommets initial et terminal de l'arc fictif u arrivant seul à un ou plusieurs arcs réels.



Figure VI. 9. Contraction des sommets initial et terminal de l'arc fictif u débutant seul d'un ou plusieurs arcs réels.

III.1. Algorithme 3

La technique vue dans la section précédente est transformée en écriture algorithmique suivante :

Début

Si G_v contient des configurations Z **alors**

Repérer les Z_i appartenant au même biparti complet BC_i ($i = 1, 2, \dots, m$)

Pour i allant de 1 à m **faire**

- Créer le sommet f_i dans G_v
- Remplacer les barres de $Z \in BC_i$ les tâches fictives correspondantes.

Fpour

FSi

- Repérer les bipartis dans G_v ;
- représenter chaque biparti B_i dans G_v par un sommet B_i dans G_e ;
- représenter les arcs de façon que :
un arc est dessiné entre 2 sommets B_i et B_j dans G_e ssi les 2 bipartis B_i et B_j dans G_v sont tels que $Y_i \cap Y_j = \emptyset$.
- Contracter, dans G_e , les sommets ayant les mêmes successeurs ou les mêmes prédécesseurs.
- Contracter, dans G_e , les sommets initial et terminal de tout arc fictif débutant seul d'un ou plusieurs arcs réels ou arrivant seul à un ou plusieurs arcs réels.
- Vérification si la table d'ordonnancement est respectée.

Fin

Figure VI.10. Algorithme (3) de passage du graphe des potentiels au graphe PERT.

III.2 Exemple

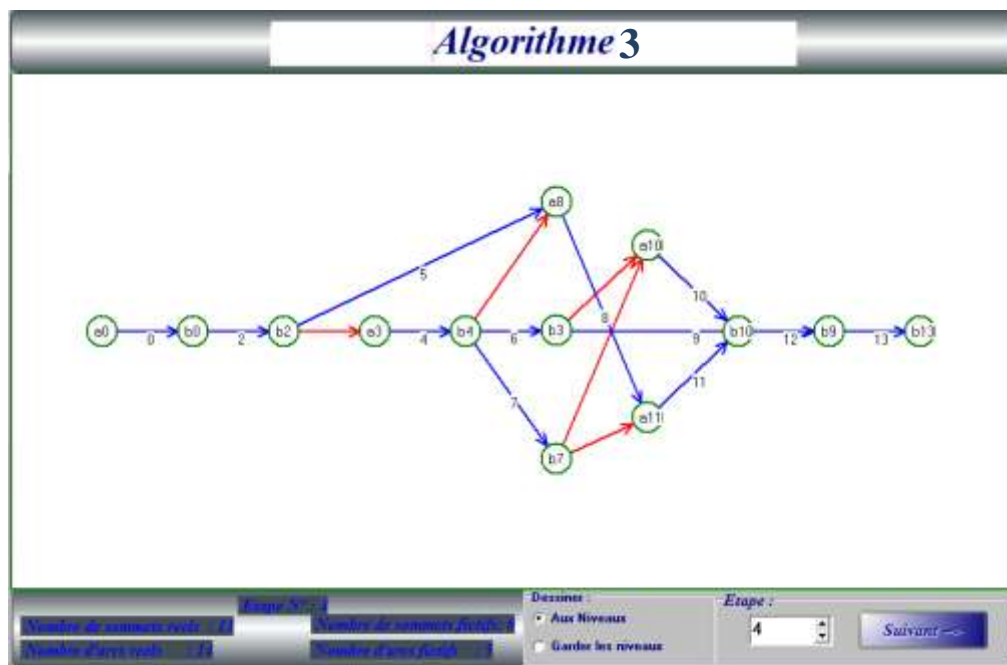


Figure VI.11. Construction du graphe PERT à partir du graphe des potentiels par l'application de l'algorithme 3 de la Table VI. 1.

IV Etude comparative

Pour réaliser une comparaison entre les trois algorithmes, on doit avoir un sujet de comparaison qui est le nombre de tâches fictives dans chaque graphe PERT obtenu par l'application d'un des trois algorithmes.

Pour cela, nous avons réalisé un logiciel qui comporte les modules suivants :

- 1 - Dessin de graphe: Ce module contient les fonctions habituelles de dessins de graphes qui sont : la création et suppression de sommets, la création et la suppression d'arcs, sauvegarde et la restauration des graphes, etc.
- 2 - Bibliothèque: Elle inclut les algorithmes de la théorie des graphes tels que: la connexité d'un graphe, détection d'un circuit, l'organisation du graphe en niveaux, etc. Pour tester le graphe dessiné dans la première étape, il doit vérifier les conditions du réseau (le réseau orienté, connexe, sans circuit, contenant une seule source α et un seul puits ω).
- 3 - Application des 3 algorithmes : Le software exécute les trois algorithmes étudiés précédemment. A chaque exécution d'un des algorithmes on fait ressortir les informations concernant le graphe PERT réalisé par l'application de l'algorithme (nombre de sommet dans G_v , nombre d'arcs dans G_v , nombre d'arcs réels dans G_e et enfin nombre d'arcs fictifs dans G_e par l'application d'un des 3 algorithmes).

Les résultats obtenus sont rangés dans une table (voir *Table VI.2.*). Le même processus des trois étapes précédentes est répété pour un nouveau graphe avec un nombre aléatoire de tâches réelles (i.e. nombre de sommets aléatoire dans G_v).

Nbre de sommets dans G_v	Nbre d'arcs dans G_v	Nbre d'arcs réels dans G_e	Nbre d'arcs fictifs dans G_e Algorithme 1	Nbre d'arcs fictifs dans G_e Algorithme 2	Nombre d'arcs fictifs dans G_e Algorithme 3	Meilleur algorithme
8	16	8	4	3	3	ALG2=ALG3
36	33	36	17	14	11	ALG3
14	23	14	8	5	4	ALG3
30	34	30	16	12	9	ALG3
24	45	24	11	9	8	ALG3
24	54	24	16	12	11	ALG3
40	65	40	20	18	14	ALG3
22	39	22	15	12	12	ALG2=ALG3
52	76	52	31	22	19	ALG3
30	60	30	13	10	9	ALG3
42	65	42	21	16	14	ALG3
26	43	26	10	10	10	ALG1=ALG2=ALG3
46	76	46	26	23	20	ALG3
58	80	58	30	25	24	ALG3
52	79	52	34	24	19	ALG3
68	90	68	41	35	31	ALG3
54	98	54	33	28	22	ALG3
40	84	40	17	14	13	ALG3
76	93	76	44	35	29	ALG3
90	122	90	42	36	30	ALG3
94	160	94	50	41	33	ALG3
82	143	82	39	32	26	ALG3
62	87	62	32	27	24	ALG3
36	68	36	16	13	11	ALG3
324	423	324	149	127	113	ALG3
348	480	348	179	160	143	ALG3
170	201	170	72	61	54	ALG3
28	45	28	12	11	9	ALG3
66	79	66	30	26	20	ALG3
38	63	38	17	15	14	ALG3
40	83	40	21	18	14	ALG3
26	72	26	9	8	6	ALG3
42	79	42	19	19	16	ALG3
14	41	14	6	6	6	ALG3
28	50	28	14	12	11	ALG3
200	297	200	90	81	69	ALG3
74	134	74	35	30	26	ALG3
148	188	148	65	60	53	ALG3
70	104	70	33	30	31	ALG3
60	89	60	28	28	28	ALG2=ALG3
40	75	40	19	17	13	ALG3
278	357	278	139	126	114	ALG3
96	170	96	45	41	39	ALG3
92	164	92	43	40	36	ALG3
84	153	84	40	37	35	ALG3
66	100	66	31	25	23	ALG3
84	145	84	43	40	34	ALG3
286	389	286	131	117	107	ALG3
299	405	299	139	121	114	ALG3
162	277	162	88	79	71	ALG3
90	186	90	40	36	31	ALG3
68	121	68	32	30	27	ALG3
66	99	66	24	21	16	ALG3
66	109	66	32	30	23	ALG3
78	116	78	31	27	22	ALG3
46	79	46	21	18	16	ALG3
118	180	118	53	47	40	ALG3
82	134	82	38	33	29	ALG3
78	145	78	34	30	27	ALG3
68	102	68	32	27	20	ALG3
66	97	66	30	28	21	ALG3
71	105	71	31	31	31	ALG1=ALG2=ALG3
192	278	192	88	76	69	ALG3
174	256	174	83	78	69	ALG3
218	341	218	104	94	81	ALG3
242	389	242	113	98	90	ALG3
150	245	150	71	63	57	ALG3
120	187	120	57	51	42	ALG3
184	290	184	90	84	76	ALG3
352	450	352	166	151	133	ALG3

Table VI.2. Tableau récapitulatif concernant l'exécution des trois algorithmes.

Une lecture des résultats figurant dans la table précédente aboutie aux conclusions suivantes (*Table VI.3.*) :

Nombre de réseaux	Meilleur résultat des 3 algorithmes				
	ALG1 = ALG2 = ALG3	ALG2 = ALG3	ALG1	ALG2	ALG3
75	02.66%	04%	0%	0%	93.33%

Table VI.3. Résultats de la comparaison.

On voit clairement que l’algorithme 3 fait les meilleurs résultats avec plus de 93%. Ce qui montre la nette supériorité de cette technique.

V Conclusion

Ce travail vient d’introduire les graphes adjoints dans les problèmes d’ordonnement de projet avec ou sans présence des « Z » dans le graphe des potentiels et ceci pour la construction du graphe PERT (voir Algorithme 1, chapitre IV). Cette nouvelle approche présentée dans ce chapitre (Algorithme 3) s’ajoute à celle citée précédemment.

Notre contribution ne s’est pas arrêtée à ce stade en réalisant l’algorithme 3 mais elle est consolidée par une comparaison basée sur la réduction des tâches fictives dans le graphe PERT, réalisée entre plusieurs algorithmes qui tentent à optimiser l’idée de départ (Algorithme 1).

Ce chapitre ouvre la porte grande ouverte à la recherche du graphe PERT minimal qui reste depuis des années un sujet d’actualité chez les chercheurs dans ce domaine de l’ordonnement de projet.

Chapitre VII

Le graphe PERT minimal

Sommaire

I Introduction	112
II Etat de l'art	113
II.1. Algorithme de F. STERBOUL et D. WERTHEIMER	114
II.1.1 Notations et définitions	114
II.1.2 Graphe arc-dual d'un graphe donné	115
II.1.3 Construction	115
II.1.4 Exemple	116
III Nouvel algorithme de construction du graphe PERT minimal	118
III. 1 Notations et définitions	118
III. 2 Construction	119
III.3 Démonstration	124
III.2.1 Lemme 1	125
III.2.2 Lemme 2	126
III.2.3 Lemme 3	126
III.2.4 Théorème	126
III.4. Exemple de projet de construction d'un building industriel	127
IV. Implémentation	130
IV. Résultats numériques	132
V Conclusion	135

I Introduction

Vu que le graphe des potentiels est très simple à dessiner, on est amené à construire le graphe PERT correspondant à partir du graphe des potentiels tout en réduisant le nombre de tâches fictives. Le problème a été abordé dans le chapitre précédent mais dans un contexte

particulier, puisque le sujet principal était de construire un graphe PERT à partir d'un graphe des potentiels donné. La réduction des tâches fictives n'était qu'un problème secondaire qui a surgi après l'amélioration faite par [AKR 07]. Cette amélioration nous a été très bénéfique du moment qu'elle nous a poussé à chercher une meilleure solution, néanmoins, le problème fondamental qui se pose maintenant de chercher un algorithme qui détermine le graphe PERT minimal. Nous savons que ce type de problème est NP-difficile mais nous tentons tout de même à chercher un nouvel algorithme facile à manipuler et meilleur que ceux qui se trouvent dans la littérature.

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

Nous présenterons ensuite notre nouvelle méthode utilisant un ensemble de résultats des graphes adjoints étudiés au chapitre 5, qui nous permettra de construire le graphe PERT en minimisant le nombre de tâches fictives.

L'importance de cette minimisation dans le graphe PERT réside dans le fait que les praticiens n'aiment pas utiliser le graphe des potentiels, mais ils préfèrent travailler avec le graphe PERT 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.

Signalons que nous travaillons, dans ce qui suit, sur des graphes non valués. On considère que le graphe des potentiels 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 (voir chap. IV.).

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 α (resp. ω).

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.

II Etat de l'art

Rappelons que les réseaux de gestion de projet, comme on a vu précédemment, sont basés sur deux types de représentation: Activités sur Arcs (AOA) qui est le graphe PERT et activités sur les nœuds (AON) qui est le graphe des potentiels. Chaque type de représentation (AOA ou AON) a les caractéristiques qui sont propres à la méthode [LEW 95].

Bien que la construction du graphe AON soit relativement facile, le réseau AOA peut contenir des arcs fictifs et sa construction est beaucoup plus difficile. Toutefois, en raison de leur importance, les réseaux AOA ont attiré de nombreux essais de recherche pour générer des réseaux AOA avec un nombre minimal d'arcs fictifs. Des exemples de méthodes de minimisation utilisant des heuristiques peuvent être trouvés dans [DIM 63], [HAY 69], [FIS 68].

Dans les années 1970 Corneil et al. [CGL 73], a affirmé avoir réalisé un algorithme optimal qui contient le minimum d'arcs fictifs, mais Syslo [SYS 81] infirmait leur résultat. Krishnamoorthy et Deo [KDE 79] ont montré que la constatation du problème de minimisation d'arcs fictifs est NP-difficile.

Dans les années 1980, [SPI 80], [SYS 81], [SYS 84] ont offert plusieurs heuristiques pour arriver au seul objectif : le graphe PERT minimal.

Dans les années 1990, [AEH, 96], [DDH 93] ont proposé des méthodes de génération de réseaux PERT, 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 [EH 80] a élaboré un indice de complexité comme un outil de mesure de la complexité dans les réseaux PERT. Kamburowski et al. [KMS 92], [KMS 00] chacun d'eux développé une méthode qui génère un AOA-index avec la complexité minimale. Noter cependant que à partir d'un seul réseau MPM on peut représenter plusieurs différents réseaux PERT avec le même indice minimal de complexité.

En 2007, Cohen et Sadeh [CS 07] ont présenté une nouvelle approche pour l'élaboration d'un graphe PERT unique et qui a, d'après leurs conclusions, le nombre minimal d'arcs fictifs.

II.1. Algorithme de F. STERBOUL et D. WERTHEIMER

II.1.1 Notations et définitions

Soit un graphe orienté $G=(X, U)$ où X est l'ensemble des sommets et U l'ensemble des arcs. (i, j) arc reliant le sommet i au sommet j .

$u \rightarrow v$ indique qu'il existe un chemin du premier arc u au dernier arc v .

$\forall i \in X$ on note :

$$P(i) = \{j \in X : (j, i) \in U\},$$

$$Q(i) = \{j \in X : (i, j) \in U\},$$

$$\tilde{P}(i) = \{j \in X, \exists \text{ un chemin de } j \text{ vers } i\}.$$

On appelle arc redondant s'il existe un chemin de longueur supérieure à un de i vers j .

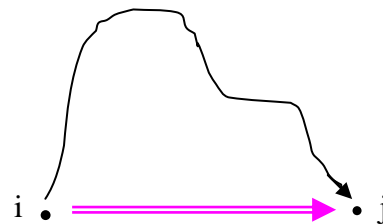


Figure VII.1. Un arc redondant (i, j) .

II.1.2 Graphe arc-dual d'un graphe donné

Soit $G = (X, U)$ un graphe donné, sans circuit et sans arc redondant, et soit $H = (Y, V)$ un graphe.

S'il existe une application f injective : $X \rightarrow V$ telle que $\forall j, \forall i \in \tilde{P}(j)$ ssi $f(i) \rightarrow f(j)$, H sera dit graphe arc-dual de G par l'application f .

Les données du problème d'ordonnancement sont représentées par le graphe « français » $G = (X, U)$, où les sommets représentent les tâches et où l'arc (i, j) appartient à U si et seulement si la tâche i précède immédiatement la tâche j . Le but du problème est donc de construire un graphe H arc-dual de G (H est le graphe PERT ou « américain ») ; on doit minimiser le nombre de sommets [SW 80].

II.1.3 Construction

A partir du graphe $G = (X, U)$, on construit un graphe arc-dual $H_0 = (Y_0, V_0)$ par l'application f de la manière suivante :

- pour chaque sommet $i \in X$, on définit deux sommets a_i et b_i ;
- on pose $Y_0 = \bigcup_{i \in X} \{a_i\} \cup \{b_i\}$
- V_0 est constitué des arcs (a_i, b_i) pour tout $i \in X$, ainsi que des arcs (b_i, a_j) si $i \in P(j)$ dans G .

Les arcs de la forme (a_i, b_i) seront dits réels et ceux de la forme (b_i, a_j) seront dits virtuels.

Posons $f(i) = (a_i, b_i)$.

Les différents chemins de H_0 sont formés d'arcs alternativement réels et virtuels.

Dans le graphe H_0 on pose : $A = \cup \{a_i\}$, $B = \cup \{b_i\}$

$$a_i R a_j \text{ ssi } P(i) = P(j)$$

$$b_i S b_j \text{ ssi } Q(i) = Q(j)$$

Les deux relations R et S sont des relations d'équivalence.

Soient $\bar{a}_1, \dots, \bar{a}_L, \bar{b}_1, \dots, \bar{b}_M$ les classes correspondantes sur A et B .

On appelle contraction de type 1 l'opération qui consiste dans H_0 à contracter, en un sommet unique, tous les sommets d'une même classe.

Dans le graphe H_0 on effectue toutes les contractions possibles et l'on note $H_I = (Y_I, V_I)$ le graphe obtenu.

L'application f induit une application de $X \rightarrow V_1$. En raison de simplifier les notations, on continue à noter cette application par f . Les sommets de H_1 seront notés $\overline{a_1}, \dots, \overline{a_L}, \overline{b_1}, \dots, \overline{b_M}$.

H_1 est un graphe arc-dual de G par l'application f .

• Un arc $(\overline{b_i}, \overline{a_j})$ est un bon arc de H_1 ssi :

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

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.

Les bons arcs de H_1 n'ont deux à deux aucun sommet commun.

Dans le graphe $H_1 = (Y_1, V_1)$ on effectue toutes les contractions possibles de type 2. Soit $H_2 = (Y_2, V_2)$ le graphe obtenu.

Pour simplifier les notations :

- on note les sommets de H_2 comme ceux de H_1 ;
- on note f l'application injective de $X \rightarrow V_2$ induite par $f : X \rightarrow V_1$.

H_2 est graphe arc-dual de G par l'application f .

Le graphe $H_2 = (Y_2, V_2)$ a le nombre minimal de sommets parmi tous les graphes arc-duaux de G .

II.1.4 Exemple

Soit le graphe des potentiels G_v suivant :

Activity	Ant.
1	-
2	-
3	-
4	1, 2, 3
5	1, 3
6	2, 3
7	4, 5
8	6
9	5, 6, 7

Table VII.1. Table d'ordonnancement.

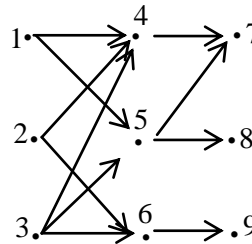


Figure VII.1. (a) Le graphe des potentiels de la Table VII.1.

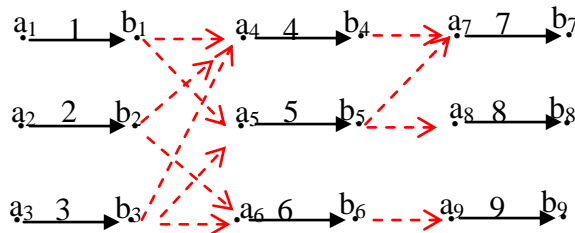


Figure VII.1. (b) Le graphe H_0 de la Figure VII.2 (b).

Les bons arcs dans H_1 , cherchés dans la *Table VII.2*, sont contractés dans le graphe H_2 comme suit (*Figure VII. 2 (c)*, *Figure VII. 2 (d)*):

Arc	$k \in$	$l \in$	Obs.
(\bar{b}_1, \bar{a}_2)	$\Gamma^-(4) = \{1, 2, 3\}$	$\Gamma^+(1) = \{4, 5\}$	Other arc
(\bar{b}_1, \bar{a}_3)	$\Gamma^-(5) = \{1, 3\}$	$\Gamma^+(1) = \{4, 5\}$	Good arc
(\bar{b}_2, \bar{a}_2)	$\Gamma^-(4) = \{1, 2, 3\}$	$\Gamma^+(2) = \{4, 6\}$	Other arc
(\bar{b}_2, \bar{a}_4)	$\Gamma^-(6) = \{2, 3\}$	$\Gamma^+(2) = \{4, 6\}$	Good arc
(\bar{b}_3, \bar{a}_2)	$\Gamma^-(4) = \{1, 2, 3\}$	$\Gamma^+(3) = \{4, 5, 6\}$	Other arc
(\bar{b}_3, \bar{a}_3)	$\Gamma^-(4) = \{1, 2, 3\}$	$\Gamma^+(3) = \{4, 5, 6\}$	Other arc
(\bar{b}_3, \bar{a}_4)	$\Gamma^-(6) = \{2, 3\}$	$\Gamma^+(3) = \{4, 5, 6\}$	Other arc
(\bar{b}_4, \bar{a}_5)	$\Gamma^-(7) = \{4, 5\}$	$\Gamma^+(4) = \{7, 8\}$	Good arc
(\bar{b}_5, \bar{a}_5)	$\Gamma^-(9) = \{4, 5\}$	$\Gamma^+(6) = \{7, 8\}$	Other arc
(\bar{b}_5, \bar{a}_6)	$\Gamma^-(9) = \{5\}$	$\Gamma^+(6) = \{7, 8\}$	Good arc
(\bar{b}_6, \bar{a}_7)	$\Gamma^-(9) = \{6\}$	$\Gamma^+(6) = \{9\}$	Good arc

Table VII.2. Recherche des bons arcs dans

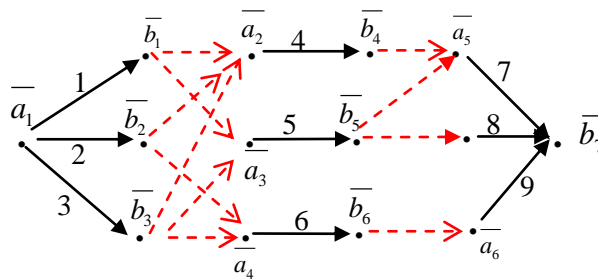


Figure VII. 2. (c) Le graphe H_1 de la figure VII.2 (a).

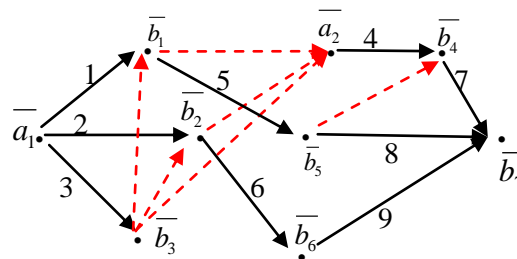


Figure VII. 2. (d) Le graphe H_2 de la figure VII.2 (a).

III Nouvel algorithme de construction du graphe PERT minimal

Rappelons que chaque tâche t est représentée par un sommet t dans le graphe des potentiels G_v et par un arc t dans le graphe PERT G_e . Si la tâche t précède la tâche u , alors (t, u) est un arc de G_v et $T(t) = T(u)$ dans le graphe PERT G_e .

Comme il n'y a pas de tâche qui se succède à elle-même, G_v et G_e ne contiennent pas de circuit.

Un des résultats étudié dans le chapitre V indique que G est le graphe adjoint d'un graphe H , ou $G \cong L(H)$, si et seulement si G ne contient aucune configuration « Z » ou « Δ ».

Une conséquence de ce résultat est que G_v ne contient pas une des configurations « Z » ou « Δ » si et seulement si $G_v \cong L(G_e)$.

Le problème est de construire le graphe PERT G_e quand le graphe des potentiels G_v contient une des configurations « Z » ou « Δ » ; autrement dit comment modifier G_v pour avoir un nouveau G_v satisfaisant les contraintes de la table T et sans la configurations « Z » et « Δ ».

Le chemin le plus simple est d'introduire des tâches fictives f_i par la subdivision de la barre de « Z » ou la base de « Δ » dans G_v selon la figure VI.2, c'est la méthode la plus simple.

Dans ce qui suit, on construit un graphe PERT G_e , avec un nombre minimal de tâches fictives, en appliquant sept règles qu'on verra ultérieurement. Le nouvel algorithme que nous proposons ne tient pas compte de l'existence ou non des arcs redondants. De plus, notre objectif est de minimiser le nombre de sommets et de tâches fictives à la fois.

III. 1 Notations et définitions

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

$G = (X, U)$ 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 is 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 $T(P) = y$.

III. 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 des potentiels G_v est représentée par un arc dans le graphe PERT 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.

Remarquons que le graphe G_{e1} n'est autre que le diagramme de Gantt en lui rajoutant les arcs d'antériorité. Pour plus de détails voir le chapitre VI paragraphe IV.

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 VII. 3).

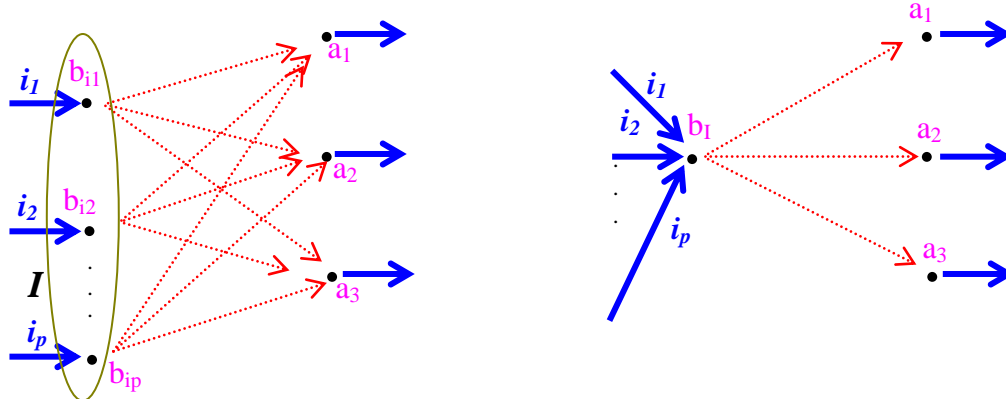


Figure VII.3. Les sommets b_i ayant les mêmes successeurs contractés en un sommet unique b_I dans G_{e1} .

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 VII. 4).

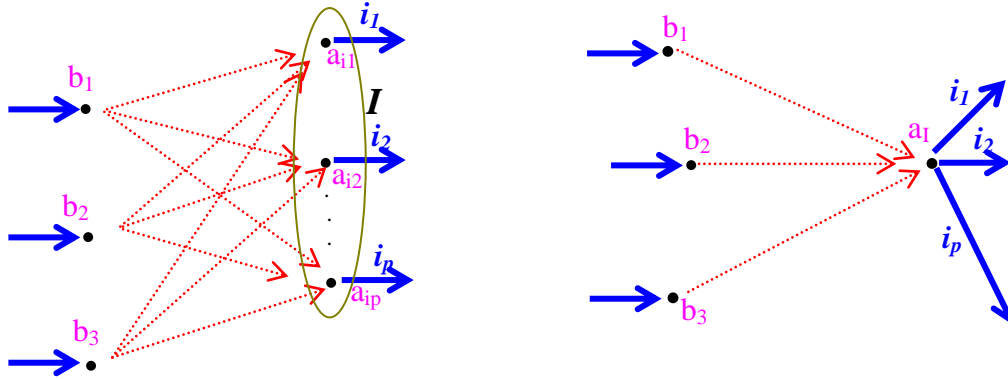


Figure VII.4. Les sommets a_i ont les mêmes prédécesseurs contractés en un sommet unique a_I dans G_{e2} .

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 ≥ 2 de même couleur.

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 $a_i = +1$ ($d_{a_i}^- = +1$).

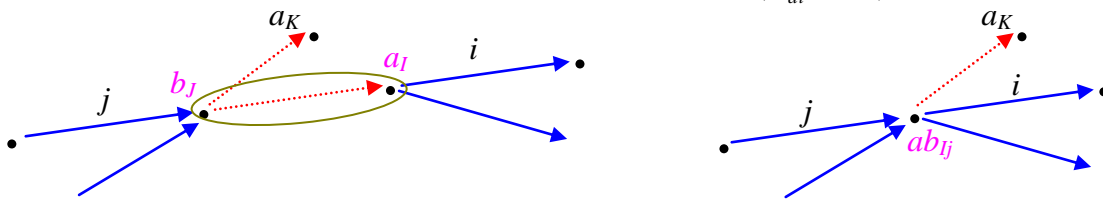


Figure VII.5. Un arc transmetteur de type 1 entre deux tâches i et j et sa contraction.

Si le sommet a_I a un seul prédécesseur b_J dans G_{e3} alors contracter a_I et b_J en un sommet unique ab_{IJ} (voir *Figure VII. 5*).

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 $b_i = +1$ ($d_{b_i}^+ = +1$).

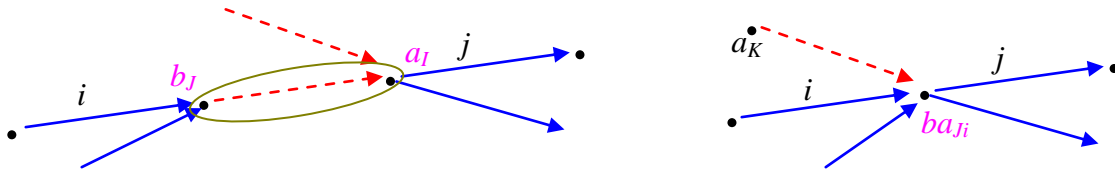


Figure VII.6. Un arc transmetteur de type 2 entre deux tâches i et j et sa contraction.

Si le sommet b_J a un seul successeur a_I dans G_{e4} alors contracter b_J et a_I en un sommet unique ba_{JI} (voir *Figure VII. 6*).

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 ≤ 2
- L'ensemble des sommets de G_{e5} sont partitionnés en quatre types : $a_i, b_i, ab_{IJ}, ba_{IJ}$.

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 rouges 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 VII. 7*).

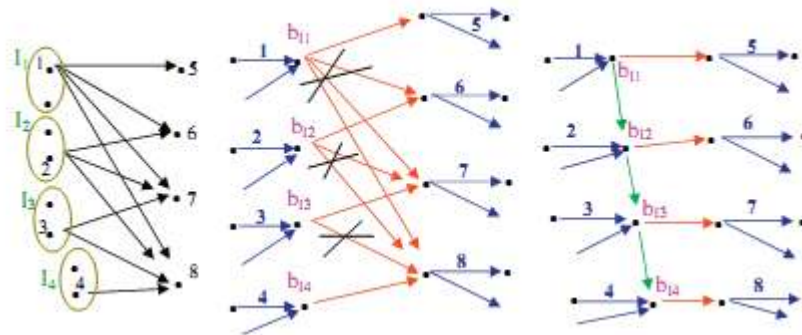


Figure VII.7. Les sommets dont les successeurs sont inclus dans l'ensemble des successeurs d'un autre sommet dans G_{e5} et la réduction des tâches fictives dans G_{e6} .

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

G_{e6} contient des chemins rouges directs de longueur ≤ 2 .

Appliquer à nouveau la règle 3 pour la suppression des arcs transmetteurs qui ont comme sommet d'extrémité terminale a_i . Le graphe obtenu sera appelé G_{e7} .

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 (rouge) de a_I à a_J (voir *Figure VII. 8*).

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

G_{e6} contient des chemins rouges directs de longueur ≤ 2 .

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

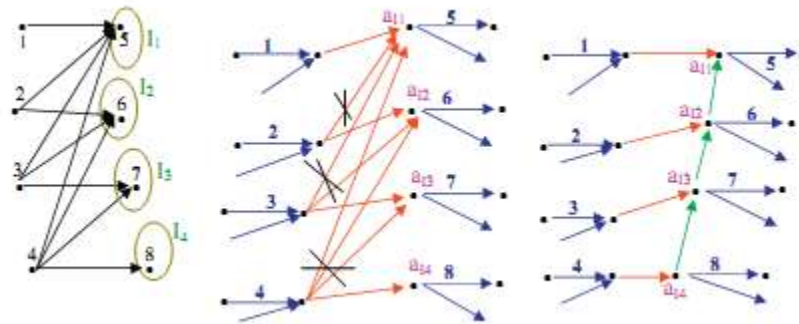


Figure VII.8. Les sommets dont les prédécesseurs sont inclus dans l'ensemble des prédécesseurs d'un autre sommet dans G_{e5} et la réduction des tâches fictives dans G_{e6} .

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

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 comme le montre la figure (Figure VII. 9):

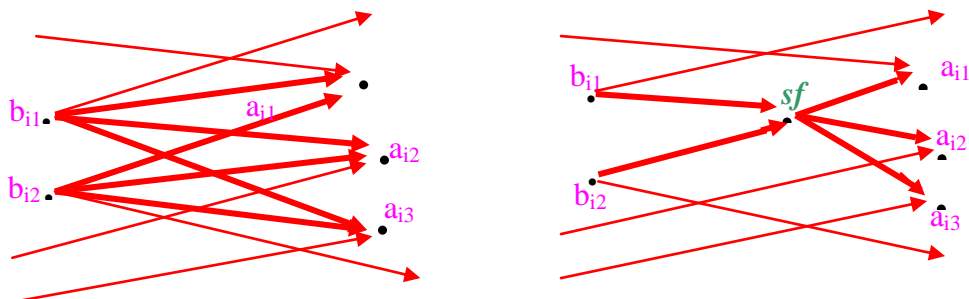


Figure VII. 9. Un biparti complet dans G_{e7} et le sommet fictif sf dans G_{e8} . La représentation dans G_{e8} contient moins d'arcs fictifs que dans G_{e7} .

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.

III.2 Démonstration

Pour montrer que notre algorithme réalise le réseau minimal PERT, laissez-nous envisager un graphe PERT quelconque G_{f1} et à lui appliquons les sept sus citées. Pour des raisons de démonstration appelons ces règles R'_1, R'_2, \dots, R'_7 (voir Figure VII. 10). G_{f8} est le graphe PERT obtenu.

Il faut prouver que le nombre d'arcs virtuels dans G_{f8} est supérieur ou égal au nombre d'arcs factifs dans G_{e8} .

La démonstration est effectuée en deux étapes: lorsque le graphe PERT ne contient pas d'arcs fictifs, donc aucune configuration «Z» ou «Δ» et lorsque le graphe PERT les contient.

Pour arriver à cet objectif, G_e est comparé à un graphe PERT G_f comme suit: G_e est le résultat de réduction des tâches fictives et des sommets dans les différents réseaux PERT $G_{e1}, \dots, G_{e8} = G_e$, par l'application des règles R_2, R_1, \dots, R_7 avec G_{e1} obtenu du diagramme de Gantt défini précédemment.

G_f est également le résultat de réduction des tâches fictives et des sommets dans les différents réseaux PERT $G_{f1}, \dots, G_{f8} = G_f$, par l'application des règles R'_2, R'_1, \dots, R'_7 avec la différence que G_{f1} est un graphe PERT quelconque et contient un nombre inconnu d'arcs factifs.

Nous devons prouver que :

$$|E(G_e)| \leq |E(G_f)|$$

Notons que le R_i et R'_i sont des règles identiques. Les R_i sont utilisés dans l'algorithme de construction du graphe PERT minimal, tandis que les R'_i sont appliquées sur des réseaux G_{fi} pour des raisons de démonstration et pour éviter toute ambiguïté (voir Figure VII. 10).

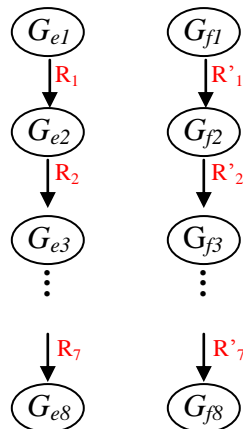


Figure VII.10. Construction de G_e et G_f .

III. 1 Lemme 1

Si $(i, j) \in E(G_v)$ n'est ni barre de 'Z' ni base of 'Δ' dans G_v , alors:

(a) $T(G_{e8}, i) = I(G_{e8}, j)$,

(b) Soit $T(G_{f6}, i) = I(G_{f6}, j)$ soit \exists un graphe PERT G_{f9} qui satisfait la table d'ordonnancement avec $|E(G_{f9})| \leq |E(G_{f8})|$, $|V(G_{f9})| \leq |V(G_{f8})|$ et $T(G_{f9}, i) = I(G_{f9}, j)$.

Preuve

Puisque (i, j) n'est pas une base de 'Δ', $\Gamma^+(G_v, i) \cap \Gamma^-(G_v, j) = \emptyset$ et puisque (i, j) n'est pas une barre of 'Z', tout prédécesseur de j précède tout successeur de i , donc on peut partitionner $\Gamma^-(G_v, j)$ en $I \cup K$ et $\Gamma(G_v, i)$ en $J \cup L$ tel que

I est l'ensemble des sommets qui ont les mêmes successeurs semblables à i ,

$$K = \Gamma^-(G_v, j) - I \text{ et}$$

J est l'ensemble des sommets qui ont les mêmes prédécesseurs semblables j ,

$$L = \Gamma^+(G_v, i) - J.$$

Si $K = L = \emptyset$, alors :

(a) $(b_I, a_J) \in RE(G_{e3})$, contracté par la règle R_3 en un seul sommet ab_{JI} pour obtenir G_{e4} ,

(b) $T(G_{f5}, i)$ and $I(G_{f5}, j)$ sont respectivement la source le puits d'un sous graphe rouge contracté par la règle R'_5 en un seul sommet, appelé ab_{JI} . Put $G_{f9} = G_{f8}$.

Supposons, sans tenir compte de généralisation que $K \neq \emptyset$, alors $\Gamma^+(G_v, i) \subset \Gamma^+(G_v, k)$, pour tout $k \in K$.

Notons K_1, \dots, K_r les sous ensembles de sommets de K avec les successeurs. Si $L \neq \emptyset$, then $\Gamma^-(G_v, j) \subset \Gamma^-(G_v, l)$, pour tout $l \in L$, notons L_1, \dots, L_s les sous ensembles de sommets de L qui ont les mêmes prédécesseurs. Signalons qu'il n'y a pas d'arc de b_I à b_{Kp} , $p = 1, \dots, r$, et aucun arc de a_{Lq} , $q=1, \dots, s$ à a_J , sinon, (i, j) doit être une base de 'Δ' dans G_v .

(a) $\Gamma^+(G_{e5}, b_I) \subset \Gamma^+(G_{e5}, b_{Kp})$, $\forall p = 1, \dots, r$, donc on peut ajouter dans G_{e5} des arcs rouges (b_{Kp}, b_I) et supprimer les arcs rouges (b_{Kp}, a_J) par application de la règle R_3 pour contracter a_J et b_I à ab_{JI} ,

(b) Dans G_{f8} , on considère l'ensemble \mathcal{P} de tous les chemins rouges de b_I à a_J . On verifie que tout arc entrant ou sortant de $V(\mathcal{P})$ fait partie d'un chemin sortant d'un certain b_{Kp} ou entrant d'un certain a_{Lq} , respectivement. Contracter $V(\mathcal{P})$ abouti au graphe PERT network G_{f9} qui satisfait la table d'ordonnancement avec le nombre d'arcs et de sommets moins que ceux de G_{f8} .

5.2.2 Lemme 2.

Si $(i, j) \in E(G_v)$ est une barre de 'Z' ou une base de 'Δ' dans G_v , alors $T(G, i) \neq I(G, j)$ dans tous les graphes PERT G et qui satisfont la table d'ordonnancement.

Preuve

Si $T(G, i) = I(G, j)$ dans un certain graphe PERT G satisfaisant la table d'ordonnancement, alors $\forall p \in \Gamma^-(G_v, j), \forall q \in \Gamma^+(G_v, i), (p, q) \in E(G_v)$, donc $(i, j) \in E(G_v)$ n'est ni barre de 'Z' ni base de 'Δ', contradiction.

5.2.3 Lemme 3

Si $(i, j) \in E(G_v)$ est une barre de 'Z' ou une base de 'Δ' et $(T(G_{e8}, i), I(G_{e8}, j)) \in RE(G_{e8})$, alors $(T(G_{f9}, i), I(G_{f9}, j)) \in RE(G_{f9})$.

Preuve

Partitionnons $\Gamma^-(G_v, j)$ en $I \cup I_1 \cup \dots \cup I_p$, des sommets qui ont les mêmes successeurs avec $I \ni i$; partitionnons $\Gamma^+(G_v, i)$ en $J \cup J_1 \cup \dots \cup J_q$, des sommets qui ont les mêmes prédécesseurs avec $J \ni j$.

Puisque (i, j) est une barre de 'Z', alors il existe $t \in \Gamma^-(G_v, j)$ et $u \in \Gamma^+(G_v, i)$ tel que $(t, u) \notin E(G_v)$, donc $p, q \geq 1$. Supposons sans perte de généralité que $t \in I_1$ et $u \in J_1$, on obtient $(r, s) \notin E(G_v), \forall r \in I_1, \forall s \in J_1$.

Une conséquence du Lemme 2, $T(G_{f9}, I) \neq I(G_{f9}, j)$, donc, on considère l'ensemble P de tous les chemins rouges de $T(G_{e8}, i) = b_I$ à $I(G_{e8}, j) = a_J$. If $\exists k \in [1, p] / \Gamma^+(G_v, i_k) \subset \Gamma^+(G_v, i)$, alors en application de la règle R₅ implique $(T(G_{e8}, i), I(G_{e8}, j)) \notin RE(G_{e8})$, contradiction. Supposons donc que $\forall k \in [1, p], \Gamma^+(G_v, i_k) \not\subset \Gamma^+(G_v, i)$, implique $b_{I_k} \notin V(\mathcal{P})$, et par symétrie, $\forall l \in [1, q], \Gamma^-(G_v, i_l) \not\subset \Gamma^-(G_v, j)$, implique $a_{J_l} \notin V(\mathcal{P})$.

III. 3 Théorème

G_{e8} a le nombre minimum d'arcs fictifs, en effet, selon les précédents lemmes:

$$|E(G_e)| \leq |E(G_f)|$$

III.4. Exemple de projet de construction d'un building industriel

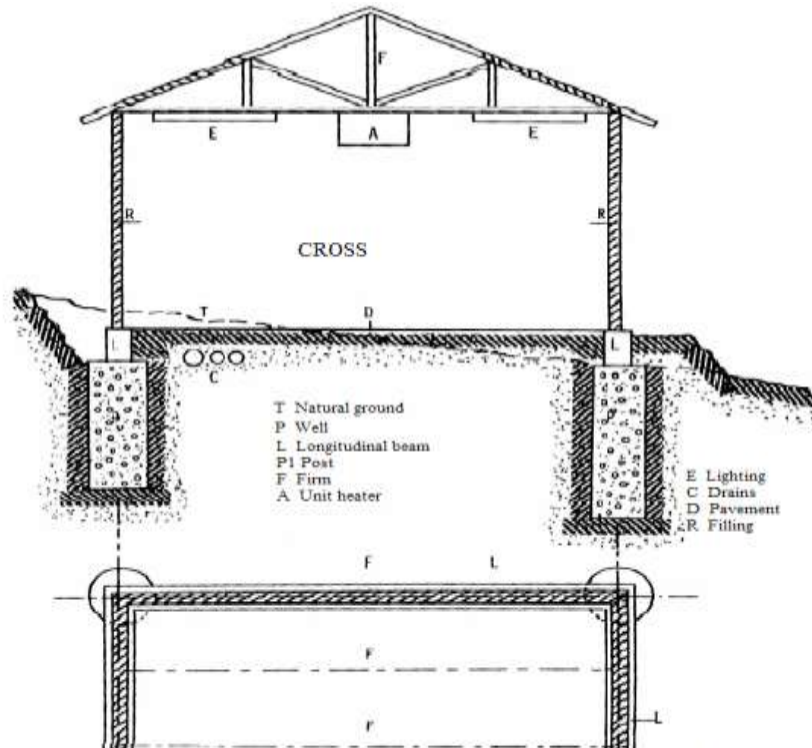


Figure VII.11. Construction d'un building industriel.

Le projet que nous considérons, tiré de [ERN 07], consiste à réaliser un bâtiment industriel comprenant cinq travées et un cadre métallique sur un sol irrégulier incliné (voir *Figure VII.11*).

<i>Activité</i>	<i>Désignation</i>	<i>Prédécesseurs</i>
α	Démarrage du projet	-
1	Étude de chauffage et de cadres	α
2	Scouring and installation of building site establishment	α
3	Décapage et l'installation du chantier	α
4	Terrassement des poutres en béton	α
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
ω	Clôture du projet	11, 12

Table VII.3. Table d'ordonnement de la Figure VII.11.

Les activités impliquées pour la construction de ce bâtiment sont définies dans le tableau VII.3 et le graphe des potentiels correspondant G_v représentant les liens entre les activités du projet est montré dans la *Figure VII.11. (a)*.

G_v doit être connexe, orienté et sans circuit. Notons que les durées des tâches ne sont pas mentionnées sur les différents réseaux.

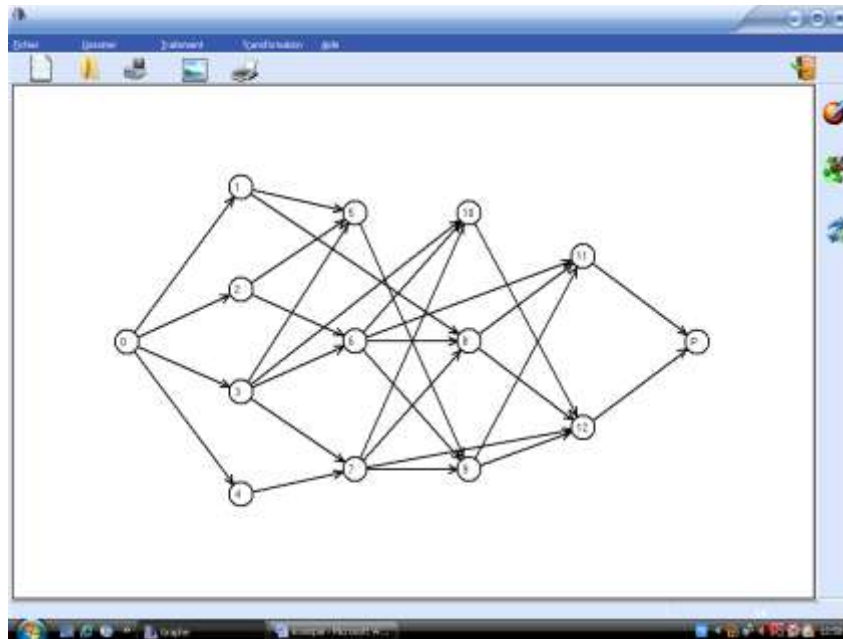


Figure VII.12. (a) Le graphe des potentiels G_v avec 14 sommets de la table d'ordonnancement VII. 3.

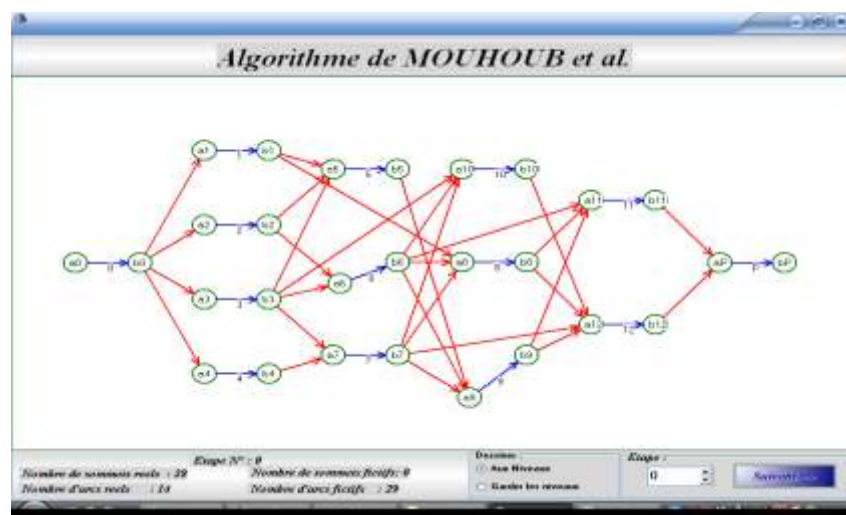


Figure VII.12. (b) Le graphe PERT G_{e1} de la Figure VII.12. (a). G_{e1} contient 29 tâches fictives et 28 sommets.

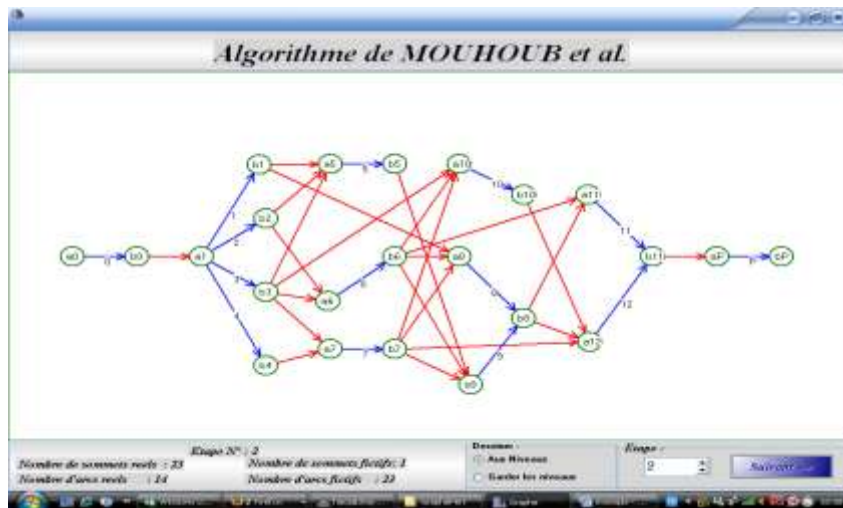


Figure VII.12. (c) Le graphe PERT G_{e3} de la Figure 15. (a) obtenu en appliquant la règle 1 ensuite la règle 2. G_{e3} contient 23 tâches fictives et 23 sommets.

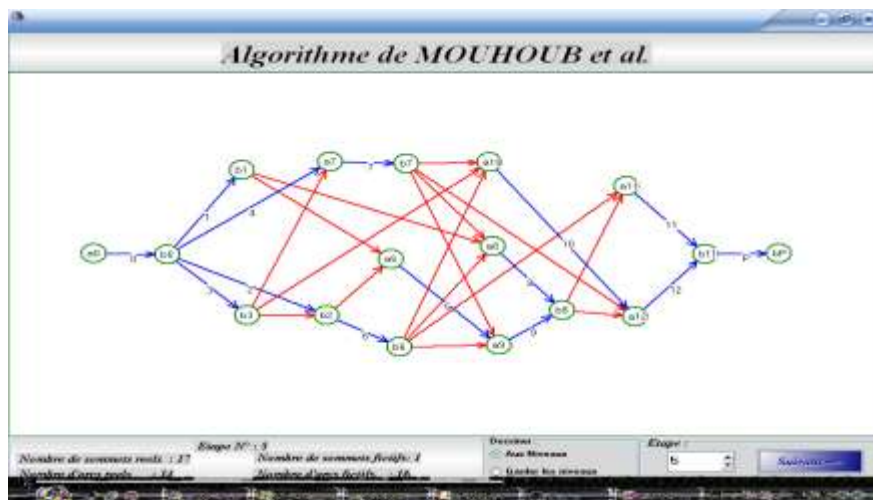


Figure VII.12 (d) Le graphe PERT G_{e5} de la Figure 12. (a) obtenu en appliquant la règle 3 ensuite la règle 4, avec 16 tâches fictives et 17 sommets.

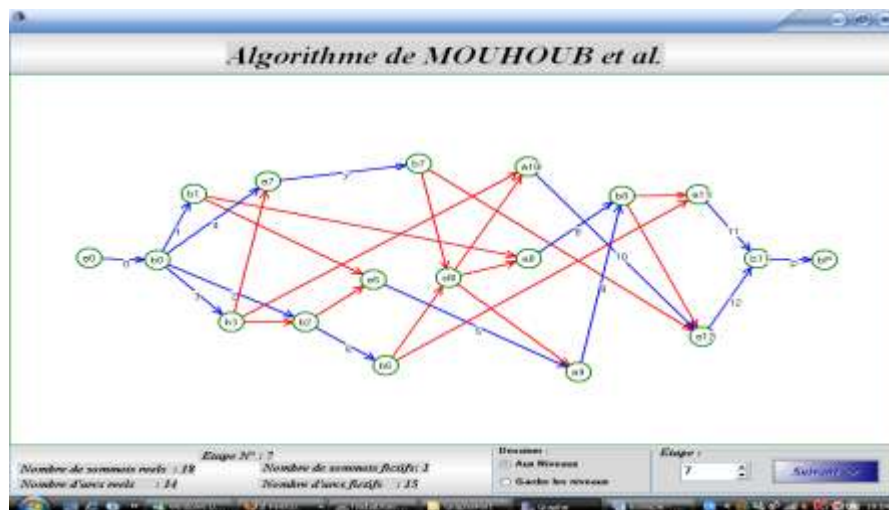


Figure VII.12 (e) Le graphe PERT minimal G_{e8} avec le sommet fictif sf_0 obtenu en appliquant les règles 5, 3 ensuite la règle 7. G_{e8} contient 15 tâches fictives et 18 sommets.

Dans l'exemple présenté (Figure VII.12. (a)), G_{e1} contient $2 | X |$ sommets et les différents chemins sont formés alternativement, d'arcs réels et fictifs (G_{e1} contient 28 sommets et 29 arcs factives). Pour obtenir G_{e1} il suffit d'ajouter au diagramme de Gantt les arcs de précédence (arcs rouges). Tous les réseaux $G_{e1}, G_{e2} \dots, G_{e8}$ satisfont les contraintes de la table d'ordonnement (Tableau VII. 3).

Dans le graphe G_{e7} (Figure VII. 12. (e)), notre algorithme a réduit le nombre de sommets à 18 et le nombre d'arcs fictifs à 15, c'est-à-dire 35,71% de réduction des sommets et 48,27% de réduction des arcs fictifs.

IV. Implémentation

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 P 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 $b_{i_1}, b_{i_2}, \dots, b_{i_p}$ en mêmes successeurs en un seul sommet b_I ;

FIN POUR;

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

I ensemble des tâches qui en les mêmes successeurs.

$I = \Phi$;

POUR P allant de 1 à N **FAIRE**

SI i_p et i_{p+1} en en 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_I ;

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 $d_{a_i}^- = +1$ **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 $d_{b_j}^+ = +1$ **Alors**

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

FIN SI ;

FIN POUR ;

FIN POUR ;

FIN POUR ;

(Construction des graphe 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_i succes $[b_i]$ inclus dans succes $[b_j]$ **FAIRE**

- supprimer les arcs fictifs qui sortent de b_i et qui sont inclus dans les successeurs de b_j .
- remplace 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 prédé $[a_i]$ inclus dans prédé $[a_j]$ **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})

$X_1 = \{ b_1, b_2, \dots, b_p \}$ et $X_2 = \{ a_1, a_2, \dots, a_p \}$

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

- créer un sommet fictif sf ;
- supprimer les arcs multiples ;
(On obtient le graphe G_{e8})

Fin Taque

FIN.

Figure VII.13. Implémentation du nouvel algorithme.

V. Résultats numériques

Nous avons utilisé C++ pour programmer cet algorithme qui est composé de trois modules suivants:

- 1 – Dessin de graphe: Ce module contient les fonctions habituelles de dessins de graphes qui sont : la création et suppression de sommets, la création et la suppression d'arcs, sauvegarde et la restauration des graphes, etc.
- 2 - Bibliothèque: Elle inclut les algorithmes de la théorie des graphes tels que: la connexité d'un graphe, détection d'un circuit, l'organisation du graphe en niveaux, etc. Pour tester le graphe dessiné dans la première étape, il doit vérifier les conditions du réseau (le réseau orienté, connexe, sans circuit, contenant une seule source α et un seul puits ω).
- 3 – A partir du graphe des potentiels, le troisième module comprend l'application de notre algorithme et celui de Sterboul et Wertheimer [SW 80]. Le programme affiche le nombre de sommets et le nombre d'arcs fictifs dans G_{e1} et H_0 . Après avoir lancé le premier algorithme, puis la seconde, le logiciel affiche à la fois G_{emin} et H_2 avec l'affichage du nombre de sommets et d'arcs fictifs dans chaque réseau. Il affiche également le nombre de sommets et d'arcs fictifs réduits.

En utilisant le même processus (obtenir le nombre de sommets et d'arcs fictifs aléatoirement pour tout nouveau réseau G_v), on obtient des données qui sont stockées dans une matrice de 75 par 12 (voir *Tableau VII.4*). Les données contenant une étoile (*) stipule que le graphe contient un ou plusieurs arcs transitifs où l'algorithme de Sterboul et Wertheimer ne fonctionne pas ou bien il donne des résultats erronés.

Nbre de sommets dans $G_{e1} H_0$	Nbre d'arcs rouges dans G_{e1} ou H_0	Nbre de sommets dans H_2	Nbre d'arcs rouges dans H_2	Nbre de sommets supprimés dans H_2	Nbre d'arcs rouges supprimés dans H_2	Nbre de sommets dans G_{emin}	Nbre d'arcs rouges dans G_{emin}	Nbre de sommets supprimés dans G_{emin}	Nbre d'arcs rouges supprimés dans G_{emin}	Meilleur résultat dans la suppression des sommets	Meilleur résultat dans la suppression d'arcs rouges
8	5	6	3	2	3	6	3	2	3	$H_2 = G_{emin}$	$H_2 = G_{emin}$
36	50	12	9	24	41	12	6	24	44	$H_2 = G_{emin}$	G_{emin}
14	10	8	4	6	6	8	4	6	6	$H_2 = G_{emin}$	$H_2 = G_{emin}$
30	25	13	5	17	20	13	5	17	20	$H_2 = G_{emin}$	$H_2 = G_{emin}$
24	22	10	8	14	14	10	5	14	17	$H_2 = G_{emin}$	G_{emin}
24	23*					11	6	13	17	G_{emin}	G_{emin}
40	53*					15	9	25	44	G_{emin}	G_{emin}
22	18*					11	5	11	13	G_{emin}	G_{emin}
52	70	35	51	17	19	35	40	17	20	$H_2 = G_{emin}$	G_{emin}
30	24	13	5	17	19	13	5	17	19	$H_2 = G_{emin}$	$H_2 = G_{emin}$
42	40	21	15	21	35	22	14	20	26	G_{emin}	G_{emin}
26	27	16	16	10	11	16	15	10	12	$H_2 = G_{emin}$	G_{emin}
56	80*					27	23	29	57	G_{emin}	G_{emin}
14	9	8	3	6	6	8	3	6	6	$H_2 = G_{emin}$	$H_2 = G_{emin}$
64	64	29	20	35	44	29	17	35	47	$H_2 = G_{emin}$	G_{emin}
128	175	49	44	79	131	48	40	80	135	G_{emin}	G_{emin}
42	53	19	22	23	31	19	19	23	34	$H_2 = G_{emin}$	G_{emin}
46	55	25	26	21	29	25	22	21	33	$H_2 = G_{emin}$	G_{emin}
58	73	28	27	30	46	28	24	30	49	$H_2 = G_{emin}$	G_{emin}
52	64	28	30	24	34	27	25	25	39	G_{emin}	G_{emin}
68	91	34	39	34	52	34	34	34	57	$H_2 = G_{emin}$	G_{emin}
54	67	33	37	21	30	32	31	22	36	G_{emin}	G_{emin}
40	47*					25	24	15	23	G_{emin}	G_{emin}
76	92	23	11	53	81	23	11	53	81	$H_2 = G_{emin}$	$H_2 = G_{emin}$
90	105	54	49	36	56	52	45	37	60	G_{emin}	G_{emin}
94	129	52	62	42	67	50	49	44	80	G_{emin}	G_{emin}
82	123	35	32	47	91	32	27	50	96	G_{emin}	G_{emin}
62	87	34	41	28	46	34	31	28	56	$H_2 = G_{emin}$	G_{emin}
36	40*					17	13	19	27	G_{emin}	G_{emin}
324	388*					164	191	160	197	G_{emin}	G_{emin}
348	392	178	223	170	169	178	218	170	174	$H_2 = G_{emin}$	G_{emin}
170	189	114	113	56	76	113	110	57	79	G_{emin}	G_{emin}
28	29	17	17	11	12	17	15	11	14	$H_2 = G_{emin}$	G_{emin}
66	67	25	18	41	49	23	15	43	52	G_{emin}	G_{emin}
38	57*					14	11	24	46	G_{emin}	G_{emin}
40	54	15	8	25	46	14	8	26	46	G_{emin}	$H_2 = G_{emin}$
26	26	15	14	11	12	15	13	11	13	$H_2 = G_{emin}$	G_{emin}
42	41*					22	14	20	27	G_{emin}	G_{emin}
14	9	9	3	5	6	8	3	6	6	G_{emin}	$H_2 = G_{emin}$
28	30	17	18	11	12	17	15	11	15	$H_2 = G_{emin}$	G_{emin}
200	313	88	111	112	202	84	85	116	228	G_{emin}	G_{emin}
74	124*					28	23	44	101	G_{emin}	G_{emin}
148	219	41	33	107	186	39	29	109	190	G_{emin}	G_{emin}
70	97	45	70	25	27	45	60	25	37	$H_2 = G_{emin}$	G_{emin}
60	77	37	53	23	24	37	44	23	33	$H_2 = G_{emin}$	G_{emin}
40	45	22	19	18	26	20	17	20	28	G_{emin}	G_{emin}
278	304*					176	179	102	125	G_{emin}	G_{emin}
84	99	54	64	30	35	54	52	30	47	$H_2 = G_{emin}$	G_{emin}
66	77	39	47	27	30	38	34	28	33	G_{emin}	G_{emin}
84	96*					41	32	43	64	G_{emin}	G_{emin}
286	307	161	208	125	99	161	204	125	103	$H_2 = G_{emin}$	G_{emin}
90	107	37	31	53	76	37	27	53	80	$H_2 = G_{emin}$	G_{emin}
68	85	38	43	30	42	38	35	30	50	$H_2 = G_{emin}$	G_{emin}
66	71	43	42	23	29	40	36	26	35	G_{emin}	G_{emin}
66	70*					41	38	25	32	G_{emin}	G_{emin}
78	86	45	44	33	42	45	39	33	47	$H_2 = G_{emin}$	G_{emin}
46	56*					26	24	20	32	G_{emin}	G_{emin}
118	126	66	62	52	64	66	57	52	74	$H_2 = G_{emin}$	G_{emin}
82	92	40	33	42	59	39	29	43	63	G_{emin}	G_{emin}
78	105*					38	36	40	69	G_{emin}	G_{emin}
68	87	40	48	28	39	40	38	28	49	$H_2 = G_{emin}$	G_{emin}
70	89	44	52	26	37	42	41	28	48	G_{emin}	G_{emin}
190	233	108	151	82	82	108	141	82	92	$H_2 = G_{emin}$	G_{emin}
177	191	99	109	78	82	96	109	81	82	G_{emin}	$H_2 = G_{emin}$
205	267*					97	145	108	122	G_{emin}	G_{emin}
245	280	178	161	67	119	178	159	67	121	$H_2 = G_{emin}$	G_{emin}
159	208	82	139	77	69	79	136	80	72	G_{emin}	G_{emin}
121	175	71	104	50	71	71	99	50	76	$H_2 = G_{emin}$	G_{emin}
180	210*					103	99	77	111	G_{emin}	G_{emin}
350	372	203	219	147	153	201	209	149	163	G_{emin}	G_{emin}

Table VII.4. Tableau récapitulatif de l'application de l'algorithme de Mouhoub et Sterboul sur 75 graphes.

L'étude de ce tableau donne les conclusions suivantes:

Nombre de réseaux	Meilleur rapport de réduction de sommets (%)			Meilleur rapport de réduction des arcs fictifs (%)		
	H_2	$G_{emin} = H_2$	G_{emin}	H_2	$G_{emin} = H_2$	G_{emin}
75	0%	46.66%	53.33%	0%	13.33%	86.66%

Table VII.5. Résultats de la comparaison.

Nous venons de voir un nouvel algorithme de construction du graphe PERT minimal qui a été comparé avec l'algorithme similaire de Sterboul. Le nouvel algorithme mis au point est meilleur avec plus de 53% dans le cas de réduction des sommets et plus de 86% dans le cas de réduction des arcs fictifs (voir *Tableau VII. 5*). Il s'agit d'une amélioration importante qui se traduit par la production facile, la lecture et la compréhension des réseaux PERT, en plus de la simplification de l'exécution de commande des projets. Il est important de garder à l'esprit que l'algorithme de Sterboul et Wertheimer n'est meilleur dans aucun des deux cas. En outre, notre algorithme tourne avec l'existence d'arcs transitifs tandis que l'autre échoue selon d'après ce qui a été écrit dans leur papier [SW 80].

VI Conclusion

Ce chapitre présente une nouvelle approche pour générer des réseaux PERT avec un nombre minimum d'arcs fictifs. En effet, en appliquant les sept règles R_1, R_2, \dots, R_7 dans l'ordre séquentiel, l'algorithme permet de réduire un certain nombre d'arcs fictifs jusqu'à la dernière règle qui donne le graphe PERT minimal avec le respect total des contraintes dans la table d'ordonnement en utilisant certains résultats des graphes adjoints.

L'algorithme a été programmé en C++ Builder et par rapport à l'algorithme de Sterboul, il s'avère plus optimal soit en terme de réduction du nombre d'arcs fictifs soit en terme de réduction du nombre de sommets.

Notre nouvelle approche est très simple à appliquer. Elle donne le PERT 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, même lorsque les réseaux sont d'une très grande taille.

Un autre avantage majeur, qui est intéressant, et mérite d'être mentionné est le fait que l'algorithme fonctionne sans aucun problème en présence des arcs transitifs.

Chapitre VIII

Conclusions et perspectives

Sommaire

I Introduction	136
II Bilan	136
III Perspectives	138

I Introduction

Les problèmes d'ordonnement de projet figurent au rang des problèmes d'ordonnement les plus étudiés depuis les origines de la théorie de l'ordonnement. Malgré les nombreux progrès réalisés par la suite, ce mémoire montre que ces problèmes, en dépit de leur simplicité apparente, constituent un challenge pour la recherche opérationnelle actuelle.

Depuis leur début d'utilisation vers la fin des années 50, PERT/CPM et MPM ont été employés intensivement pour aider les chefs de projet dans la planification, la programmation, et le commandement de leurs projets. Avec le temps, ces deux techniques graduellement ont bénéficié des avantages des uns et des autres.

L'application de PERT/CPM commence en cassant le projet vers le bas en ses différentes activités, en identifiant les prédécesseurs immédiats de chaque activité, et en estimant la durée de chaque activité. Un réseau de projet alors est construit pour montrer visuellement toute cette information. Le type de réseau qui devient de plus en plus populaire à cette fin est le réseau des potentiels, où chaque activité est représentée par un noeud.

II Bilan

PERT/CPM et MPM produisent beaucoup d'informations d'établissement du programme qui sont utiles pour le chef de projet, y compris la date de début au plus tôt, la date de début au plus tard ainsi que les différentes marges pour chaque activité. Elle identifie également le chemin critique des activités tel que tout retard touchant une des activités le long de ce chemin

retarde l'accomplissement du projet. Puisque le chemin critique est le plus long chemin dans le réseau, sa longueur détermine la durée totale du projet, en supposant que toutes les activités s'exécutent dans les délais.

Cependant, il est difficile que toutes les activités restent dans les délais parce qu'il y a fréquemment des incertitudes considérables au sujet de ce qu'elle sera la durée d'une activité. PERT à trois estimations présente une approche qui mérite toute l'admiration. Elle offre trois genres différents d'évaluations (optimistes, pessimistes, et les plus probables) pour la durée de chaque activité et à partir de là, en utilisant une distribution de probabilité, la méthode calcule la durée pour une activité, ce qui est appliqué ensuite pour chaque activité du projet.

PERT/CPM a quelques insuffisances importantes. Celles-ci incluent les approximations incertaines faites en estimant les durées des activités. Ces erreurs qui entachent le calcul des durées ont été étudiés, dans cette thèse, le sujet d'une analyse suivie d'une critique ensuite d'un ensemble de suggestions qui peuvent apporter des corrections à cette merveilleuse méthode.

La méthode PERT-coût permet au chef de projet d'étudier l'effet sur le coût total de changer la durée estimée du projet en diverses valeurs alternatives. Les données importantes pour chaque activité sont le temps et le coût lorsqu'elles sont faites de la manière normale et puis quand elles sont entièrement accélérées.

Néanmoins, le graphe PERT/CPM reste difficile à dessiner en présence des tâches fictives qu'il engendre et qui représente plus de 40% des arcs du graphe. Notre travail s'est focalisé sur le sujet et nous a offert deux nouvelles méthodes de dessin du graphe. La première est une technique qui balaye la table d'ordonnancement ligne par ligne et à chaque fois elle ajoute un arc au graphe construit au fur et à mesure. La présence de tâche fictive sera vite détectée et introduite dans le graphe.

La deuxième méthode utilise un certain ensemble de concepts des graphes adjoints de graphes pour dessiner, à partir du graphe des potentiels, et à travers une série de transformations, le graphe PERT. L'idée consiste : en la recherche des sous graphes sous la forme de « Z » dans le graphe des potentiels, éliminer ces « Z » en introduisant des arcs fictifs supprimant toute barre de « Z ». Les bipartis complets sont ensuite localisés dans le graphe des potentiels. Chaque biparti représente, dans le nouveau graphe qui est appelé PERT, une étoile avec un sommet au milieu recevant et donnant des arcs incidents intérieurement et extérieurement. Le graphe est dessiné au fur et à mesure, étoile par étoile, et à la fin on obtient un graphe PERT respectant les contraintes figurant dans la table d'ordonnancement.

Cette nouvelle technique originale a réussi (pour la première fois) à introduire les graphes adjoints dans les problèmes d'ordonnancement de projet. Elle offre un algorithme simple avec une complexité polynomiale.

Cet algorithme est revu à dans le chapitre VI en vue d'une optimisation dans le nombre des tâches fictives se trouvant dans le graphe PERT. Deux approches ont été présentées. La première celle de [AKR 07] se basant sur le principe de regroupement des arcs fictifs appartenant au même biparti complet. Or, le graphe ne contient pas toujours des bipartis contenant des arcs fictifs. Il fallait chercher un autre moyen pour réduire les tâches fictives. Une deuxième approche venait donc, pour apporter une solution nouvelle à ce problème embêtant pour les managers de projet. Une étude comparative a été réalisée entre les deux approches. Cette étude montre la supériorité de notre algorithme dans la majorité des cas.

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

Notre technique de réduction nous a servi comme une ébauche à une méthode nouvelle plus concise englobant tous les cas possibles.

La présentation de cette méthode a été suivie d'une démonstration prouvant qu'elle offre le graphe PERT minimal en termes de tâches fictives.

Notre recherche ne s'est pas arrêtée à ce stade. La thèse présente un des algorithmes célèbres et qui très connu dans la littérature. En le comparant avec notre nouvelle méthode, les résultats étaient très satisfaisants. Ils sont en notre faveur.

III Perspectives

Il serait dommage de s'arrêter là. De nombreuses pistes, pouvant apporter une véritable aide, restent à explorer. Voici ci-dessous celles qui nous semblent les plus prometteuses.

Le PERT flou très peu connue des utilisateurs contrairement à son analogue stochastique pour lequel il existe de nombreux programmes commerciaux, est un axe de recherche très attirant.

Nous avons l'idée que le PERT flou fait appel à moins d'hypothèses de calcul que le PERT stochastique, qu'il est également plus simple à utiliser, ne nécessitant aucun calcul de moyenne ou variance, et que contrairement au PERT stochastique, il permet de tenir compte des changements de chemin de durée maximale en fonction de l'incertitude sur la durée totale.

C'est ce dernier point qui le démarque le plus du PERT stochastique.

Par beaucoup d'aspects, comme la fidélité à la réalité sachant qu'une estimation de durées est vague et non probabiliste, la commodité d'utilisation et la qualité des résultats, le modèle du PERT flou est plus proche de la réalité et plus efficace qu'un modèle stochastique.

Le PERT flou répond aux conditions de planning “tendu”, ce qui n’est pas le cas du PERT stochastique qui allonge la durée des tâches. Toutefois, cette méthode du PERT flou ne permet pas (ou pas encore) de déterminer des marges de temps dans tous les cas, ce problème étant particulièrement difficile à résoudre et toujours en cours de recherche.

Nous savons que les contraintes potentielles sont de deux types : les contraintes d’antériorité et les contraintes de localisation temporelle. Le premier type a été pris en charge dans cette thèse mais le second, reste à conquérir. La modélisation par les graphes peut résoudre ce cas. Elle en a les moyens nécessaires pour offrir une solution globale concernant la transformation d’un graphe des potentiels (même avec les contraintes de durée) en un graphe PERT.

Un autre axe de recherche qui mérite la prise en charge, c’est celui de l’ordonnancement répétitif dans un projet. En effet, il ne faut jamais oublier que quelque soit la taille d’un projet, il existe des tâches qui se répètent jusqu’à n fois lors de l’exécution. Jusqu’à présent, on ne tient pas compte de ce cas puisque le graphe obtenu ne doit pas contenir de circuits (une tâche ne peut pas succéder à elle-même). Avec la règle 7 que nous avons vu précédemment par la création d’un sommet fictif (voie chapitre VII), ce problème peut être pris en charge mais ça nécessite une étude approfondie pour le cas général.

Un dernier axe qui me paraît intéressant est celui de la modélisation par les graphes d’un projet, en tenant en compte la présence des ressources qui sont d’ordre limité. Nous savons que la théorie des graphes tire toutes ses forces par sa représentation binaire, ce qui est permis actuellement, c’est la modélisation par un graphe disjonctif mais je reste quand même optimiste pour la recherche d’une solution radicale à ce type de problème en utilisant cet outil puissant qui est : la théorie des graphes.

Bibliographie

A

[AAC 75] J. ABDELNOUR, A. H-S ANG, & A.A. CHAKER, Analysis of activity networks under uncertainty, *Journal of the Engineering Mechanics Division, ASCE*, 101, 373-387, 1975.

[ABCCG, 06] K. ALTISEN, P. BOUYER, T. CACHAT, F. CASSEZ & G. GARDEY, Introduction au contrôle des systèmes temps-réel, *Site web* : <http://www.lsv.ens-cachan.fr/aci-cortos/>, 2006.

[AEH, 96] M. K. Agarawal, S. E. Elmaghraby, W. S. Herroelen, DAGEN a generator for test sets for project activity nets, *European Journal of Operational Research*, 90, pp.376-382, 1996.

[AKR 07] S. AKROUF, Approche optimisée dans les problèmes d'ordonnancement, ANL'07, Colloque international sur l'analyse non linéaire, Université de Sétif, ALGERIE, 2007.

[ARM 01] J.S. ARMSTRONG, The forecasting dictionary. Principles of forecasting: A handbook for researchers and practitioners. Ed. J. S. Armstrong. Boston, Kluwer Academic Publishers, p. 761-824.

[AS 94] R. J. AARTS & S. F. SMITH, A high performance scheduler for an automated chemistry workstation, In [Eib94], pages 3-7, 1994

[AV 67] R.D. ARCHIBALD, & R.L. VILLORIA, *Network-based management systems (PERT/CPM)*, New York: John Wiley & Sons, 1967.

B

[BAC 00] B. BACHELET, Recherche opérationnelle, version numérisée, 2001.

[BBK 98] T. BAAR, P. BRUCKER & S. KNUST, Tabu-search algorithms and lower bounds for the resource constrained project scheduling problem. In S. Voss, S. Martello, I. Osman, and C. Roucairol, editors, *Meta-heuristics : advances and trends in local search paradigms for optimization*, pages 1–18. Kluwer, 1998.

[**BDF 97**] J. C. BECK, A. DAVENPORT & M. S. FOX, Five pitfalls of empirical scheduling research, 1997. *Principles and Practice of Constraint Programming*.

[**BEN 07**] I. BENKERMI, Modèle et algorithme d'ordonnement pour architectures reconfigurables dynamiquement. Thèse, Université de RENNES 1, Janvier 2007.

[**BEN 01**] A. BENHOCINE, Théorie des graphes et applications, Cours de post-graduation, Université de Sétif, 2001.

[**BEPSW 96**] J. BLAZERWICZ, K.H. ECKER, E. PESCH, G. SCHMIDT & J. WEGLARZ, Scheduling in computer and manufacturing processes. Springer Verlag, 1996.

[**BIM 07**] F. BIMBARD, *Dimensionnement temporel de systèmes embarqués : application à OSEK*. PhD thesis, Conservatoire National des Arts et Métiers, 2007.

[**BRU 93**] R. BRUNS, Direct chromosome representation and advanced genetic operators for production scheduling, In [For93], pages 352-359, 1993.

[**BDMNP 99**] P. BRUCKER, A. DREXEL, R. MOHRING, K. NEUMANN & E. PESCH, Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, vol. 112, pages 36-41, 1999.

[**BOR 07**] K. M. BORGWARDT, *Graph Kernels*, PhD thesis, Faculté de Mathématique, d'Informatique et de Statistique de l'université de Ludwig Maximilians, Munich, 2007.

[**BUM 91**] S. BAGCHI, S. UCKUN, Y. MIABE, & K. KAWAMURA, Exploring problem-specific recombination operators for job shop scheduling, In [BB91], pages 10-17, 1991.

C

[**CC 97**] J. CARLIER & P. CHRETIENNE, Problèmes d'ordonnement, modélisation, complexité, algorithmes, MASSON, ISBN 2-225-81275-6.

[**CDB 10**] CARDOEN, E. DEMEULEMEESTER & J.BELIEN, Operating room planning and scheduling: A literature review, *European Journal of Operational Research* 201 : 921-932, 2010.

- [CDF 92] Y. CRAMA, L. DUPONT & G. FINKE, Recherche opérationnelle et gestion de la production, Nouvelles de la science et des technologies, 1997.
- [CFT 09] A.CAUVIN, A. FERRARINI, E. TRANVOUEZ, Disruption management in distributed enterprises: A multiagent modelling and simulation of cooperative recovery behaviours, *International Journal of Production Economics*, 122 : 429-439, 2009.
- [CGH 96] I. CHARON, A. GERMA & O. HUDRY, Méthodes d'optimisation combinatoire, Masson, Paris, France, 1996.
- [CGL 73] 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.
- [CGP 1998] J. COUILLARD, D. GODCHARLES & J-P PAQUIN, La qualité acquise: pour un meilleur contrôle des projets, *Revue Internationale en gestion et management de projet*, vol 3, n0 1, 46-59, 1998.
- [CGR 08] R. CASSAGNE, M. GOURGAND, & S. RODIER, Un outil d'aide au dimensionnement, à la planification et à la visualisation d'un programme opératoire. In *Gestion et Ingénierie des Systèmes Hospitaliers (GISEH)* : 807-814. Lausanne, Suisse, 4-6 septembre 2008.
- [CH 08] S-T. LO, R-M. CHEN, Y-M. HUANG & C-L. WU, Multiprocessor system scheduling with precedence and resource constraints using an enhanced ant colony system. *Expert Syst. Appl.*, 34(3):2071.2081, 2008.
- [CHA 97] F.T.S. CHAN, Resource management in project scheduling through simulation, *International Journal of Computer Applications in Technology*, vol. 10, no1 / 2, 81 – 89, 1997.
- [CHE 92] H. CHETTO, Ordonnancement tolérant les fautes pour les applications temps réel, MOAD'92, Vol. 1, pages 67-74, Université de Bejaia, ALGERIE, 1992
- [CHU 95a] C. CHU, A new class of scheduling criteria and their optimization, 1995.
- [CHU 95b] C. CHU, Ordonnancement de la production: Approches théoriques nouvelles, Thèse d'habilitation à diriger des recherches, Université de Metz, Metz, France, Juillet 1995.
- [CMGL 08] S. CHAABANE, N. MESKENS, A. GUINET & M. LAURENT, Comparison of two methods of operating theatre planning : application in belgian hospital, *Journal of Systems Science and Systems Engineering* 17(2) : 171-186, 2008.

[COF 76] E.G. COFFMAN, Computer and job-shop scheduling theory, John Wiley and Sons, 1976.

[CS 07] Y. Cohen, A. Sadeh, A New Approach for Constructing and Generating AOA Networks, Journal of computer science, 1-1, 2007.

<http://www.scientificjournals.org/journals2007/articles/1049.htm>

[CP 96] C. CHU & J. M. PROTH, L'ordonnancement et ses applications, MASSON, 1996, ISBN 2-225-85193-X.

[CS 89] G. A. CLEVELAND, & S. F. SMITH, Using genetic algorithms to schedule flow shop releases. In [Sch89], pages 160-169, 1989.

[CT 94] H. M. CARTWRIGHT & A. L. TUSON, Genetic algorithms and flow shop scheduling: towards the development of a real time process control system, In [FOG 94a], pages 277-290, 1994.

D

[DDH 93] E. Demeulemeester, B. Dodin, and W. Herroelen, A random activity network generator, *Operations Research*, 41 (5), pp. 972-980, 1993.

[DIE 00] R. DIESTEL, Graph theory, Springer-Verlag, 2000.

[DEM 01] S. DEMASSEY, Méthode exacte pour l'ordonnancement de projets à moyens limités, Laboratoire d'informatique d'Avignon, 2001.

[DIM 63] D. Dimsdale, Computer construction of minimal project network, *IBM systems journal*, 2, 24-36, 1963.

[DOR 10] F DORIN, Contributions à l'ordonnancement et l'analyse des systèmes temps réel critiques, Thèse de doctorat, ECOLE NATIONALE SUPERIEURE DE MECANIQUE ET D'AEROTECHNIQUE, 2010.

[DP 01] J. M. DALBARADE & B. PETIT-JEAN, Ordonnancement et gestion des projets, MASSON, Paris, 2001.

[DUM 92] J. DUMOND, In a multi-resource environment, how much is enough, *International Journal of Production research*, Vol. 30, N°. 2, pp. 395-410, 1992.

[DUV 00] D. DUVIVIER, Etude de l'hybridation des méta-heuristiques, application à un problème d'ordonnancement de type job shop, thèse de Doctorat, Université de Littoral Côte d'Opale, 2000.

E

[ECO 11] F. ECOTO, Initiation à la recherche opérationnelle, ELLIPES, Paris, France, 2011.

[EH 80] S. E. Elmaghraby, and W. S. Herroelen, On measurement of complexity in activity networks, *European Journal of Operational Research*, 5, pp.223-234, 1980.

[ERN 07] M. E. Ernould, Comparaison des approches PERT stochastique et PERT flou, application à un projet industriel, Thèse, université libre de Bruxelles, 2007.

[ESS 03] C. ESSWEIN, Un apport de flexibilité séquentielle pour l'ordonnancement robuste, Mémoire de thèse, Université de Tours, décembre 2003.

[EL 99] P. ESQUIROL & P. LOPEZ, l'ordonnancement, ECONOMICA, Paris, France, ISBN 2-7178-3798-1.

F

[FIN 02] G. FINK, Recherche opérationnelle et réseaux, Lavoisier, Paris, 2002.

[FIS 68] A.C. Fisher, D.S. Lieberman, and G.L. Nemhauser, Computer construction of project networks, *Communications of the ACM*, 11, pp.493-497, 1968.

[FRC 93] H. FANG, P. ROSS & D. CORNE, A promising hybrid genetic algorithm approach to job-shop scheduling, rescheduling, and open shop scheduling problems In [For 93], pages 488-493, 1993.

[FSKM 03] T. FOSS, E. STENSRUD, B. KITCHENHAM & I. MYRTVEIT, A Simulation Study of the Model Evaluation Criterion MMRE, *IEEE Transactions on Software Engineering*, November 2003.

G

[GIA 91] V. GIARD, Gestion des projets, Economica, Paris, 1991.

[GIA 03] V. GIARD, Gestion de la production et des flux, 3^e édition, Economica, Paris, 2003.

[GJ 79] Michael R. GAREY & D. JHONSON, Computers and intractability; A guide to the theory of NP-Completeness, W.H. Freeman and company, 1979, ISBN: 0-7167-1045-5.

[GM 96] V. GIARD & C. MIDLER, Management et gestion des projets : bilan et perspectives, IAE, Paris, 1996.

[GOT 93] A. GOTH, Groupe d'ordonnancement théorique et appliqué, Les problèmes d'ordonnancement, *Operations research*, 27(1) : 77-150, 1993

[GLL 79] R.L. GRAHAM, E.L. LAWLER, J.K. LENSTRA & A.H.G. RINNOO KAN. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, vol. 5, pages 287-326, 1979.

[GOL 88] D. GOLENKO-GINZBURG, On the distribution of activity time in PERT, *Journal of Operational Research Society*, XXXIX, vol. 39, n°8, 767 –771, 1988.

[GS 78] T. GONZALEZ & S. SAHNI, Flow shop and Job shop schedules: complexity and approximation, *Operations research*, 26(1): 36-52, 1978.

[GUI 70] M. GUIGNARD, Ordonnancement, Laboratoire de calculs, Université de Lille, France, 1970.

H

[HABT 07] A. HAIT, C. ARTIGUES, P. BAPTISTE, M. TREPANIER, Ordonnancement sous contraintes d'énergie et de ressources humaines. In: 11e congrès de la Société Française de Génie des Procédés, 9-11 Oct 2007, Saint-Etienne, France.

[HAY 69] M. Hayes, The role of activity precedence relations in node-oriented networks. *In H.J.M. Lombaers (ed.), Project planning by network analysis*, North-Holland Publishing Company, Amsterdam, pp.128-146, 1969.

- [**HCDJ 08**] P-E. HLADIK, H. CAMBAZARD, A-M. DEPLANCHE, & N. JUSSIEN, Solving a real-time allocation problem with constraint programming. *J. Syst. Softw.*, 81(1):132-149, 2008.
- [**HD 90**] W. HAN et P. DEJAX, Une heuristique pour le problème d'ordonnancement de type $n/m//F/C_{max}$ avec la présence de machines goulots, *Operations research*, 24(4) : 315-330, 1990.
- [**HDD 99**] W. HERROELEN, E. DEUMEULEMEESTER & B. DE REYCK, A classification scheme for project scheduling. Dans *Project Scheduling – Recent Models, Algorithms and Applications* sous la direction de J. Weglarz, pages 1-26, Kluwer's International Series, 1999.
- [**HB 83**] R. L. HEMMINGER & L. W. BEINEKE, Line graphs and line digraphs, in *Selected topics in graph theory I* (1983), 270-304.
- [**HEN 08**] C. HENDERCKSON, *Project Management for Construction*, Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, Version 2.2, 2008.
- [**HH 97**] K. W. HANSMANN & M. HOEK, Production control of a flexible manufacturing system in a job shop environment, *International Transactions in Operational Research*, 4(5/6): 341-351, Sep/Nov 1997, ISSN: 0969-6016.
- [**HIL 04**] HIL, *Project Management with PERT/CPM*, Holden-Day, San Francisco, 2004.
- [**HL 80**] F.S. HILLIER, G.J. LIEBERMAN, *Operations Research*, 3rd ed., Holden-Day, San Francisco, 1980.
- [**HLA 04**] P. E. HLADIK, Ordonnabilité et placement des systèmes temps réel distribués préemptifs et à priorités fixes, Thèse de Doctorat, Université de Nantes, 2004.
- [**HOA 05**] T. HOANG, Utilisation d'ordres partiels pour la caractérisation de solutions robustes en ordonnancement, Thèse de Doctorat, Institut national des sciences appliquées de Toulouse, 2005.
- [**HKSW 94**] S.V. HOESEL, R. KUIK, L. M. SALOMON & N.V. WASSENHOVE, The single-item discrete lot sizing and scheduling problem: Optimization by linear and dynamic programming, *Discrete Applied Mathematics*, 48(3), 289-303, 1994.

[HSS 92] A HATCHUEL, D. K. SAIDI & J. C. SARDAS, Planification des productions complexes : une approche hiérarchisée et dynamique, MOAD'92, Vol. 1, pages 51-57, Université de Bejaia, ALGERIE, 1992.

[HW 96] A. HERTZ, & M. WIDMER, An improved tabu search approach for solving the job shop scheduling problem with tooling constraints, Discrete Applied Mathematics, 65: 319-345, 1996.

I

[IHI 96] M. MC ILHAGGA, P. HUSBANDS & R. IVES, A comparison of optimization techniques for integrated manufacturing planning and scheduling, In [VERS96], pages 604-613, 1996.

[IKA 04] L. A. IKA, L'analyse de la valeur acquise en contexte d'interdépendance des chemins : une analyse à explorer, Université du Québec en Outaouais, ASAC, 2004.

[IZU 90] J. IZUCHUKWU, Shortening the critical path, *Mechanical Engineering*, vol. 112, n° 2, 59-60, 1990.

J

[JOR 04a] M. JORGENSEN, A Review of Studies on Expert Estimation of Software Development Effort, Journal of Systems and Software, Vol. 70 Nr. 1-2, 2004.

Online reference:

http://www.simula.no/departments/engineering/.artifacts/expertsubmitnovember2002_copy.pdf

[JTM 04b] M. JORGENSEN, K. H. TEIGEN & K. MOLOKKEN-OSTVOLD, Better sure than safe? Overconfidence in judgment based software development effort prediction intervals, Journal of Systems and Software, Vol. 70 Nr. 1-2, 2004.

Online reference:

http://www.simula.no/departments/engineering/.artifacts/pireviseddesember_copy.pdf

[JOR 04c] M. JORGENSEN, Realism in Assessment of Effort Estimation Uncertainty: It Matters How You Ask, IEEE Transactions on Software Engineering, April 2004.

Online reference (draft)

www.simula.no/departments/engineering/.artifacts/framingsubmit.pdf

[JRR 04] S. JEANNENOT, P. RICHARD & F. RIDOUARD, Ordonnancement temps réel avec profils variables de consommation d'énergie, Laboratoire d'Informatique Scientifique et Industrielle, École Nationale de Mécanique et d'Aérotechnique, France, 2004.

K

[KD 00] J. P. KINDINGER, & J. L. DARBY, Risk Factor Analysis – A New Qualitative Risk, Management Tool, *Proceedings of the Project Management Institute Annual Seminars & Symposium*, Houston, Texas, USA, 2000.

[KDE 79] M. S. Krishnamoorthy, and N. Deon, Complexity of minimum-Dummy-Activities Problem in a Pert Network, *Networks*, 9, pp.189-194, 1979.

[KER 09] O. KERMIA, Ordonnancement temps réel multiprocesseur de tâches non-préemptives avec contraintes de précédence, de périodicité stricte et de latence, Thèse de doctorat, université Paris XI, 2009.

[KIM 01] A. KIMMS, Mathematical Programming and Financial Objectives for Scheduling Projects, Kluwer Academic Publishers, Boston, 2001.

[KLO 09] T. J. KLOPPENBORG, Contemporary Project Management, South-western CENGAGE Learning, Mason, OH, 2009.

[KMS 92] J. D. Kamburowski, J. Michael, and M. Stallman, Optimal construction of Project Activity Networks, *Proceeding of the Annual Meeting of the Decision Sciences Institute*, San-Francisco, CA, pp.1424-1426, 1992.

[KMS 00] J. D. Kamburowski, J. Michael, and M. Stallman, Minimizing the Complexity of an Activity Network, *Networks*, 36 (1), pp. 47-52, 2000.

[KSX 08] 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.

[KUM 07] R. KUMAR, Evolutionary multiobjective combinatorial optimization. In *GECCO '07: Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation*, pages 3366.3390, New York, NY, USA, 2007. ACM.

L

[LA 05] LA Hoang Trung, Utilisation d'ordres partiels pour la caractérisation des solutions robustes en ordonnancement, Thèse de Doctorat d'état, l'Institut National des Sciences Appliquées de Toulouse, France, 2005.

[LCM 03] P. LACOMME, C. PRINS & M. SEVAUX, Algorithmes de graphes, Eyrolles, France, 2003.

[LEW 95] P. J. Lewis, *Project Planning, Scheduling and Control*, McGraw-Hill, 1995.

[LIN 03] H. LINDO, Project Planning with PERT/CPM, Application Survey Paper, Washington University, 2003.

[LIT 06] T. LITTLE, Schedule Estimation and Uncertainty Surrounding the Cone of Uncertainty, IEEE Software, May/June 2006.

[LIZ 05] C. LIZZARAGA, Contribution au pilotage des projets partagés par des PME en groupement basée sur la gestion des risques, Thèse de Doctorat d'état, Institut national polytechnique de Toulouse, 2005.

[LL 08] C-H. LIN & C.-J. LIAO, Makespan minimization for multiple uniform machines. *Comput. Ind. Eng.*, 54(4):983-992, 2008.

[LOP 03] P. LOPEZ, Approche par contraintes des problèmes d'ordonnancement et d'affectation : structures temporelles et mécanismes de propagation, Thèse d'habilitation à diriger des recherches, Institut polytechnique de Toulouse, 2003

[LSA 99] L.J. LEBLANC, A. SHTUB & ANANDALINGRAM, Formulating and solving production planning problems, *European Journal of Operational Research*, 112, 54-80, 1999.

[LSS 93] I. LEE, R. SIKORA & M. J. SHAW, Joint lot sizing and sequencing with genetic algorithms for scheduling: Evolving the chromosome structure, In [For 93], pages 383-389, 1993.

M

[MBBO 06] N. E. MOUHOUB, H. BELOUADAH & A. BOUBETRA, Algorithme de construction d'un graphe Pert à partir d'un graphe des potentiels donné, *STUDIA UNIV. BABES BOLYAI, INFORMATICA*, Volume LI, Number 2, 2006, ISSN 1224-869x;

[**MBBO 10**] N. E. MOUHOUB, H. BELOUADAH & A. BOUBETRA, Towards a Better Assessment of the Durations in PERT Method, (IJCSIS) International Journal of Computer Science and Information Security, ISSN :1947-5500, Vol. 8, No. 9, Décembre 2010.

[**MBBE 11**] N. E. MOUHOUB, A. BENHOCINE. & H. BELOUADAH, A new method for constructing a minimal PERT network, (APM) Applied Mathematical modelling, Elsevier ISSN: 0307904X, Vol. 35, Issue: 9, 4575-4588, 2011.

[**MH 87**] P.W.G MORRIS & G.H. HOUGH, *The Anatomy of Major Projects: A study of the Reality of Project Management*, Wiley, New York, 1987.

[**ML 93**], B.L. MACCATHY & J. LIU. Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. International Journal of Production Research, vol. 31, no. 1, pages 59-79, 1993.

[**MIN 94**] R. MINICIARDI, M. PAOLUCCI & P.P. PULIAFITO, Development of a heuristic project scheduler under resource constraints, European Journal of Operational Research, Vol. 79, pp. 176-182.

[**MN 08**] J. MAKI-TURJA & M. NOLIN, Efficient implementation of tight response-times for tasks with offsets. Real-Time Systems Journal, Springer Netherlands, 16 February 2008.

[**MOJ 05**] K. MOLOKKEN-OSTVOLD & M. JORGENSEN, A Comparison of Software Project Overruns— Flexible versus Sequential Development Models, IEEE Transactions on Software Engineering, September 2005.

[**MOU 02**] N. E. MOUHOUB, Les problèmes d’ordonnancement de projet, Thèse de Magistère, Université de Sétif, Algerie, 2002.

[**MP 96**] K.R MACLEOD & P.F. PETERSEN, Estimating the tradeoff between resource allocation and probability of on – time completion in project management, *Project Management Journal*, vol. 27, n0 1, 26– 33, 1996 .

N

[**NA 94**] W.P.M NUIJTEN & E.H.L. AARTS, Constraint satisfaction for multiple capacitate job shop scheduling, In [Eib94], pages 635-639, 1994.

[**NOP 99**] Y. NORBERT, R. OUELLET & R. PARENT, La recherche opérationnelle, 2^{ème} éd., Gaëtan Morin Éditeur, 1999.

[**NSZZ 03**] K. NEUMANN, C. SCHMIDT, & J. ZIMMERMAN, Project scheduling with time windows and scarce resources: Temporal and resource constrained project scheduling with regular and nonregular objective functions, Springer-Verlag, 01 August, 2003.

[**OUE 88**] J.C. OUELLET, Evaluation et analyse par la simulation du projet de réfection Léopard A2 du ministère de la défense nationale, rapport de projet (M. Ing.), Ecole Polytechnique de Montréal, 1988.

O

[**ÖZD 96**] L. ÖZDAMAR, Capacitated family desegregations in hierarchical production planning, In [FUC96], pages 86-89, 1996.

P

[**PAR 05**] N. PARROD, Analyse d'un processus de coopération entre donneur d'ordres et sous-traitant au sein d'une chaîne logistique projet : une approche par simulation, Thèse de Doctorat, Ecole nationale supérieure de l'aéronautique et de l'espace, 2005.

[**PES 07**] C. PESCIO, Realistic and Useful: Toward Better Estimates, Draft version 1.2 - April, 2007.

Online reference: <http://www.eptacom.net/betterestimate/TheMathBehind.pdf>

[**PG 95**] M. PORTMAN & F. GHEDJATI, Affectation et ordonnancement par des algorithmes génétiques, 1995.

[**PIN 95**] M. PINEDO, **Scheduling: Theory, algorithms and systems, Prentice Hall, 1995.**

[**PX 99**] M. PINEDO & C. XIULI, "*Operations Scheduling with Applications in Manufacturing and Services*", Irwin/Mac Graw-Hill, 1999.

[**PIN 02**] M. PINEDO, *Scheduling, Theory, Algorithms and Systems*, Prentice Hall, 2002.

[**POR 91**] M. PORTMAN, Planification et ordonnancement, Cours, Ecole des mines de Nancy, Nancy, France, 1991.

[PRI 74] W. L. PRICE, Introduction aux graphes et aux réseaux, MASSON, Paris, France, 1974.

R

[RAE 96] F. RIANE, A. ARTIBA, & S.E. ELMAGHRABY, Sequencing on hybrid two-stages flow shop to minimise makspan, In [FUC 96], pages 93.1-93.8, September 1996.

[RS 75] G.T. ROSS & R.M. SOLAND, A branch and bound algorithm for the generalized assignment problem, *Mathematical Programming*, 8, 91-103, 1975.

[ROB 97] P.B ROBINSON, The Performance Measurement Baseline- A statistical view, *Project Management Journal*, vol. 28, no 2, 47 –52, 1997.

[ROY 70] B. ROY, Algèbre moderne et théorie des graphes, tome 2, Fascicule 3, Problèmes d'ordonnement et ensembles de potentiels sur un graphe, DUNOD, Paris, France, 1970.

[ROY 93] B. ROY, Aide multicritère à la décision : méthodes et cas, Economica, Paris, 1993.

[RUL 00] G. RULE, Bees and the Art of Estimating, *IEEE Software*, November/December 2000, p. 23.

S

[SDM 08] E. SAULE, P-F. DUTOT & G. MOUNIE, Scheduling With Storage Constraints. In *Electronic proceedings of IPDPS 2008*, Miami, Florida USA, APR 2008.

[SAK 84] M. SAKAROVICH, Optimisation combinatoire, HERMANN, Paris, France, 1984.

[SF 96] K.J. SHAW & P.J. FLEMING, Initial study of multi-objective genetic algorithms for scheduling the production of chilled ready meals, In proceedings of the second international Mendel conference on genetic algorithms – Mendel'96, Juin 1996.

[SOR 94] H.M, SOROUSH, The most critical path in a PERT network: A heuristic approach, *European Journal of Operational Research*, vol 78, no1, 93-105, 1994.

[SPI 80] J. Spinard, The Minimum Dummy Task Problem, *Networks*, 16, 331-348, 1980.

[ST 94] Y. N. SOTSKOV & V. S. TANAEV, Scheduling theory and practice : Minsk group results, Intelligent systems engineering, Spring, 1994.

[SW 80] F. STERBOUL, & D. WERTHEIMER, Comment construire un graphe PERT minimal, RAIRO, France, 1980.

[SYS 81] M.M. Syslo, Optimal Construction of Event-Node Networks, *RAIRO*, 15, 241-260, 1981.

[SYS 84] M.M. Syslo, On the Computational Complexity of the Minimum dummy Activities Problem in PERT Network, *Networks*, 14, pp.37-45, 1984.

T

[TRU 05] L. H. TRUNG, Utilisation d'ordre partiel pour la caractérisation des solutions robustes en ordonnancement, Thèse de Doctorat, Institut national des sciences appliquées de Toulouse, 2005.

[TB 02] V. TKINDT & J-C. BILLAUT, Multicriteria scheduling. Springer-Verlag, 2002.

V

[VDDP 96] A. VIGNIER, D. DARDILHAC, D. DEZALAY & C. PROUST, An optimal approach to solve a k-stage hybrid flow shop, In [FUC 96], pages 315-319, septembre 1996.

[VGP 00] L ; VILLENEUVE, A. GHARBI & R. PELLERIN, Nouvelle approche d'ordonnancement multi-projet pour l'industrie de la réfection, Canada, 2000

W

[WOE 08] G. J. WOEGINGER, Open problems around exact algorithms. *Discrete Appl. Math.*, 156(3):397.405 , 2008.

[WSL 95] A. M. WAIKAR, B. SARKER & A. M. LAL, A comparative study of some priority dispatching rules under different shop loads, *Production planning & control*, 6(4):301-310, 1995. ISSN: 0953-7287.

X

[XUO 91] N. XUONG, *Mathématiques discrète et informatique*, Masson, Paris, 1991.

Y

[YN 96] T. YAMADA & R. NAKANO, Scheduling by genetic local search with multi-step crossover, In [VERS96], pages 960-969, 1996

[YS, 07] P. M. YOMSI & Y. SOREL, Extending rate monotonic analysis with exact cost of preemptions for hard real-time systems. In *Proceedings of 19th Euromicro Conference on Real-Time Systems, ECRTS'07*, Pisa, Italy, July 2007.

Annexe 1

Liste des tableaux

Tableau II.1 Exemple de 3 algorithmes avec leurs complexités.

Tableau II.2 Tableau récapitulatif de classification des problèmes combinatoires.

Tableau II.3 Exemple de contraintes cumulatives

Tableau II.4 Quelques valeurs du champ α_1

Tableau II.5 Quelques valeurs du champ β

Tableau II.6 Quelques valeurs du champ γ

Tableau II.7 Valeurs du champ α en gestion de projet

Tableau III.1. Table initiale d'ordonnancement T .

Tableau III.2. La table initiale d'ordonnancement T pour les traitements algorithmiques, désignations non reprises.

Tableau III.3. (a) Un sous-tableau des antériorités de c et d .

Tableau III.3. (b) Nouveau sous-tableau des antériorités c , d et f .

Tableau III.4. La table d'ordonnancement T et l'organisation des sommets en niveaux.

Tableau III.7. Une table d'ordonnancement T .

Tableau III.8. (a) Dates de début au plus tôt du tableau de la figure III.4.

Tableau III.8. (b) Dates de début au plus tard du tableau de la figure III.4.

Tableau III.9. Tableau des données et des résultats.

Table IV.1. Table des antériorités et les durées calculées à partir de a , b et m .

Tableau IV.2. Récapitulation de la probabilité correspondante à chaque durée totale choisie pour le projet.

Table V. 1. Une table d'ordonnancement

Table V. 2. La table d'ordonnancement, désignations non prises.

Table VI. 1. La table d'ordonnancement T avec les désignations non prises.

Table VI. 2 Tableau récapitulatif concernant l'exécution des trois algorithmes.

Table VI.3 Résultats de la comparaison.

Table VII.1. Table d'ordonnancement.

Table VII.2. Recherche des bons arcs dans H_1 .

Table VII.3. Table d'ordonnancement de la Figure VII.11.

Table VII.4. Tableau récapitulatif de l'application de l'algorithme de Mouhoub et Sterboul sur 75 graphes.

Table VII.5. Résultats de la comparaison.

Annexe 2

Liste des figures

Figure II.1 Typologie des problèmes d'ordonnancement par les ressources

Figure II.2 Machine unique réalisant des travaux.

Figure II.3 Machines parallèles réalisant des travaux.

Figure III.4. (a) Atelier à cheminement unique (Flowshop).

Figure II.4. (b) Cas le plus général du « flow shop » : on peut sauter des machines, mais on les visite toujours par numéros croissants des machines.

Figure II.5 « Flow shop » hybride : Machines en série / gammes identiques et machines en parallèle.

Figure II.6 Atelier à cheminement multiples : machines en série (M_i) /gammes non identiques (Jobshop).

Figure II. 7 Machines en série / gammes non linéaires : « Open shop ».

Aucune contrainte sur l'ordre de visite des machines

Figure III.1. Les objectifs du projet.

Figure III.2. Cycle de vie d'un projet.

Figure III.3. Le diagramme de Gantt de la table III.1.

Figure III.4. La tâche u , de durée $t(u)$, précède la tâche v

Figure III.5. Le graphe des potentiels avec les sommets réorganisés en niveaux.

Figure III.6. Le passage du diagramme de Gantt au graphe PERT.

Figure III.7. La tâche u précède la tâche v dans le graphe PERT.

Figure III.8. Le graphe PERT construit par ajout successif des tâches, les durées n'étant pas inscrites.

Figure III.9. (a) Problème de représentation dans le graphe PERT.

Figure III.9. (b) Introduction de la tâche fictive f et représentation dans le graphe PERT.

Figure III.10. Le graphe PERT de la table III.4 avec un minimum de 5 tâches fictives.

Figure III.11. La modularité dans le graphe PERT.

Figure III.12 Calcul de do_A dans le cas d'un ou plusieurs arcs.

Figure III.13. Calcul de da_A dans le cas d'un ou plusieurs arcs.

Figure III.14. L'intervalle de placement d'une tâche A , les dates et les marges associées à A représentées par le diagramme de Gantt. A_1 , A_2 et A_3 sont les tâches qui succèdent à A .

Figure III.15. Les dates de début au plus tôt et au plus tard dans le graphe des potentiels.

Figure III.16. Les dates de début au plus tôt et au plus tard dans le graphe PERT

Figure III.17. Le diagramme de Gantt.

Figure IV.1. Distribution du temps des activités tiré de [KIM 01].

Figure IV.2. La loi Bêta.

Figure IV.3. Graphe des potentiels correspondant au tableau IV.1.

Figure IV.4 Courbe de la distribution Bêta tiré de [IKA 04].

Figure IV.5. Exemple de distributions possibles.

Figure IV.6. Courbe des coûts γ d'une tâche A en fonction de sa durée.

Figure V.1. Un graphe G et son graphe adjoint $L(G)$.

Figure V.2. Exemple d'utilisation des graphes adjoints dans la coloration d'arêtes.

Figure V.3. Les graphes G_1 , G_2 et G_3 du graphe adjoint H .

Figure V.4. Les configurations « Z » et « Δ ».

Figure V.5. Un biparti complet B de H l'étoile de G associée à B

Figure V.6. (a) un graphe H .

Figure V.6. (b) La partition des arcs de H en bipartis complets. Les arcs de chaque couleur représentent un biparti complet.

Figure V.6. (c) Le graphe G construit à partir de son graphe adjoint H de la figure V.6. (a).

Figure V.7. Sous graphes contenant de la configuration « Z ».

Figure V.8. La table d'ordonnancement et le graphe des potentiels dont les arcs sont partitionnés en bipartis complets.

Figure V.9. Le graphe PERT construit à partir des bipartis complets du graphe des potentiels.

Figure V.10. Correspondance entre un biparti complet du graphe des potentiels et l'étoile dans le graphe PERT.

Figure V.11. La configuration « Z » et sa transformation dans le graphe des potentiels.

Figure VI.12. Algorithme 1 de passage du graphe des potentiels au graphe PERT.

Figure 13. (a). Le graphe des potentiels correspondant à la table VII.2. avec les barres de « Z » en gras.

Figure 13. (b). Le graphe des potentiels modifié par l'introduction des tâches fictives f_i .

Figure 13. (c). Le graphe des potentiels modifié par l'introduction des tâches fictives f_i avec réorganisation des tâches en niveaux et partition des arcs en bipartis complets.

Figure 13. (d). Construction du graphe PERT à partir du graphe des potentiels.

Figure V.14 Introduction de la tâche fictive dans le diagramme de Gantt et modifié en un graphe PERT

Figure V. 15 Le diagramme de Gantt de la figure IV.15.(a)

Figure V.16. Le graphe des potentiels et le diagramme de Gantt résultant.

Figure VI. 1. La configuration « Z » et sa transformation dans le graphe des potentiels.

Figure VI. 2. Les sommets a et b ayant des successeurs communs et des successeurs non communs, des prédécesseurs communs et des prédécesseurs non communs d'où la présence des Z.

Figure VI. 3. Regroupement des barres de "Z", tâches fictives et partition des arcs en bipartis complets.

Figure VI. 4. Algorithme (2) de passage du graphe des potentiels au graphe PERT.

Figure VI. 5. (a) Le graphe des potentiels correspondant à la table V.1.

Figure VI. 5. (b) Le graphe des potentiels avec l'introduction des tâches fictives (sommets 14, ..., 18) avec regroupement des barres de Z appartenant au même biparti et organisation des sommets en niveaux.

Figure VI. 5. (c) Construction du graphe PERT à partir du graphe des potentiels par l'application de l'algorithme 2 avec 5 tâches fictives (1, 2), (3, 6), (4, 7), (5, 7), (5, 8).

Figure VI. 6. Les sommets ayant les mêmes successeurs contractés en un sommet

Figure VI. 7. Les sommets ont les mêmes prédécesseurs contractés en un sommet unique dans G_e .

Figure VI. 8. Contraction des sommets initial et terminal de l'arc fictif u arrivant seul à un ou plusieurs arcs réels.

Figure VI. 9. Contraction des sommets initial et terminal de l'arc fictif u débutant seul d'un ou plusieurs arcs réels.

Figure VI. 10. Algorithme (3) de passage du graphe des potentiels au graphe PERT.

Figure VI. 11. Construction du graphe PERT à partir du graphe des potentiels par l'application de l'algorithme 3 de la Table VI. 1.

Figure VII.1. Un arc redondant (i, j).

Figure VII.1. (a) Le graphe des potentiels de la Table. VII.1.

Figure VII.1. (b) Le graphe H_0 de la Figure VII.2 (b).

Figure VII. 2. (c) Le graphe H_1 de la figure VII.2 (a).

Figure VII. 2. (d) Le graphe H_2 de la figure VII.2 (a).

Figure VII.3. Les sommets b_i ayant les mêmes successeurs contractés en un sommet unique b_1 dans G_{e1} .

Figure VII.4. Les sommets a_i ont les mêmes prédécesseurs contractés en un sommet unique a_1 dans G_{e2} .

Figure VII.5. Un arc transmetteur de type 1 entre deux tâches i et j et sa contraction.

Figure VII.6. Un arc transmetteur de type 2 entre deux tâches i et j et sa contraction.

Figure VII.7. 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} .

Figure VII.8. 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} .

Figure VII.9. Un biparti complet dans G_{e7} et le sommet fictif sf dans G_{e8} . La représentation dans G_{e8} contient moins d'arcs fictifs que dans G_{e7} .

Figure VII.10. Construction de G_e et G_f .

Figure VII.11. Construction d'un building industriel.

Figure VII.12. (a) Le graphe des potentiels G_v avec 14 sommets de la table d'ordonnancement VII. 3.

Figure 12. (b) Le graphe PERT G_{e1} de la Figure VII.12. (a). G_{e1} contient 29 tâches fictives et 28 sommets.

Figure VII.12. (c) Le graphe PERT G_{e3} de la Figure 15. (a) obtenu en appliquant la règle 1 ensuite la règle 2. G_{e3} contient 23 tâches fictives et 23 sommets.

Figure VII.12 (d) Le graphe PERT G_{e5} de la Figure 12. (a) obtenu en appliquant la règle 3 ensuite la règle 4, avec 16 tâches fictives et 17 sommets.

Figure VII.12 (e) Le graphe PERT minimal G_{e8} de la figure 12. (a) avec le sommet fictif sf_0 obtenu en appliquant la règle 5, la règle 3 ensuite la règle 7. G_{e8} contient 15 tâches fictives et 18 sommets.

Figure VII.13. Implémentation du nouvel algorithme.

Résumé de la thèse

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

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

Ce travail présente deux idées originales de dessin du graphe PERT. La première consiste à balayer la table d'ordonnancement ligne par ligne et dessiner, à chaque étape, un arc qui s'ajoute à ce qui a été construit avant. La deuxième consiste à localiser les bipartis complets dans le graphe des potentiels et les transformer au fur et à mesure en étoiles adjacentes constituant le graphe PERT à l'aide d'un ensemble de concepts et de résultats sur les graphes adjoints de graphe. Cette idée a été optimisée à deux reprises en vue d'avoir un graphe PERT moins encombrant, facile à lire et à contrôler.

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

Abstract of the thesis

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

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

This work presents two novel ideas for drawing PERT graph. The first is to scan the scheduling table line by line and drawing, at each stage, an arc in addition to what has been built before. The second is to locate the complete bipartites in the potentials graph to transform them progressively into adjacent stars forming the PERT graph with a set of concepts and results on line graphs of graphs. This idea has been optimized on two occasions to have a PERT graph more compact, easy to read and to control.

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

ملخص الأطروحة

دراسة نماذج جدولة المشروع من خلال مخطط غانت و PERT و MPM، تبين أن مديري المشاريع يفضلون العمل بطريقة PERT على الرغم من أنه من الصعب تحقيقها، في حين أن MPM توفر إمكانيات أكثر بساطة. في طريقة PERT يتم حساب مدة الأشغال باستعمال تقنية تبدو لأول وهلة صحيحة، غير أن دراسة تحليلية معمقة للموضوع، مثلما هو وارد في هذه الأطروحة، تظهر اختلافات متعددة حيث قمنا بتقديم اقتراحات عديدة من أجل تصحيح هذه الاختلالات.

تقدم هذه الأطروحة طريقتين جديدتين لرسم بيان PERT. وقد تم تحسين الطريقة الثانية على مرحلتين وتم مقارنتها بطريقة مشابهة أظهرت تفوق فكرتنا. أخيراً، تم طرح مشكلة الأشغال الوهمية في مخطط PERT حيث أنجزنا خوارزمية يقدم أحسن وسيلة لتقليصها إلى الحد الأدنى. وبمقارنة هذا الخوارزمية مع آخر يقوم بنفس العمل أظهرت النتائج المحصل عليها أفضلية طريقتنا الجديدة.