



HAL
open science

Ordonnements coopératifs pour les chaînes logistiques

Zerouk Mouloua

► **To cite this version:**

Zerouk Mouloua. Ordonnements coopératifs pour les chaînes logistiques. Informatique [cs]. Institut National Polytechnique de Lorraine - INPL, 2007. Français. NNT : . tel-00606329

HAL Id: tel-00606329

<https://theses.hal.science/tel-00606329>

Submitted on 6 Jul 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Ecole doctorale IAEM Lorraine



Ordonnements coopératifs pour les chaînes logistiques

THÈSE

Sera soutenue le 21 novembre 2007
Pour l'obtention du

**Doctorat de l'Institut National Polytechnique de Lorraine
(Spécialité Informatique)**

par

Zerouk MOULOUA

Composition du jury

Rapporteurs :

Jin-Kao HAO

Professeur - Université d'Angers

Emmanuel NERON

Professeur – Université de Tours

Examineurs :

Christian ARTIGUES

Chargé de recherche - CNRS

Françoise SIMONOT

Professeur - INPL

Directrice de Thèse

Marie Claude PORTMANN

Professeur - INPL

Co-directeur de thèse :

Ammar OULAMARA

Maître de conférences - INPL



TABLE DES MATIERES

INTRODUCTION GENERALE	1
CHAPITRE 1 : Etat de l'art	5
1.1 Introduction	7
1.2 Les chaînes logistiques	7
1.2.1 Définitions	8
1.2.2 Types de chaînes logistiques	10
1.2.3 Fonctions de la chaîne logistique	13
1.2.3.1 L'approvisionnement	13
1.2.3.2 La production	13
1.2.3.3 Le stockage	13
1.2.3.4 Distribution et transports	14
1.2.3.5 La vente	14
1.2.4 Les décisions dans la chaîne logistique	14
1.2.4.1 Les décisions stratégiques	16
1.2.4.2 Les décisions tactiques	17
1.2.4.3 Les décisions opérationnelles	18
1.2.5 La gestion de la chaîne logistique ou supply chain management	18
1.2.6 Optimisation de la chaîne logistique	20
1.2.7 Modélisation des chaînes logistiques	21
1.2.7.1 Les modèles conceptuels	22
1.2.7.2 Modèles mathématiques	22
1.2.7.3 Modèles par simulation	22
1.2.8 Les outils d'aide à la décision	23
1.2.8.1 Entreprise Resource Planning (ERP)	23
1.2.8.2 Advanced Planning and Scheduling APS	24
1.2.9 Motivations et exemples de chaînes logistiques	25
1.2.9.1 Chaîne logistique de DELL	26
1.2.9.2 L'exemple de Chrysler	27
1.2.9.3 L'exemple de Amazon.com	27
1.2.10 Les mesures de la performance de la chaîne logistique	27
1.3 Ordonnancement de la production	30
1.3.1 Introduction	30
1.3.2 Gestion de la production	31
1.3.3 Ordonnancement	31
1.3.3.1 Définitions	31
1.3.3.2 Ordonnancement d'ateliers	33
1.3.3.3 Les méthodes de résolution	35
1.3.4 Le concept du Juste à Temps JAT (Just In Time JIT)	37
1.4 Ordonnancement et coopération dans la chaîne logistique	39
1.5 Conclusion	44
CHAPITRE 2: Le Problème et l'Approche retenue	45
2.1 Introduction	47
2.2 Définition du problème	49
2.2.1 Fonctionnement général	49
2.2.2 Exemple	51
2.3 Problème de JIT particulier à la thèse	56
2.3.1 Hypothèses générales	56
2.3.2 Le problème d'ordonnancement en juste à temps	57

2.3.3	Notations	58
2.4	Approche de résolution du système global	60
2.4.1	Description Générale	60
2.4.2	Processus de négociation	61
2.5	Conclusion	64
CHAPITRE 3 : Méthode de résolution		65
3.1	Introduction	67
3.2	Approche de résolution par décomposition et méta-heuristique	67
3.3	Résolution du JIT-Seq	69
3.3.1	Ordonnancement de projet	69
3.3.2	Problème original du Pert coût	70
3.3.3	Algorithme du JIT adapté du Pert coût	70
3.3.4	Bilan de la complexité des évaluations	79
3.4	Méta-heuristiques	80
3.4.1	Introduction : les métaheuristiques	80
3.4.2	Les Algorithmes Génétiques	81
3.4.2.1	Principes	82
3.4.2.2	Schémas	83
3.4.2.3	Opérateurs classiques	86
3.4.3	Recherche de solutions avec les AG	90
3.4.3.1	Le Codage	90
3.4.3.2	Les opérateurs	90
3.4.3.3	Schéma proposé	92
3.4.4	Le recuit simulé	93
3.4.4.1	Introduction	93
3.4.4.2	Principe de fonctionnement	94
3.5	Conclusion	95
CHAPITRE 4 : Experiences numériques		96
4.1	Introduction	98
4.2	Génération de données	98
4.2.1	Les temps d'exécutions	98
4.2.2	Génération des pénalités d'avance et de retard	99
4.2.3	Génération des dates souhaitées et des dates impératives	99
4.3	Evaluation du Pert Coût	100
4.4	Evaluation des métaheuristiques	103
4.4.1	Pourcentage de faisabilité	103
4.4.2	Qualité des solutions trouvées	104
4.4.3	Evaluation des opérateurs génétiques	107
4.5	Conclusion	107
CHAPITRE 5 : Problèmes de transports dans les chaînes logistiques		109
5.1	Introduction	111
5.2	Problème 1 : La coopération entre fournisseurs pour minimiser les coûts de possession des stocks	112
5.2.1	Description du problème	112
5.2.2	Hypothèses et Notations	112
5.2.2.1	Hypothèses :	112
5.2.2.2	Notations	113
5.2.3	Le problème Mono Produit	114
5.2.3.1	Le problème à 2 fournisseurs	114
5.2.3.2	Le problème à N fournisseurs	118
5.2.3.3	Heuristique pour rendre les W_i entiers	119
5.2.4	Le problème Multi Produits	119

5.2.4.1	Le problème à 2 fournisseurs- 2 produits	119
5.2.4.2	Le problème à N fournisseurs- M produits	121
5.2.4.3	Algorithme pour rendre les W_i^j entiers	121
5.3	Problème 2 : ordonnancement des transports dans une supply chain élémentaire	123
5.3.1	Description du problème mono fournisseur et mono produit	123
5.3.2	Notations	125
5.3.3	Problème de faisabilité	125
5.3.4	Minimisation des transports : capacité illimitée	127
5.3.5	Minimisation des transports : capacité limitée	128
5.4	Problème 3: Optimisation des transports dans une supply chain spécifique	130
5.4.1	Optimisation des transports dans une supply chain mono-produit multi-clients avec planning de production fixé	130
5.4.1.1	Notations	130
5.4.1.2	Représentation graphique	131
5.4.1.3	Approche de résolution	133
5.4.2	Optimisation des transports dans une supply chain multi-produit multi-clients avec planning de production fixé	135
5.4.2.1	Notations	135
5.4.2.2	Représentation graphique	136
5.4.2.3	Approche de résolution	138
5.4.3	Optimisation conjointe de la production et des transports dans une supply chain multi-produit multi-clients	139
5.4.3.1	Notations	139
5.4.3.2	Approche de résolution	140
5.5	Conclusion	146
	CHAPITRE 6 : Conclusions et perspectives	147
6.1	Conclusions	149
6.2	Perspectives	151
	ANNEXE 1	154
	BIBLIOGRAPHIE	164
	Liste des figures	170

Introduction générale

Cette thèse a été effectuée au sein de l'équipe MACSI (Modélisation, Analyse et Conduite des Systèmes Industriels) devenue en 2007 ORCHIDS (Operations Research for Complex Hybrid Decision Systems) du LORIA (Laboratoire Lorrain de Recherche en informatique et ses Applications) sous la direction de Marie Claude Portmann et d'Ammar Oulamara. Elle a été menée dans le cadre d'une allocation de recherche du ministère de l'enseignement supérieur et de la recherche. L'équipe MACSI-Nancy devenue ORCHIDS s'intéresse aux problèmes liés à la recherche opérationnelle, à l'optimisation, et plus spécifiquement aux problèmes d'ordonnancements. Plus récemment, l'équipe a orienté une partie de ses recherches vers l'étude des problèmes liés aux chaînes logistiques.

Les chaînes logistiques sont un résultat, ou plutôt une conséquence, de la déferlante mondialisation et de la globalisation des marchés. En effet, dans un environnement de plus en plus concurrentiel, la course vers la réduction des coûts de production tout en gardant une bonne qualité de produits finis et en améliorant la qualité de service aux clients, plus exigeants que jamais, n'en finit plus. Les entreprises et organisations diverses dépassent désormais les frontières, deviennent encore plus complexes, et de ce fait de nouveaux défis pour les gérer ont vu le jour. Le regroupement des entreprises ayant des intérêts en commun dans une structure de chaîne logistique pour optimiser les opérations et maximiser les bénéfices et profits conduit vers un nouveau type de concurrence : la concurrence entre chaînes logistiques. Dans ces nouveaux schémas d'entreprises, plusieurs entités doivent collaborer afin d'obtenir les meilleurs résultats possibles. Cette coordination n'est pas chose facile à réaliser car elle implique un changement culturel profond, ce qui est connu pour être un élément très difficile à mettre en œuvre car impliquant le changement de règles sociales et de comportement. L'être humain est par nature hostile au changement, ou du moins il en a peur.

L'intérêt de la communauté scientifique aux problématiques au sein des chaînes logistiques est relativement récent (les dix à quinze dernières années), mais va en grandissant. Cela s'explique par l'intérêt économique que représente le fait de développer de nouvelles méthodes, outils et techniques pour aider les entreprises à gérer des flux de matières, d'informations, et de finances d'une manière optimale. Parmi ces outils, on retrouve notamment les outils d'aide à la décision incorporés dans des progiciels tel que les ERP (*Enterprise Resource Planning*) et surtout les APS (*Advanced Planning and Scheduling*).

Au sein d'une chaîne logistique, il y'a trois niveaux de prise de décisions : le niveau stratégique, le niveau tactique, et le niveau opérationnel. Ces problèmes sont des problèmes rencontrés au quotidien dans, par exemple, les ateliers de production ou d'assemblage, où l'on doit être capable à tout moment de décider quelles sont les opérations à exécuter, dans quel ordre et en utilisant quelles ressources.

Dans le présent mémoire nous étudions des problèmes d'ordonnancement au sein des chaînes logistiques. Nous nous intéressons dans le problème qui constitue la majeure partie de la thèse à la coordination et la coopération des ordonnancements entre plusieurs partenaires appartenant à la même chaîne logistique mais indépendants les uns des autres. Nous matérialisons cette coordination via l'établissement et la construction d'ordonnancements qui tiennent compte de négociations menées entre les différents centres de décisions appartenant à la chaîne. Cela implique dans notre cas de faire des ordonnancements en juste à temps (Just In Time ou JIT).

Dans la deuxième partie de la thèse nous nous intéressons à certaines problématiques d'ordonnancement et de coopération pour notamment optimiser les coûts de transport, qui constituent une grande partie des coûts logistiques.

Dans cette thèse, nous avons combiné l'utilisation de méthodes exactes (programmation dynamique) et de méthodes heuristiques pour résoudre les problèmes étudiés. Nous proposons notamment de nouveaux opérateurs génétiques spécifiques à notre problème.

Ce mémoire est organisé comme suit :

Dans le premier chapitre, nous présentons un état de l'art sur les principaux thèmes liés à cette thèse. Ce chapitre peut être divisé en trois parties. Dans la première, nous commençons par introduire la notion de chaîne logistique et tous les concepts qui y sont liés. Nous définissons la gestion de la chaîne logistique, ainsi que les principaux types ou architectures de chaînes logistiques, décrivons quelques outils et notions tel que : les mesures de performance, les différentes fonctions, et les niveaux de décisions. Nous donnons enfin quelques exemples de chaînes logistiques connues à l'échelle mondiale. Dans la deuxième partie nous introduisons les concepts d'ordonnancement et de juste à temps. Nous rappelons brièvement les notions d'ordonnancement d'atelier et de production, ainsi que les méthodes de résolution. Enfin, dans la troisième partie, nous présentons une recherche bibliographique sur les travaux scientifiques menés sous le thème de l'ordonnancement et de la coopération dans les chaînes logistiques. Nous avons voulu que notre mémoire soit accessible aux non spécialistes en ordonnancement ou en chaînes logistiques.

Dans le second chapitre, nous décrivons le problème étudié et nous présentons l'approche globale de résolution. Le problème dans sa forme la plus générale y est présenté. Nous commençons par décrire le fonctionnement global, puis nous nous intéressons à certains problèmes de juste à temps liés à la thèse. Ce problème de juste à temps est présenté sous sa forme la plus générique et une approche de résolution du système global basé sur un processus itératif de négociation y est décrite.

Dans le troisième chapitre, nous présentons l'approche de résolution détaillée pour les problèmes d'ordonnancement en juste à temps décrits dans le chapitre précédent. Nous présentons le schéma global de l'algorithme de résolution basé sur deux étapes. La première étant l'utilisation de la méthode du Pert (Program Review and Evaluation Technique) coût. Nous en donnons un bref rappel puis montrons comment notre problème se ramène à un problème équivalent du Pert coût. Nous décrivons ensuite les différentes phases de l'algorithme tout en mettant l'accent sur les étapes que nous avons nous mêmes développés et qui optimisent sa durée d'exécution. La deuxième étape de la méthode de résolution est l'utilisation de méta heuristiques. Nous commençons par rappeler les définitions des méta heuristiques. Comme métaheuristique, nous avons choisi d'utiliser les algorithmes génétiques et le recuit simulé. Nous rappelons donc ce que sont les algorithmes génétiques, et nous introduisons les opérateurs génétiques développés dans le cadre de cette thèse. Enfin, nous

rappelons brièvement la méthode du recuit simulé, sans aller dans le détail car nous l'utilisons pour des fins de comparaisons.

Dans le quatrième chapitre, nous présentons les résultats des expériences menées pour valider l'approche de résolution décrite dans le chapitre trois. Nous introduisons le schéma des expériences menées, puis nous expliquons la méthode de génération de données que nous avons utilisé. Les performances de la méthode du Pert coût utilisée, ainsi que des métaheuristiques sont données.

Dans le cinquième chapitre, nous présentons des problématiques d'ordonnancement au sein des chaînes logistiques sans liens directs avec le problème étudié dans les précédents chapitres. Néanmoins, il s'agit toujours de problèmes au niveau opérationnel de chaînes logistiques impliquant la coordination entre partenaires pour optimiser le fonctionnement général. On se concentre ici beaucoup plus sur la réduction des coûts de transport. En premier, nous étudions un problème dans la chaîne logistique qui met en œuvre la coopération entre fournisseurs pour satisfaire les demandes d'un client. Nous commençons d'abord par un problème simplifié avec un nombre réduit de fournisseurs et de types de produits fabriqués, pour ensuite généraliser à un nombre quelconque de fournisseurs et de produits fabriqués. Dans le deuxième problème, nous étudions les problèmes de transport dans une chaîne logistique élémentaire à deux niveaux. Le premier niveau est un système de production simplifié, et le deuxième niveau est le système de transport. Dans un premier temps nous proposons un algorithme pour décider de la faisabilité ou pas d'un problème donné. Nous ajoutons ensuite des contraintes sur les capacités des transports. Nous proposons des algorithmes polynomiaux pour l'optimisation des coûts de transport. Dans le troisième problème nous traitons trois cas d'optimisation dans une chaîne logistique à deux niveaux (production et transport) avec des clients en aval et des courbes cumulées de demandes de livraisons, et des fournisseurs en amont avec des courbes cumulées d'arrivées des matières premières. Nous supposons dans les deux premiers cas que le planning de production est fixé à l'avance et est optimal. Relativement à ce planning de production, nous proposons un algorithme de programmation dynamique qui minimise les coûts de transports. Dans le premier cas on ne considère qu'un seul type de produit, une généralisation à plusieurs types de produits est étudiée dans le deuxième cas. Dans le troisième cas on traite un problème de coopération entre les fonctions production et transport de la chaîne logistique pour lequel nous développons un algorithme à base de programmation dynamique pour l'optimisation conjointe et simultanée des coûts de production et de transport.

Nous concluons en présentant des développements qui pourraient être faits dans le prolongement de cette thèse.

1 CHAPITRE 1 : Etat de l'art

Résumé

Les notions de chaînes logistiques, d'ordonnancements, et de coopération dans les chaînes logistiques sont présentées dans ce chapitre. Une étude qui se veut la plus complète possible y est présentée afin de permettre aux non spécialistes de comprendre les problématiques considérées dans les chapitres qui suivent. Notre étude bibliographique montre la pauvreté des publications scientifiques concernant les aspects ordonnancement et coopération au niveau opérationnel.

1.1 Introduction

Dans ce chapitre nous présentons l'état de l'art des différents domaines de recherches qui touchent aux problèmes étudiés dans cette thèse. Nous présentons les chaînes logistiques : leurs définitions, leurs conceptions, leur gestion, et de la prise de décisions aux différents niveaux de la chaîne. Nous nous intéressons aussi aux problèmes d'ordonnancement en ateliers d'une manière générale, et plus particulièrement des flowshops et flowshops de permutation. Nous présentons ensuite les travaux concernant l'ordonnancement dans la chaîne logistique et nous terminons par les problématiques du juste à temps (just in time). Bien sûr notre présentation n'est pas exhaustive, mais nous essayons de donner les notions nécessaires pour comprendre notre problématique sans pour autant être un spécialiste de chaîne logistique ou d'ordonnancement.

1.2 Les chaînes logistiques

De nos jours, les entreprises doivent s'adapter à la dynamique du marché pour espérer survivre dans un environnement très concurrentiel et très compétitif. Les clients sont beaucoup plus exigeants en raison des nombreux choix qui s'offrent à eux. La satisfaction des clients combinée à la réduction des coûts est difficile à réaliser. Ajoutons à cela les nouvelles technologies de l'information et de la communication qui ont révolutionné la manière avec laquelle doivent être gérées les entreprises. Celles-ci deviennent de plus en plus grandes, englobent de plus en plus de filiales, et sont de plus en plus de dimension internationale, elles n'ont quasiment plus de nationalité. Cette nouvelle situation a créé des nouveaux défis. Le défi de s'adapter à la globalisation et d'en tirer les bénéfices. Depuis une quinzaine d'années la notion de modélisation des entreprises ou des processus aboutissant à la création, la production ou le développement de nouveaux produits ou services en chaîne logistique a émergé et est devenue incontournable dans les organisations à grande échelle. Aujourd'hui, aucune entreprise ne peut ignorer que la gestion de production classique laisse la place à la gestion de la chaîne logistique pour faire face aux nouvelles attentes du marché, aux nouveaux concurrents, nouveaux liens entre les entreprises et leurs partenaires. En clair : tout change ! Et les entreprises doivent s'adapter à cette nouvelle réalité, ce qui signifie :

- Une réactivité très supérieure.
- Une baisse significative des coûts.
- Une nette amélioration de la qualité et du service du client.
- De meilleures performances.

La notion de chaîne logistique inclut le terme « logistique » qui vient d'un mot grec qui signifie l'art du raisonnement et du calcul (Pons, 1996). La logistique est apparue en premier lieu dans un contexte militaire qui concernait tout ce qui est nécessaire (physiquement) à l'application sur le terrain des décisions stratégiques et tactiques. Certains remontent même jusqu'au temps d'Alexandre le grand (Engles, 1978) et ils mettent en évidence le sens qu'il avait pour gérer la chaîne logistique de son armée. On peut dire qu'Alexandre le grand était un précurseur.

Après la logistique militaire vint la logistique industrielle, celle-ci repose plus particulièrement sur les activités de soutien à la production. Elle est apparue à la fin de la seconde guerre mondiale, notamment avec la reconversion dans les entreprises des spécialistes militaires de la logistique. Le concept de logistique a évolué depuis, avec l'évolution du marché et des systèmes industriels. Aujourd'hui, le terme « logistique »

recouvre des interprétations diverses, et certains pensent que le concept de la logistique est une problématique en soi (Moller, 1995).

Un synonyme de « logistique » pourrait être « gestion des flux » mais la logistique ne se limite pas à la gestion des flux, elle nécessite également la conception de systèmes physiques qui vont créer et déplacer les flux, de manière interne sur un même site de production ou de manière externe entre plusieurs sites de production au sein d'un groupe ou d'un réseau d'entreprises.

Dans un contexte économique instable, et sous la pression d'une concurrence croissante, des cycles de produits de plus en plus court et de systèmes informatiques, juridiques et financiers beaucoup plus complexes, les entreprises doivent améliorer leurs performances en optimisant leurs ressources et en réduisant leurs coûts. Fournir le produit ou le service désiré par le client le plus rapidement possible, moins cher et plus performant que ceux de ses concurrents sur le marché est le souci majeur de chaque entreprise opérant à l'échelle locale et/ou internationale. Quel que soit le domaine d'activité (santé, industrie, transport..) la performance de l'entreprise a une influence sur la satisfaction des clients et leur fidélisation, ainsi que sur les résultats de l'entreprise.

La globalisation étant devenue une réalité, les professionnels prédisent que dans l'avenir, si ce n'est déjà commencé, la concurrence ne sera plus entre entreprises mais entre chaînes logistiques (Christopher, 92).

Nous allons voir dans ce qui suit les différentes définitions des chaînes logistiques, on étudie aussi ses fonctionnalités, les participants et les systèmes de gestion de la chaîne logistique.

1.2.1 Définitions

La chaîne logistique englobe l'ensemble des opérations réalisées pour la fabrication d'un produit ou d'un service allant de l'extraction de la matière première à la livraison au client final, en passant par les étapes de transformation, de stockage, et de distribution. De nos jours, de plus en plus on regarde la chaîne logistique comme une toile regroupant plusieurs des activités citées, cela est dû à la complexité des organisations actuelles et à leur dimension internationale. Ajoutés aux flux des matières, la chaîne logistique inclut les flux d'information et les flux financiers. Chaque étape de transformation ou de distribution peut impliquer de nouveaux acteurs, soit de nouveaux fournisseurs ou de nouveaux clients intermédiaires, avec également des nouveaux flux d'informations.

Il existe énormément de définitions de la chaîne logistique dans la littérature. C'est un sujet passionnant car nouveau et porteur de progrès. Tayur et al (Tayur et al, 1999) définissent la chaîne logistique comme « un système de sous-traitants, de producteurs, de distributeurs, de détaillants et de clients entre lesquels s'échangent les flux matériels dans le sens des fournisseurs vers les clients, et des flux d'information dans les deux sens ». Lambert et al (Lambert et al, 1998) proposent une définition simple : « Une chaîne logistique est l'ensemble des firmes qui amènent des produits ou des services sur le marché », ce qui est une définition très généraliste. Lee et Bellington (Lee et al, 93) donnent une vision plus opérationnelle : « un réseau d'installations qui assure les fonctions d'approvisionnement en matières premières, de transformation de ces matières premières en composants puis en produits finis, et de distribution du produit fini vers le client ». Le parallélisme entre la chaîne logistique et l'organisation en réseau est très significative, ça met en évidence la complexité de sa gestion étant donné qu'elle est le point qui rassemble plusieurs acteurs qui participent à un même projet ou à défaut ils participent tous au processus de création, ou développement d'un service

ou d'un produit. New et al (New et al, 1995) vont dans le même sens et proposent de représenter les activités et les entreprises impliquées dans cette chaîne qui commence à l'extraction de la matière première en passant par les entreprises de production, les grossistes, les détaillants jusqu'au client final.

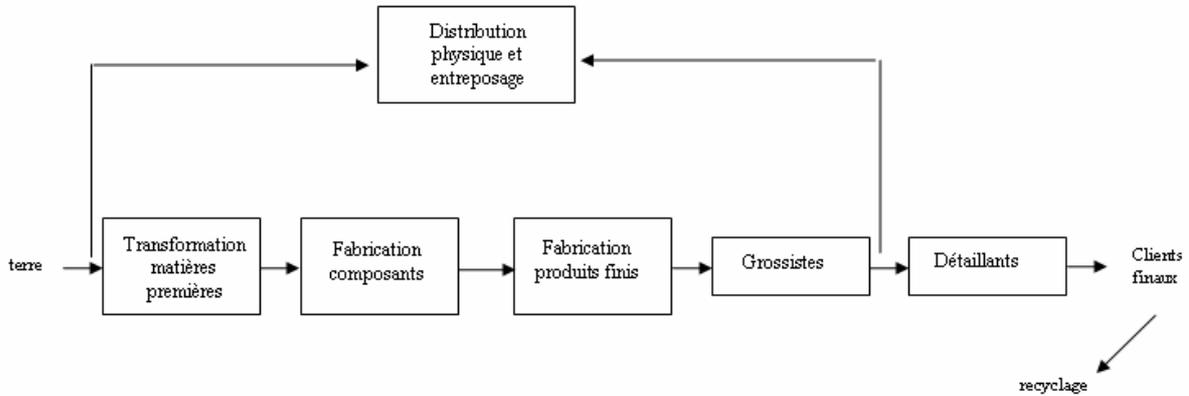


Figure 1.1 modèle de chaîne logistique selon (new et al, 1995).

Govil et Proth (Govil et Proth, 2002) donnent la définition suivante : « la chaîne logistique est un réseau global d'organisations qui coopèrent afin d'améliorer les flux des matériels et des informations entre les fournisseurs et les clients au plus bas coût et à la plus grande vitesse. L'objectif de la chaîne logistique est la satisfaction du client ». Cette définition suggère que la chaîne logistique englobe des partenaires indépendants mais vus comme une seule entité ou une seule stratégie.

Dans l'un des livres les plus importants concernant les chaînes logistiques, Chopra et Meindil (Chopra et Meindil, 2007) donnent la définition suivante : « une chaîne logistique consiste en toutes les étapes impliquées directement ou indirectement dans la satisfaction de la requête d'un client. La chaîne logistique inclut non seulement le fabricant et ses fournisseurs, mais aussi les transporteurs, les centres d'entreposage, les détaillants et les clients eux mêmes ».

En résumant ces définitions on peut voir qu'il existe deux visions de la chaîne logistique, l'une basée sur l'entreprise, et la seconde basée sur le produit. Un modèle de chaîne logistique basée sur une entreprise peut être donné par la figure suivante (Kearney, 1994)

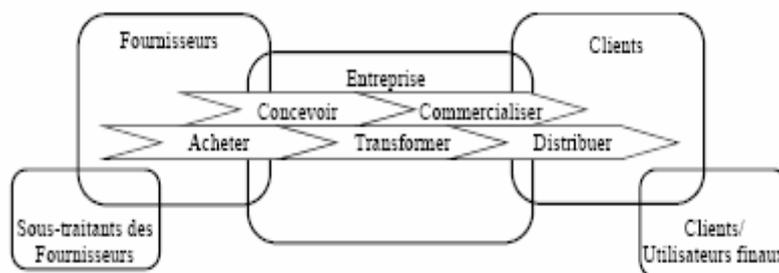


Figure 1.2 modèle de chaîne logistique (Kearney, 1994)

Lorsque le concept de chaîne logistique est abordé du point de vue de l'entreprise, alors on considère toutes les chaînes logistiques qui incluent l'entreprise considérée en se limitant

parfois aux fournisseurs et aux clients de l'entreprise, voire aux fournisseurs des fournisseurs et aux clients des clients (Thierry, 2003).

Au delà d'une entreprise donnée, Rota et al (Rota et al, 2001) définissent la chaîne logistique d'un produit donné (ou d'une famille de produits) comme « l'ensemble des entreprises qui interviennent dans les processus de fabrication, de distribution et de vente du produit, du premier des fournisseurs au client ultime ». Ceci nous amène à voir les différents types d'organisation des chaînes logistiques.

1.2.2 Types de chaînes logistiques

On a vu que la chaîne logistique est constituée d'un ensemble d'acteurs ou de partenaires qui achètent, produisent (développent, ou transforment), éventuellement stockent, transportent et vendent des biens ou des services de telle sorte que, sous des contraintes pouvant faire intervenir des facteurs humains, sociaux, juridiques ou encore environnementaux, la marge bénéficiaire revenant à chaque partenaire soit satisfaisante pour chacun d'eux et que la somme des marges soit la plus grande possible. Les typologies des chaînes logistiques diffèrent selon les propriétés des acteurs qui y interviennent.

Si les sites sont localisés dans différent pays, on parle alors de chaîne logistique globale. Dans ce cas, les aspects relatifs à l'importation et à l'exportation comme le taux de change, les taxes douanières, les assurances, et les législations doivent être pris en compte. De nos jours, une grande partie des chaînes logistiques sont globales, c'est une des conséquences de la globalisation. La figure suivante montre un exemple d'une chaîne logistique typique.

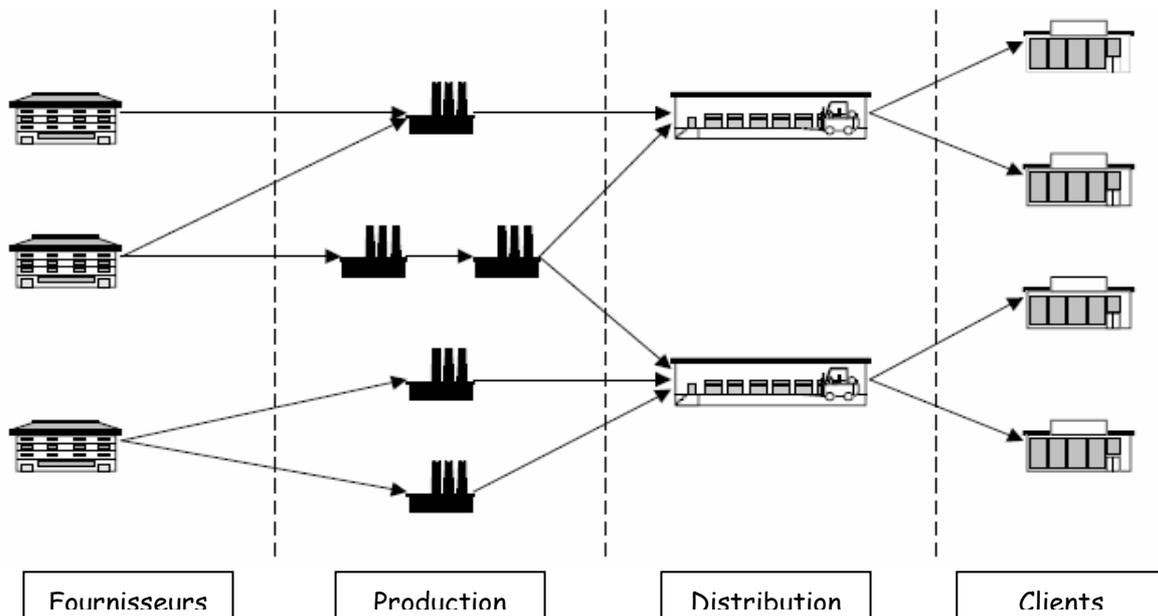


Figure 1.3 chaîne logistique globale.

Si les partenaires appartiennent tous à la même entité juridique (même si l'entreprise est multi sites) alors on parle de chaîne logistique interne.



Figure 1.4 chaîne logistique interne. Source (Portmann, ISDP32).

Si plusieurs entreprises collaborent au sein de la chaîne logistique, mais l'une d'entre elle joue un rôle dominant et central, on parle alors d'une entreprise étendue. Browne et al (Browne et al, 1995) ont défini l'entreprise étendue comme étant une organisation ou une forme d'entreprise représentée en tout ou en partie par les clients, les fournisseurs et les sous traitants engagés d'une façon collaborative pour la conception, le développement, la production et la livraison de produits pour les utilisateurs finaux. Pour un fonctionnement correct, le donneur d'ordre (l'entreprise dominante) doit communiquer sur ses prévisions, en général, des contrats sont signés entre les partenaires sur des fourchettes de volumes par période ou par cumul de périodes. Il développe ainsi des relations de partenariat à long terme avec des clients potentiels, il les traite comme d'importants partenaires. En outre, il sous traite tout ce qui sort de ses compétences à des fournisseurs externes ou à des prestataires de services, et se concentre sur les compétences de l'entreprise (Browne et al, 1999). La figure suivante montre un schéma d'une entreprise étendue.

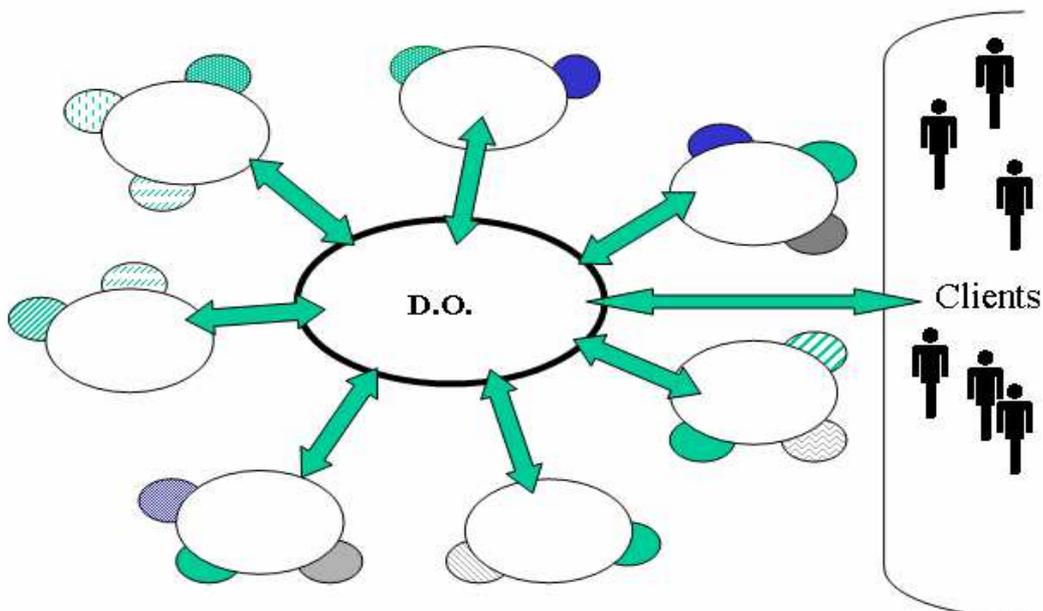


Figure 1.5 entreprise étendue. Source (Portmann, ISDP32).

Dans le cas où plusieurs entreprises collaborent au sein de la chaîne logistique, mais où le pilotage est décentralisé ou est au minimum semi décentralisé avec des négociations bilatérales entre couples de partenaires, on parle alors d'une entreprise virtuelle. Makastoris et al (Makastoris et al, 1996) ont défini l'entreprise virtuelle comme un ensemble d'unités et de processus au sein d'une chaîne logistique se composant d'un ensemble d'unités de production coopérant entre elles, des magasins, et des unités de transport qui se comportent comme une entreprise simple avec une forte coordination pour atteindre des objectifs communs. La réussite d'une telle organisation dépend de l'efficacité du système de partage d'informations utilisé afin d'assurer l'intégration des différents partenaires pendant une courte durée (Browne et Zang, 1999). Des entreprises peuvent se regrouper en une entreprise virtuelle pour faire face à la concurrence ou bien pour atteindre des capacités de production qu'elles ne peuvent atteindre seules (Monteiro, 2001). L'entreprise virtuelle cherche à exploiter des opportunités volatiles, à accéder à de nouveaux marchés et à partager les coûts et les risques, ceci sans superstructure organisationnelle importante, en recourant aux nouvelles possibilités offertes par les technologies de l'information et de la communication. Le concept d'entreprise virtuelle peut être résumé comme le regroupement d'entreprises dans le sens d'alliances stratégiques ou opportunistes. Une telle organisation implique des relations de confiance et une compréhension mutuelle de la manière à traiter les affaires, ainsi que de partager sans restrictions des informations confidentielles.

A la différence des entreprises étendues où le contrôle est purement hiérarchique, au niveau de l'entreprise virtuelle le contrôle est non hiérarchique et dans certains cas, pour éviter des conflits, il est possible de trouver un coordinateur. La figure suivante montre un schéma d'une entreprise virtuelle à niveaux.

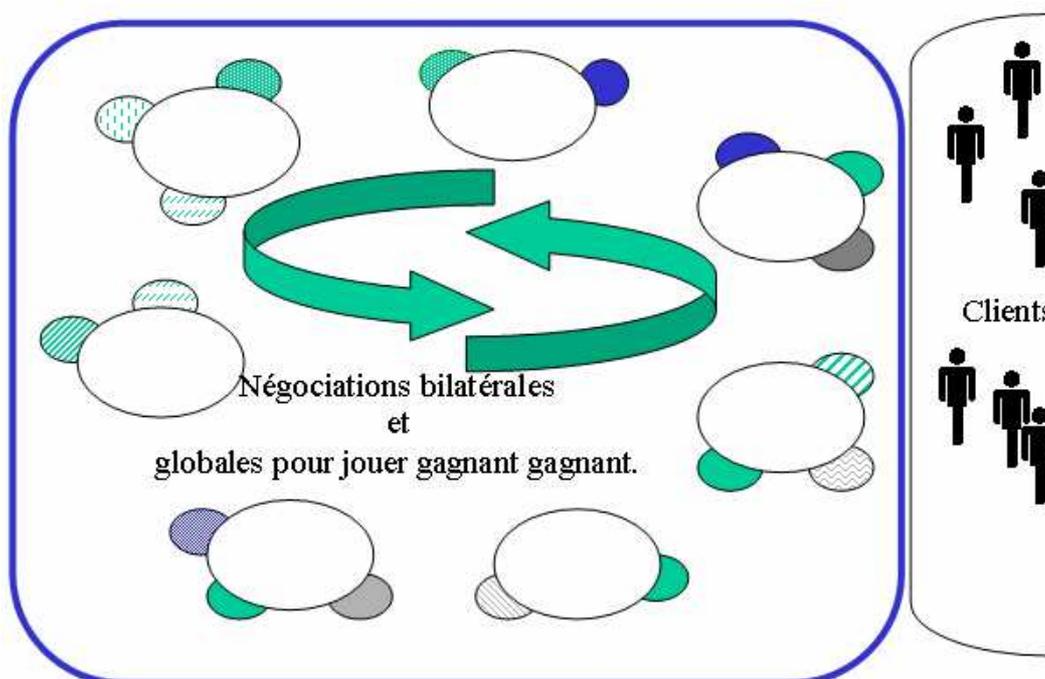


Figure 1.6 entreprise virtuelle. Source (Portmann, ISDP32).

1.2.3 Fonctions de la chaîne logistique

La définition suivante de la chaîne logistique donnée par Ganeshan and Harisson (Ganeshan et Hrisson, 1995) donne un aperçu des fonctions de la chaîne logistique : « une chaîne logistique est le réseau des moyens de production et de distribution qui assurent les tâches d'approvisionnement en matières premières, la transformation de ces matières premières en produits semi finis et en produits finis, et la distribution de ces produits finis aux clients ». Plus généralement, les fonctions d'une chaîne logistique vont de l'achat des matières premières à la vente des produits finis en passant par la production, le stockage et la distribution.

1.2.3.1 L'approvisionnement

Il constitue la fonction la plus en amont de la chaîne logistique. Les matières et les composants approvisionnés constituent de 60% à 70% des coûts des produits fabriqués (Ouzizi, 2005) dans une majorité d'entreprises. Réduire les coûts d'approvisionnement contribue à réduire les coûts des produits finis, et ainsi à avoir plus de marges financières. Les délais de livraison des fournisseurs et la fiabilité de la distribution influent plus que le temps de production sur le niveau de stock ainsi que la qualité de service de chaque fabricant (Harmon, 92). La tendance générale des relations clients/fournisseurs va vers plus de coopération via un partage d'informations plus rapide en utilisant les nouveaux systèmes d'informations basés sur les technologies de l'information et de communication qui ont révolutionné les pratiques du passé où on était plutôt dans une configuration de face à face plutôt qu'une collaboration bénéfique pour l'ensemble des participants.

1.2.3.2 La production

La fonction de production est au cœur de la chaîne logistique, il s'agit là des compétences que détient l'entreprise pour fabriquer, développer ou transformer les matières premières en produits ou services. Elle donne quelle capacité a la chaîne logistique pour produire et donne ainsi un indice sur sa réactivité aux demandes fluctuantes du marché. Si les usines ont été construites avec une grande capacité de production, parfois excessive, alors on peut être réactif à la demande en présence de quantités supplémentaire à faire, cet environnement a l'avantage d'être disponible pour des clients en cas de demandes urgentes, mais d'un autre côté une partie de la capacité de production peut rester inactive ce qui engendrent des coûts et dépenses en plus. D'un autre côté si la capacité de production est limitée, la chaîne logistique a du mal à être très réactive et donc peut perdre des parts du marché vu qu'elle n'est pas capable de répondre favorablement à certaines demandes. Il faut donc trouver un équilibre entre réactivité et coûts.

1.2.3.3 Le stockage

Le stockage inclut toutes les quantités stockées tout au long du processus en commençant par le stock de matières premières, le stock des composants, le stocks des en-cours et finalement le stock des produits finis. Les stocks sont donc partagés entre les différents acteurs : les fournisseurs, le producteurs et les distributeurs. Ici aussi se pose la question de l'équilibre à trouver entre une meilleure réactivité et la réduction des coûts. Il est évident que plus on a de stocks, plus la chaîne logistique est réactive aux fluctuations des demandes sur le marché. Cependant, avoir des stocks engendre des coûts et des risques surtout dans le cas de produits périssables ou bien des produits dont la rapidité d'innovations est telle qu'une nouvelle gamme du même produit mise sur le marché par un concurrent puisse rendre obsolètes les quantités de ce produit en stock et ainsi une perte importante. La gestion des stocks est l'une des clés de la réussite et l'optimisation de toute une chaîne logistique. Une

meilleure gestion de cette fonction peut engendrer des économies importantes, l'expérience de DELL dans ce domaine est un exemple très connu. En outre, avec l'avènement des techniques de management dite de « juste à temps » (just in time) beaucoup d'entreprises tendent vers avoir un stock nul, ou bien « juste » ce qu'il faut pour produire et satisfaire les commandes. Mais ceci n'est évidemment pas sans risques.

1.2.3.4 Distribution et transports

La fonction transport intervient tout au long de la chaîne, le transport des matières premières, le transport des composants entre les usines, le transport des composants vers les centres d'entreposage ou vers les centres de distribution, ainsi que la livraison des produits finis aux clients. Le rapport entre la réactivité de la chaîne et son efficacité peut être aussi vu par le choix du mode de transport. Les modes de transport les plus rapides comme par exemple les avions, sont très coûteux, mais permettent de réagir très vite et ainsi de satisfaire les demandes non prévisibles. Les modes de transport par voies ferrées ou par camions sont plus efficaces du point de vue des coûts engendrés mais moins rapides. L'ensemble des partenaires peut choisir de combiner ces modes de transport et de les adapter à certaines situations selon l'importance de la demande et le gain total engendré.

Les problèmes liés à la distribution et au transport peuvent être vus sous plusieurs angles. On peut chercher à trouver les meilleures routes possibles pour visiter les point de collecte et/ou de distribution (Vehicle routing problems, problèmes de tournées des véhicules), ou bien, comme dit plus haut, chercher les meilleurs modes de transports, ou bien les quantités des produits qui doivent être transportées aux clients tout en minimisant le coût global des transports et des stocks. En effet, selon les études (Hugos, 2003) les coûts de transport et distributions constituent le tiers des coûts opérationnels globaux d'une chaîne logistique, ce qui rend leur optimisation un défi majeur pour les entreprises.

1.2.3.5 La vente

La fonction de vente est la fonction ultime dans une chaîne logistique, son efficacité dépend des performances des fonctions en amont. Si on a bien optimisé pendant les étapes précédentes, alors on facilite la tâche du personnel chargé de la vente, car ils pourront offrir des prix plus compétitifs que la concurrence, sinon les marges seront très étroites et les bénéfiques pas très importants, voire même engendrer des pertes.

1.2.4 Les décisions dans la chaîne logistique

Une décision peut être définie comme étant le problème de donner une valeur à une variable inconnue et dont la connaissance permet au décideur de sortir d'une situation de jugement ou d'incertitude (Ouzizi, 2005). La conception d'une chaîne logistique nécessite de prendre un ensemble de décisions. Cet ensemble de décisions peut s'envisager sur trois niveaux hiérarchique : décisions stratégiques, décisions tactiques, et décisions opérationnelles. La figure suivante (1.7) montre un tel schéma. Une telle hiérarchie est basée sur la portée temporelles des activités et sur la pertinence des décisions.

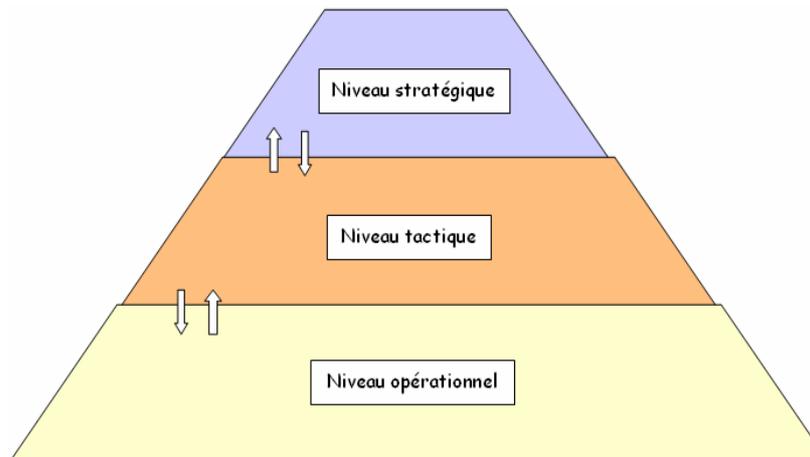


Figure 1.7 pyramide des niveaux de décisions.

Il n'existe pas de méthode générique valable pour toutes les chaînes logistiques et toutes les industries pour classer les décisions qui doivent être prises. La première différence entre ces trois niveaux de décision concerne la portée temporelle de l'application de la décision. Les décisions stratégiques sont à long terme, les décisions tactiques sont à moyen terme, et enfin les décisions opérationnelles sont à court terme. La deuxième différence entre ces niveaux de décisions est le niveau d'agrégation : les décisions stratégiques sont au niveau de l'ensemble de l'entreprise, les décisions tactiques sont prises au niveau de l'usine, et les décisions opérationnelles sont prises au niveau de l'atelier. La troisième différence est le niveau de responsabilité des décideurs. Les décisions stratégiques sont prises par la direction générale de l'entreprise, les décisions tactiques sont prises par les cadres, et les décisions opérationnelles sont prises par les responsables d'ateliers.

A cause de la complexité du problème d'optimisation des décisions, les trois types de décisions sont traités de manière séquentielle et hiérarchique. Néanmoins, il est important de prendre en compte l'impact des décisions stratégiques sur les niveaux tactiques et opérationnel. En effet, elles déterminent les solutions admissibles des niveaux tactique et opérationnel. Autrement dit, la solution optimale d'une décision tactique ou opérationnelle dépend de la solution prise au niveau stratégique. De la même manière, les décisions opérationnelles et tactiques peuvent influencer la prise de décisions au niveau stratégique lors de la conception même de la chaîne. Cette coordination est souhaitée mais difficile à mettre en œuvre comme nous allons le voir dans la section dédiée à la coordination et la coopération dans les chaînes logistiques. La figure suivante (1.8) illustre les différences entre les trois niveaux de décisions.

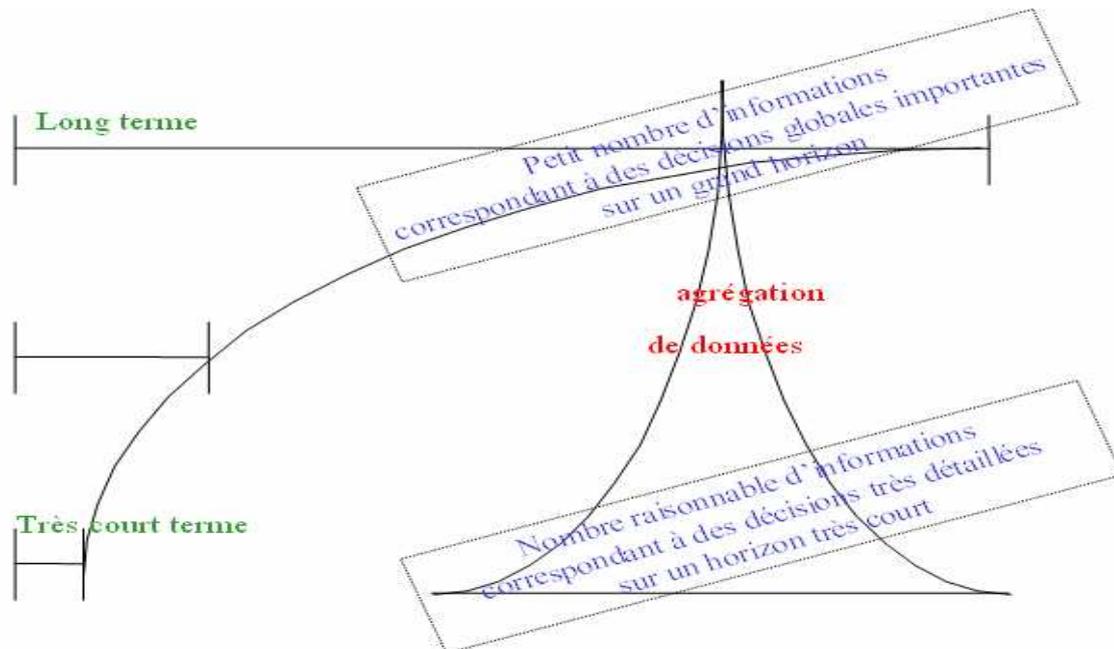


Figure 1.8 différence entre les niveaux de décisions. Source (Portmann, ISDP32).

1.2.4.1 Les décisions stratégiques

Les décisions stratégiques définissent la politique de l'entreprise sur le long terme, une durée s'étalant souvent sur plusieurs années (la durée de l'horizon dépend du cycle de vie des produits). Elles comprennent toutes les décisions de conception de la chaîne logistique et de ce fait, elles ont une influence importante sur la stratégie concurrentielle et donc sur la viabilité à long terme de l'entreprise. Elles sont prises normalement par la direction de l'entreprise. Les décisions stratégiques configurent la chaîne logistique. Nous donnons dans ce qui suit une liste non exhaustive des décisions stratégiques :

- Choisir les partenaires de la chaîne logistique (cas d'entreprises étendues ou virtuelles). Recherche de la complémentarité des compétences (toutes les fonctions doivent pouvoir être assumées en interne ou en externe (sous-traitance à l'extérieur de la chaîne logistique constituée)).
- Faire ou faire-faire : l'entreprise a le choix entre utiliser ses propres moyens pour réaliser en interne certaines fonctions (faire), ou bien passer par une entreprise extérieure et indépendante (faire-faire), ou bien déléguer ces tâches à une entreprise qui sous une forme ou une autre a des liens privilégiés avec elle.
- Choix et nombre de fournisseurs : l'entreprise peut avoir un seul fournisseur ou un nombre réduit de fournisseurs pour augmenter le niveau de coopération, ou bien avoir un grand nombre de fournisseurs pour jouer sur la concurrence. Les fournisseurs sont choisis en fonction des prix, qualités de service, délais de livraison...etc. Barbarosoglu et Yazgac (Barbarosoglu et Yazgac, 1997) regroupent les critères de choix des fournisseurs en trois catégories : i) la capacité technique et l'état financier du fournisseur; ii) l'historique des performances du fournisseur ; iii) la qualité du système du fournisseur.
- Choisir les implantations des sites de production et des entrepôts. Cela inclut aussi la décision d'affecter les activités aux sites. Les décisions concernant la localisation des sites de production sont très importantes et très stratégiques car elles conditionnent les décisions de transport et de distribution. Plusieurs facteurs doivent

être pris en compte lors de la prise de telles décisions comme la proximité par rapports aux clients et aux fournisseurs, les taxes et tarifs, et la disponibilité de la main d'œuvre.

- Déterminer le nombre de sites : un nombre élevé de sites de productions ou de stockage engendre des coûts colossaux, en même temps cela réduit les coûts de transports. Les entreprises doivent choisir entre des politiques de groupages de sites ou au contraire des politiques de dégroupage.
- Capacité des sites : cette problématique est liée à celle du nombre de sites. Une capacité très grande engendre une réactivité très grande mais aussi des coûts très grands (surtout en cas de sous-production).
- Choisir les moyens de transport (éventuellement multiples) entre les différentes localisations. Cela dépend aussi du nombre et de la localisation des sites. Plus le nombre de sites est grand, plus on est proche des clients, et plus on utilise des modes de transport économiques.
- Le choix des technologies utilisées dans les sites de production et d'entreposage. Cette décision est liée à la capacité des sites car celle ci dépend de la technologie utilisée. Cette décision dépend de certains critères économiques, sociaux et financiers.

1.2.4.2 Les décisions tactiques

Les décisions tactiques sont prises sur un horizon de moins de 18 mois en général. Il s'agit de produire au moindre coût pour les demandes prévisibles, donc avec connaissance des ressources matérielles et humaines. Il s'agit en effet de faire la planification dépendant de la structure conçue au niveau stratégique. Nous donnons dans ce qui suit une liste non exhaustive des décisions tactiques :

- Obtenir les prévisions les plus fiables possibles. Les quantités à produire pour chaque produit et les quantités des matières premières nécessaires.
- Choisir les modes d'utilisation des ressources (par exemple: ouverture en 2 huit ou en 3 huit, faire ou non appel à de la sous-traitance ou à des heures supplémentaires).
- Trouver une allocation optimale des fournisseurs aux sites de production.
- Allocation des produits aux sites de production et déterminer les quantités à produire sur chaque site en tenant compte de la capacité de production de chaque site et des moyens de transports qui le desservent.
- Planifier la production à tous les niveaux (lissage et équilibrage de charge et minimisation des coûts) et les transports associés, ainsi que la maintenance des outils de production et des moyens de transport.
- Gérer tous les stocks induits (matières premières, encours, produits finis, pièces de rechanges, outils ...) ainsi que trouver l'allocation optimale des produits aux sites de stockage en prenant en compte le coût de stockage de chaque produit dans chaque site, les capacités des sites de stockage, et les coûts de transports entre les sites de production des produits et les sites de stockage.
- Allocation des sites de stockage aux clients pour optimiser le plus souvent les coûts de transports ou de livraisons.
- Définir la politique de transport : on doit décider si les livraisons aux clients se font de manière individuelle pour chaque client ou bien on essaye de regrouper les livraisons pour livrer le plus de clients possibles lors d'une même tournée. Cette décision dépend

du mode de transport et de la quantité demandée par chaque client. Le transport des matières premières et des produits semi finis est traditionnellement séparé du transport des produits finis car ils sont réalisés par deux entités différentes. L'intégration de ces deux entités ou bien la coordination entre elles peut permettre une meilleure utilisation des ressources en transport de l'entreprise.

1.2.4.3 Les décisions opérationnelles

Les décisions opérationnelles sont prises pour un horizon de très court terme pour assurer la gestion des moyens et le fonctionnement au jour le jour de la chaîne logistique. Dans le cadre des chaînes logistiques, les entreprises ont besoin à tout moment de prendre des décisions avec un temps de réponse très court. La réactivité de la prise des décisions opérationnelles est un élément de mesure de la performance de la chaîne logistique. Au niveau opérationnel, la configuration de la chaîne logistique est déjà fixée et les politiques de planifications déjà définies. Il y a moins d'incertitudes sur les informations sur la demande car on doit prendre les décisions opérationnelles en un laps de temps très court (minutes, heures, jours). Avec moins d'incertitudes, l'objectif à ce niveau est de répondre aux requêtes des clients d'une façon optimale en respectant les contraintes établies par les configurations et les politiques de planification choisies aux niveaux stratégiques et tactiques. Nous donnons dans ce qui suit une liste non exhaustive des décisions opérationnelles :

- Ordonnancement et pilotage en temps réel des systèmes de production.
- Tournée de véhicules ou programme des livraisons qui donne les produits, la destination et les quantités à livrer.
- Allocation des moyens de transports : ces moyens étant limités, cette allocation est basée sur le programme des livraisons.
- Placement plus précis des activités de maintenance préventive.
- Affectation des ressources aux tâches (matérielles et humaines) de manière dynamique en fonction des aléas. Il s'agit de faire l'emploi du temps des employés en tenant compte des contraintes sociales et juridiques.

1.2.5 La gestion de la chaîne logistique ou supply chain management

Le terme « supply chain management » qui signifie gestion de la chaîne logistique, la version anglaise du terme est celle utilisée au niveau international, même dans les publications francophones, nous utilisons donc la version anglaise. Il est apparu dans les années 1980 et s'est largement répandu dans les années 1990. Avant, on utilisait les termes de « logistique » et de « gestion des opérations » (operations management). Dans un environnement logistique complexe émerge le besoin d'un outil pour appréhender et modéliser cette complexité et apporter une aide à la décision. En effet, les anciens outils de gestion de production, de planification et de pilotage des entreprises sont devenus insuffisants car dépassés par les demandes nouvelles et la réorganisation des entreprises en réseaux. En effet, l'émergence de la chaîne logistique a fait naître des besoins en matière d'intégration des entreprises et de coordination des flux des matières, des flux d'informations et des flux financiers à des niveaux jamais atteints auparavant. C'est le supply chain management qui englobe les approches, les méthodes et les outils permettant de satisfaire ces besoins.

L'adoption de la démarche supply chain management apparaît comme un outil de performance pour l'entreprise, puisque son ambition affichée est de répondre au triple objectif d'amélioration des niveaux de services, de réduction des coûts et de création de valeur, en gérant les relations, tant en amont qu'en aval, avec les fournisseurs et les clients. La création

de valeur qui est la finalité du supply chain management devient la résultante de l'optimisation d'un processus destiné à vendre plus en répondant davantage aux attentes des clients mais aussi à organiser au mieux le processus de production et de circulation des flux afin de réduire les coûts. Il est difficile d'identifier une définition unique et acceptée par tous du supply chain management, la démarche peut être envisagée comme une philosophie, une orientation et une logique de gestion, une vision de l'entreprise en réseau où la coordination parmi les acteurs garantit la baisse des coûts et augmente la qualité de service au consommateur final. Une des raisons de l'absence d'une définition universelle du supply chain management est l'origine et l'évolution multidisciplinaire de cette notion. Nous donnons dans ce qui suit les définitions les plus courantes.

Mentzen et al (Mentzen et al, 2001) définissent le supply chain management comme « la coordination systémique, stratégique des fonctions opérationnelles classiques et de leurs tactiques respectives à l'intérieur d'une même entreprise et entre partenaires au sein de la chaîne logistique, dans le but d'améliorer la performance à long terme de chaque entreprise membre et de l'ensemble de la chaîne ». Cette définition est générale centrée sur l'entreprise, et considère donc le cas d'une entreprise qui appartient à plusieurs chaînes logistiques, ce qui est souvent le cas dans l'économie mondiale actuelle. Dans ce cas on peut se poser la question légitime de qui gère la chaîne logistique ? dans le cas d'une organisation avec une entreprise dominante (donneur d'ordre), c'est clair que c'est cette dernière qui impose ses contraintes, mais de nos jours les chaînes logistiques intègrent plus de systèmes de partenariat d'égal à égal, et dans ce cas on parle alors de chaîne logistique collaborative.

(Hugos, 2003) donne la définition suivante : « le supply chain management est la coordination de la production, du stockage, de la localisation, et des transports à travers tous les participants à la chaîne logistique afin d'avoir la meilleure combinaison de réactivité et d'efficacité par rapport au marché desservi ».

La différence entre le concept de supply chain management et le traditionnel concept de logistique est, d'une part, que la logistique se limitait au cadre d'une seule entreprise alors que le supply chain management gère un réseau d'entreprises, et d'autre part la logistique traditionnelle se concentrait sur les fonctions d'approvisionnement, distribution, gestion de stock, alors que le supply chain management reprend ces fonctions et inclut les activités de marketing, développement de nouveaux produits, les finances et le service du client. La logistique devient donc l'une des fonctions de la gestion de la chaîne logistique, certainement l'une des plus importantes (New et Payne, 1995). Ceci nous amène à une définition du supply chain management centrée sur les activités de distribution physique des produits finis aux clients finaux intégrant la gestion de la logistique. Le terme « supply chain management » a été probablement utilisé pour la première fois dans de telles configurations (Lamming, 1996). Pour plus de précisions sur les définitions (Tan, 2001) propose un état de l'art intéressant.

Toutes ces définitions ont un point commun qui est que le supply chain management est le processus qui intègre toutes les fonctions de la chaîne logistique, et avec une vision globale des choses, c'est-à-dire qu'il voit le tout comme une seule entité même s'il s'agit d'une organisation hétérogène juridiquement. L'intégration de toutes ces fonctions dans le même système permet de faire une optimisation tout au long de la chaîne, alors que, prises individuellement, les différentes fonctions de la chaîne logistique présentent des objectifs différents. Par exemple, pour avoir une meilleure qualité de service pour les clients il est nécessaire d'avoir un taux de stockage important, alors que pour réduire les coûts opérationnels on veut réduire les quantités en stock. L'optimisation simultanée de la qualité de service et les coûts opérationnels internes en trouvant un compromis est l'un des critères de mesure de l'efficacité d'un système de management de la supply chain.

On a vu que le supply chain management gère entre autres le flux des informations, ceci est primordial pour la prise de décisions. La qualité des décisions et leurs impacts sur la suite des décisions (on a vu dans la section des décisions dans la chaîne logistique que les décisions peuvent avoir des conséquences en cascade sur les autres niveaux de décisions) sur les autres niveaux dépend de la qualité de l'information. Celle-ci est critique quant au succès de tout système de supply chain management. Beaucoup de décisions aux niveaux tactiques sont basées sur les prévisions, si celles-ci sont loin de la réalité cela engendre des pertes colossales pour la chaîne logistique. Il est important aussi que les partenaires puissent partager des informations et qu'ils aient une démarche de confiance et non de méfiance (Hadfield, 2002). Plusieurs recherches ont montré l'impact du partage du flux d'information sur l'amélioration de plusieurs critères de la chaîne comme la minimisation des niveaux de stocks (Chopra et Meindil, 2007), (Bourland et al, 1996) ; l'amélioration de la qualité de service (Thonemann, 2002) (Zhao et al, 2002) ; l'amélioration des planning de toute la chaîne (D'amours, 1999) (Johansson, 2002) (Kovacs et al, 2003). L'une des clés de la réussite du management global de la chaîne logistique est la capacité des partenaires à partager l'information et la synchronisation de leurs activités (Ouzizi, 2005). Le système d'informations utilisé joue donc un rôle primordial.

1.2.6 Optimisation de la chaîne logistique

La chaîne logistique optimise la gestion des flux, du fournisseur du fournisseur au client du client pour le cas le plus élargi, autrement dit de l'approvisionnement en matières premières au consommateur final. Il s'agit dans le contexte actuel de globalisation des marchés, de coordonner tous les processus pour atteindre le niveau de performance qui convient à l'entreprise. Comme la chaîne logistique rassemble plusieurs partenaires, il est évident que c'est complexe de trouver une optimisation qui satisfasse tout le monde surtout que les objectifs sont souvent conflictuels entre les différentes fonctions de la chaîne. La première remarque qu'on peut faire dans ce sens est qu'un optimum global est rarement obtenu en rassemblant des optima locaux. Néanmoins, si les gains que l'on obtient localement n'ont pas d'incidence sur les coûts des autres services ou partenaires, alors cela vaut la peine de rechercher un optimum local (sous contraintes ou en pénalisant les incidences néfastes sur l'environnement ou sur les partenaires internes ou externes à la chaîne logistique).

Il est évident que l'intérêt général de tous les partenaires est de se positionner dans une posture collaborative plutôt qu'une posture concurrentielle. Il faut savoir minimiser les coûts globaux et maximiser les gains globaux. Cette minimisation peut demander des efforts d'organisation ou de réorganisation qui peut passer par un développement d'un alignement stratégique, c'est à dire la recherche de procédures collaboratives de gestion pour mettre à la disposition du client le bon produit au bon moment (Paché, 2004).

Cette optimisation rencontre des problèmes dans sa mise en œuvre qui freine la démarche gagnant-gagnant et rendent l'approche optimisante. Dans ce qui suit nous allons énumérer certains de ces problèmes (très fortement inspiré de (Portmann, ISDP 32)) :

- La qualité de l'information : Trop d'informations sont imprécises, incertaines et même inconnues ou imprévisibles comme les durées des opérations et des transports qui sont seulement des valeurs estimées plus ou moins précises, les pannes partielles ou totales des machines, absences du personnel, grèves des routiers ou des transports ferroviaires, ou encore des matières premières non arrivées ou non conformes.
- Les décisions concernent de nombreux services dans la (ou les) entreprise(s) concernée(s) ce qui peut engendrer des problèmes de conflits et d'indicateurs de

performance qui doivent conduire à un comportement global gagnant-gagnant, problèmes de confiance, de sécurité, de partage et de circulation d'information.

- Les critères de performance et de qualité sont antagonistes en raison des différents acteurs concernés: actionnaires, clients, employés.
- Il y a une multitude de systèmes de production et de distribution différents qui doivent être considérés avec leurs hypothèses particulières ce qui nuit à l'existence de solution générique pour les outils d'aide à la décision. La diversité des procédés manufacturiers: fabrication continue, en série, production de produits personnalisés à la demande ou bien de prototypes. La diversité des ressources de production : renouvelables, non renouvelables. La diversité des gammes et des structures d'ateliers : flowshop, openshop, jobshop. La diversité des machines : une machine pour chaque type d'opérations, des machines en parallèle pour le même type d'opérations. La diversité des critères de performances : « makespan » (ou durée totale), volume en sortie, critères de retard (maximum, somme, somme pondérée, nombre de travaux en retard), minimisation de coûts (heures supplémentaires, sous-traitance ...).
- Contraintes de production : préemption autorisée ou non autorisée, chevauchement autorisé ou non entre les opérations (fabrication en séries), fabrication par lots ou « batches » de type maximum (produits en parallèles dans un four) ou de type somme (exécution successive où les produits doivent arriver et repartir ensemble), temps de transport plus ou moins précis à prendre en compte, limitation des places de stockages (jusqu'à pas de stock entre deux machines consécutives), production cyclique ou à la commande.
- Diversité des systèmes de transport et de distribution.
- Vouloir absolument un optimum global parfait conduit souvent à des calculs exponentiels qui ont des durées tellement grandes que ça devient inimaginable de les prendre comme solutions (problèmes NP-difficiles à résoudre), alors que l'on travaille souvent sur des données imprécises. Ce qui implique que les méthodes de résolutions exactes ne sont considérées que pour des problèmes de petite taille, des méthodes approchées le plus souvent pour les problèmes de taille industrielle.
- La séparation entre la conception de la chaîne logistique et les décisions opérationnelles : les changements de la structure d'un réseau ont des répercussions sur l'efficacité des décisions au niveau opérationnel, celles-ci doivent être considérées dès la conception de la chaîne.
- Chaîne logistique incomplète : il faut aller au delà de la chaîne logistique interne et considérer les clients des clients et les fournisseurs des fournisseurs, ceci permet à la chaîne d'avoir une meilleure connaissance sur les besoins des clients des clients et ainsi de s'adapter afin d'améliorer sa qualité de service.

Pour faire face à ces problèmes, des nouvelles relations de partenariats entre fournisseurs et clients sont à définir. Ces relations doivent être durables, et permettre aux entreprises de s'intégrer et coordonner la prise de décisions au niveau global de la chaîne, ceci dans le souci d'être réactif et efficace face aux mouvements des marchés qui restent très imprévisibles.

1.2.7 Modélisation des chaînes logistiques

La chaîne logistique étant un système complexe et dynamique, ajouté à cela un environnement instable qui génère de nombreuses incertitudes. La difficulté de la prise en

compte de ces incertitudes fait que la plupart des modèles proposés pour modéliser les chaînes logistiques utilisent des hypothèses restrictives, et parfois simplistes (Vidal et Goetschalckx, 1997). La modélisation de ces systèmes complexes permet une meilleure compréhension et une meilleure gestion de ces systèmes. Un modèle n'est qu'une représentation simplifiée d'un système réel, qui permet de l'analyser, le contrôler et le piloter. Ils sont à la base des systèmes d'aide à la décision. Nous allons voir trois types de modélisations : modèles conceptuels, modèles mathématiques, et modèles par simulation (Taylor, 2003).

1.2.7.1 Les modèles conceptuels

Les modèles conceptuels sont de loin les plus simples. Il s'agit en fait d'une description basique d'un système économique comme la chaîne logistique qui peut s'exprimer sous formes de diagrammes ou d'explications verbales. Le format utilisé dépend en grande partie de l'expérience du modélisateur, ceux avec une grande expérience font des diagrammes détaillés pour réduire l'ambiguïté, tandis que ceux avec une moindre expérience se basent sur une analyse par scénario. Dans ces modèles, il faut trouver un bon équilibre entre précision et aisance de communication. Ces modèles sont limités car difficiles à mettre en œuvre dans le cas d'organisations très complexes, et surtout ils ne donnent pas d'orientations quant au contrôle et au pilotage de la chaîne.

1.2.7.2 Modèles mathématiques

Les modèles mathématiques sont très utilisés pour la conception des chaînes logistiques et pour l'optimisation des coûts. Ils consistent à modéliser un système réel par un ensemble d'équations exprimant les contraintes et les objectifs. Contrairement aux modèles conceptuels qui eux aident seulement à la compréhension du système, les modèles mathématiques résolvent les problèmes d'optimisation. Une autre différence avec les modèles conceptuels est que l'utilisation des modèles mathématiques requiert des compétences spéciales dans les mathématiques et la recherche opérationnelle.

L'une des techniques les plus utilisées est la programmation linéaire et la programmation dynamique. Ces outils de recherche opérationnelle sont à la base de beaucoup de systèmes d'optimisation des supply chain management. L'inconvénient avec les modèles mathématiques est qu'ils font des restrictions trop importantes sur certaines hypothèses. Un autre inconvénient, et pas des moindres, est le temps d'exécution nécessaire pour résoudre des problèmes de tailles réalistes. Ainsi, pour des problèmes avec de grandes tailles, les industriels préfèrent utiliser des solutions approchées obtenues dans des délais raisonnables.

1.2.7.3 Modèles par simulation

Les modèles par simulations sont très pratiques dans le cas de systèmes où il est difficile de représenter toutes les hypothèses par des équations, et de ce fait, on ne peut pas utiliser les modèles mathématiques. Ces modèles essaient d'imiter le comportement des composants d'un modèle et donc de pouvoir faire des prévisions et des évaluations de performances. Ils ont la capacité de capturer les incertitudes et de traiter l'aspect dynamique des systèmes complexes et des systèmes à grandes échelles.

De nombreux modèles ont été proposés pour la simulation et la modélisation des chaînes logistiques. Hermann et al (Hermann et al, 2003) proposent un nouveau cadre de simulation et des modèles hiérarchiques pour capturer les activités spécifiques au sein de la chaîne logistique. Dans sa thèse, Ding (Ding, 2004) propose une approche d'optimisation basée sur la simulation pour la conception des chaînes logistiques appliquée à l'industrie automobile et textile. Beaucoup de travaux de recherches se sont intéressés à ces modèles, nous pouvons citer (Jain et al, 2001) et (Bhasharan, 1998).

Le tableau ci dessous illustre les différences entre les trois types de modélisation.

	Modèles conceptuels	Modèles mathématiques	Modèles par simulation
Représente la chaîne comme	<i>Diagrammes et descriptions</i>	<i>Formules et équations</i>	<i>Objets et interactions</i>
Solutions trouvées par	<i>Raisonnement verbal</i>	<i>Les solveurs (comme Cplex ou Express)</i>	<i>Expériences (monte carlo)</i>
Meilleure application pour	<i>Partage de la compréhension</i>	<i>Performances optimales</i>	<i>Prévisions réalistes</i>

Figure 1.9 différence entre les types de modélisations des chaînes logistiques (Taylor, 2003)

1.2.8 Les outils d'aide à la décision

Nous avons vu dans les sections précédentes la complexité des entreprises actuelles et leur organisation en chaînes logistiques. Celles-ci comprennent plusieurs partenaires et réalisent plusieurs fonctions. Nous avons vu que le supply chain management est le processus qui intègre toutes ces fonctions, tous les partenaires, coordonnent leurs activités en vue de l'obtention du meilleur résultat pour le collectif. La taille immense d'un tel système, la diversité des données, et l'antagonisme des objectifs des différentes fonctions font sentir le besoin de systèmes d'aide à la décision capable de prendre en compte tous ces paramètres. De nos jours, il existe une très grande offre dans ce secteur parce que ces outils sont devenus indispensables pour les managers de chaînes logistiques. Par ailleurs, les nouvelles technologies permettent une ouverture de ces systèmes pour une collaboration interne ou externe. Nous allons étudier deux de ces systèmes, qui à notre sens sont les plus importants : les Entreprise Resource Planning ERP, et les Advanced Planning and Scheduling APS.

1.2.8.1 Entreprise Resource Planning (ERP)

Les ERP sont des systèmes transactionnels qui couvrent les différentes fonctions d'une entreprise (production, distribution, mais aussi comptabilité/finances, ressources humaines...). L'organisation des traitements et des données des ERP répond pour l'essentiel à des préoccupations de découpage analytique par fonction, de suivi des flux, des performances financières et d'exécution, de suivi des principales activités et tâches. Il se concentrent généralement sur les opérations internes principalement liées à la gestion de la production. Ces systèmes héritent généralement de la logique MRP2 (Manufacturing Resource Planning) qui traite l'information de manière séquentielle, par lots, et avec des capacités de production supposées infinies (i.e. pouvant toujours répondre à la demande planifiée). L'idée principale de la méthode MRP est que la connaissance de la demande pour le produit en termes de quantités demandées et de dates de livraisons souhaitées, et la nomenclature de fabrication de ce produit nous permet de calculer exactement les besoins en composants et matières premières et par conséquent planifier de manière optimale les réapprovisionnements (Ould Louly, 2001). La figure suivante montre les modules d'un ERP.

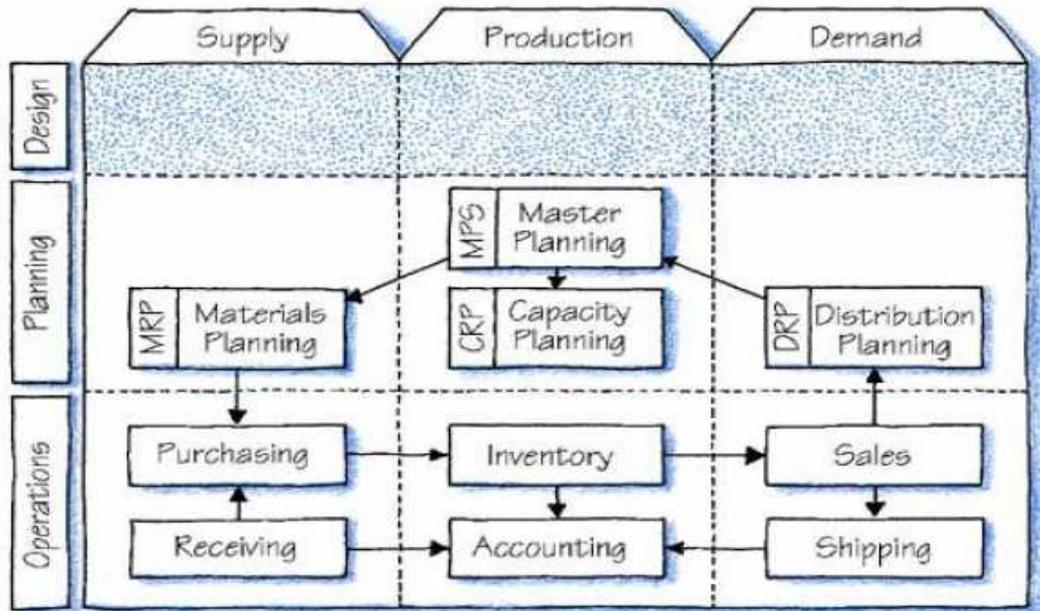


Figure 1.10 modules d'un ERP (Taylor, 2003)

Le DRP (Distribution Requirements Planning) utilise l'historique et les prévisions de ventes comme données pour construire des programmes de distribution qui indiquent les quantités de chaque produit à transporter pour chaque localisation. Ces derniers plans sont passés comme entrée au module MPS (Master Production Scheduling) qui réalise l'ordonnancement de la production afin de satisfaire le plan de distribution. Ensuite, le MPS appelle le MRP (Material Requirements Planning) pour s'assurer que toutes les matières premières et les composants nécessaires à la production sont bien disponibles, et le CRP (Capacity Requirements Planning) assure que la capacité de production est suffisante pour satisfaire la demande.

Comme son architecture l'indique, les ERP ont été conçus à la base pour piloter la gestion de la production au sein des ateliers et des usines, c'est à dire des environnements assez contrôlés, et ne sont donc pas aptes, tout seuls, à gérer la totalité de la chaîne logistique. Logiquement, une nouvelle génération d'applications de gestion des ressources est donc apparue, les APS.

1.2.8.2 Advanced Planning and Scheduling APS

L'APS est une application destinée à la planification de la supply chain. En fonction de la demande, elle permet d'analyser la capacité des ressources et les contraintes afin de proposer un horaire détaillé et adaptable pour une production optimale. Elle intervient à tous les niveaux :

- **la demande** : détermine combien de produits doivent être fabriqués.
- **les achats** : vérifie la disponibilité des matières premières et des composants suivant la nomenclature du produit.
- **la production** : analyse les contraintes et la capacité à développer un plan de production optimal.
- **le stockage** : prévoit les espaces nécessaires.
- **le transport et la distribution** : optimise les coûts et assure la qualité de service à la clientèle.

La figure suivante illustre les différents modules qui composent un système de type APS. Notons que cette représentation n'est pas commune à tous les APS, en effet il n'existe pas d'architecture générique de tous les modèles. Celle représentée ci dessous comporte néanmoins les modules importants. Cette critique doit cependant être relativisée par le fait que les éditeurs développent aujourd'hui des outils développés adaptés aux secteurs d'activité de leurs clients.

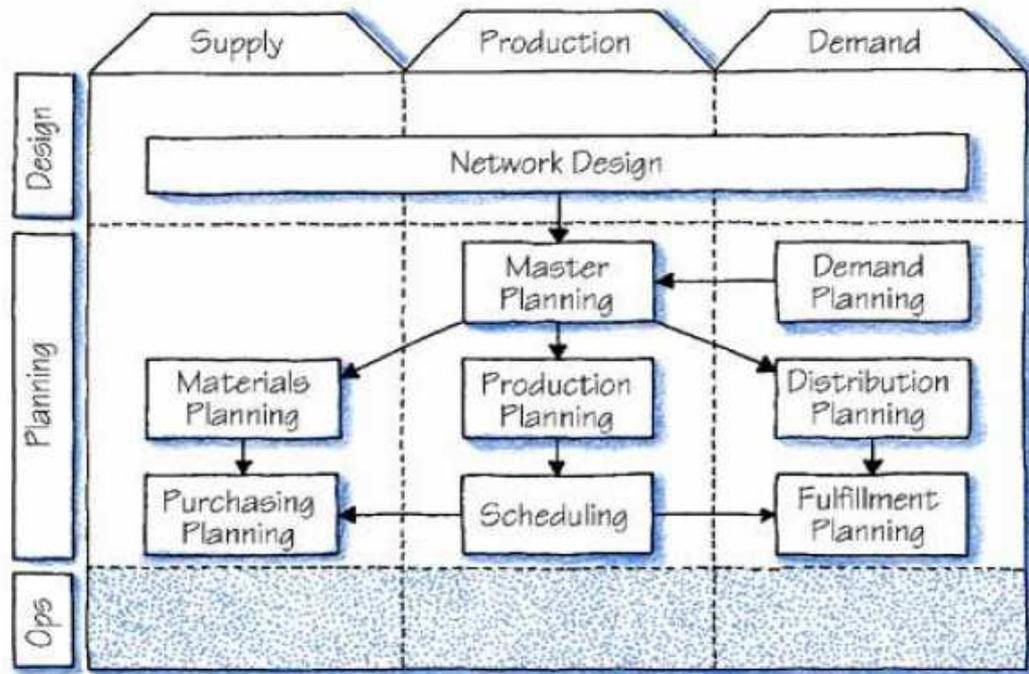


Figure 1.11 modules d'un APS (Taylor, 2003)

La différence majeure que l'on peut constater en comparant les deux figures précédentes est que les ERP se situent plutôt à un niveau opérationnel d'actions, alors que les APS se situent au niveau tactique.

Niveau	Application	Tâches
Tactique	APS	Planification des achats, de la production, de la distribution et des transports.
Opérationnel	ERP	Gestion de la matière première, ordonnancement de la production, des stocks, et des transports

Figure 1.12 différences entre APS et ERP.

1.2.9 Motivations et exemples de chaînes logistiques

Dans cette section nous allons illustrer par des exemples réels l'impact et les conséquences qu'a une bonne gestion de la chaîne logistique sur les performances de certaines entreprises. Comme nous l'avons dit au début de ce chapitre, la nouvelle donne économique, sociale, et technologique font que la compétition se passe maintenant entre chaînes logistiques et non plus entre entreprises. C'est un changement stratégique, les compagnies ont compris l'importance de se restructurer et de collaborer avec leurs partenaires pour survivre dans un environnement très concurrentiel. Il existe de nos jours des multinationales qui doivent leur succès à l'architecture de la chaîne logistique qu'ils ont mis en place. Ceci leur a permis - comme on va le voir plus loin dans cette section- de faire des bénéfices colossaux et ainsi

d'être en bonne santé financière ce qui leur permet de mieux répondre aux besoins de leurs clients, de développer de nouveaux produits, et d'avoir un temps d'avance sur leurs concurrents.

1.2.9.1 Chaîne logistique de DELL

Dell offre certainement le meilleur exemple de réussite et de succès d'une entreprise grâce à une vision optimisée et optimisante de la chaîne logistique. DELL est une compagnie de fabrication d'ordinateurs personnels. Avant DELL, les ordinateurs étaient fabriqués en volumes, acheminés vers des centres de stock, puis vendus individuellement aux clients. Cette méthode nécessitait d'avoir un niveau assez élevé de stock. DELL a conçu une chaîne logistique innovante où les clients sont un partenaire important. Elle inclut le site web de DELL, les centres d'assemblage, les fournisseurs et les fournisseurs des fournisseurs. DELL a adopté une stratégie de vente directement liée aux clients via son site internet. Les clients passent leurs commandes via le web en connaissance des prix et des produits disponibles. DELL se passe donc des centres de distribution et des détaillants comme illustré par la figure 1.13. Cette proximité vis à vis de ses clients lui a permis de mieux comprendre leurs besoins et ainsi d'ajuster ses stratégies commerciales et de développement de nouveaux produits. L'entreprise a centralisé les usines d'assemblage et les centres de stock dans quelques endroits près de ses fournisseurs. Le résultat de cette approche est que DELL est capable d'offrir un large choix de configurations d'ordinateurs pour ces clients tout en gardant un niveau de stock très bas. Pour certains composants, comme les moniteurs, DELL ne garde aucun stock. L'avantage de ce bas taux de stock est la réactivité dans le cas de l'introduction de nouvelles technologies, par exemple, si Intel met sur le marché un nouveau processeur, DELL pourrait rapidement l'ajouter à son offre plus vite que ses concurrents. Vu les fluctuations du marché, les prix pourraient soudainement chuter, DELL aura moins de stock qui perdra de la valeur par rapport à ses concurrents. Aussi, dans le cas de défauts de fabrication, il n'y aura qu'une petite quantité de produits qui sera affectée.

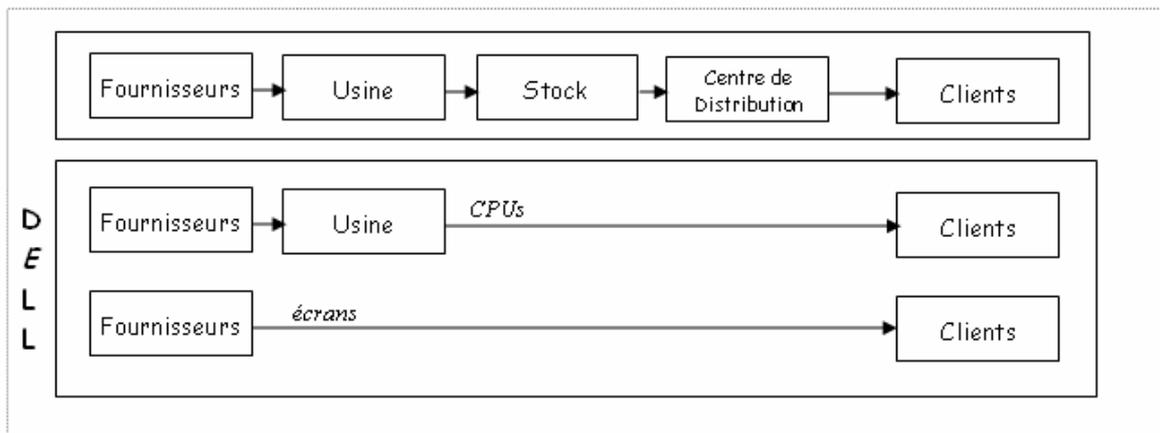


Figure 1.13 Chaîne logistique de DELL.

Grâce à sa chaîne logistique DELL est devenu le numéro un mondial de la fabrication d'ordinateurs personnels. En 2004, la compagnie a fait des bénéfices de 2.6 milliards de dollars (Chopra et Meindil, 2007). Ces bénéfices sont dus en grande partie à la gestion des flux de production, d'informations, et financiers à l'intérieur de la chaîne logistique.

1.2.9.2 L'exemple de Chrysler

Chrysler est un géant américain de construction automobile. A la fin des années 1980, Chrysler était à la limite du dépôt de bilan. Pour redresser l'entreprise, on a formé des équipes composées de gens appartenant aux divers services pour mettre en œuvre en commun la conception, l'ingénierie, la production, le marketing, l'approvisionnement, et les finances réinventant ainsi sa chaîne logistique. Ces équipes de travail ont gardé la moitié des fournisseurs et ont liés avec eux de nouvelles relations basées sur la confiance, et ont inclus ces fournisseurs dans la conception de nouveaux modèles d'automobiles. L'entreprise a même offert à ses fournisseurs de partager les bénéfices. Cette nouvelle orientation des relations entre Chrysler et ses fournisseurs lui a permis de réduire ses coûts de développement et de conception et a encouragé les fournisseurs à la considérer comme client privilégié. Chrysler a appelé ce programme « l'effort de réduction des coûts des fournisseurs » ou SCORE pour « the supplier reduction cost effort ». Ce programme a été lancé en 1990. En 1995, l'entreprise avait déjà implémenté 5300 des idées fournies par ses fournisseurs faisant ainsi des économies de l'ordre de 1.7 milliard de dollars. Le coût de développement d'un nouveau véhicule a été réduit de 40%, et le temps du processus de développement est passé de 234 semaines à 160, et le profit fait par véhicule a augmenté de 844% du milieu des années 1980 au milieu des années 1990 (Hugos, 2003).

Ces chiffres illustrent qu'une bonne conception de la chaîne logistique peut réduire les coûts d'une manière importante et engendrer des profits colossaux, et montrent l'efficacité d'un système où les partenaires agissent en concertation.

1.2.9.3 L'exemple de Amazon.com

Amazon.com est l'un des pionniers de la vente sur internet. Ce portail vend des livres et de la musique ainsi que d'autres produits. Amazon.com reçoit donc les commandes des clients via son site Internet comme dans le cas de DELL, et répond à ces commandes par les livres achetés chez les distributeurs. Cette approche diffère de celle utilisée par les vendeurs de livres traditionnels qui achètent les livres directement chez la maison d'édition et les gardent en stock anticipant ainsi les demandes des clients. Avant de connaître son succès actuel, Amazon.com était passé par des années de disette notamment à cause des défections dans sa chaîne logistique. En effet, 12% des produits destinés au stockage étaient acheminés vers les mauvais entrepôts ce qui engendrait une perte de temps, d'énergie et de moyens. L'entreprise a donc dû repenser sa chaîne logistique. Une meilleure gestion de l'acheminement des marchandises en les regroupant d'une manière optimale a engendré des réductions dans les quantités en stock et des économies dans les transports.

Les exemples vus ci-dessus montrent qu'en ayant une meilleure gestion de la chaîne logistique, l'entreprise peut gagner en compétitivité et générer de grands profits. De l'autre côté, si la gestion de la chaîne n'est pas optimale, alors cela peut générer des pertes de temps, d'énergie, et un manque de compétitivité qui seraient fortement préjudiciables. Il faut donc à tous les niveaux de la chaîne penser à optimiser le fonctionnement. Cela dépend de la nature de la chaîne (produits ou services). Néanmoins ces exemples montrent qu'il n'existe pas de configuration ou conception de chaîne logistique qui soit optimale pour toutes les entreprises et pour tous les produits.

1.2.10 Les mesures de la performance de la chaîne logistique

Il y a un adage qui dit : « you can't manage what you don't measure » dont la traduction en français pourrait être « vous ne pouvez pas gérer ce que vous ne mesurez pas ». La question qui est posée est donc : comment mesurer l'efficacité d'une chaîne logistique ?.

Traditionnellement, la pratique la plus courante est de mesurer l'aspect financier qui est facile à faire et regarder la balance entre les revenus et les dépenses. Le problème est que les métriques financières sont inadéquates pour mesurer la performance d'une chaîne logistique car elles n'ont pas une vue précise sur l'efficacité au niveau opérationnel (Camirenelli et Cantu, 2006), et ne prennent pas en compte la qualité de service du client. On peut même dire que les décisions au niveau opérationnel sont mieux gérées si on ne prenait pas en compte l'aspect financier. Mais pour une optimisation globale de la chaîne logistique, on doit trouver un équilibre entre les métriques financières et non financières que ce soit au niveau stratégique, tactique ou opérationnel. Aussi, pouvoir faire les bonnes mesures dans toutes les fonctions de la chaîne permettrait de mieux la comprendre et ainsi de pouvoir l'améliorer là où les besoins se font sentir.

Nous devons donc définir un ensemble d'indicateurs pour mesurer cette performance. Un indicateur de performance est défini par (Courtois et al, 1996) comme « une donnée quantifiée qui mesure l'efficacité de tout ou partie d'un processus ou d'un système, par rapport à une norme, un plan ou un objectif qui aura été déterminé et accepté, dans le cadre d'une stratégie d'ensemble ».

Chopra et Meindil (Chopra et Meindil, 2007) identifient six indicateurs de performance :

- Les infrastructures : ce sont les localisations physiques où les produits sont stockés, assemblés, ou fabriqués. Les décisions concernant leurs rôle, localisation, capacité et flexibilité (technologie utilisée) influencent la performance de la chaîne logistique.
- Les stocks : ceci comprend tous les stocks de matières premières, les encours, et les produits finis. Changer les politiques de stocks influence la réactivité de la chaîne logistique.
- Les transports : incluant aussi les transport internes de stocks ou de produit semi finis. La politique de transport est une combinaison de choix de modes de transport et d'établissement de programmes de livraison. La performance de la fonction transport sur la réactivité et l'efficacité de la chaîne est importante surtout que pour la plupart des compagnies les coûts de transports représentent le tiers des coûts globaux.
- L'information : elle est constituée des données concernant les infrastructures, le stock, les transports, les coût, les prix, les client. Ce peut être l'indicateur majeur de performances de la chaîne logistique car la performance de cette fonction influe directement sur l'ensemble des autres indicateurs. Une information qui « circule » bien et qui est synchronisée entre les différents partenaires peut améliorer l'efficacité de la chaîne, tandis qu'une information difficilement transmissible d'un partenaire à un autre, ou bien non communiquée, peut avoir des retombées néfastes sur l'ensemble de la chaîne et pénaliser les autres fonctions.
- Sourcing : il s'agit de déterminer qui doit réaliser une activité à travers la chaîne comme la production, le stockage ou les transports. Au niveau stratégique, il s'agit de déterminer quelles activités l'entreprise va faire elle même et quelles activités elle va sous-traiter.
- Les prix : il s'agit de donner une valeur et un prix aux biens et services que la chaîne logistique produit. Les prix influencent le comportement des clients et ainsi les performances de la chaîne.

Gunasekaran et al (Gunasekaran et al, 2004) donnent un état de l'art des mesures de performances dans la chaîne logistique. Ils identifient six indicateurs importants pour lesquels ils développent des métriques.

Métriques de la planification des commandes

- La méthode de réception des commandes : cette méthode détermine comment les spécifications des clients sont converties en données échangées tout au long de la chaîne.
- Temps de latence de la commande (order lead time) : le cycle total de commande est le temps écoulé depuis la réception de la commande jusqu'à la livraison du produit au client. La réduction du temps de ce cycle induit une meilleure réponse de la chaîne aux commandes de ses clients ce qui lui donne un avantage certain en terme de compétitivité.
- Le chemin de la commande : il inclut tous les canaux par lesquels la commande est passé. Cet indicateur identifie les étapes où il n'y a pas eu de valeur ajoutée, et ainsi permettre à la chaîne de prendre les décisions nécessaires pour éliminer les étapes sans valeurs ajoutés.

Evaluation des fournisseurs

L'évaluation des fournisseurs implique des mesures à tous les niveaux de la chaîne. Cette évaluation a souvent été basée sur les variations des prix et sur les délais de livraison. La compétition entre les fournisseurs était une compétition basée sur les prix proposés en négligeant d'autres aspects tout aussi importants comme la qualité, la réactivité, la disponibilité et la satisfaction des clients. Cette analyse des fournisseurs doit être faite de manière périodique et projetée sur le long terme.

Les métriques au niveau de la production

La performance au niveau de la fonction de production a un grand impact sur le coût du produit fabriqué, sa qualité, et la rapidité de sa distribution aux clients. Les auteurs proposent les métriques suivantes pour la mesurer.

- La gamme de produits et/ou de services offerts et produites.
- La capacité de production : son rôle est important vu qu'elle détermine les niveaux d'activités tout au long de la chaîne. Elle influence directement la vitesse de réponse aux commandes (réactivité de la chaîne) et le temps de cycle d'un produit dans la chaîne.
- Efficacité des techniques d'ordonnancement : détermine la façon avec laquelle les ressources sont allouées aux tâches.

Evaluation des livraisons

La livraison constitue le lien direct qu'a la chaîne avec ses clients. La performance de cette fonction détermine en grande partie la satisfaction ou non du client, et ainsi la compétitivité de la chaîne. La performance optimale des livraisons serait que les clients soient livrés aux dates convenues. La capacité de cette fonction à être rapide dépend de certains paramètres comme le mode de transport choisi, la fréquence des livraisons, et la localisation des dépôts. Une autre mesure de cette fonction serait la capacité de la chaîne à répondre favorablement à une demande spécifique d'un client (mode de livraison particulier à un endroit en particulier sous certaines conditions). Une meilleure réponse à ce genre de demandes a l'avantage de fidéliser les clients.

Evaluation de la qualité de service

- La flexibilité : c'est à dire la capacité de la chaîne logistique de pouvoir répondre favorablement à des demandes individuelles des clients. La flexibilité peut être mesurée par le temps de cycle de développement d'un produit et les temps de réglage des machines ou outils.
- Le temps de réponse aux requêtes des clients concernant par exemple le suivi de l'état de leurs commandes.
- La qualité du service après vente.

Evaluation des coûts de la logistique

C'est l'évaluation de tous les coûts liés à la logistique. C'est un indicateur financier très important ; les flux financiers ayant une grande influence sur les flux des produits. L'un de ses indicateurs est la mesure du coût des risques entrepris par la chaîne.

Taylor (Taylor, 2003) fait une classification des indicateurs de performances de la chaîne logistique en quatre catégories :

- *Mesures du temps* : incluant entre autres le temps de cycle d'une commande, le temps de cycle du développement d'un produit, la ponctualité des livraisons.
- *Mesures des coûts* : incluant entre autres les coûts des matières premières, la masse salariale, la maintenance, les retours de produits défectueux, les transports, le stockage, et le management des infrastructures.
- *Mesures d'efficacité* : concerne le taux d'utilisation d'un bien de la chaîne comme les taux d'utilisation des centres d'entreposage, le taux de la capacité de production utilisée, et le taux des capitaux utilisés.
- *Mesures de qualité de service* : comme les taux des livraisons effectuées à temps, des commandes satisfaites, des retours en usine, des plaintes des clients, et des clients qui passent de nouvelles commandes.

Comme on vient de le voir, il existe une multitude d'indicateurs de performances de la chaîne logistique. Prendre trop d'indicateurs peut engendrer un nombre important de données qu'ils seraient difficile de gérer et qui ne donneraient pas assez de visibilité sur ce que doivent être les décisions qu'il faut prendre pour améliorer la qualité de la chaîne logistique. En revanche, prendre peu d'indicateurs pourrait avoir des conséquences néfastes car on pourrait négliger certains facteurs importants. Le challenge ici est de prendre les « bons » indicateurs. Encore une fois, il n'y a pas un ensemble défini et précis d'indicateurs valables pour toutes les chaînes logistiques. L'ensemble des indicateurs choisis doit dépendre de la nature des activités réalisées par la chaîne.

1.3 Ordonnancement de la production

1.3.1 Introduction

Les décisions d'ordonnancement se situent au niveau opérationnel de la chaîne logistique. ce sont des décisions qui se prennent sur un très court terme. A ce niveau de la chaîne plusieurs types d'ordonnancement sont établis : l'ordonnancement de la production, l'ordonnancement des livraisons, l'ordonnancement des tournées du personnel,

l'ordonnancement des emplois du temps...etc. Apporter des solutions efficaces et performantes aux problèmes d'ordonnancement constitue un enjeu économique important au sein de la chaîne.

Dans le cadre de notre travail nous nous intéressons à l'ordonnancement de la production au niveau de la chaîne logistique pour lequel nous développons des techniques de coopération et de coordination entre les différents partenaires. Nous nous intéressons en particulier aux problèmes d'ateliers de type flowshop (nous modélisons les lignes de production dans le problème principal étudié dans cette thèse par un flowshop) et au concept du juste à temps. L'objectif de l'ordonnancement de la production est l'organisation de l'utilisation des ressources humaines et matérielles des ateliers de l'entreprise pour satisfaire les demandes des clients (Lopez et Roubellat, 2001). Dans ce qui suit nous allons voir les différentes structures d'ateliers et les méthodes de résolution pour les problèmes d'ordonnancement.

1.3.2 Gestion de la production

La gestion de la production se situe au niveau tactique et opérationnel de la chaîne logistique. Il s'agit de faire la planification de la production en termes de quantités à produire et de types de produits à faire. Elle consiste en l'ensemble des activités suivantes :

- La conception ;
- La planification des ressources ;
- L'ordonnancement de la production;
- Pilotage des activités de production.

La fonction ordonnancement de la production récupère les données nécessaires à son fonctionnement à partir des plans de production décidés au niveau de la fonction de gestion.

La gestion de la production occupe une place importante dans une entreprise, et a naturellement des relations avec les autres services ou fonction de l'entreprise ce qui engendre des objectifs contradictoires.

1.3.3 Ordonnancement

1.3.3.1 Définitions

Pour (Carlier et Chrétienne, 1988), un problème d'ordonnancement consiste à « programmer l'exécution d'une réalisation en attribuant des ressources aux tâches et en fixant leurs dates d'exécutions ». D'une manière plus précise, un problème d'ordonnancement peut être défini comme suit : « étant donné un ensemble de travaux à accomplir et un ensemble de ressources permettant leur réalisation, le problème d'ordonnancement consiste à déterminer, d'une part l'affectation des ressources aux différentes tâches composant les travaux, et d'autre part, les dates d'exécution de ces tâches, de telle sorte qu'une fonction objectif donnée soit optimisée » (Aggoune, 2002). Cette affectation doit prendre en compte certaines contraintes temporelles (délais, contraintes d'enchaînement...etc.) et des contraintes portant sur l'utilisation et la disponibilité des ressources requises pour la réalisation des tâches. Un ordonnancement (qui est la solution d'un problème d'ordonnancement) doit répondre aux questions suivantes : quand ? Avec quels moyens ? Ou encore sous quelle modalités, la production doit être réalisée.

D'après les définitions précédentes, les composantes d'un problème d'ordonnancement sont : un ensemble de ressources, un ensemble de tâches à réaliser, et une fonction objectif à optimiser.

Les tâches sont des entités élémentaires généralement désignées sous le nom d'opérations et sont regroupés en travaux (jobs). Elles sont caractérisées par une date de début d'exécution ou une date de fin d'exécution, et par une durée opératoire. Selon les problèmes, les tâches peuvent être exécutées sans aucune interruption (tâches non préemptives), ou par morceaux (tâches préemptives). En outre, il existe des relations et contraintes de précédences entre les tâches qui définissent l'ordre total ou partiel de leurs exécutions. Deux autres contraintes temporelles peuvent être associées aux tâches : la date de disponibilité de la tâche qui définit la date à partir de laquelle on peut commencer l'exécution d'une tâche, et la date butoir qui détermine la date avant laquelle la tâche doit être terminée. Les dates de disponibilité et les dates butoirs peuvent aussi s'exprimer en fenêtres de temps. Dans l'ordonnancement de la production, les jobs à faire correspondent aux commandes des clients.

Les ressources sont les moyens techniques, matériels ou humains nécessaires à l'exécution des tâches. Elles sont de types différents comme par exemple les matières premières et les composants, la main d'œuvre, et les machines et outils. On dit qu'une ressource est une ressource consommable si sa quantité diminue au fur et à mesure de l'exécution de la tâche, c'est le cas pour les composants et les matières premières. Une ressource est dite renouvelable si sa quantité reste constante pendant l'exécution des tâches (les ressources humaines, équipements...etc.). Dans les ressources renouvelables, on distingue les ressources disjonctives qui ne peuvent être utilisées que par une tâche à la fois, et les ressources cumulatives qui peuvent être utilisées par un ensemble de tâches en nombre limité. Dans une structure d'atelier, les ressources sont les machines. Comme pour les tâches, l'ensemble des ressources peut être régi par certaines contraintes liées principalement à la disponibilité des ressources. C'est le cas par exemple des tâches qui nécessitent que plusieurs machines fonctionnent en parallèle.

Il existe deux grandes stratégies concernant l'objectif d'un ordonnancement : chercher des solutions admissibles (respectant les contraintes), ou bien viser l'optimalité des solutions. Le plus souvent c'est l'optimalité des solutions qui est visée. Les fonctions à optimiser sont les critères (basés sur des indicateurs de performance) permettant d'apprécier la qualité d'un ordonnancement. On parle de critère régulier lorsque l'avancement de l'exécution d'une tâche, sans en retarder d'autres, ne dégrade pas la valeur du critère. Les critères s'expriment généralement en termes de durée, de retard, et de coût. Parmi les critères les plus utilisés :

- Minimisation de la date de fin d'ordonnancement (makespan), $C_{\max} = \max(C_j / 1 \leq j \leq n)$ tel que n est le nombre de jobs et C_j est la date de fin d'exécution du job j . Ce critère permet de minimiser le temps de séjour total de l'ensemble des produits et donc de maximiser la productivité.
- La somme des dates de fin des jobs $\sum C_j$ qui minimise le temps de séjour moyen.
- Le plus grand retard $L_{\max} = \max(C_j - d_j / 1 \leq j \leq n)$ tel que d_j désigne la date butoir associée au job j .
- La somme des retards $\sum_j T_j = \sum_j \max(0, C_j - d_j)$.
- Le nombre de travaux en retard $\sum_j U_j$ tel que $U_j = 1$ si $C_j > d_j$ (job j en retard) et $U_j = 0$ sinon.

Il existe une notation introduite par (Graham et al, 1979) pour décrire tous les problèmes d'ordonnancement. C'est une notation en trois champs $\alpha/\beta/\gamma$ tel que le champs α décrit la structure du système à ordonnancer, le champs β décrit les contraintes liées à ce système, et le champs γ décrit la fonction objectif. Le moyen le plus simple et le plus utilisé pour représenter un ordonnancement est le diagramme de Gantt où des barres horizontales représentent le déroulement des tâches selon l'axe des temps, et les lignes correspondent aux ressources.

1.3.3.2 Ordonnancement d'ateliers

Dans un problème d'atelier les ressources (les machines) sont des ressources renouvelables disjonctives, c'est-à-dire que les machines ne peuvent réaliser qu'une opération à la fois, et chaque opération nécessite une seule machine pour sa réalisation. Les contraintes de précédences entre les opérations des différents travaux sont données par une gamme linéaire représentant l'ordre de fabrication, qui peut être fixée ou non.

Selon la disposition des machines et leur nombre, on peut distinguer les deux situations suivantes :

- Les problèmes à une machine : dans ce cas chaque travail comprend une seule opération à laquelle il est assimilé. Dans ce cas, le problème d'ordonnancement se résume au séquençement des opérations. Ce genre de problèmes sert de base pour le développement de solutions aux problèmes plus complexes.
- Les problèmes à machines parallèles : dans ce cas également chaque travail est constitué d'une seule opération, néanmoins plusieurs machines sont disponibles pour la réalisation d'un travail qui n'en nécessite qu'une. Il faut donc décider sur quelle machine effectuer chaque opération et déterminer la séquence d'opérations sur chaque machine. De plus, les machines peuvent être identiques ou non. Dans le cas de machines non identiques, l'ensemble des opérations qu'elles peuvent traiter, ainsi que leurs durées opératoires, peuvent varier d'une machine à une autre.

Selon la gamme de fabrication, on distingue trois types d'ateliers.

1.3.3.2.1 Flowshops

Dans les problèmes d'atelier de type flowshop, appelés aussi « atelier à cheminement unique », chaque travail visite chaque machine de l'atelier et l'ordre de passage d'un travail sur les différentes machines est le même pour tous les travaux. Cet ordre de passage ou gamme de fabrication est linéaire et est fixé à l'avance, c'est donc une donnée du problème. En pratique, un atelier structuré en flowshop peut correspondre par exemple à une chaîne de traitement ou de montage. Un cas particulier et simplifié du flowshop est le flowshop de permutation. Dans cette sous classe, l'ordre de passage des travaux est identique sur toutes les machines (gamme unique pour tous les travaux).

Les problèmes de type flowshop sont, parmi les problèmes d'atelier, les problèmes les plus étudiés dans la littérature. Différentes méthodes ont été proposées pour résoudre ce type de problèmes. Ces problèmes étant généralement NP-difficiles, des heuristiques sont proposées pour les problèmes à tailles industrielles. (Ruiz et al, 2006) proposent deux algorithmes génétiques pour le problème de flowshop de permutation avec l'objectif de minimisation du makespan. Ils ont développé de nouveaux opérateurs de croisement et une procédure de redémarrage d'une partie de la population s'il n'y a pas amélioration du critère pendant un certain nombre de générations. (Wang and Zheng, 2003) proposent aussi un algorithme génétique dans lequel l'opérateur de mutation est remplacé par une procédure de recuit

simulé. Bulbul et al (Bulbul et al, 2004) ont étudié le problème avec comme fonction objective la minimisation de la somme des retards, avances, et les coûts de stocks intermédiaires dans une entreprise de fabrication de biens. Ce problème a des applications industrielles comme par exemple dans le cas d'une entreprise de biotechnologie où les stocks intermédiaires coûtent chers et nécessitent un traitement spécial avant l'étape suivante. Ils ont utilisé des méthodes de programmation mathématique avec la génération de colonnes pour relaxer les contraintes du programme linéaire.

Nearchou (Nearchou, 2004) propose une approche basée sur le recuit simulé combiné aux algorithmes génétiques. Au lieu de partir d'une seule solution, et faire la recherche selon le principe traditionnel du recuit simulé, l'auteur propose de partir d'une population de solutions. Il propose aussi des opérateurs pour la recherche de voisinage.

Fondrevelle (Fondrevelle, 2005) dans sa thèse effectuée au sein de l'équipe MACSI du Loria propose des méthodes de résolution exacte basée sur les procédures par séparation et évaluation pour le problème de flowshop de permutation en présence de contraintes d'écart temporel entre les opérations. Deppner (Deppner, 2004) pour sa part traite le même genre de problèmes mais de tailles industrielles pour lesquels il propose des méthodes approchées à base d'algorithmes génétiques. Il propose entre autres un algorithme d'autodétermination pour le choix automatique des opérateurs pendant le processus évolutionnaire.

Yang et al (Yang et al, 2000) étudient le problème de minimisation de la somme pondérée des retards dans un flowshop flexible. Un flowshop flexible est une généralisation du flowshop traditionnel dans lequel il y a un nombre limité d'étages, chacun est composé de machines identiques en parallèle. Les auteurs proposent trois heuristiques pour résoudre ce problème. La première procède par décomposition de l'atelier en étage et trouve successivement des solutions aux sous problèmes indépendants. La deuxième se base sur une méthode de recherche locale, et enfin, la troisième est une hybridation des deux premières approches.

Noorul Haq et al (Noorul Haq et al, 2004) propose, pour le problème du flowshop avec comme fonction objective la minimisation du makespan, un algorithme hybride dans lequel un algorithme génétique agit en amont d'un algorithme à base de système de colonie de fourmis, pour l'amélioration de la solution obtenue par ce dernier. Le système de colonie de fourmis est aussi utilisé par (Rajendran et Ziegler, 2005). Le même problème, (nous remarquons qu'il s'agit du problème de flowshop le plus étudié dans la littérature), avec la prise en compte de contraintes de disponibilité des machines est étudié par (Aggoune, 2004). L'indisponibilité des machines est due à des activités de maintenance préventive. L'auteur considère deux cas d'indisponibilité : le cas où les dates de début des tâches de maintenance sont fixés, et le cas où ils sont compris dans une fenêtre de temps. Il propose des solutions basées sur les algorithmes génétiques.

(Lin et al, 2007) étudient un système de production à deux étages structuré en flowshop. Dans le premier étage deux machines en parallèle fabriquent des composants qui sont assemblés et groupés en lots sur la troisième machine. Le critère considéré est la minimisation du makespan. Les auteurs démontrent que le problème général est NP-difficile au sens fort et fournissent des heuristiques et un algorithme polynomial pour le cas spécifique où la séquence de traitement des opérations est fixée.

Deux autres familles d'ateliers ne sont qu'évoquées ci-dessous car nous avons choisis de travailler sur des flowshops.

1.3.3.2 Jobshops

Les problèmes d'atelier de type jobshop, appelés aussi « problèmes d'atelier à cheminements multiples », sont une généralisation du problème du flowshop. La différence entre les deux structures réside dans le fait que l'ordre de passage des travaux sur les machines est différent d'un travail à un autre.

1.3.3.3 Openshops

Les problèmes d'atelier de type openshop, appelés aussi « problèmes d'atelier à cheminements libres », sont des problèmes où la gamme de fabrication n'est pas fixée à l'avance. Dans ce cas, le problème d'ordonnancement consiste à déterminer le cheminement de chaque travail et à ordonnancer les travaux en fonction des gammes trouvées. C'est le problème le moins étudié dans la littérature parmi les trois types d'ateliers. Cette forme de structure d'atelier se rencontre néanmoins dans divers domaines d'application comme pour le test d'activités de maintenance, dans les services comme pour les examens de patients hospitalisés, ou encore dans les télécommunications (les travaux sont l'ensemble de paquets à émettre par une station terrienne, les opérations sont modélisées par les paquets d'une station émettrice vers un répéteur du satellite qui forment l'ensemble des machines).

1.3.3.3 Les méthodes de résolution

Après la description des problèmes d'ordonnancement et la présentation des structures d'ateliers, nous nous intéressons maintenant aux méthodes de résolutions de ces problèmes. Les problèmes d'ordonnancement étant des problèmes d'optimisation, il existe deux grandes familles de méthodes de résolutions : les méthodes exactes, et les méthodes approchées. L'utilisation d'une méthode de l'une ou de l'autre de ces familles dépend de la taille du problème et de sa complexité.

1.3.3.3.1 Les méthodes exactes

L'approche intuitive simpliste pour résoudre un problème d'optimisation combinatoire est de lister toutes les solutions faisables, évaluer leurs fonctions objectives et choisir la meilleure solution. Cette énumération complète permet de trouver les solutions optimales. Néanmoins, ce genre de méthodes montre ses limites en pratique à cause du très grand nombre de solutions qui existent pour un problème de taille raisonnable (on suppose que l'on considère des problèmes NP-difficiles). Les méthodes exactes les plus utilisées pour les problèmes d'ordonnancement sont :

- Les procédures par séparation et évaluation (branch and bound) : Les PSE procèdent par l'énumération implicite de l'ensemble des solutions. Elles sont basées sur la décomposition du problème de façon arborescente en sous problèmes. Avec cette méthode on est assuré de trouver la meilleure solution puisqu'on analyse tous les sous-ensembles de solutions possibles. La manière d'effectuer la décomposition constitue le principe de séparation qui conditionne la performance de la méthode avec le calcul de bornes permettant d'éliminer des sous-ensembles de solutions.
- La programmation dynamique : le principe est de procéder à une décomposition des décisions associées aux problèmes en sous ensembles considérés successivement, ce qui conduit à résoudre des sous problèmes plus grands en utilisant la solution de sous problèmes plus petits. Ceci n'est possible que si la fonction objective possède la propriété de décomposition. On traite le problème par étapes commençant par le plus petit sous problème jusqu'au problème complet. On résout optimalement chaque sous problème en tenant compte des informations obtenues lors des précédents étapes.

L'exemple le plus simple de l'utilisation de la programmation dynamique est la recherche du plus court chemin dans un graphe orienté. Comme les PSE, la programmation dynamique énumère implicitement toutes les solutions, et de ce fait elle est très exigeante en termes de temps de calculs et de mémoire pour les problèmes NP-difficiles. Elle ne permet pas alors d'aborder des problèmes de grandes tailles.

- La programmation linéaire : Elle formule le problème avec un critère et des contraintes exprimés en fonction linéaires de variables de décisions. Les variables sont entières ou binaires pour la plupart des problèmes d'ordonnancement. Ceci rend l'ensemble des problèmes pour lesquels on peut trouver des solutions exactes en un temps raisonnable, limité à des problèmes de tailles relativement faibles.

1.3.3.2 Les méthodes approchées

Puisque les approches exactes ne peuvent traiter que des problèmes beaucoup plus petits que ceux envisagés dans les applications réelles, les méthodes approchées (heuristiques ou méta heuristiques) sont largement utilisées car elles fournissent un bon compromis entre la qualité d'une solution et son coût. Ce sont des méthodes qui recherchent de bonnes solutions (que l'on espère proche de l'optimum) en un temps raisonnable, sans toutefois garantir l'optimalité de la solution, ni même sa possible non admissibilité. Souvent on ne connaît pas la distance entre la solution trouvée et la solution optimale.

Les méthodes approchées les plus utilisées sont :

- Les algorithmes génétiques : ce sont des méthodes à base de populations qui évoluent pendant un processus dit évolutionniste inspiré de la théorie de l'évolution. Les individus sont améliorés à bases d'opérations de croisement ou de mutations. Ces méthodes sont très pratiques car faciles à implémenter et parce qu'elles génèrent un ensemble de solutions et non une seule solution. On reviendra sur les algorithmes génétiques avec plus de détails dans le chapitre 4.
- Les méthodes par voisinage : appelés aussi méthodes à base de recherche locale. A partir d'une solution initiale obtenue d'une façon aléatoire ou grâce à une heuristique spécifique au problème étudié, on explore l'espace de solution on passant d'un voisin à un autre. Le choix d'une solution parmi les solutions voisines est basé sur des critères qui diffèrent d'une méthode à une autre. Les deux méthodes par voisinage les plus connues et utilisées sont la recherche taboue (Glover, 1989, 1990) et le recuit simulé. La recherche taboue est basée sur la notion de mouvements interdits. Elle examine les solutions voisines de la solution courante et choisit la meilleure parmi celles-ci. En acceptant de détériorer la valeur de la solution courante, cela permet de s'éloigner d'un optimum local. Pour éviter de faire un cycle, on utilise une liste taboue conservant en mémoire les derniers mouvements pendant un nombre limité d'itérations. Le recuit simulé est basé sur un algorithme de simulation de recuit des métaux et s'inspire des méthodes de la physique statistique (Metropolis et al, 1953). Cette méthode est utilisée en aval d'une méthode de recherche locale afin de l'améliorer, surtout par rapport à l'optimum local trouvé par cette dernière. Le recuit simulé accepte de faire des échanges, avec une certaine probabilité, qui n'améliorent pas la solution courante. Cette probabilité est déterminée par un paramètre de contrôle appelé *température*, allusion au processus de refroidissement "cooling schedule".

1.3.4 Le concept du Juste à Temps JAT (Just In Time JIT)

La notion du Juste à temps a été inventé par Taiichi Ohno qui était ingénieur chez le géant japonais de l'automobile Toyota, le but qu'il cherchait était d'éliminer le gaspillage tout au long de la chaîne de production de Toyota. Le gaspillage du temps et des moyens était la seule raison que trouva Ohno pour expliquer la meilleure productivité des américains par rapport aux japonais. Le gaspillage est défini par Mr Toyoda, président honoraire de Toyota, comme «tout ce qui dépasse la quantité minimale requise en matériel, équipement, espace et temps pour ajouter de la valeur au produit.»(Desrosiers, 2006). Les sources du gaspillage peuvent être :

- Produits défectueux.
- Mouvements ou transports inutiles.
- Les temps d'attente.
- La surproduction qui implique des surplus de stocks.
- Mauvaise qualité de communication entre les différents participants.

Depuis l'adoption de cette méthode par Toyota, elle s'est répandue dans le monde entier.

Le juste à temps est une philosophie qui permet de livrer les produits juste au moment voulu, en quantité suffisante pour la micro période et à l'endroit voulu (Ménard, 2005). Cette philosophie s'appuie sur l'amélioration continue de la qualité et de la productivité dans toutes les activités de la chaîne. On parle de philosophie dès lors que cela nécessite parfois une réorganisation de l'entreprise et la manière d'appréhender la production.

C'est un moyen de pilotage de la production par l'aval. Cela veut dire que le juste à temps permet de satisfaire la demande juste au moment où elle se manifeste et dans la qualité et la quantité demandées. Cette méthode du juste à temps répond de ce fait aux quatre objectifs de la production que sont la qualité, les délais, et les coûts.

Dans le cas habituel d'une entreprise industrielle qui transforme des matières premières en pièces fabriquées, puis regroupe ces pièces dans des sous-ensembles et, enfin, réunit les sous-ensembles pour constituer des produits finis, le principe du Juste à Temps peut s'exprimer ainsi : il faut produire et livrer de telle sorte que

- les produits finis juste à temps pour qu'ils soient vendus ;
- les sous ensembles juste à temps pour qu'ils soient montés dans les produits finis ;
- les pièces fabriquées juste à temps pour être assemblées en sous-ensembles ;
- les matières premières juste à temps pour être transformées en pièces fabriquées.

Beaucoup de gens voient dans le concept du juste à temps un ensemble de méthodes pour minimiser les stocks. En réalité, la minimisation des stocks n'est qu'une partie des objectifs du JAT qui sont le zéro attente, zéro délais, zéro défauts, zéro stocks, et zéro pannes.

Dans le contexte de la chaîne logistique, le juste à temps est un processus de l'amélioration des flux matériels des fournisseurs vers les centres de production, et vers les clients. C'est à dire que le système de production doit fonctionner en flux tendus. Dans les systèmes à flux tendus de type « tiré » (*pull systems*), la production est « tirée » de l'aval vers l'amont. Chaque étape ne travaille que si le poste en aval a besoin de produits ou composants. Une commande d'un client déclenche le système. Dans ce genre de logique de production les flux matériels, d'information, et de produits finis sont des flux continus. D'une certaine façon, on ne produit que ce qu'on a déjà vendu, et on essaye de produire avec les coûts les plus bas possibles, toujours en supprimant les gaspillages.

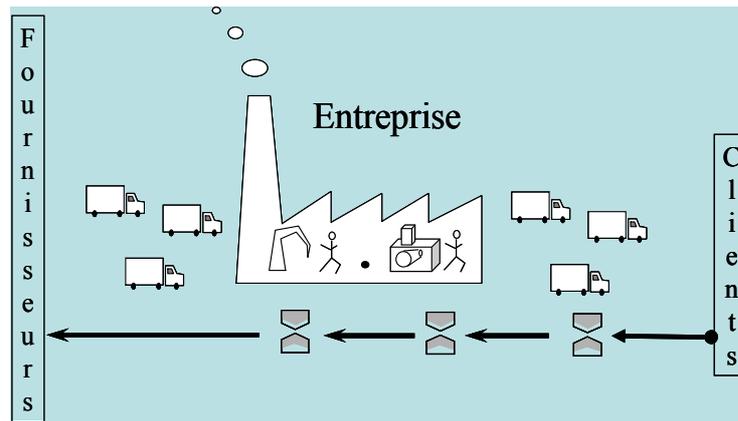


Figure 1.14 flux tirés.

Selon certaines études, Les entreprises adoptant le juste à temps peuvent faire des :

- réduction de temps de cycle de 80% à 90%.
- réduction des stocks (matières premières, encours, composants, produits finis) de 35% à 70%.
- réduction du coût de la main d'œuvre de 10% à 50%.
- réduction des espaces de 40% à 80%.
- réduction du nombre de défauts de 25% à 60%.

Ces chiffres varient d'une entreprise à une autre et de l'intensité et l'intelligence avec lesquels chaque entreprise a implémenté son système de juste à temps. Mais pour arriver à de tels résultats, les entreprises doivent faire des sacrifices, ou vu sous un autre angle, satisfaire certaines conditions parmi lesquelles :

- avoir une main d'œuvre flexible capable de s'adapter rapidement aux nouveaux objectifs de l'entreprise.
- avoir des délais de fabrication très courts et une grande souplesse pour pouvoir répondre aux souhaits de la clientèle.
- avoir un système d'information rapide et efficace.
- établir des relations basées sur le JAT avec ses fournisseurs. Ceci est primordial et nécessaire au bon fonctionnement du système.
- changer de mode de transport car avec le JAT la fréquence des livraisons augmente, mais la taille des lots diminue.
- assurer la fiabilité des équipements. Des machines en pannes génèrent une perte de temps, et de moyens qui vont se traduire en non respect des délais de livraison, ou tout simplement d'annulation de commandes, ce qui est très préjudiciable pour une entreprise vis à vis de sa clientèle.

La compétitivité de l'entreprise est considérablement renforcée, pas seulement par la baisse des coûts, mais aussi par la très forte réduction des délais et par l'amélioration de la qualité de la production. Convertir une entreprise au Juste à Temps, c'est donc mettre son organisation industrielle totalement au service de sa compétitivité et réduire considérablement ses besoins financiers. Pour bien mettre en œuvre le Juste à Temps, plusieurs méthodes techniques comme le Kanban, le SMED, ou encore la méthode des « 5S » sont utilisées.

Le Kanban, mot japonais pour carte, est un système d'information et une méthode d'organisation et de gestion de l'atelier qui fait remonter les besoins de l'aval vers l'amont. Elle superpose aux flux physiques de produits, un flux inverse d'information. Le Kanban est utile pour s'assurer que chaque poste de travail reçoit les matières premières ou composants

au bon moment, et livrer au poste suivant au bon moment. Ni trop tôt pour ne pas créer du stock, ni trop tard pour ne pas créer une rupture de produits finis qui va influencer négativement sur le flux du processus. Ce système est basé sur un outil visuel, une carte en l'occurrence, qui indique à un opérateur sur un poste de travail les besoins d'un poste dans le maillon suivant de la chaîne.

Le SMED (Single Minute Exchange Die) est utilisé pour rentabiliser la fabrication en petits lots qui est l'une des exigences d'une production en juste à temps. Elle vise à réduire les délais de réglage (mise en marche) des équipements, et ainsi d'éliminer des pertes de temps.

La méthode des 5S (pour Seiri, Seiton, Seiso, Seiketsu et Shitsuke) vise à :

- débarrasser le poste de travail des choses inutiles qui s'y trouvent;
- s'assurer qu'il reste bien rangé et en ordre;
- le nettoyer de façon régulière;
- y instaurer la rigueur nécessaire à l'exécution d'un bon travail.

En ordonnancement, le juste à temps est mis en œuvre par la prise en compte de critères induisant des pénalités d'avance et/ou de retard par rapport à des dates butoirs ou des dates de disponibilité; le plus souvent c'est par rapport à des dates butoirs.

(Hendel et Sourd, 2006a) étudient un problème d'ordonnancement à une machine avec comme fonction objective la somme des coûts de chaque job. Chaque job ayant des coûts d'avance ou de retard. Les problèmes avec des critères d'avance et/ ou de retard étant le plus souvent NP-difficiles, les auteurs proposent un algorithme en $O(n \log n)$ (en ayant déjà choisi l'ordre sur les machines) qu'ils généralisent pour le cas de flowshop de permutation. Une méthode à base de recherche locale est présentée dans (Hendel et Sourd, 2006b), et une autre basée sur les procédures par séparation et évaluation est donnée dans (Sourd et Kedad-Sidhoum, 2003). (Sourd, 2005) étudie le même problème en intégrant des coûts deoisiveté. Pour cela il propose un algorithme à base de programmation dynamique. Dans un autre article, (Sourd, 2006) intègre des contraintes relatives aux temps de réglages des machines et aux coûts de ces réglages. Un algorithme pseudo polynomial basé sur la technique de recherche de voisinage à grande échelle est proposé.

(Shabtay et Steiner, 2006) étudient les problèmes d'ordonnancement à deux machines avec comme fonction objective la minimisation de la somme pondérée des pénalités d'avance, de retard, et d'affectation de dates butoirs, ainsi que le problème de minimisation du nombre pondéré de jobs en retards et les coûts d'affectation de dates butoirs. Les auteurs prouvent que les deux problèmes sont NP-difficiles et proposent des algorithmes polynomiaux pour des cas spécifiques des deux problèmes. D'autres problèmes avec critères de pénalités sur les avances et les retards, mais aussi avec prise en compte d'un nouveau critère qui est la pénalisation des tâches non exécutées sont étudiés dans (Hassin et Shani, 2005).

1.4 Ordonnancement et coopération dans la chaîne logistique

La chaîne logistique implique plusieurs entités ou partenaires qui doivent réaliser des activités afin de produire un bien ou un service. Ces partenaires doivent collaborer, coopérer, travailler ensemble pour fournir le meilleur résultat. La coordination au sein de la chaîne logistique améliore la qualité de la chaîne si toutes les fonctions de la chaîne prennent les décisions pour leurs activités en tenant compte des activités des autres dans le but de maximiser le profit de l'ensemble de la chaîne. Cela peut être comparé à une équipe de relais 4×100 mètres d'athlétisme (Cooper et Ellram, 1997). Les relations sont fortes entre les quatre athlètes, mais elles sont encore plus fortes entre les athlètes qui doivent se passer le témoin. Ils doivent « négocier » sur la façon et le moment exact de passage du témoin. Dans la chaîne

logistique la stratégie doit être similaire. Les relations sont plus fortes entre les fonctions et services qui ont des relations directes et qui doivent s'échanger des informations. Mais cela n'est pas aussi facile que pour l'équipe d'athlétisme à cause de l'hétérogénéité des participants à la chaîne. Certains peuvent appartenir à des entités juridiques différentes ce qui pourrait impliquer un conflit d'objectifs. Chaque entité chercherait donc à maximiser ces profits, ce qui résulterait, le plus souvent, en la diminution des profits globaux de la chaîne. Le manque de coordination entre les activités visant à améliorer un même objectif peut diminuer la compétitivité de toute la chaîne. Cela se transforme en des coûts plus importants, une marge bénéficiaire plus étroite, des retards de livraison, pertes de commandes, et une qualité de service aux clients de mauvaise qualité.

L'un des phénomènes les plus observés (et redoutés) dans la gestion de la chaîne logistique est le « bullwhip effect ». Le bullwhip effect peut se produire lorsque les fluctuations de la demande augmentent à travers toute la chaîne : elle augmente des détaillants aux grossistes, des grossistes aux fabricants, et des fabricants à leurs fournisseurs. Dans ce phénomène l'information sur la quantité de la demande est altérée d'une fonction à une autre au sein de la chaîne à cause des différences d'estimation de la demande entre les différents services. Bien entendu, un tel phénomène est possible à cause d'un manque de coordination entre les différentes entités composant la chaîne. Ce phénomène s'est déjà produit dans plusieurs entreprises et est très étudié dans la littérature (Lee et al, 1997).

Les raisons du manque de coordination sont diverses, parmi elles :

- Le manque de partage de l'information : c'est la raison principale du manque de coordination. Elle peut se produire à cause d'un système d'information pas très efficace, ou d'une volonté délibérée de ne pas partager certaines informations avec les autres partenaires. Néanmoins, cette dernière hypothèse est de plus en plus rare car les entreprises ont compris qu'elles sont forcées de travailler en collaboration avec leurs partenaires pour faire face à la concurrence accrue des marchés.
- La qualité de l'information : certaines informations qui sont passées à travers la chaîne peuvent être erronées. L'une des sources d'erreurs peuvent être le fait de l'existence d'un grand nombre de produits, de composants, de matières premières, de locations d'infrastructures. Seul un système d'information efficace peut garantir la qualité de l'information.
- Le comportement : la chaîne logistique est une idée assez neuve, et la coopération en son sein l'est encore plus. Certains comportements relatifs à la rétention de l'information sont dus aux anciennes méthodes de gestion.
- Vue locale de la chaîne : chaque composante ou fonction de la chaîne voit la chaîne d'un point de vue local sans le remettre dans le contexte général, comme par exemple réagir à certaines situations locales sans mesurer l'impact de ces décisions sur le reste de la chaîne.
- Les activités opérationnelles : souvent, les activités au niveau opérationnel sont gérées localement car on ne dispose pas d'assez de temps pour de plus larges consultations. Les décisions doivent être rapides. A ce niveau de la chaîne, il est difficile de voir l'aspect chaîne logistique de l'activité. Ceci peut entraîner des ordonnancements de la production, des transports, ou d'approvisionnement en matières premières qui ne sont pas optimaux pour l'ensemble de la chaîne. La coordination des ordonnancements est un des problèmes étudiés dans le problème principal de notre thèse.

Pour (Thomas et Griffin, 1996), la mise en œuvre de la coordination peut se faire entre les partenaires de la chaîne, ou bien entre les fonctions de la chaîne. Ils en déduisent trois catégories de coordination opérationnelle.

Coordination acheteur-vendeur

L'activité la plus en aval de la chaîne logistique est l'approvisionnement en matières premières. Cette tâche est très importante pour le reste de la chaîne car les coûts des matières premières représentent une partie importante du coût de revient du produit fini. Beaucoup de travaux sur la coordination à ce niveau s'intéressent à trouver les quantités optimales à approvisionner. Cette quantité influe sur le reste de la chaîne comme les stocks et les transports. Elle détermine aussi la politique de « lot-sizing » utilisée. Aussi, l'un des éléments de la gestion de la chaîne logistique moderne étant le juste à temps, les liens entre les producteurs et leurs fournisseurs doivent être privilégiés à cause de la fréquence des livraisons et pour permettre un flux de matières premières optimal pour le reste de la chaîne et éviter ainsi des phénomènes comme la rupture de stocks. Cette coopération entre le client (producteur) et ses fournisseurs doit être continue dans le temps.

Coordination Production-Distribution

Les fonctions de production et de distribution sont des fonctions qui se suivent directement et qui sont donc fortement liées. En effet, on peut soit choisir d'adapter la politique de production aux moyens de transports pour en réduire les coûts, ou bien adapter la politique de transport en fonction de la politique de production. L'une ou l'autre de ces méthodes n'est pas nécessairement la meilleure. Maximiser localement (au niveau de la fonction) les profits, ou on peut plutôt dire minimiser les coûts, n'induit pas forcément la réduction des coûts globaux de la chaîne, ça peut même produire l'effet inverse.

La coordination entre la production et la distribution est difficile à mettre en œuvre car il s'agit de coordonner les efforts au niveau opérationnel, c'est-à-dire à un niveau où la vision de la globalité de la chaîne n'est pas aisée à faire. De plus, la production et la distribution peuvent être séparées par des stocks intermédiaires, et sont gérées par des départements différents au niveau de la chaîne ou d'une entreprise. Le challenge ici est d'échanger suffisamment d'informations et à un débit suffisant pour pouvoir mettre en œuvre des ordonnancements des machines, et des transports (programme de livraison, tournées des véhicules) qui visent à réduire l'ensemble des coûts. En outre, le système de transport conditionne en grande partie les performances du juste à temps. En pratique, à notre avis, les liens entre production et transport déterminent l'efficacité opérationnelle de la chaîne.

Coordination Stocks-Distribution

La gestion des stocks a été beaucoup étudiée dans la littérature. Les chercheurs s'y sont intéressés car les praticiens de la gestion de production considéraient les surplus de stocks comme l'ennemi principal de la productivité de leurs entreprises. Cependant, la gestion des stocks n'est pas, à elle seule, un gage de performance pour l'entreprise. Celle-ci doit savoir gérer ses stocks en liens avec les autres fonctions. Déterminer le niveau optimal des stocks est conditionné par la capacité de production, les prévisions de la demande et des moyens de transport. La fonction de distribution constitue le niveau en amont de la fonction de gestion de stocks dans la chaîne logistique. Comme pour l'équipe de relais, les relations entre les gestionnaires des stocks et les développeurs et planificateurs des programmes de distributions doivent être renforcées. De plus, le juste à temps impliquant un changement de philosophie, ces deux fonctions doivent être très réactives ; les performances de l'une influençant l'autre. Cette nouvelle organisation induit une plus grande fréquence des livraisons par petits lots. Pour éviter de gaspiller en coûts de transports les économies effectuées dans les étapes précédentes, l'entreprise doit coordonner ces décisions au niveau de la gestion de stocks et

prendre en compte les contraintes de transport dépendant de la localisation du client, de la quantité livrée, des conditions de transport (qui dépendent de la nature du produit transporté, certains produits peuvent nécessiter un mode de transport particulier). Pour être réactif, et ainsi ne pas perdre en efficacité, le service des transports doit avoir les bonnes informations au bon moment de la part du service de stocks.

Dans notre thèse, le but recherché est d'améliorer les performances d'une chaîne logistique par le biais de la coordination des ordonnancements au niveau opérationnel. L'importance de l'optimisation à ce niveau de la chaîne a été pointée par (Thomas and Griffin, 1996). Dans la littérature, les problèmes d'ordonnement dans la chaîne logistique commencent à intéresser la communauté scientifique. Cet intérêt est motivé par les besoins de coordination des efforts à ce niveau.

(Steiner et Salvarajah, 2006) étudient le problème d'ordonnement par lots dans une chaîne logistique à deux niveaux du point de vue du fournisseur. Le système du fournisseur est modélisé par un système à une machine où le traitement d'un lot nécessite un temps de réglage de la machine s'il s'agit d'un nouveau type de produits. Ils proposent un algorithme polynomial pour la minimisation de la somme des coûts de livraisons et de possession de stocks. (Agnētis et al, 2006) considèrent deux niveaux de la chaîne logistique avec un fournisseur et un ou plusieurs fabricants. Ils comparent les résultats entre le développement de solutions localement optimales pour chacun des participants en prenant en compte séparément leurs contraintes spécifiques. Le problème de coordination de l'ordonnement au sein de cette chaîne est d'ordonner simultanément les activités de tous les participants. Ils donnent en outre certaines conditions nécessaires au bon fonctionnement d'un processus de coordination.

(Hall et Potts, 2003) étudient les bénéfices et l'intérêt de la coordination des décisions d'ordonnement au sein de la chaîne logistique. Ils s'intéressent à la coordination des décisions d'ordonnement, de la taille des lots, et de livraison au sein du même niveau de la chaîne, et entre les différents niveaux. Ils présentent des algorithmes à base de programmation dynamique pour montrer l'intérêt de la coordination comparée à des systèmes soit du point de vue du fournisseur, soit du point de vue du fabricant. Les aspects de coordination de l'ordonnement des machines et de tailles de lots sont traités plus spécifiquement dans (Hall et Potts, 2005). (Ertogral et al, 2006) quant à eux, intègrent la taille des lots pour exprimer les coûts de transport dans un système impliquant deux parties (un acheteur et un vendeur).

(Chen et Hall, 2002) comparent les effets d'un système de coopération, et d'un système non coopératif dans le cadre d'un problème de système d'assemblage où les composants sont fournis par plusieurs fournisseurs. (Dawande et al, 2006) étudient des scénarios de coopération à différents niveaux entre un fabricant et un distributeur. Ils étudient les effets de la coopération sur la minimisation des temps de non productivité chez le fabricant et les coûts de transport chez le distributeur. Ils démontrent que la coopération génère des effets plus importants que dans le cas où les deux parties ne coordonnent pas leurs ordonnancements. D'autres aspects de la coordination entre fabricants et fournisseurs relatifs au partage des bénéfices sont discutés dans (Gupta et Weerawat, 2005).

(Li et Xiao, 2004) analysent et développent des mécanismes de coordination au sein d'une chaîne logistique constituée de plusieurs fournisseurs et de plusieurs producteurs. Ces mécanismes sont développés dans le but de générer des profits optimaux pour l'ensemble de la chaîne en coordonnant les décisions de détermination des tailles des lots des biens produits.

(Kreipel et Pinedo, 2004) étudient des modèles qui intègrent la coordination entre les niveaux de planification et d'ordonnement pour le développement d'outils d'aide à la

décision dans les chaînes logistiques. Cette coordination a pour but d'optimiser les coûts de possessions de produits intermédiaires, les coûts de transports, les coûts de retards de livraisons, et les coûts de mise en marche des équipements. (Dauzere-Pérès et Lasserre, 2002) montrent les limites des approches traditionnelles de planification et d'ordonnancement de la production. Ils observent que souvent, les décisions prises au niveau de la planification sont conflictuelles avec les décisions au niveau de l'ordonnancement. Ils proposent une procédure itérative à deux étapes pour la minimisation des coûts de stocks et de production. Ils intègrent un module d'ordonnancement au niveau de la planification, non dans le but de déterminer l'ordonnancement à ce niveau de la chaîne, mais pour guider le module de planification à considérer les contraintes d'ordonnancement.

(Neiro et Pinto, 2004) proposent un cadre général pour l'optimisation d'une chaîne logistique pétrolière avec l'intégration de plusieurs aspects du niveau opérationnel lors de l'établissement de la planification. Ce modèle intégré est valable pour plusieurs périodes de planification. Le modèle proposé prend en compte à ce niveau la sélection des matières premières et l'établissement de leurs plans de transports, les quantités à produire sous les contraintes de qualité et de capacité, la gestion des stocks, et les programmes de livraisons des produits finis. Ce modèle est formulé par un programme non linéaire en nombres entiers mixtes. Un modèle de coordination au niveau tactique est proposé par (Meijboom et Obel, 2007) dans le cas d'une chaîne logistiques globale avec des infrastructures réparties dans plusieurs pays et ayants plusieurs niveaux de fonctionnement.

(Chang et Lee, 2004) étudient le problème d'intégration dans le même modèle de l'ordonnancement de la production et l'ordonnancement des livraisons. Ils considèrent une chaîne logistique à deux étages où le premier étage fabrique des biens, et le deuxième étage s'occupe de l'expédition de ces biens aux clients. Ils traitent en particulier le problème où les différents produits nécessitent des volumes de stocks différents dans les moyens de transports. Ils proposent des heuristiques avec des analyses du pire cas.

(Li et al, 2005) étudient le problème d'intégration de la production et des transports à des clients se trouvant dans différentes localisations géographiques. Le problème étant NP-difficile, ils proposent des algorithmes à base de programmation dynamique pour certains cas spécifiques.

(Wang et Lee, 2005) étudient un nouveau problème émergent dans la chaîne logistique qui consiste en la sélection du mode des transports simultanément avec l'ordonnancement de la production. Ils supposent que, au niveau des transports, il y a deux modes de transport avec des coûts et des délais différents. Ils proposent une approche qui intègre ces deux fonctions de la chaîne pour maximiser le profit global en minimisant la somme des coûts de transport et la somme pondérées des retards. Pour la résolution du problème, ils ont développé un algorithme exact basé sur les procédures par séparation et évaluation. (Pundoor et Chen, 2005) proposent des résultats de complexité et des heuristiques pour les cas NP-difficiles du problème de l'ordonnancement conjoint de la production et des transports avec l'objectif de minimiser la somme des coûts de transport, ainsi que le plus grand retard.

(Naso et al, 2007) étudient la coordination entre la production et les transports, dans un réseau d'infrastructures partiellement indépendantes pour garantir des livraisons en juste à temps à des clients avec des localisations géographiques distribuées. Il y a des contraintes dures qui interdisent les avances et retards de livraisons par rapport à une fenêtre de temps donnée pour chaque commande. Ils proposent une approche basée sur les algorithmes génétiques.

(Huang et Iravani, 2007) étudient le cas de deux niveaux de la chaîne logistique qui sont un producteur pour le premier niveau, et deux détaillants pour le second niveau. Le producteur

ayant des entrepôts mais avec des capacités limités et un coût de stockage. Ils proposent des politiques pour avoir une coordination optimale entre la production et le niveau des stocks en utilisant des outils de partage d'informations.

1.5 Conclusion

Dans ce chapitre, nous avons exposé les concepts liés à la chaîne logistique et à l'ordonnancement dans une moindre mesure. Nous avons vu la complexité des chaînes logistiques, leurs composantes, les participants, les mesures de performances, les méthodes proposées pour leurs modélisations, et le supply chain management. A travers l'ensemble des définitions que nous avons donné, nous remarquons la difficulté de situer la notion de chaîne logistique. Cette dernière dépend de la nature du produit ou du service fourni, et de la nature des relations entre les participants.

Nous avons ensuite introduit les problèmes d'ordonnancement, et tout particulièrement le flowshop. La notion du juste à temps, ses exigences, les facteurs de sa mise en œuvre et comment y arriver, sont exposés. Enfin, nous avons clôturé ce chapitre par une étude bibliographique sur les problèmes de coordination dans la chaîne logistique au niveau opérationnel, c'est à dire au niveau de l'ordonnancement.

A notre connaissance, il n'existe pas dans la littérature de travaux de recherche bibliographique qui prennent en compte en même temps la coordination et l'aspect ordonnancement simultanément. Cela est dû au fait que les chercheurs ne se sont intéressés à l'ordonnancement dans la chaîne logistique que tout récemment.

Vu la complexité des chaînes logistiques, beaucoup de problèmes ont été étudiés notamment pour la conception et la modélisation de la chaîne. Le problème principal étudié dans cette thèse est original, et n'a jamais été étudié dans la littérature. Dans le chapitre suivant, nous donnons la formalisation du problème et nous expliquons l'approche de résolution adoptée.

2 Chapitre 2: Le Problème et l'Approche retenue

Résumé

Dans ce chapitre, est présenté le principal problème étudié dans cette thèse. Nous présentons l'architecture générale de la chaîne logistique dans laquelle nous étudions des problèmes d'ordonnancement en juste à temps. Pour un système globale semi décentralisé, nous proposons un mécanisme de négociation entre les partenaires pour permettre la mise en œuvre du juste à temps, et pour garantir la convergence du système global dans le respect des contraintes de dates de disponibilité des matières premières, et des dates de livraisons des produits finis.

2.1 Introduction

Dans le présent chapitre, nous allons détailler le problème étudié qui constitue la majeure partie de cette thèse. Il s'agit du problème de négociation s'appuyant sur des fenêtres de temps strictes ou souples au niveau de l'ordonnancement dans la chaîne logistique.

Comme on l'a déjà dit dans ce mémoire, la globalisation a forcé les entreprises et les groupes d'entreprises à devenir plus compétitives au risque de perdre des parts de marché. De plus en plus d'entreprises « grossissent » en fusionnant avec d'autres entreprises pour former un ensemble beaucoup plus performant. Dans cet environnement fortement concurrentiel, les entreprises se doivent de réduire leurs cycles de développement, leurs délais de livraisons, leurs coûts de fonctionnement. La chaîne logistique étant un réseau d'organisations regroupant différents partenaires qui peuvent appartenir à diverses entités indépendantes les unes des autres, dont le but commun est de faire du profit (Goldratt, 2007). Pour atteindre cet objectif, chacun des partenaires réalisent des activités dans la chaîne pour donner de la valeur ajoutée au produit ou au service qui lui est fourni par le niveau en amont de la chaîne. Ces activités doivent être coordonnées pour garantir un fonctionnement global optimal. Les participants à la chaîne ne doivent pas se comporter comme des concurrents mais comme des partenaires partageant le même but et prenant des risques en commun. En effet, si les services de vente n'arrivent plus à écouler leurs produits ou services, ils ne feront plus de commandes aux producteurs, qui eux mêmes ne feront plus de demandes en matières premières ou en composants à leurs fournisseurs, et ainsi de suite. En outre, un manque de coordination peut engendrer une mauvaise qualité de service, détériorer la qualité du produit, augmenter les coûts de fonctionnement et surtout ternir l'image de l'entreprise.

Les cadres et dirigeants d'entreprises ont commencé à admettre la nécessité de développer des outils de coordination avec leurs autres partenaires aux niveaux des chaînes logistiques auxquelles ils appartiennent. Il reste donc à définir ces outils, et les moyens de leur mise en œuvre. Cette coordination est matérialisée par un processus de *négociation* avec les partenaires. Le plus souvent, dans la littérature, la négociation a été étudiée pour les niveaux les plus hauts de la chaîne, c'est à dire les niveaux stratégiques et tactiques. Aussi, la négociation est étudiée dans le contexte de prise de décisions centralisé. Cette hypothèse restreint le champs d'utilisation pratique de ces recherches car destinées à la coordination de parties isolées de la chaîne (appartenant à la même entité juridique), et n'ont donc pas de vision globale de la chaîne. Ces modèles de négociations peuvent générer de bonnes solutions pour les parties concernées, seulement, au niveau de la chaîne, ces solutions sont *locales*. On rappelle ici que la somme d'optima locaux ne donne pas généralement une solution globale optimale, le plus souvent ça a des effets néfastes sur le reste des composantes. Dans ce contexte il faut trouver une stratégie *gagnant-gagnant* pour l'ensemble des participants de la chaîne.

Plusieurs modèles de négociation centralisée ou semi-centralisée ont été proposés récemment dans la littérature. (Dudek et Stadtler, 2005) étudient le problème de synchronisation des planifications entre deux entités indépendantes d'une chaîne logistique liées par des flux de matières. Le modèle de négociation est basé sur un processus itératif pour déterminer les quantités à approvisionner ou à livrer, ou bien à modifier des plannings déjà établis. Les planifications sont générées par des modèles de programmation mathématique. Jung et al. (2006) proposent un processus itératif de négociation entre un acheteur (réseau de distribution) et un vendeur (réseau de production) dans une chaîne logistique à prise de décisions décentralisée car les deux entités appartiennent à des organisations différentes. Cette chaîne est dirigée par le distributeur. Ce dernier formule des commandes avec des quantités et des prix fixés par le marché, et émet ses commandes au producteur qui lui communique la

quantité de produits (de différents types) qu'il peut approvisionner. Les auteurs montrent que leur modèle fournit des résultats proches de ceux générés par des systèmes de prise de décisions centralisées nécessitant un partage complet d'information, alors que dans leur modèle seules les informations concernant les quantités demandées et les quantités fournies sont partagées.

L'une des techniques souvent utilisées pour modéliser une chaîne logistique ou bien pour modéliser un processus de négociation en son sein est le système multi-agents. Lin et Lin (Lin et Lin, 2006) les utilisent pour intégrer des techniques de négociation pour le problème de satisfaction de commandes dans une chaîne logistique. Ce problème est modélisé par un problème de satisfaction de contraintes distribué. La négociation est utilisée pour coordonner les décisions distribuées concernant la satisfaction de commandes. Jiao et al. (Jiao et al., 2006) utilisent un système multi-agents pour élaborer un cadre de négociations au sein d'une chaîne logistique manufacturière où il existe plusieurs centres de décisions autonomes ou semi autonomes. Pour leur part, Monteiro et al. (Monteiro et al, 2007) proposent un système multi-agents pour une chaîne logistique multi-sites.

Neubert et al. (Neubert et al, 2004) proposent une architecture avec des agents pour développer un processus automatique de négociations de contrats avec les fournisseurs dans le cadre d'un réseau d'unités de production autonomes. Ce modèle est basé sur des propositions et des contre proposition sur les prix, les dates de livraisons et les quantités à livrer, le tout chapeauté par des agents pour garantir l'harmonisation du système global.

Dans sa thèse au sein de l'équipe Macsi du Loria, (Ouzizi, 2005) propose des outils pour les partenaires d'une chaîne logistique qui désirent « se structurer en entreprise virtuelle pour mieux coordonner leurs activités de planification de la production et de la distribution », ainsi qu'une architecture informatique comme support pour la négociation entre les différents partenaires pour établir la planification co-décidée. Elle utilise une stratégie de négociation itérative et convergente dans un environnement semi distribué pour calculer des plannings de production cohérents à partir de données qui sont les courbes cumulées d'approvisionnement et de livraison qui peuvent être souhaitées, imposées, ou calculées. Une architecture en système multi agents est proposée pour modéliser le système global. Parmi les méthodes de résolutions utilisées on trouve les algorithmes génétiques et la programmation linéaire.

Le problème de négociation étudié dans cette thèse est une continuation du travail effectué par (Ouzizi, 2005) et (Ouzizi, 2006). Les travaux précédents se situent au niveau tactique de la chaîne logistique. Les modèles développés pour la planification ne peuvent être efficaces que si les ordonnancements et les décisions prises au niveau opérationnel sont cohérents avec les plannings provenant du niveau supérieur. Au niveau opérationnel de la chaîne, rares sont les travaux réalisés, car dans la communauté scientifique on semble préférer se focaliser sur les niveaux stratégique et tactique. Dans une logique de continuation dans notre équipe de recherche, et après avoir étudié et proposé des modèles de planification et de négociation au niveau tactique, nous étudions dans la présente thèse les problèmes de coordination au niveau opérationnel. Cette coordination est mise en œuvre via un processus de négociation. L'architecture de la chaîne logistique considérée est décrite dans la section 2 de ce chapitre avec les relations entre les différents partenaires, les hypothèses de fonctionnement, ainsi que les contraintes et l'objectif à optimiser. Dans la section 3 est décrit le schéma global de la négociation et de la méthode de résolution proposée.

2.2 Définition du problème

2.2.1 Fonctionnement général

Nous considérons une chaîne logistique composée de plusieurs entreprises indépendantes. L'indépendance des partenaires implique des centres de décisions indépendants. Nous nous retrouvons donc dans le cas de prise de décision distribuée (décentralisée ou semi-décentralisée). La prise de décision distribuée est le scénario le plus plausible dans la réalité. En effet, rares sont les réseaux ou les chaînes logistiques dont toutes les composantes appartiennent à la même entité juridique. Nous distinguons trois composantes principales du système global de chaîne logistique étudié : les fournisseurs externes (n'appartenant pas à la chaîne), la chaîne logistique, et un ensemble de clients externes (n'appartenant pas à la chaîne). La figure 3.1 illustre son schéma général.

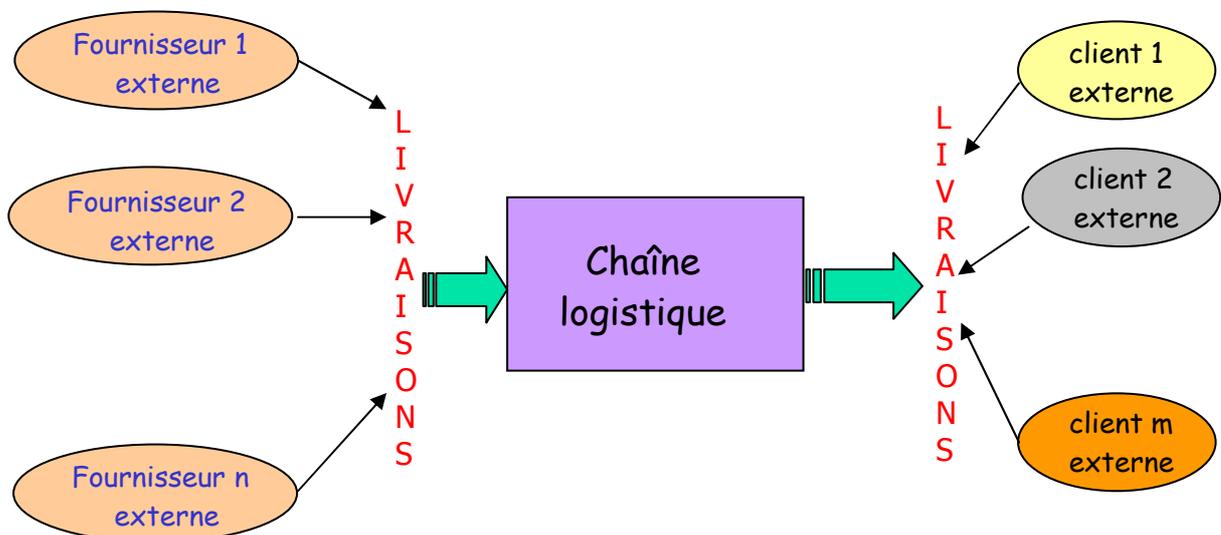


Figure 2.1 Schéma général de la chaîne logistique.

Les fournisseurs approvisionnent le système de chaîne logistique en matières premières et composants. La chaîne logistique au centre de ce système est elle même composée de plusieurs nœuds élémentaires juridiquement indépendants. Comme le montre la figure 2.2, il peut y avoir des liens entre n'importe quel couple de nœuds, on n'impose pas de structure linéaire entre les nœuds. A l'intérieur de la chaîne logistique on dispose de systèmes de production qui fabriquent des produits finis qui sont délivrés aux clients externes selon un programme de livraison et des quantités établis en avance (au niveau planification).

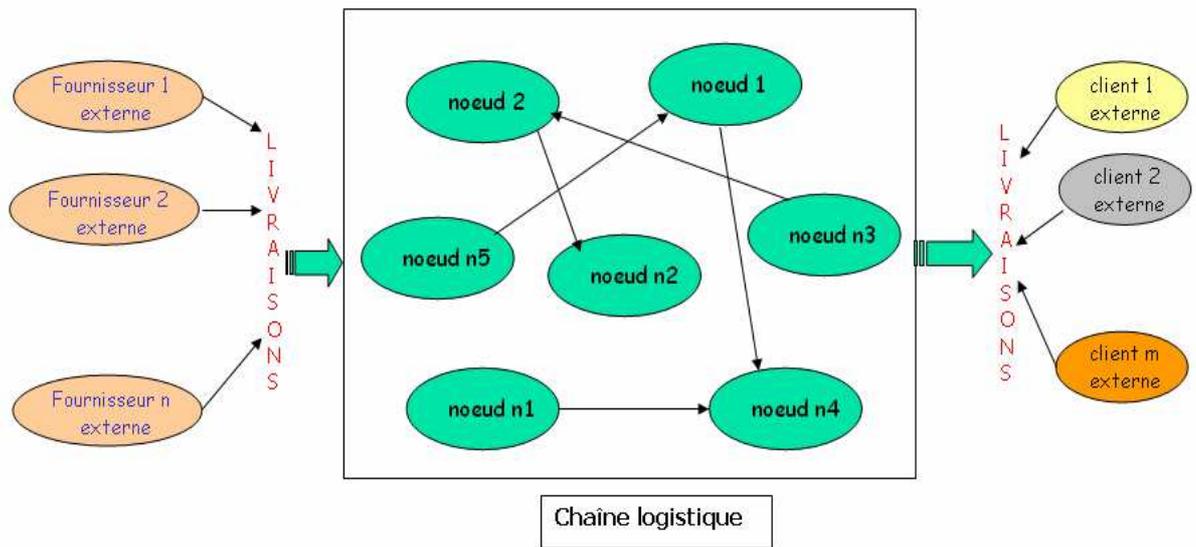


Figure 2.2 les éléments de la supply chain.

Chacun des nœuds élémentaires est un système de production capable de fabriquer des produits semi-finis, ou des produits finis. Il peut avoir des liens avec des fournisseurs externes, comme il peut être approvisionné en composants ou sous produits par d'autres nœuds de la chaîne logistique. Une telle structure est illustrée par la figure 2.3 suivante :

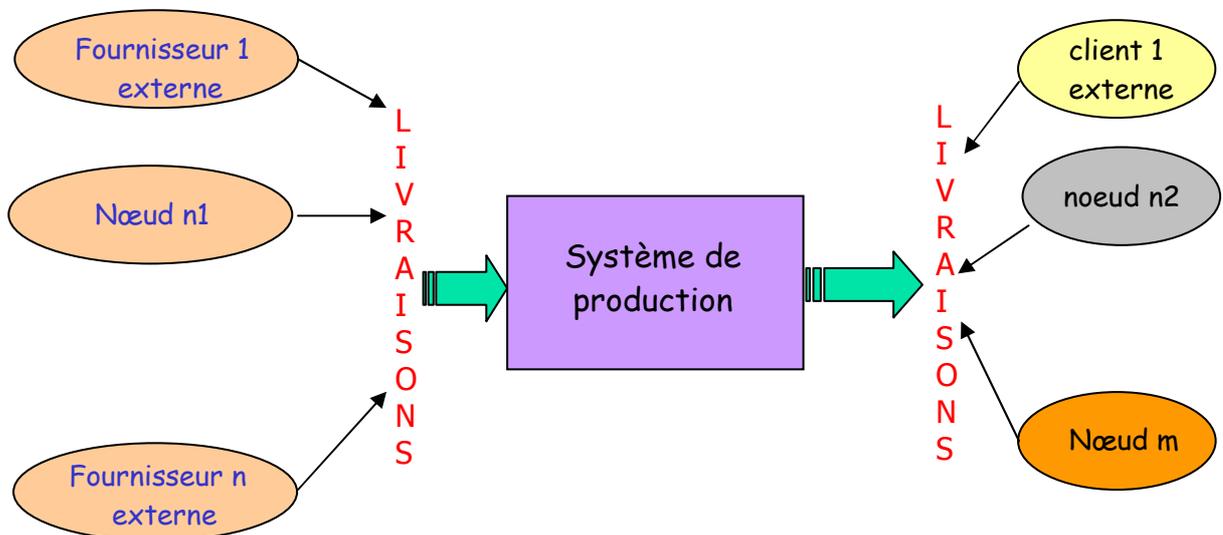


Figure 2.3 Interactions entre les différents éléments de la supply chain.

Les centres de décisions des centres de production dans le système global de production étant indépendants, chacun d'entre eux construit son ordonnancement par rapport à son plan de production. Une telle approche totalement décentralisée peut mener vers un ordonnancement global complètement déséquilibré. A certains moments, on peut avoir des quantités en surplus de certains composants et un manque dans d'autres types de composants, ce qui peut mener vers un niveau de stocks élevé, des retards de livraisons et à des temps d'oisiveté (donc de non productivité). Une synchronisation des activités au sein du système

global de production s'avère donc nécessaire. Nous choisissons donc un système de décision distribué semi-décentralisé dans lequel les partenaires se partagent une mémoire commune et se mettent d'accord sur un ensemble de règles de fonctionnement (aux niveaux tactiques et opérationnels) pour garantir la convergence du processus global de négociation (éventuellement par la relaxation de certaines contraintes dans le cas de l'absence de solutions faisables pour le problème initial). Un schéma du processus de pilotage semi-décentralisé est donné par la figure 2.4.

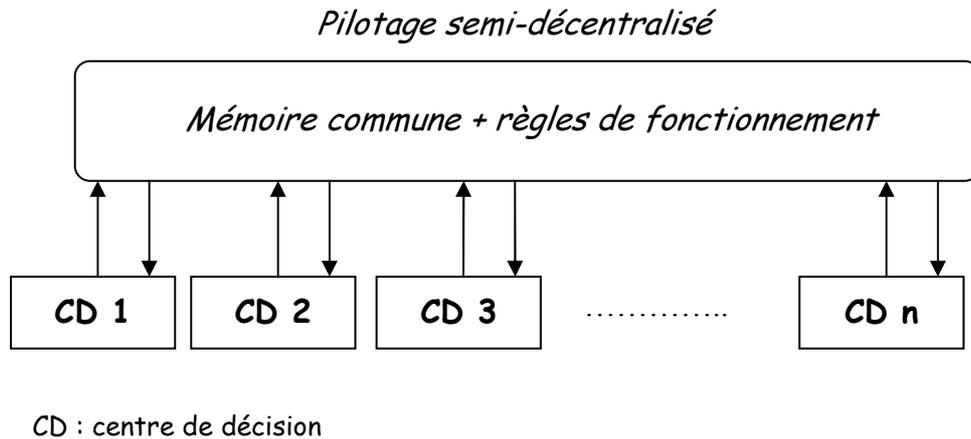


Figure 2.4 schéma du processus de pilotage semi-décentralisé.

2.2.2 Exemple

Pour mieux comprendre le système global, nous illustrons par un cas particulier où le système de production, à l'intérieur de chaque nœud élémentaire, est composé de deux étages. Le premier étage est un ensemble de centres de production indépendants (avec des centres de décisions indépendants) qui reçoivent les matières premières et les composants des fournisseurs externes et les transforment en composants ou produits semi-finis. Les produits semi-finis fabriqués par le premier étage sont acheminés vers le deuxième étage qui consiste en un centre d'assemblage de produits finis. Nous supposons que les temps de transport entre les centres de productions sont pris en compte par des contraintes d'écart temporel (time lags).

Les centres de production du premier et du deuxième étages sont modélisés par des flowshop (ou éventuellement jobshop). Nous avons choisi la modélisation par des structures d'ateliers à acheminement unique (flowshop) car c'est la structure la plus utilisée dans l'industrie surtout en ce qui concerne l'assemblage de composants.

Au premier étage, les centres de productions peuvent être vus comme des lignes de flowshop en parallèle comme illustré par la figure 2.5.

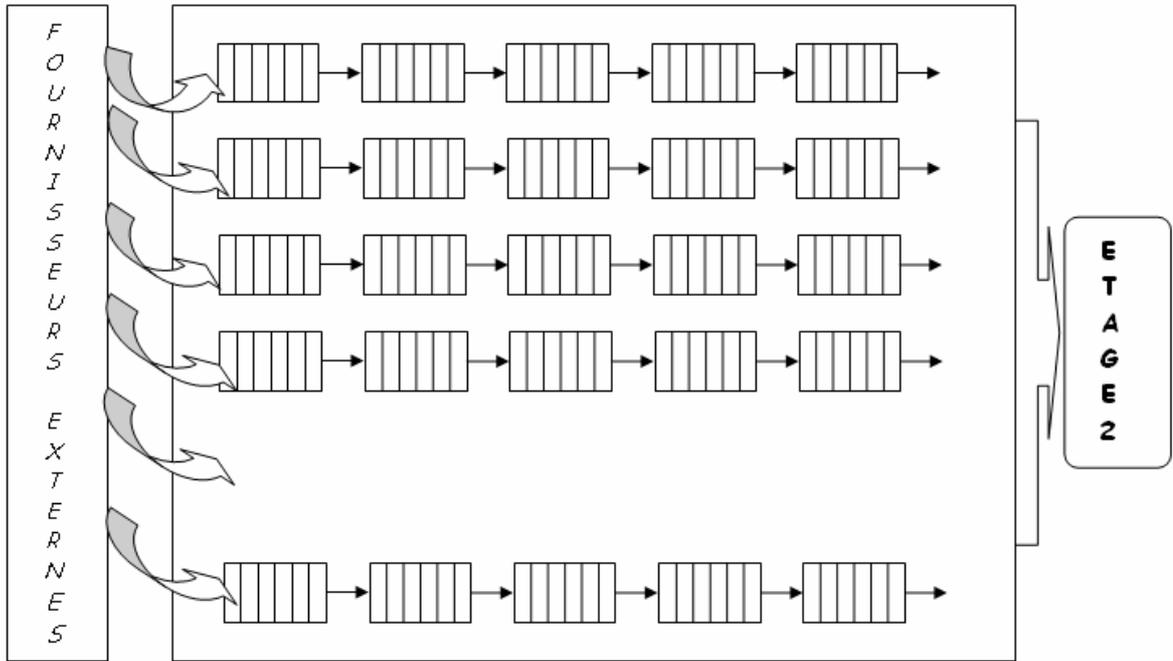


Figure 2.5 premier étage du système de production.

Les flowshops sont formés par des machines en série sur lesquelles passent les composants pour ajouter de la valeur. Le nombre de machines diffère d'un flowshop à un autre. A chaque flowshop est associé un centre de décisions qui négocie les dates d'approvisionnement avec les fournisseurs des composants et des matières premières. Le deuxième étage est aussi un flowshop, un atelier d'assemblage de produits finis à partir des produits semi-finis fournis par les centres de production. Un schéma du fonctionnement général de cet étage est illustré par la figure 2.6.

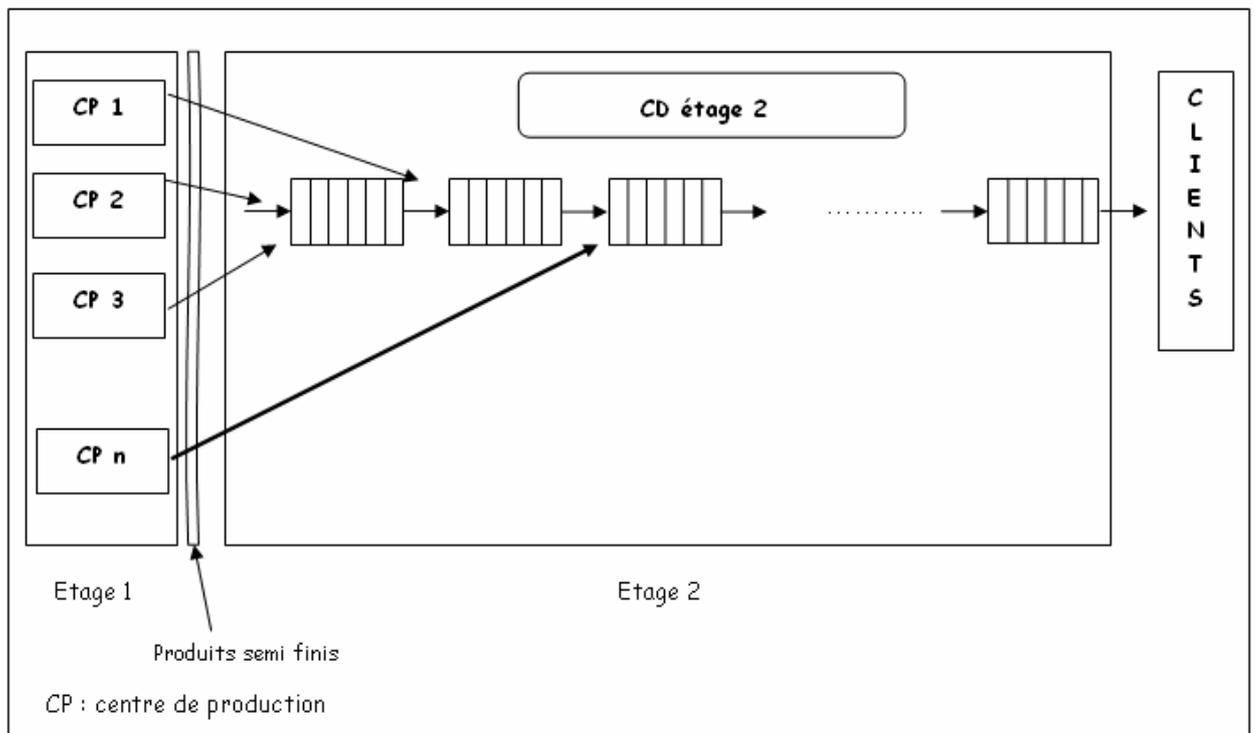


Figure 2.6 fonctionnement du deuxième étage du système de production.

Une opération dans le premier étage ou bien dans le deuxième n'est réalisée que si tous les composants ou les produits semi-finis nécessaires à son exécution sont disponibles. Ceci implique des dépendances entre ces différents centres pour établir leurs ordonnancements. Des négociations sont donc souhaitées pour mener l'ensemble du système vers un état de fonctionnement optimal. Les centres de décisions négocient des dates de disponibilité (release dates) avec leurs fournisseurs, et des dates de livraisons (due dates) avec les clients finaux. Des dates dures (impératives) de disponibilité des composants (hard release dates), et des dates dures de livraison des produits finis (hard due dates) sont des contraintes impératives, leur non respect implique une infaisabilité de la solution proposée par le processus de négociation. Les dates flexibles (souples ou souhaitées) de disponibilité et de livraison induisent des pénalités d'avances et de retards par rapport à ces dates. Il s'agit donc de fenêtres de temps associées aux demandes d'approvisionnement et aux commandes. Cet intervalle représente la période de temps pendant laquelle une livraison ou un arrivage de composants est considéré comme faisable (c'est à dire respectant les contrats établis préalablement, ou des contraintes physiques incompressibles). La somme totale des pénalités d'avance et de retard de chaque arrivage ou livraison par rapport à sa date souhaitée représente le coût total de la solution.

Comme nous nous intéressons aux ordonnancements au niveau opérationnel, nous supposons que les planifications des transports, de la production, des volumes associés à chaque famille de produits, de composants ou de produits semi-finis sont pris en compte au niveau tactique. La transformation de ces plannings en commandes, et l'affectation de ces commandes aux centres de production sont également établis entre le niveau tactique et le niveau opérationnel.

Dans ce qui suit, nous donnons un récapitulatif des hypothèses, de l'architecture, et des règles de fonctionnement de l'ensemble de la chaîne.

Le premier étage

- Les centres de production dans cet étage négocient les dates des arrivées des composants avec les fournisseurs externes.
- Le premier étage du système de production fabrique les sous-produits (ou produits semi finis).
- Tous les composants et les matières premières associées à une opération doivent être disponibles avant de la commencer sur les lignes du premier étage.
- Il est organisé en lignes de production, une ligne pour chaque type de sous produit. Chaque ligne étant un flowshop.
- Chaque ligne contient un nombre donné de machines.
- Chaque type de sous produit utilise un nombre donné de composants.
- Chaque composant est utilisé sur une seule machine.
- Les lignes de production fonctionnent en parallèle, et dépendent de centres de décisions différents.
- Chaque sous produit en cours d'élaboration passe un temps d'exécution différent d'une machine à une autre qui dépend du sous produit et de la ligne de production sur laquelle il se trouve.
- Les flowshops peuvent avoir des localisations géographiques différentes.

- Les composants et les produits semi finis fabriqués sont acheminés vers le centre d'assemblage (le deuxième étage).
- Les sous produits sont regroupés en série et une série de sous produits n'est jamais interrompue. Le découpage en séries est effectué avant l'ordonnancement.

Le deuxième étage

- Il s'agit d'une seule ligne de production composée d'un nombre donné de machines pour l'assemblage de sous produits en produits finis (fenêtres de temps pour les release dates).
- On y fabrique différents types de produits finis.
- On a besoin des sous produits résultants du premier étage de production pour fabriquer les produits finis.
- Chaque produit a besoin d'un certain nombre de sous produits, chaque sous ensemble de sous produits est assemblé sur une machine dédiée.
- Chaque famille de produits finis est définie par le nombre et le type de chaque sous produit qu'elle utilise.
- Tous les sous produits doivent être disponibles avant de commencer l'exécution (assemblage) d'un produit.
- Les produits finis fabriqués sont expédiés vers les clients.
- Le centre de décision négocie des dates de livraisons avec les clients (fenêtres de temps pour les due dates).
- Une série de produits correspond à un travail (job), et n'est jamais interrompue, mais une commande a pu être découpée en plusieurs séries au niveau supérieur. Chaque série de sous produits du premier étage est affectée à une et une seule série de produits du deuxième étage.

La figure suivante (2.7) montre les différents participants de la chaîne logistique et leurs interactions.

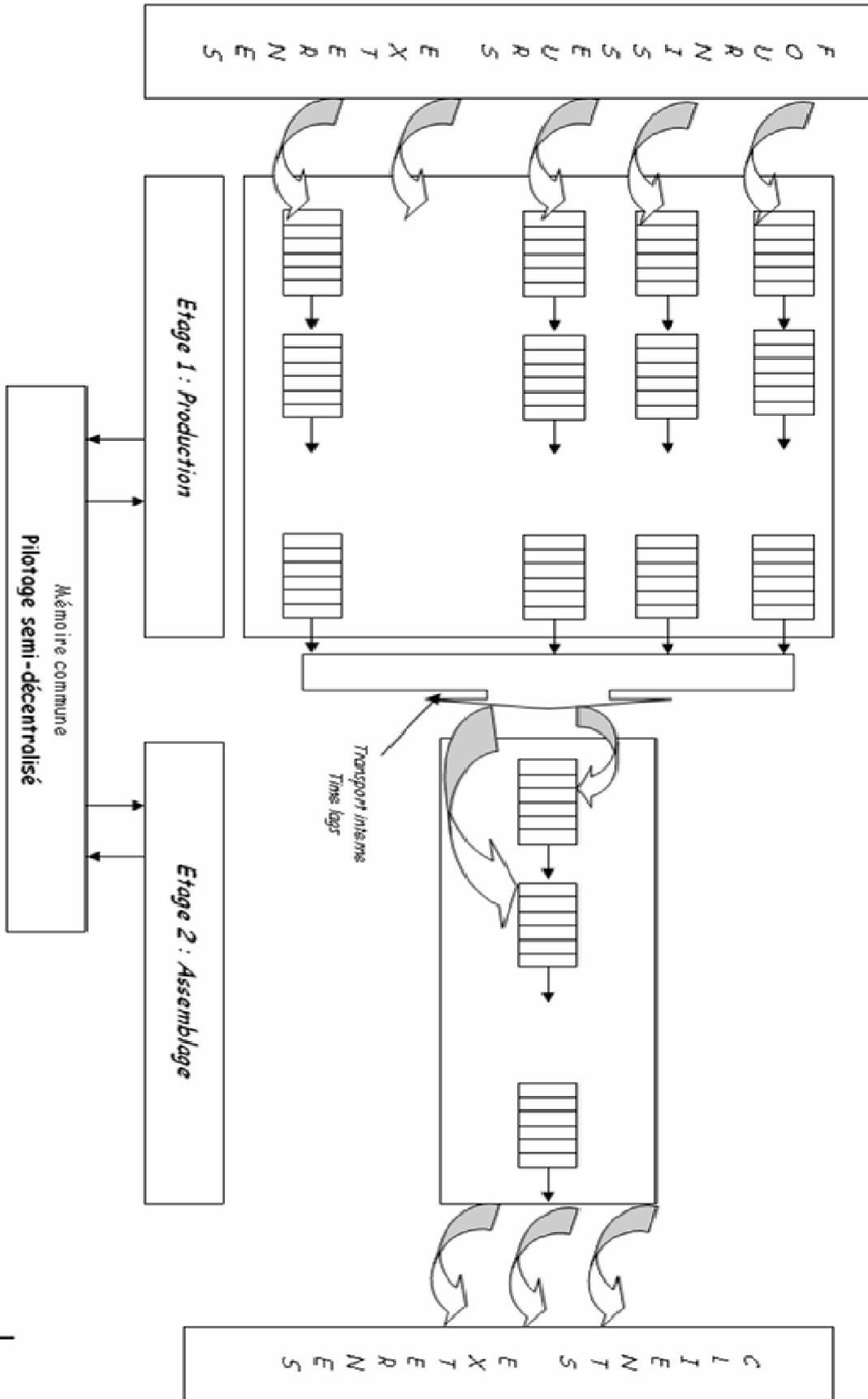


Figure 2.7 interactions entre partenaires et pilotage semi-décentralisé.

2.3 Problème de JIT particulier à la thèse

2.3.1 Hypothèses générales

Outre les notations $\alpha/\beta/\gamma$ que nous continuons d'utiliser et qui définissent toujours parfaitement tout problème d'ordonnancement, nous proposons des sigles spécifiques aux problèmes d'ordonnancement d'atelier en juste à temps que nous considérons dans cette thèse de manière à mieux mettre en évidence les problèmes les plus généraux et leurs cas particuliers.

Notre sigle commence toujours par les trois lettres JIT et signifie que les critères que nous utilisons sont exclusivement des critères d'avance et de retard. Ces trois lettres sont suivies de la partie α décrivant la structure d'atelier considéré si les travaux sont définis par des gammes, comme par exemple :

JIT-1 : pour les problèmes à une machine,

JIT-F : pour les problèmes de type flow-shop,

JIT-Fperm : pour les problèmes de type flow-shop de permutation

JIT-J : pour les problèmes de type job-shop

JIT-prec : pour les problèmes dont les gammes sont données par des contraintes de précédence entre les opérations (exemple : les problèmes d'assemblage).

Lorsque les éventuels conflits entre les opérations qui utilisent les mêmes ressources ont été levés en choisissant l'ordre de passage (ou séquence de passage) des opérations en conflit sur les ressources (en supposant que l'arbitrage effectué n'a pas créé de circuit), on obtient des cas particulier des problèmes précédents que l'on distingue en ajoutant « Seq » au sigle retenu, comme par exemple :

JIT-1-seq, JIT-F-seq, JIT-Fperm-seq, JIT-prec-seq et pour désigner le problème général d'atelier où les conflits ont été levés : JIT-seq (la partie description de l'atelier n'est plus précisée).

La deuxième partie de notre sigle concerne la manière de construire le critère d'avance et de retard à minimiser.

Dans le cas le plus général qui correspond aux problèmes résolus dans cette thèse, toute opération (considérée ici comme non interruptible) n'induit aucun coût d'avance (ou de retard) si son début est placé sur un intervalle dit « idéal ». Elle induit un coût proportionnel à l'avance si elle commence strictement avant le début de l'intervalle idéal et un coût proportionnel au retard si elle commence strictement après la fin de l'intervalle idéal. Il est à noter que si on définit plusieurs intervalles « idéaux » que l'on associe à la même opération, on peut définir ainsi une fonction de pénalité convexe, continue et linéaire par morceaux autour de l'intervalle idéal associé à l'opération (éventuellement réduit à un point).

Le problème général sera noté JIT/op($\Sigma E-T$) ou encore JIT/op($\Sigma E-T, w=1$) ou JIT/op($\Sigma E-T, w=k$) qui signifie que tout opération peut avoir un ou plusieurs intervalles « idéaux » et on peut préciser s'il n'y en a au plus 1 ou s'il peut y en avoir plusieurs. Par ailleurs, le signe Σ rappelle qu'il s'agit d'une somme de pénalités d'avance et de retard.

Il s'agit ici d'intervalle flexible ou mou et on pourrait, en principe, placer l'opération n'importe où (y compris sur les valeurs négatives de l'axe temps). On suppose en fait que toute opération doit être placée sur un horizon défini par l'intervalle de temps $[0, H]$. On suppose toujours que les instants de début des opérations sont positifs, il n'est donc pas

nécessaire de l'inclure dans le sigle. Par contre, s'il est important de rester dans l'horizon H , on l'indique dans le sigle : $JIT/op(\Sigma E-T, H)$, de même, s'il existe des contraintes strictes à respecter pour le début au plus tôt et au plus tard des opérations, on l'indique dans le sigle : $JIT/op(\Sigma E-T, hw)$ où hw signifie Hard Windows (ou date au plus tôt et au plus tard impératives ou fenêtres impératives).

Des cas particuliers ont été considérés dans cette thèse, lorsque l'on considère que seules les premières et les dernières opérations d'un travail impliquent des négociations avec l'amont et/ou l'aval de la chaîne logistique. Nous utilisons des notations pour définir ces cas particuliers qui peuvent permettre d'utiliser des méthodes de résolution simplifiée.

$JIT/job(\Sigma E1-Tm)$ correspond au cas très particulier où on ne considère des pénalités d'avance que pour la première opération de chaque travail et des pénalités de retard que pour la dernière opération de chaque travail.

$JIT/job(\Sigma E1-T1-Em-Tm)$ correspond au cas particulier où on considère des pénalités d'avance et de retard pour la première et pour la dernière opération de chaque travail, car on peut négocier en amont et en aval de la chaîne logistique des avances et/ou des retards.

En ce qui concerne les méthodes de résolution que nous présentons dans le chapitre 4, on peut déjà signaler que le problème $JIT-seq/job(\Sigma E1-Tm, w=k)$ peut être résolu avec la méthode classique connue du Pert Coût, mieux, on peut accélérer l'algorithme primal en tenant compte de la position particulière dans le graphe de précedence des arcs qui peuvent être compressés.

Par contre le problème $JIT/job(\Sigma E1T1-Em-Tm)$ et plus généralement $JIT/op(\Sigma E-T)$ nécessite une adaptation de la méthode du Pert Coût où il faut gérer non seulement des arcs que l'on contracte, mais également des arcs que l'on dilate.

Néanmoins, les arcs compressés et les arcs dilatés sont toujours adjacents à la source et au puits, ce qui permet de conserver l'amélioration de la mise en œuvre de l'algorithme primal du Pert Coût.

2.3.2 Le problème d'ordonnement en juste à temps

On a vu qu'après l'introduction des fenêtres de temps souples et dure (éventuellement virtuelles) entre les différents centres de décisions, les ordonnancements de ces derniers deviennent complètement indépendants. A l'intérieur de chaque centre de décisions on a donc un problème d'ordonnement en juste à temps avec comme fonction objective de minimiser la somme des avances et des retards par rapport aux délais négociés. Nous ne prenons pas en compte ici d'autres critères comme la minimisation des stocks, ou la somme des heures supplémentaires, ou les coûts de sous-traitance, maintenance...

Pour le problème particulier où on n'a des pénalités d'avance et/ou de retards que sur les premières et dernières opérations de chaque job, on considère que les centres de production sont organisés en atelier de type flowshop, et que les dates souhaitées et impératives de disponibilité (soft and hard release dates) sont associées aux premières opérations de chaque job (ceci est logique si tous les composants externes sont assemblés sur la première machine puis transportés sur les machines suivantes. C'est la date d'exécution de la première opération de chaque job qui détermine alors s'il est en avance ou en retard par rapport à sa date de disponibilité) ; et que les dates souhaitées et impératives de livraison (soft et hard due dates) sont associées à la dernière opération de chaque job (c'est la date d'exécution de la dernière opération de chaque job qui détermine si le produit correspondant est en avance ou en retard par rapport à sa date de livraison). En utilisant la notation classique en ordonnancement

$\alpha / \beta / \gamma$, nous pouvons donner la notation suivante à chaque problème d'ordonnancement interne : $F / \overline{r}_i, r_i, d_i, \overline{d}_i / \sum \alpha_{r,i} E_i^r + \sum \beta_{r,i} T_i^r + \sum \alpha_{d,i} E_i^d + \sum \beta_{d,i} T_i^d$. Tel que :

- F : décrit une structure d'atelier en flowshop.
- E_i^r et E_i^d sont les avances par rapport à r_i, d_i respectivement.
- T_i^r et T_i^d sont les retards par rapport à r_i, d_i respectivement.

Il est à noter qu'évidemment, une opération ne peut être à la fois en avance et en retard à la fois.

D'une manière plus générale, et si nous supposons que les dates souhaitées et/ou impératives peuvent être associées à n'importe quelle opération, alors nous décrivons le problème par la notation suivante : $J / t \min_j, t_j^*, t \max_j / \sum \alpha_j E_j + \sum \beta_j T_j$ tel que :

- t_j^* : est la date d'exécution idéale de l'opération j .
- $t \min_j, t \max_j$ sont respectivement les dates minimales et maximales autorisées pour l'opération j .

Pour résoudre ce problème localement au sein de chaque centre de décision, nous proposons une méthode par décomposition. Une solution du problème est donnée par la séquence (permutation ou ordre) de l'exécution des jobs. Si nous supposons que cette séquence est donnée ou fixée, alors nous devons calculer la date de début d'exécution de chaque opération sur toutes les machines de l'atelier avec autorisation de temps morts (idle times) car notre critère est irrégulier (on peut l'améliorer en retardant des opérations). Les séquences de jobs sont explorées par une méta-heuristique et sont modélisées par des *chromosomes*.

Dans le cas où une séquence est donnée par un chromosome, nous avons les notations suivantes des problèmes simplifiés $F_{seq} / \overline{r}_i, r_i, d_i, \overline{d}_i / \sum \alpha_{r,i} E_i^r + \sum \beta_{r,i} T_i^r + \sum \alpha_{d,i} E_i^d + \sum \beta_{d,i} T_i^d$ ou bien $j_{seq} / t \min_j, t_j^*, t \max_j / \sum \alpha_j E_j + \sum \beta_j T_j$. Ce problème est équivalent à un problème de *PERT coût*. Nous présentons avec plus de détail l'évaluation des chromosomes dans le chapitre 4.

La recherche de bonnes solutions faisables s'effectue avec des méthodes approchées. Nous avons choisi des méta-heuristiques du type évolutionniste (les algorithmes génétiques) et aussi du type recherche par voisinage (le recuit simulé). Nous générons des séquences avec ces méthodes que nous évaluons avec la méthode du *PERT coût*. Nous détaillons ces approches de résolutions dans le chapitre 4.

2.3.3 Notations

Pour ce problème nous adoptons les notations suivantes tout au long du présent chapitre. Pour le problème général :

Les données

- L : le nombre de lignes (centres de production) au premier étage.
- $l = \overline{1, \dots, L}$: indice de la ligne.

- $L+1$: indice de la ligne d'assemblage (deuxième étage).
- M_l : nombre de machines sur la ligne l .
- M_{L+1} : nombre de machines sur la ligne d'assemblage.
- $m_{j,l}$: indice de la machine j de la ligne l .
- N_l : nombre de jobs sur la ligne l .
- $P_{i,j,l}$: temps d'exécution de l'opération du job i sur la machine j de la ligne l .
- $r_{i,j}$: date de disponibilité souhaitée (soft release date) associée à l'opération du job i sur la machine j .
- $\overline{r_{i,j}}$: date de disponibilité impérative (hard release date) associée à l'opération du job i sur la machine j .
- $d_{i,j}$: date de livraison souhaitée (soft due date) associée à l'opération du job i sur la machine j .
- $\overline{d_{i,j}}$: date de livraison impérative (hard due date) associée à l'opération du job i sur la machine j .
- $\alpha_{r_{i,j}}$: pénalité d'avance par unité de temps par rapport à r_i associée à l'opération du job i sur la machine j .
- $\alpha_{d_{i,j}}$: pénalité de retard par unité de temps par rapport à r_i associée à l'opération du job i sur la machine j .
- $\beta_{r_{i,j}}$: pénalité d'avance par unité de temps par rapport à d_i associée à l'opération du job i sur la machine j .
- $\beta_{d_{i,j}}$: pénalité de retard par unité de temps par rapport à d_i associée à l'opération du job i sur la machine j .

Pour le cas particulier du JIT/job($\Sigma E1-T1-Em-Tm$)

- M : nombre de machines.
- m_j : indice de la machine j .
- N : nombre de jobs.
- $P_{i,j}$: temps d'exécution de l'opération du job i sur la machine j .
- r_i : date de disponibilité souhaitée (soft release date) associée à la première opération du job i .
- $\overline{r_i}$: date de disponibilité impérative (hard release date) associée à la première opération du job i .
- d_i : date de livraison souhaitée (soft due date) associée à la dernière opération du job i .

- \bar{d}_i : date de livraison impérative (hard due date) associée à la dernière opération du job i .
- α_{r_i} : pénalité d'avance par unité de temps par rapport à r_i associée au job i .
- α_{d_i} : pénalité de retard par unité de temps par rapport à r_i associée au job i .
- β_{r_i} : pénalité d'avance par unité de temps par rapport à d_i associée au job i .
- β_{d_i} : pénalité de retard par unité de temps par rapport à d_i associée au job i .

Variables

- $a_{i,j}$: date de début d'exécution de l'opération correspondant au job i sur la machine j .

Fonction objective

- minimiser la somme des avances et des retards par rapports aux dates souhaitées de disponibilités des composants, et de livraison des produits finis.

2.4 Approche de résolution du système global

2.4.1 Description Générale

Nous cherchons une solution globale au problème en utilisant une approche itérative de négociation basée sur une décomposition spatiale du problème global. Cette approche inclut des négociations bilatérales entre les centres de décisions des centres de production et la mémoire commune (pilotage semi-décentralisé) qui est chargée de garantir la convergence du processus global via la satisfaction des contraintes impératives sur les dates des arrivées des composants et les dates de livraisons des produits finis. Cette approche semi-décentralisé assure le contrôle du système global en respectant l'indépendance des entités participantes. L'objectif de la coordination des ordonnancements au sein de la chaîne globale est de minimiser une fonction des pénalités d'avance et de retards par rapports aux dates souhaitées (que ce soit avec les fournisseurs externes ou bien avec les clients externes). Une telle fonction objective implique un ordonnancement en juste à temps à l'intérieur de chaque centre de production. Le schéma global du processus de négociation est montré par la figure 2.8.

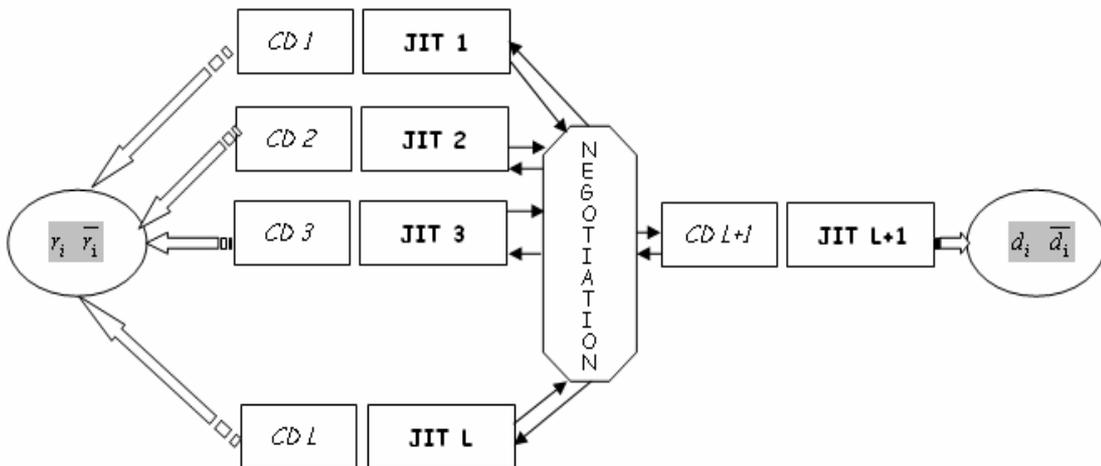


Figure 2.8 schéma global du processus de négociation.

Cette figure montre que chacun des centres de décisions a un problème d'ordonnancement en juste à temps qu'il doit gérer indépendamment, dans une certaine mesure, des autres centres. Les dates de disponibilités ayant été négociées entre les centres de décisions du premier étage et les fournisseurs externes, et les dates de livraisons entre les clients externes et le centre d'assemblage ; le tout piloté par les décisions prises au niveau de la planification (en termes de volumes des composants). Le but est de coordonner les ordonnancements en juste à temps à l'intérieur du système global de production afin de satisfaire les commandes et de minimiser les coûts induits par les avances/retards par rapport aux dates fixées.

2.4.2 Processus de négociation

Nous reprenons ici l'idée utilisée au niveau tactique (planification) de dates souhaitées et dates impératives sur les courbes cumulées des arrivées de composants et de livraison des produits finis avec des pénalités d'avance et de retard négociées par les partenaires de la chaîne logistique, en remplaçant les dates sur les courbes cumulées par des fenêtres de temps sur les opérations formant les travaux (jobs) à ordonnancer dans chaque centre de production.

Chaque commande émise par un client est convertie en un ou plusieurs jobs si la fabrication se fait par lots (ce qui est très souvent le cas dans l'industrie). Un job correspond donc à un produit, ou bien à des séries inséparables de produits, pour lesquels le détail des paramètres de l'ordonnancement sont connus : les centres de production à visiter, la nomenclature du produit (les sous produits et composants nécessaires à sa fabrication, et les dates de disponibilités sur la première machine de l'atelier dans lequel ils vont être traités). Les contraintes de dates souhaitées ou impératives de disponibilités ne sont associées qu'aux premières opérations de chaque job. De la même façon, les contraintes de dates souhaitées ou impératives de livraisons ne sont associées qu'aux dernières opérations de chaque job. Nous supposons que les dates de livraisons (impératives et souhaitées) ont été promises aux clients externes (qui n'appartiennent pas à la chaîne logistique), et que les dates de disponibilités (impératives et souhaitées) ont été négociées (des dates fermes pour la plupart à cause des frais de transport et des délais) avec les fournisseurs externes (qui n'appartiennent pas à la chaîne logistique).

Les méthodes itératives avec décomposition spatiale pour traiter des problèmes d'ordonnancement ne sont pas des méthodes nouvelles, elles ont déjà été utilisées pour des

problèmes avec des critères comme la somme pondérée des retards (Portmann, 1988) (Chu et al, 1992). L'idée générale du processus itératif que nous proposons est d'orienter l'ensemble de la chaîne logistique vers une stratégie gagnant-gagnant via le partage de certaines informations non critiques pour la stratégie concurrentielle de chacun des partenaires.

L'idée d'associer des fenêtres de temps aux jobs est illustrée par la figure 2.9 sur un exemple simple d'assemblage d'un produit.

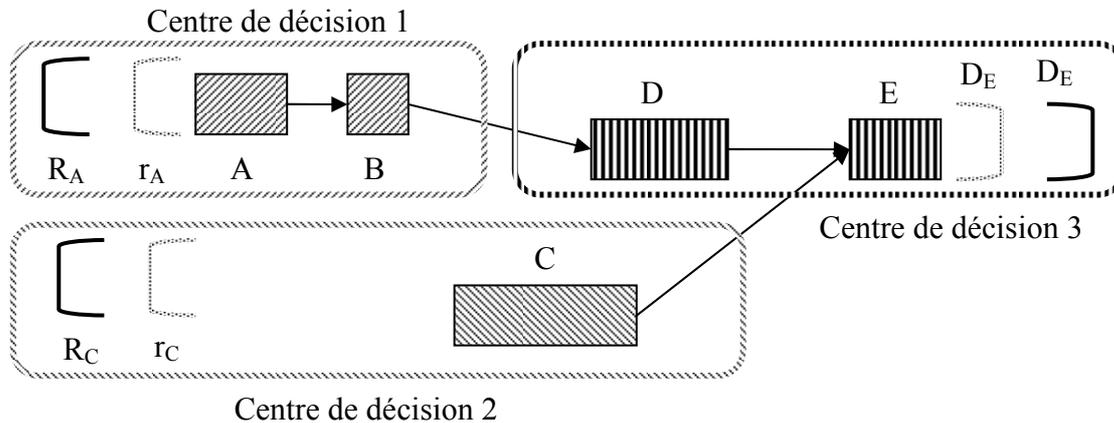


Figure 2.9 dépendances entre les centres de décisions (Portmann et Mouloua, 2007).

La figure 2.9 montre un système de production composé de deux centres de production indépendants et d'un centre d'assemblage. Les liens sont illustrés sur un type de produit particulier, le produit E. Le produit E nécessite deux sous produits (A et C). Le sous produit A est fabriqué par le premier centre de production et nécessite deux opérations. Le sous produit C ne nécessite qu'une seule opération dans le deuxième centre de production. Enfin, l'assemblage du produit fini E nécessite deux opérations dans le centre d'assemblage, l'une qui modifie B et l'autre qui assemble B et C (on est donc dans le cas général de fenêtres de temps sur toutes les opérations). Les dates souhaitées et impératives de livraison du produit E sont données par d_E et D_E . Les dates souhaitées et impératives de disponibilité des sous produits nécessaire à l'assemblage du produit fini sont données par r_A , R_A , r_C et R_C .

Si on était dans un système de chaîne logistique à pilotage centralisé, on n'aurait eu qu'un seul centre de décision qui construirait un seul ordonnancement en juste à temps englobant tous les centres de production et d'assemblage dans le même problème, avec comme objectif la minimisation des coûts de production et la somme totale des pénalités d'avance et de retards, en respectant les contraintes sur les dates d'arrivées et de livraisons avec les partenaires externes. Comme nous nous mettons dans un système à pilotage semi-décentralisé du système global, les centres de décisions restent autonomes et chacun d'entre eux construit seul son ordonnancement en juste à temps et négocie les dates souhaitées et les dates impératives avec les autres partenaires de la chaîne logistique afin d'arriver à la meilleure solution globale de tout le système qui prenne en compte les intérêts de chacun d'eux. La négociation entre les différents centres de décisions est nécessaire car ils sont interdépendants. Le résultat final des ventes de produits finis dépend de *tous* les partenaires.

Pour organiser le processus global de production, les partenaires au sein du système doivent partager des informations relatives à la livraison et la disponibilité de leurs sous produits pour le centre d'assemblage. Il faut donc introduire des fenêtres de temps pour déterminer les dates souhaitées (soft) de livraisons de sous produits par les centre de production du premier étage, ce qui correspond aux dates de disponibilité souhaitées (soft) de

ces sous produits pour le deuxième étage (système d'assemblage). Ces fenêtres de temps serviront à organiser le processus global de négociation. Elles seront appelées « virtuelles » parce qu'elles ne sont pas imposées, et elle varieront au cours du temps durant le processus de négociation pour ajuster les ordonnancements en juste à temps de chaque centre de décision.

Sur l'exemple de la figure 2.9, les dépendances entre le centre d'assemblage et les deux autres centres de production impliquent la communication des dates auxquelles les sous produits sont disponibles pour l'assemblage. L'introduction de fenêtres virtuelles *soft* donne le schéma illustré par la figure 2.10.

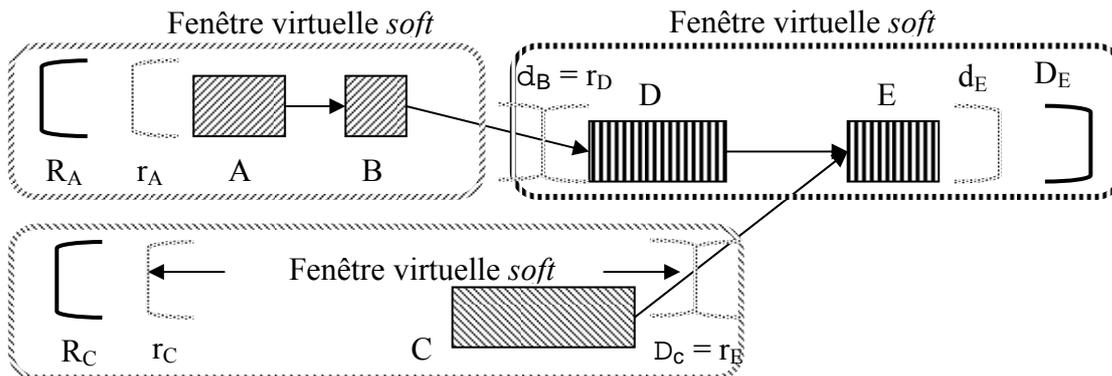


Figure 2.10 association des fenêtres de temps virtuelles aux jobs (Portmann et Mouloua, 2007).

Dans cette figure, la ligne d'assemblage permet le début de la fabrication même s'il y a des sous produits qui ne sont pas encore disponibles. En effet, le traitement du sous produit D peut commencer car il ne nécessite que le sous produit B, on n'a besoin du sous produit C qui est fabriqué par le deuxième centre de production que sur la deuxième machine.

Pour déterminer les fenêtres de temps virtuelles *soft* initiales pour chaque centre de production, nous pouvons utiliser les dates souhaitées de disponibilité et les dates de livraisons pour chaque sous produit du produit final et leurs temps d'exécution à travers une fonction qui, par exemple, va calculer la longueur des fenêtres proportionnellement à la somme des temps d'exécution. Les fenêtres de temps virtuelles *hard* initiales se calculent aussi en suivant certaines règles. Par exemple, au début du processus, elles peuvent être grandes (pour que le système puissent trouver une solution initiale admissible qui sera améliorée dans chaque itération), ceci peut être obtenu en considérant que les ateliers ne traitent que ce produit en effectuant un ordonnancement calé à gauche sur sa date de disponibilité impérative ou bien sur la date de disponibilité impérative de l'opération qui la précède pour les dates de disponibilité. On peut utiliser un processus analogue en remplaçant « calé gauche » par « calé droite » pour les dates de livraisons. Les dates de disponibilité ou de livraisons peuvent être associées à n'importe quelle opération dont les prédécesseurs ou les successeurs appartiennent à un autre centre de décision. Dans ce cas, on peut considérer des contraintes d'écart temporel minimales pour modéliser les temps de transport. Des pénalités virtuelles seront associées aux fenêtres virtuelles de façon uniforme (sur les partenaires) avec des possibilités d'ajustement dans les centres autonomes par exemple pour soulager une machine surchargée.

Après l'attribution des fenêtres virtuelles (*soft* et *hard*) initiales, les ordonnancements des différents centres de décisions deviennent indépendants, car les dépendances entre eux ont été prises en compte à travers les fenêtres de temps. Chaque centre de production peut utiliser une méthode de résolution exacte ou approchée pour calculer l'ordonnancement en juste à temps

pour minimiser la somme des pénalités d'avances et de retards. Nous venons ainsi de terminer la première itération du processus global de négociation. Après chaque étape (itération) du processus itératif, une analyse des solutions calculées par les différents centres de production s'avère indispensable pour prendre des décisions pour orienter le système global vers la cohérence et vers des solutions de meilleure qualité.

La convergence du processus global out être garantie par la manière avec laquelle ces négociations sont organisées, et par le fait qu'il existe une possibilité de relaxation des contraintes impératives dans le cas où le système initial n'admet pas de solutions faisables.

2.5 Conclusion

Nous avons décrit dans ce chapitre le problème principal étudié dans cette thèse. Il s'agit du problème de coordination au niveau des ordonnancements en juste à temps entre différents partenaires de la chaîne logistique pour satisfaire des commandes de clients. Nous avons décrit le fonctionnement global du système, ainsi que les dépendances entre les différents centres de décision. Les centres de décisions étant autonomes, nous avons proposé un processus global de négociations pour coordonner les décisions afin de satisfaire les clients externes à la chaîne. C'est une méthode itérative d'amélioration des solutions globales basée sur la notion de fenêtres de temps souhaitées et fenêtres de temps impératives vis à vis aussi bien des clients externes, que des fournisseurs externes. Cette négociation ne requiert pas le partage d'informations « sensibles » entre les centres de décisions. En effet, ces derniers ne doivent se partager que des informations générales, mais surtout communiquer des fenêtres de temps virtuelles. La convergence du processus global de négociation devra être assurée par un pilotage semi-décentralisé qui oriente l'ensemble des partenaires de la chaîne vers une solution globale qui est supposée satisfaire les intérêts de chacun d'entre eux : une stratégie gagnant-gagnant.

Dans les chapitres 3 et 4 nous développons des méthodes approchées pour la mise en œuvre des ordonnancements en juste à temps pour les problèmes décrits par le processus global de négociation. La validation de l'approche de négociation ne peut être assurée que par des expérimentations à travers la simulation du comportement global du système. Dans cette thèse nous nous intéressons essentiellement au développement d'outils d'ordonnement pour la coordination des décisions opérationnelles pour un pilotage semi-décentralisé de la chaîne logistique. De ce fait, nous n'avons pas mené les simulations concernant le système global de négociation. Nous proposons les outils nécessaires à sa mise en œuvre.

3 Chapitre 3 : Méthode de résolution

Résumé

Nous présentons dans ce chapitre la méthode de résolution que nous développons pour les problèmes d'ordonnancement en juste à temps générés par la problématique détaillé dans le précédent chapitre. Nous présentons en premier lieu l'architecture globale de la solution, nous développons ensuite la technique Pert utilisée, ainsi que les métaheuristiques

3.1 Introduction

Nous avons vu dans le chapitre précédent le schéma global du processus de négociations entre, d'une part, les partenaires au sein de la chaîne logistique pour coordonner leurs ordonnancements en juste à temps, et d'une autre part entre les entités de la chaîne logistique et les fournisseurs et clients externes. Notre approche par décomposition temporelle et spatiale du problème nous permet de considérer chaque centre de décision autonome séparé des autres centres. Nous proposons dans ce chapitre des outils pour mettre en œuvre l'ordonnancement en juste à temps local à chaque centre de décision. Nous procédons là encore par une approche par décomposition. Dans la quatrième section de ce chapitre, nous présentons des méthodes approchées à base de méta-heuristiques pour l'exploration de l'espace des solutions, ce qui constitue la deuxième étape de l'approche par décomposition. La première étape étant l'évaluation de toute solution. Une solution est représentée par un chromosome (une permutation, une séquence) contenant l'ordre de passages des opérations (travaux pour le flowshop de permutation) sur les machines.

Comme le problème que nous considérons est NP-complet au sens fort, une méthode de résolution exacte est à écarter car très consommatrice en temps d'exécution, ce qui est une contrainte lourde dans un milieu industriel où on cherche à toujours raccourcir les temps de prise de décisions. De ce fait, nous développons deux méthodes à bases de métaheuristiques : les algorithmes génétiques, et le recuit simulé. Nous développons avec plus de détails la solution basée sur les algorithmes génétiques, le recuit simulé utilisé étant plus standard.

3.2 Approche de résolution par décomposition et méta-heuristique

Comme expliqué au chapitre 2, le processus de négociation considéré nous permet de séparer les centres de décisions, et ainsi nous pouvons calculer les ordonnancements de chacun d'eux indépendamment de ceux des autres. Nous considérons donc les données relatives à un centre de production (ou de décision) donné, et nous cherchons les permutations de travaux qui minimisent le coût total de l'ordonnancement. L'espace de recherche de solutions est très large et dépend du nombre de travaux à ordonnancer. Nous devons donc trouver une façon intelligente de l'explorer, une façon qui privilégie l'exploration de solutions intéressantes, au lieu de « perdre du temps » à évaluer de mauvaises solutions.

Le schéma illustré par la figure 3.1 est exécuté en boucle jusqu'à un certain critère d'arrêt qui peut être par exemple une durée limite d'exécution, ou l'évaluation d'un certain nombre de permutations faisables.

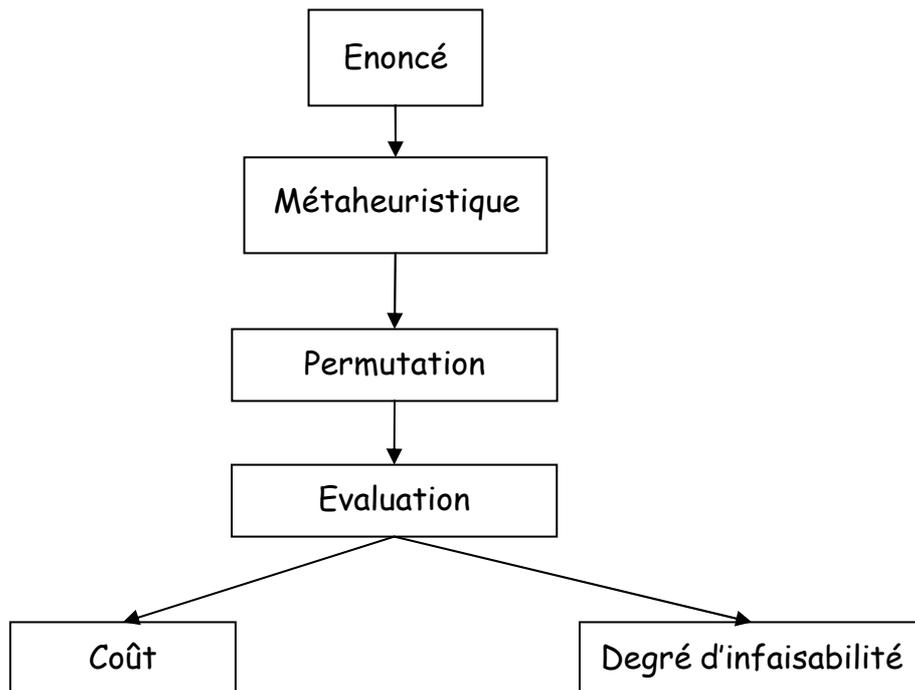


Figure 3.1 Schéma général de la méthode de résolution.

L'évaluation d'une permutation est illustrée par l'algorithme de la figure 3.2. Cet algorithme commence d'abord par déterminer si la permutation en entrée est faisable (i.e. respectant les contraintes impératives du problème). Si la permutation est faisable, alors nous l'évaluons avec l'algorithme du Pert coût, sinon nous définissons un degré d'infaisabilité.

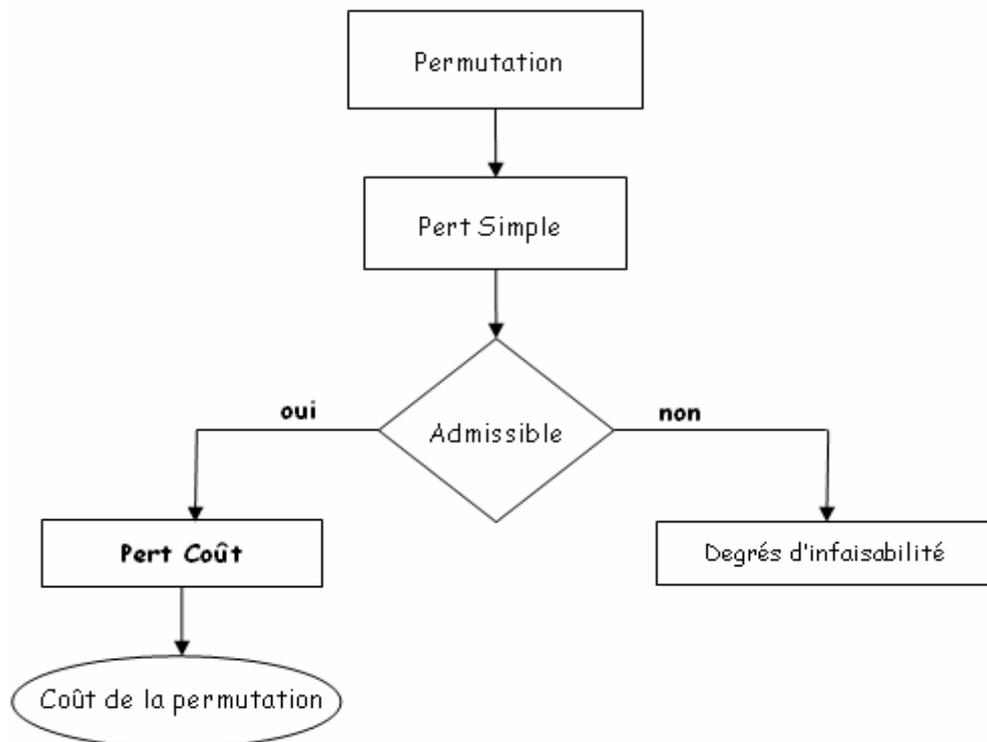


Figure 3.2. Schéma général de l'évaluation d'une permutation.

Pour le test d'admissibilité, nous appliquons l'algorithme du Pert simple en ne considérant que les contraintes impératives. Le test d'admissibilité est polynomial et se fait en calculant l'ordonnancement calé droite (au plus tard par rapport à la date maximale de fin du projet) et l'ordonnancement calé gauche (au plus tôt).

Le degré d'infaisabilité est égal au nombre de fois où la date au plus tôt d'une opération est strictement plus grande que sa date au plus tard.

En amont du processus illustré par la figure précédente, on retrouve le mécanisme de génération de permutations, c'est à dire le mécanisme chargé de l'exploration de l'espace des solutions en vue de la recherche de solutions intéressantes. Pour ce faire nous utilisons les métaheuristiques que nous décrivons dans la quatrième section du présent chapitre.

3.3 Résolution du JIT-Seq

3.3.1 Ordonnement de projet

L'ordonnement de projets consiste à programmer (ou organiser) l'exécution des différentes opérations le constituant et d'affecter des ressources à ces tâches en tenant compte des différentes contraintes entre les tâches et sur les ressources, afin d'optimiser certains critères comme la durée totale du projet, les performances techniques, et le coût du projet. Le problème dans l'ordonnement de projet est de déterminer les dates d'exécution de chaque tâche du projet de telle sorte que tout le projet soit terminé dans les délais.

Plusieurs techniques sont utilisées pour manager les projets dont les plus répandues sont le Pert, le CPM (Critical Path Method), et la méthode des potentiels. Ces méthodes sont apparues pour la plupart à la fin des années cinquante et ont connu un succès grâce au raccourcissement spectaculaire de certains grands projets dans lesquels elles ont été utilisées (projet du Queen Mary 2, et la construction du paquebot France pour la méthode des potentiels).

La méthode du Pert (Technique d'évaluation et d'examen de programmes (projets)) est l'une des méthodes les plus utilisées par les industriels pour l'ordonnement de projets. Comme bon nombre de technologies de l'ère moderne, elle a trouvé ses origines dans la recherche militaire. En effet, elle a été mise au point lors du projet POLARIS qui coordonnait les activités de milliers d'organisations.

C'est une méthode d'ordonnement ou une technique de contrôle qui permet l'organisation des activités d'un projet dans le but d'optimiser certains critères, généralement la minimisation de la durée totale de réalisation du projet. Le résultat de l'application de cette technique est l'obtention d'un calendrier d'exécution des activités du projets lorsque l'affectation des ressources est donnée et n'est pas restrictive (ressources considérées comme illimitées) en prenant en compte les contraintes entre activités (contraintes de précédence). L'algorithme correspondant est un algorithme polynomial issu de la programmation dynamique.

La différence majeure entre la méthode Pert et la méthode des potentiels réside dans la manière de concevoir le graphe associé sur lequel on calcule les plus longs chemins en utilisant la programmation dynamique. Dans le Pert, les opérations sont associées aux arcs (les sommets correspondent à des événements) alors que dans la méthode des potentiels, les opérations sont associées aux sommets (les arcs représentent les délais d'attente entre les débuts des opérations). Dans le Pert Coût, on utilise un graphe étendu où les arcs peuvent correspondre aussi bien aux opérations (dont on peut compresser la durée en payant) qu'aux délais entre les opérations qui se suivent. Nous utilisons plutôt un graphe potentiel tâche pour

le calcul de la faisabilité, mais nous avons conservé le terme générique de Pert comme sigle simplifié désignant l'algorithme.

Pour plus de détails sur cette méthode, nous renvoyons le lecteur à l'annexe 1 où on donne l'algorithme et on l'illustre sur un exemple détaillé.

3.3.2 Problème original du Pert coût

La méthode du Pert Coût est une extension de la méthode du Pert. Dans le Pert classique l'objectif considéré est la minimisation de la durée totale du projet. Dans le Pert coût on ajoute un autre critère : minimiser le coût total de l'ordonnancement. A chaque tâche est associé un coût relatif aux moyens qu'elle requiert, à la date de son exécution (par exemple, le coût varie selon la date de son exécution) ou par les compétences qu'elle requiert. Dans ce problème on suppose que les durées de certaines tâches peuvent être diminuées en contre partie d'un financement supplémentaire. Le Pert Coût est une méthode itérative qui utilise alternativement la méthode du Pert simple et la recherche d'une coupe à coût minimal dans le sou graphe critique obtenu par application du Pert. Chaque itération de la méthode est polynomiale par rapport au nombre de tâches à ordonnancer et au nombre de contraintes de précédence, mais le nombre d'itération est majoré par le nombre d'unités de temps que l'on cherche à gagner qui est lié aux valeurs numériques de chaque énoncé considéré. La démonstration de l'optimalité de la méthode est faite en utilisant le théorème de la dualité de la programmation linéaire. C'est un problème de programmation linéaire continue ou mixte selon que le temps gagné est défini comme un entier ou comme un réel (Portmann, ISDP32).

Plusieurs études ont été menées sur des problématiques liées au Pert. Castro et al (Castro et al, 2007) étudient le problème du partage des coûts des retards dans des réseaux Pert. Ils définissent la règle du partage du coût en série pondérée pour le problème de l'allocation des coûts. Cette règle est implémentée en un algorithme de complexité polynomiale. Azaron et Tavakkoli-Moghaddam (Azaron et Tavakkoli-Moghaddam, 2006) développent un modèle multi objectifs pour le problème de l'allocation des ressources dans un réseau de type Pert dynamique où les durées des activités sont des variables aléatoires avec une distribution exponentielle, et où les arrivées de projets sont générées selon la loi de poisson. Les objectifs considérés sont le coût direct du projet, la date de fin de projet moyenne, et la variance des dates de fin des projets. Le modèle proposé est transformé en un programme non linéaire à résoudre. Pour une variante du problème précédent, et en utilisant le même genre de modèle multi objectifs, Azaron et al (Azaron et al, 2005) proposent un algorithme génétique.

Notre problème d'ordonnancement en juste à temps avec des coûts d'avances et/ou retards par rapports aux dates souhaitées de disponibilité de composants, et des dates souhaitées de livraisons des produits finis, est équivalent à un problème de Pert coût car on a tous les éléments d'un tel problème : un ensemble de tâches à ordonnancer, un ensemble de contraintes de précédences entre tâches, un délai maximum pour la réalisation du projet, et des coûts correspondant à la diminution de la durée de certaines tâches additionnelles correspondant au non respect des délais souhaités (avance ou retard par rapport au souhait).

Nous utilisons cette méthode pour résoudre notre problème d'ordonnancement en juste à temps au sein des centres de production après avoir montré comment nos problèmes à permutations fixées se ramènent à des Pert Coût.

3.3.3 Algorithme du JIT adapté du Pert coût

Nous détaillons dans ce qui suit l'utilisation que nous faisons du Pert Coût pour évaluer une permutation du problème de juste à temps.

Après le test d'admissibilité (Pert simple), et si la permutation est faisable, alors nous appliquons un algorithme du Pert coût pour évaluer la permutation (somme des pénalités d'avances et de retards).

Comme dans tout problème Pert, le graphe comporte une source S et un puits T . Des arcs entre la source et des points en amont de chaque opération ayant une position idéale (la première et la dernière opération de chaque travail dans le cas simplifié) permettent de représenter les avances par rapport à la position idéale (quand l'arc est compressé). Des arcs entre des points en aval des opérations ayant des positions idéales (la première et la dernière opération de chaque travail dans le cas simplifié) permettent de représenter les retards par rapport à la position idéale (quand l'arc est compressé).

Pour assurer le placement exact des opérations, en avance ou en retard par rapport à leur position idéale, un chemin issu de S et passant par un arc direct amont de l'opération, se poursuit par un arc aval associé qui va à T et a une longueur exactement égale à la durée de l'horizon H . De manière symétrique, le chemin qui passe par un arc aval d'une opération, se poursuit en amont de l'opération par un arc associé issu de S et a une longueur exactement égale à la durée de l'horizon H . Quand on considère un couple d'arcs associés, si un des deux est compressé, alors l'autre est dilaté.

Nous agissons donc par compressions ou dilatations sur les arcs sortants de la source S et les arcs arrivant au puits T . Pour le cas simplifié, les avances et retards par rapport au dates souhaitées de disponibilité sont données par le vecteur des X_i (pour la première opération du travail i), alors que les avances et retards par rapports aux dates souhaitées de livraison sont données par le vecteur des Y_i (pour la dernière opération du travail i). Si X_i est nul alors la première opération du travail i est idéalement placée. De la même manière, Si Y_i est nul alors la dernière opération du travail i est idéalement placée. Si $X_i > 0$ (resp $Y_i > 0$) alors la première (resp. dernière) opération du travail i est en avance, et si $X_i < 0$ (resp. $Y_i < 0$) alors l'opération est en retard. Quand X_i est positif alors nous compressons l'arc correspondant sortant de la source et cela induit un coût $\alpha_{r,i} \cdot X_i$, et nous dilatons l'arc dual correspondant arrivant au puit et cela induit un coût $-\beta_{r,i} \cdot X_i$. Quand Y_i est positif alors nous compressons l'arc correspondant sortant de la source et cela induit un coût $\alpha_{d,i} \cdot Y_i$, et nous dilatons l'arc dual correspondant arrivant au puit et cela induit un coût $-\beta_{d,i} \cdot Y_i$.

Le schéma général du processus de l'évaluation par la méthode du Pert Coût est donné par l'algorithme 3.3.

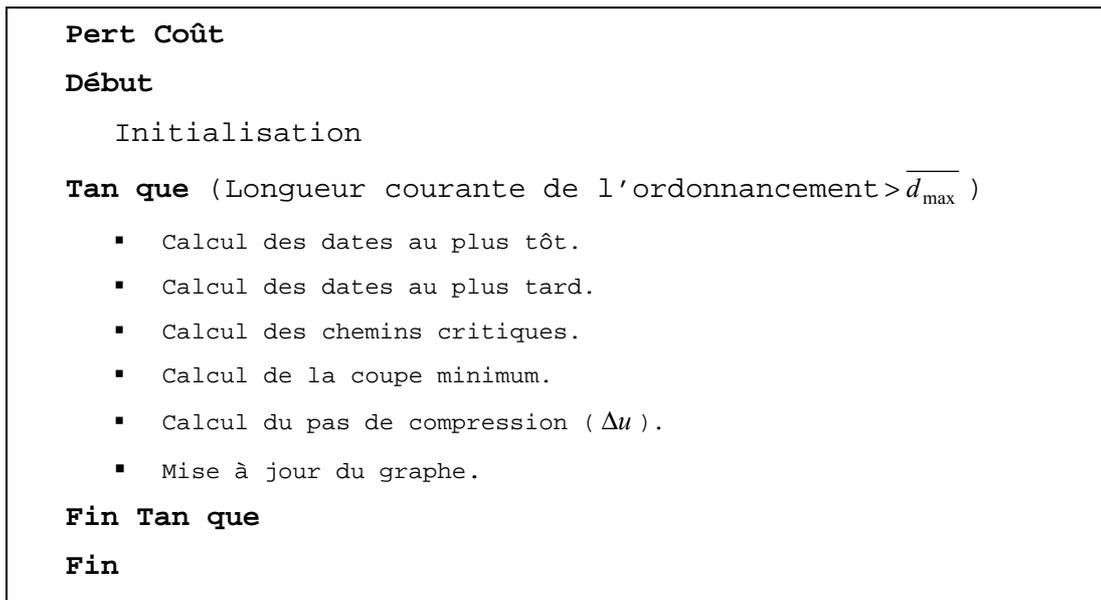


Figure 3.3 Algorithme d'évaluation d'une permutation.

L'algorithme commence par initialiser les valeurs sur les arcs, et des différentes variables. Ensuite, tant que la longueur totale de l'ordonnancement est supérieure à la date de livraison impérative maximum (égale à la longueur de l'horizon H) alors on exécute les opérations qui constituent une itération. Une itération commence par le calcul des dates au plus tôt, puis les dates au plus tard. On calcule ensuite les chemins critiques, et la coupe minimale. Le calcul de la coupe minimale détermine les arcs qui doivent être compressés, et les arcs associés qui doivent être dilatés, ainsi que les arcs qui ne subissent pas de changements. Le pas de compression/dilatation dans une itération est déterminé par Δu (dont l'évaluation maximale polynomiale est détaillée plus loin). A la fin de chaque itération, on lance la procédure de mise à jour des valeurs sur les arcs en tenant en compte les compressions/dilatations effectuées.

Nous supposons que nous avons une permutation de jobs fournie par la méta-heuristique. Nous construisons alors le graphe Pert correspondant de la manière suivante :

- 1) Le graphe a une source S de laquelle sortent des arcs vers toutes les opérations (ou bien juste vers les opérations de chaque job sur la première machine pour le cas spécifique).
- 2) Le graphe a un puit T auquel arrivent des arcs depuis toutes les opérations (ou bien juste les opérations des jobs sur la dernière machine pour le cas spécifique).
- 3) Hors de la source et du puits on retrouve les contraintes classiques de précédences entre les opérations.

Chaque nœud du graphe représente une opération identifiée par le numéro du job et le numéro de la machine. Si on considère que l'on autorise qu'il y ait des contraintes temporelles sur les dates de disponibilité et/ou de livraison sur les opérations intermédiaires, alors chaque opération du graphe est entourée des nœuds $BS_{i,j}$ et $ES_{i,j}$ comme illustré sur la figure 3.4 suivante.

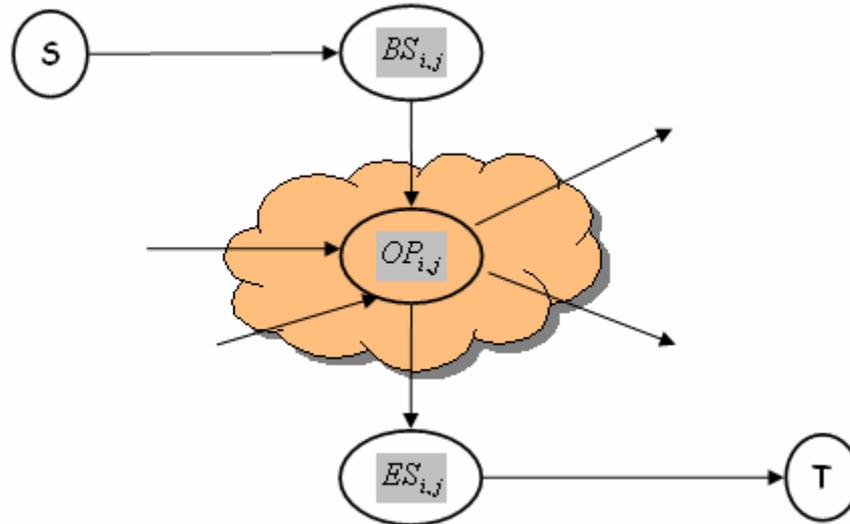


Figure. 3.4. Modèle du graphe pour chaque opération.

Le nœud $BS_{i,j}$ représente l'événement de début d'exécution de l'opération $OP_{i,j}$, tandis que le nœud $ES_{i,j}$ représente l'événement de fin de l'exécution.

Dans le cadre des expériences de cette thèse nous considérons des structures d'atelier en flowshop, plus spécifiquement les flowshops de permutation pour lesquels l'ordre de passages des opérations est le même pour toutes les machines. Dans ce cas spécifique, l'avance ou le retard d'un job par rapport aux dates de disponibilité est déterminé par la première opération du job, alors que son retard ou avance par rapport aux dates de livraison est déterminé par la dernière opération de chaque job. Les nœuds BS_i et ES_i entourent donc la première opération de chaque job i . Par souci de clarté (on supprime l'indice j) nous introduisons les nœuds BC_i et EC_i qui entourent la dernière opération de chaque job $i = \overline{1, \dots, N}$. Les arcs arrivants aux nœuds BS_i sont originaires de la source S , et les arcs sortants des nœuds ES_i arrivent au puits T . De même, les arcs arrivants aux nœuds BC_i sont originaires de la source S , et les arcs sortants des nœuds EC_i arrivent au puits T .

Les valeurs sur les arcs

- 1) Les arcs sortants d'une opération $OP_{i,j}$ ont la valeur $P_{i,j}$.

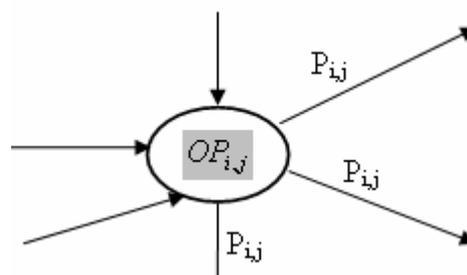


Figure. 3.5. Les arcs sortants d'une opération.

- 2) Les arcs partants des BS_i et arrivants aux opérations $OP_{i,1}$ ont une valeur nulle. Les arcs partants des $OP_{i,1}$ aux nœuds ES_i ont une valeur nulle.

- 3) La valeur sur les arcs sortant de S , arrivant aux BS_i , est donnée par la formule suivante : $a_{i,1}^* - X_i$ tel que $a_{i,1}^*$ est la date de début idéale de la première opération du job i , et X_i est la déviation, en terme de temps, entre la date d'exécution réelle de l'opération et sa date idéale.
- 4) La valeur sur les arcs sortants des ES_i , arrivants au puit T , est donnée par la formule suivante : $\overline{d_{\max}} - a_{i,1}^* - P_{i,1} + X_i$. Ainsi, si S a le potentiel 0, T le potentiel $\overline{d_{\max}}$, et $X_i = 0$ alors l'opération $(i,1)$ est placée à sa date idéale.

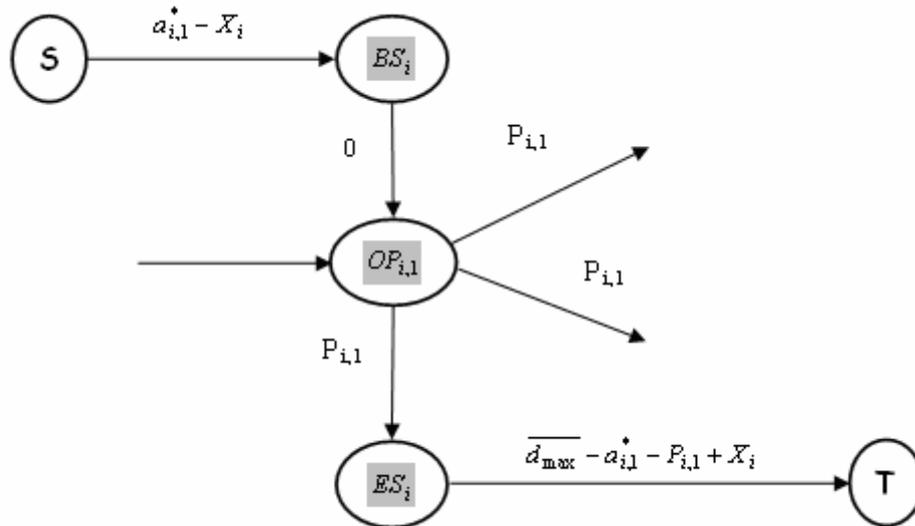


Figure 3.6. Valeurs sur les arcs.

- 5) Les arcs partants des BC_i et arrivant aux opérations $OP_{i,M}$ ont une valeur nulle. Les arcs partants des $OP_{i,M}$ aux noeuds EC_i ont une valeur égale à $P_{i,M}$.
- 6) La valeur sur les arcs sortants de S , arrivant aux BC_i , est donnée par la formule suivante : $a_{i,M}^* - Y_i$ tel que $a_{i,M}^*$ est la date de début idéale de la dernière opération du job i , et Y_i est la déviation, en terme de temps, entre la date d'exécution réelle de l'opération et sa date idéale.
- 7) La valeur sur les arcs sortants des EC_i , arrivants obligatoirement au puit T , est donnée par la formule suivante : $\overline{d_{\max}} - a_{i,1}^* + Y_i - P_{i,M}$. Ainsi, si S a le potentiel 0, T le potentiel $\overline{d_{\max}}$, et $Y_i = 0$ alors l'opération (i,M) est placée à la date idéale.
- 8)

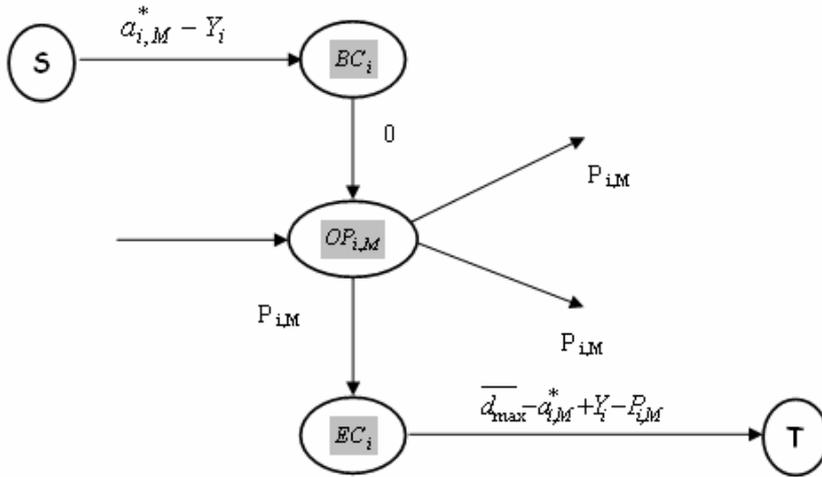


Figure 3.7. Arcs sortants.

Il est important de remarquer que la longueur du chemin allant de la source S au puits T et passant par les noeuds BS_i , $OP_{i,1}$ et ES_i est égale à $\overline{d_{max}}$ (la longueur de l'horizon). Il en est de même pour les chemins allant de la source S au puit T et passant par les noeuds BC_i , $OP_{i,M}$ et EC_i .

Calcul du pas de compression/dilatation Δu

Les arcs appartenant à la coupe minimale sont compressés avec le pas de compression Δu . Cela revient à soustraire Δu de la valeur sur l'arc compressé. Les arcs duaux des arcs compressés sont dilatés. C'est à dire qu'on ajoute Δu à la valeur sur l'arc dilaté. Le reste des arcs restent inchangés. Pour calculer le pas de compression maximale possible sans changer de coupe, nous définissons les six graphes suivants :

- i. Dans le graphe G^1 , on ne garde que les arcs partant de la source et qui n'appartiennent ni à la coupe minimale, ni aux arcs duaux de ceux appartenant à la coupe minimale (arcs neutres).

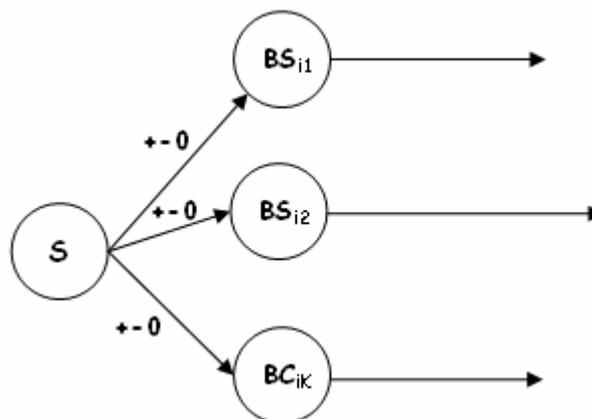


Figure 3.8. Graphe G^1 .

- ii. Dans le graphe G^2 , on ne garde que les arcs sortant de la source et qui appartiennent à la coupe minimale (arcs compressés).

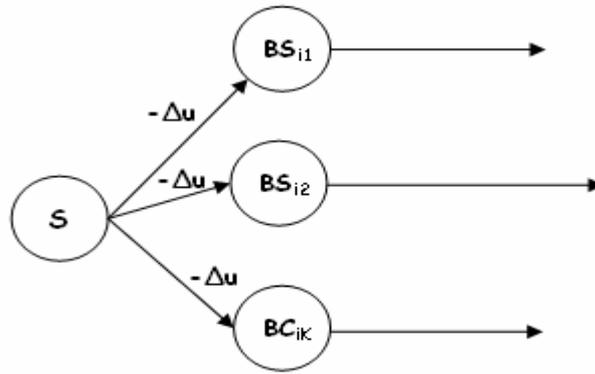


Figure 3.9. Graphe G^2

- iii. Dans le graphe G^3 , on ne garde que les arcs sortant de la source et qui sont des arcs duaux à ceux appartenant à la coupe minimale (arcs dilatés).

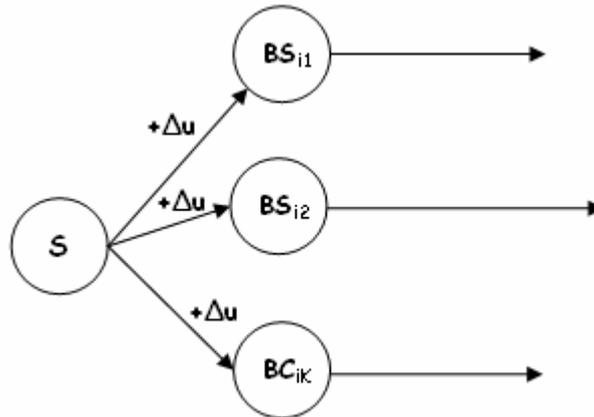


Figure. 3.10. Graphe G^3

Dans chacun des trois graphes précédents, le puits T n'en fait pas partie. Pour chaque graphe, on calcule pour chaque nœud le plus long chemin depuis la source jusqu'à ce nœud. Cette valeur est notée par λ_k^0 pour le premier graphe, λ_k^1 pour le deuxième, et λ_k^2 pour le troisième.

- iv. Dans le graphe G^4 , on ne garde que les arcs arrivant au puits T et qui n'appartiennent ni à la coupe minimale, ni aux arcs duaux de ceux appartenant à la coupe minimale (arcs neutres).

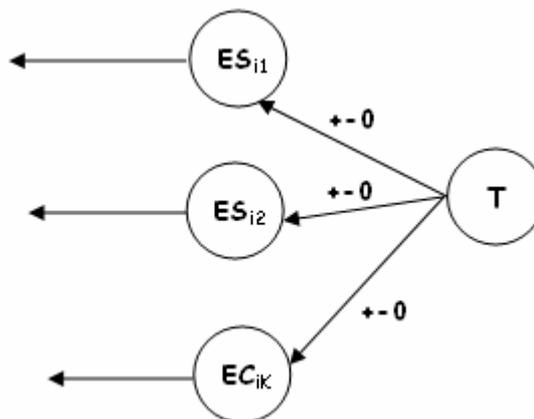


Figure. 3.11. Graphe G^4

- v. Dans le graphe G^5 , on ne garde que les arcs arrivant au puits et qui sont des arcs duaux à ceux appartenant à la coupe minimale (arcs compressés).

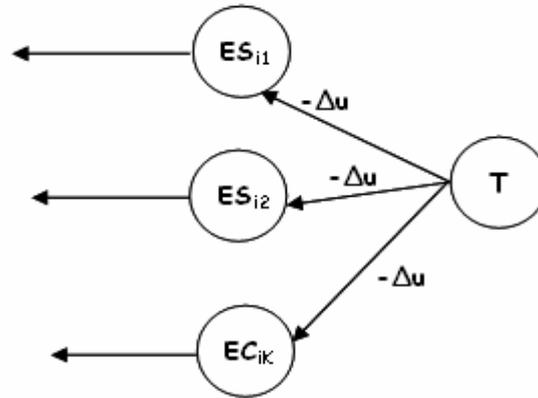


Figure 3.12. Graphe G^5

- vi. Dans le graphe G^6 , on ne garde que les arcs arrivant au puits et qui sont des arcs duaux à ceux appartenant à la coupe minimale (arcs dilatés).

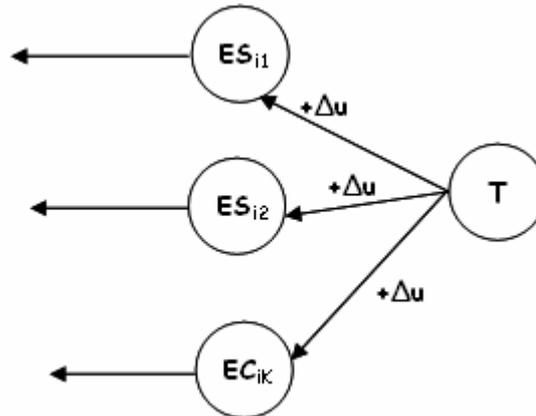


Figure 3.13. Graphe G^6 .

Dans chacun des trois graphes précédents, la source S n'en fait pas partie. Comme pour les trois premiers graphes, on calcule pour chaque nœud le plus long chemin depuis le puits jusqu'à ce nœud (dans chaque graphe). Cette valeur est notée par μ_k^0 pour le premier graphe, μ_k^1 pour le deuxième, et μ_k^2 pour le troisième.

On aura calculé six valeurs pour chaque nœud du graphe initial. Pour déterminer le plus long chemin de la source vers un nœud donné k , et le plus long chemin du puits vers ce même nœud, on définit les deux équations suivantes :

$$\begin{cases} \lambda_k(\Delta u) = \text{Max}(\lambda_k^0, \lambda_k^1 - \Delta u_k, \lambda_k^2 + \Delta u_k) \\ \mu_k(\Delta u) = \text{Max}(\mu_k^0, \mu_k^1 - \Delta u_k, \mu_k^2 + \Delta u_k) \end{cases} \quad (3.1)$$

Pour chaque nœud du graphe initial, l'équation suivante doit être vérifiée :

$$\lambda_k(\Delta u) + \mu_k(\Delta u) + \Delta u_k \leq \lambda_T \quad (3.2)$$

L'équation (3.2) exprime le fait, que pour chaque nœud du graphe initial, la somme des plus long chemins à partir de la source et du puits jusqu'à ce nœud, ajoutée à une certaine marge Δu_k est toujours inférieure ou égale à la longueur maximale de l'ordonnancement λ_T .

Pour calculer la marge Δu_k la plus grande possible, on examine toutes les combinaisons possibles de remplacement de l'équation (3.1) dans l'équation (3.2). Nous avons donc neuf cas à examiner :

1. Cas 1 : $\lambda_k(\Delta u) = \lambda_k^0$ et $\mu_k(\Delta u) = \mu_k^0$.
 $\lambda_k^0 \leq \lambda_T - \Delta u_k - \mu_k^0 \Rightarrow \Delta u_k \leq \lambda_T - \lambda_k^0 - \mu_k^0$. (3.3)

2. Cas 2 : $\lambda_k(\Delta u) = \lambda_k^0$ et $\mu_k(\Delta u) = \mu_k^1 - \Delta u_k$.
 $\lambda_k^0 \leq \lambda_T - \Delta u_k - \mu_k^1 + \Delta u_k \Rightarrow \lambda_k^0 + \mu_k^1 \leq \lambda_T$. Ce cas n'engendre pas de contraintes pour le calcul de la marge.

3. Cas 3 : $\lambda_k(\Delta u) = \lambda_k^0$ et $\mu_k(\Delta u) = \mu_k^2 + \Delta u_k$.
 $\lambda_k^0 \leq \lambda_T - \Delta u_k - \mu_k^2 - \Delta u_k \Rightarrow \Delta u_k \leq (\lambda_T - \lambda_k^0 - \mu_k^2) / 2$. (3.4)

4. Cas 4 : $\lambda_k(\Delta u) = \lambda_k^1 - \Delta u_k$ et $\mu_k(\Delta u) = \mu_k^0$.
 $\lambda_k^1 - \Delta u_k \leq \lambda_T - \Delta u_k - \mu_k^0 \Rightarrow \lambda_k^1 \leq \lambda_T - \mu_k^0$. Ce cas n'engendre pas de contraintes pour le calcul de la marge.

5. Cas 5 : $\lambda_k(\Delta u) = \lambda_k^1 - \Delta u_k$ et $\mu_k(\Delta u) = \mu_k^1 - \Delta u_k$.
 $\lambda_k^1 - \Delta u_k \leq \lambda_T - \Delta u_k - \mu_k^1 + \Delta u_k \Rightarrow \Delta u_k \geq \lambda_k^1 + \mu_k^1 - \lambda_T \leq 0$. Ce cas n'engendre pas de contraintes pour le calcul de la marge.

6. Cas 6 : $\lambda_k(\Delta u) = \lambda_k^1 - \Delta u_k$ et $\mu_k(\Delta u) = \mu_k^2 + \Delta u_k$.
 $\lambda_k^1 - \Delta u_k \leq \lambda_T - \Delta u_k - \mu_k^2 - \Delta u_k \Rightarrow \Delta u_k \leq \lambda_T - \lambda_k^1 - \mu_k^2$. (3.5)

7. Cas 7 : $\lambda_k(\Delta u) = \lambda_k^2 + \Delta u_k$ et $\mu_k(\Delta u) = \mu_k^0$.
 $\lambda_k^2 + \Delta u_k \leq \lambda_T - \Delta u_k - \mu_k^0 \Rightarrow \Delta u_k \leq (\lambda_T - \lambda_k^2 - \mu_k^0) / 2$. (3.6)

8. Cas 8 : $\lambda_k(\Delta u) = \lambda_k^2 + \Delta u_k$ et $\mu_k(\Delta u) = \mu_k^1 - \Delta u_k$.
 $\lambda_k^2 + \Delta u_k \leq \lambda_T - \Delta u_k - \mu_k^1 + \Delta u_k \Rightarrow \Delta u_k \leq \lambda_T - \lambda_k^2 + \mu_k^1$. (3.7)

9. Cas 9 : $\lambda_k(\Delta u) = \lambda_k^2 + \Delta u_k$ et $\mu_k(\Delta u) = \mu_k^2 + \Delta u_k$.
 $\lambda_k^2 + \Delta u_k \leq \lambda_T - \Delta u_k - \mu_k^2 - \Delta u_k \Rightarrow \Delta u_k \leq (\lambda_T - \lambda_k^2 - \mu_k^2) / 3$. (3.8)

La marge de chaque nœud Δu_k est égale au minimum des valeurs calculées par les formules (3.3), (3.4), (3.5), (3.6), (3.7), (3.8). Elle est donnée par la formule suivante :

$$\Delta u_k = \text{Min}((\lambda_T - \lambda_k^0 - \mu_k^2) / 2, \lambda_T - \lambda_k^1 - \mu_k^2, (\lambda_T - \lambda_k^2 - \mu_k^0) / 2, \lambda_T - \lambda_k^2 + \mu_k^1, (\lambda_T - \lambda_k^2 - \mu_k^2) / 3) \quad (3.9)$$

ainsi que par :

$$\Delta V_C = \text{Maximum de compression encore autorisé sur un arc comprimé de la coupe minimale } C. \quad (3.10)$$

Le pas de compression maximum est égal à la marge minimum sur l'ensemble du graphe. Il est donné par : $\Delta u = \text{Min}(\text{Min}_k(\Delta u_k), \text{Min}_C(\Delta V_C))$ (3.11).

Mises à jour

A la fin de chaque itération, on lance la procédure de mise à jours des valeurs sur les arcs en fonction de la coupe minimale et du pas de compression calculés. Si un arc est comprimé alors on soustrait Δu de la valeur sur cet arc (dans la limite de la durée restante sur l'arc donnée par ΔV_C , on ne peut pas aller dans des valeurs négatives, dans ce cas l'arc devient non

compressible. Son coût est infini, et ne pourra plus appartenir à la coupe minimale), et un coût. Si un arc est dilaté alors on ajoute Δu à la valeur sur cet arc. Les arcs neutres gardent la même valeur. On met à jour aussi les valeurs des X_i et Y_i (ajout ou soustraction du Δu selon que l'opération correspondant a été avancée ou retardée, ne rien faire sinon).

La valeur de la solution est donnée par la somme sur tous les travaux, des avances ou retards de toutes les opérations multipliées par la pénalité correspondante.

3.3.4 Bilan de la complexité des évaluations

Chacune des procédures à l'intérieur de l'algorithme de Pert coût décrit par la figure 3.3 est polynomiale. Il nous reste donc à évaluer le nombre d'itérations nécessaires à l'exécution du Pert coût.

Nous conjecturons que dans le cas le plus simple JIT/job($\Sigma E1-T1-Em-Tm$) qui correspond au cas où l'on considère des pénalités d'avance et de retard pour la première et la dernière opération de chaque travail, le nombre d'itérations est limité à $16n^2$. Ceci est vrai si tout arc qui commence à compresser ne se dilate pas ultérieurement. C'est intuitivement vrai, vérifié sur toutes les expériences faites, mais nous ne l'avons pas prouvé théoriquement. Cela reste donc une conjecture.

3.4 Méta-heuristiques

3.4.1 Introduction : les métaheuristiques

Nous avons vu dans le chapitre précédent l'algorithme d'évaluation d'une solution (permutation) du problème général. Cet algorithme ne peut être appliqué que si une permutation est donnée en entrée. L'espace des solutions pour le problème considéré est si vaste que l'on ne peut pas faire une recherche exhaustive de toutes les solutions existantes, les évaluer, et en prendre la meilleure. Si l'on considère qu'on a un nombre de travaux à ordonnancer égal à 10, alors on a $10!$ solutions à explorer. On aurait pu essayer de construire une procédure par séparation et évaluation (méthode exacte) pour ce problème général (NP-difficile) mais ce serait irréaliste en milieu industriel, car elle nécessite un temps d'exécution phénoménal. Pour parcourir cet espace de solutions dans des délais raisonnables, et en essayant d'avoir de « bonnes » solutions, on a généralement recouru à des méthodes approchées.

Par opposition aux méthodes exactes (programmation linéaire, programmation dynamique, approche par séparation et évaluation...), les méthodes approchées ne garantissent pas de trouver la solution optimale à un problème donné, et tendent à chercher des solutions qui s'en approchent sans avoir (généralement) une idée sur la distance entre la meilleure solution trouvée et la solution optimale.

Les méthodes approchées suffisamment générales pour être appliquées à de nombreux problèmes d'optimisation sont appelées *métaheuristiques*. Le mot métaheuristique est dérivé de la composition de deux mots grecs : *heuristique* qui signifie « trouver », et *meta* qui est un suffixe voulant dire « dans un niveau supérieur ».

Osman et Laporte (Osman et Laporte, 1996) donnent la définition suivante d'une métaheuristique : « Une métaheuristique est formellement définie comme un processus de génération itérative guidant une heuristique subordonnée par la combinaison de différents concepts intelligents pour l'exploration et l'exploitation de l'espace de recherche, les stratégies d'apprentissage sont utilisées pour structurer l'information dans le but de trouver avec efficacité des solutions proches de l'optimalité ».

Une autre définition est donnée par (Voss et al, 1999) : « Une métaheuristique est un processus itératif maître qui guide et modifie les opérations des heuristiques subordonnées pour produire efficacement des solutions de haute qualité. Elle peut, à chaque itération, manipuler une solution complète ou incomplète, ou bien une collection de solutions. Les heuristiques subordonnées peuvent être des procédures de bas ou de haut niveau, de simples recherches locales, ou juste des méthodes constructives ».

Les métaheuristiques utilisent un haut niveau d'abstraction qui leur permet d'être adaptées à une large gamme de problèmes différents. Le but d'une métaheuristique est de chercher une solution (sous forme d'une permutation ou d'un vecteur), qui minimise ou maximise une fonction objective décrivant la qualité d'une solution. L'ensemble des solutions possibles forme l'espace de recherche. Les métaheuristiques progressent dans cet espace en suivant un processus itératif manipulant une ou plusieurs solutions pour essayer de converger vers un optimum global. Cette recherche itérative est mise en œuvre par, soit des procédures simples comme la recherche du voisinage, soit à base d'algorithmes complexes généralement inspirés de processus naturels ou physiques.

On peut résumer le fonctionnement des métaheuristiques par les règles suivantes (non exhaustives) :

- L'objectif est d'explorer efficacement l'espace de recherche afin de trouver les meilleures solutions.
- Le concept de base d'une métaheuristique peut être décrit d'une manière abstraite sans faire appel à un problème spécifique.
- Les métaheuristiques font appel à des heuristiques qui tiennent compte de la spécificité du problème étudié. Ces heuristiques sont contrôlées par un processus maître de niveau supérieur.
- Les métaheuristiques font usage de l'expérience accumulée durant la recherche pour mieux guider la suite du processus.
- Les métaheuristiques ne donnent aucune garantie sur l'obtention de la solution optimale.
- Les métaheuristiques contiennent en général des mécanismes qui permettent d'éviter de se bloquer dans un optimum local.

On peut distinguer les métaheuristiques qui ne manipulent qu'une seule solution de celles qui manipule une famille de solutions (une population). Parmi les méthodes qui manipulent une seule solution en essayant itérativement de l'améliorer, on peut citer, par exemple, la recherche tabou, le recuit simulé, et la méthode à voisinage variables. En ce qui concerne les métaheuristiques manipulant plusieurs solutions, les algorithmes génétiques sont l'exemple le plus rencontré.

Il existe d'autres classifications des métaheuristiques selon la manière d'utiliser la fonction objectif, ou bien en fonction du nombre de structures de voisinages utilisées, ou encore selon l'utilisation de l'historique de la recherche durant le processus d'optimisation. Nous allons utiliser deux approches de métaheuristiques, il s'agit des algorithmes génétiques, et du recuit simulé.

Dans la section suivante nous présentons les algorithmes génétiques, leurs principes, leurs schémas élémentaires, ainsi que les opérateurs nécessaires à leurs applications. Nous développons ensuite notre propre algorithme génétique avec les opérateurs et le schéma que nous proposons pour résoudre le problème décrit dans le chapitre 3. La section 4.4.4 est consacrée à la méthode du recuit simulé. Après un bref rappel de cette méthode, nous décrivons la manière dont nous l'utilisons.

3.4.2 Les Algorithmes Génétiques

Sur la base des travaux qu'il a mené entre 1960 et 1970, Holland (Holland, 1975) a développé les principes fondamentaux des algorithmes génétiques pour les problèmes d'optimisation mathématique. Les premiers travaux sur les algorithmes génétiques ont commencé dans les années cinquante par des biologistes ayant simulé des structures biologiques sur ordinateur. Puis, vinrent les travaux de Holland qui ont adapté cette structure pour des problèmes d'optimisation mathématiques, mais les ordinateurs de l'époque n'avaient pas assez de puissance pour les mettre en œuvre. L'ouvrage de Goldberg (Goldberg, 1989) a permis de populariser cette approche dans la communauté scientifique.

Les algorithmes génétiques ont pour base les concepts de la génétique et la théorie de l'évolution de Darwin. La génétique représente chaque individu par son génome qui est un ensemble d'informations identifiant complètement l'individu, ce sont les *chromosomes*. L'ensemble des individus forme une *population*. Ces individus vivent dans le même milieu, et se partagent donc les mêmes ressources, et font face aux mêmes dangers. Ainsi, seuls les

individus les plus forts survivront. Les plus faibles et les moins adaptés auront tendance à disparaître au fil des générations. La *force* d'un individu est une mesure de sa qualité. Une population d'individus évolue au cours des générations par l'application de certains opérateurs génétiques. Ainsi, la reproduction est un mixage des chromosomes de deux individus parents qui donne naissance à des individus enfants qui héritent de certaines propriétés et caractéristiques des parents. Au fil des générations, les meilleurs gènes vont donc se propager en échangeant les meilleures caractéristiques. Cette opération est appelée *croisement*. L'apparition de gènes nouveaux chez un enfant, qui ne sont pas hérités de l'un des parents, est appelée *mutation*. Ce phénomène est rare mais permet d'expliquer des changements de morphologie chez certaines espèces. La *sélection* naturelle qui détermine les individus qui survivent et ceux qui disparaissent conduit, de génération en génération, à avoir une population composée d'individus de plus en plus forts et adaptés au milieu dans lequel ils doivent vivre.

Ces règles dans le milieu du vivant, ont été introduites dans des algorithmes (évolutions simulées sur l'ordinateur) et adaptées aux problèmes d'optimisation combinatoire.

3.4.2.1 Principes

Les algorithmes génétiques tentent de simuler le processus d'évolution des humains. Ainsi, on retrouve les termes : sélection, croisement et mutation qui sont les opérations de base des algorithmes génétiques. Elles s'appuient dans leur fonctionnement sur les processus naturels. Les algorithmes génétiques partent d'une population de base, qui est améliorée au fil des générations par l'introduction de nouveaux éléments dans la population. L'opération de sélection choisit dans la population un ensemble de couples d'individus selon un certain critère, pour être utilisés comme parents afin de produire de nouveaux individus. Le croisement combine les parents choisis par l'opération de sélection, pour générer un ou plusieurs enfants. Enfin, la mutation apporte un facteur de diversification à la population, en modifiant les enfants générés par le croisement, avec une certaine probabilité généralement très petite, vu le fait qu'un tel phénomène est rare en milieu naturel (selon la théorie de Darwin).

La qualité d'une solution est donnée par la valeur de la fonction de fitness. Celle-ci est généralement une fonction de gain associée à la solution. La solution est codée sous forme de chromosome, cette codification joue un rôle prépondérant dans l'efficacité de l'algorithme, car l'application des opérateurs génétiques peut changer selon le codage retenu.

Les algorithmes génétiques ont été appliqués avec succès aux problèmes d'optimisation combinatoire. Ils tiennent leur succès de leur rapidité et la qualité des solutions qu'ils produisent. Ils permettent aussi d'avoir des familles de solutions, ce qui laisse une marge de flexibilité aux décideurs pour choisir la solution qui convient le mieux à un moment donné, se basant sur leurs expériences. De plus, on peut intégrer dans la population des solutions ou des parties de solutions qu'on juge préalablement intéressantes. Ce type de méthodes permet en principe une meilleure exploration de l'espace de recherche, un balayage intelligent en somme.

Comme pour toutes les méthodes approchées, les algorithmes génétiques ne donnent aucune garantie sur la qualité des solutions qu'ils fournissent. Tout repose sur les opérateurs génétiques appliqués et leurs capacités à transmettre les *bonnes* informations de générations en générations. Un autre paramètre sur leurs performances pourrait être la durée pendant laquelle on les exécute. Il est clair qu'avec ce genre de méthodes, plus on les laisse travailler, plus on a des chances de trouver de meilleures solutions. Ils ne contiennent aucune règle interne qui permet de prouver qu'une solution obtenue dans un temps fini est optimale (il n'y

a que des preuves de convergences asymptotiques, c'est à dire en temps infini). C'est donc un compromis à faire entre qualité/durée qu'il appartient aux décideurs dans les organismes d'application de choisir.

3.4.2.2 Schémas

On a dit ci-dessus que les algorithmes génétiques partent d'une population initiale qu'on améliore au fil du temps par le biais des opérateurs génétiques. Une population est composée d'un ensemble de chromosomes, pour reprendre les termes du processus naturel duquel sont issus les algorithmes génétiques. Un chromosome représente une solution. La façon dont la solution est représentée par le chromosome est très importante pour l'efficacité de l'algorithme. Pour améliorer la population de base, les opérations de sélection, de croisement et de mutation sont appliqués d'une manière itérative jusqu'à un certain critère d'arrêt, qui peut être le nombre maximum d'itérations permises ou, le nombre maximum d'itérations sans amélioration de la meilleure solution, ou même la durée d'exécution permise. La figure suivante montre une itération de base d'un algorithme génétique.

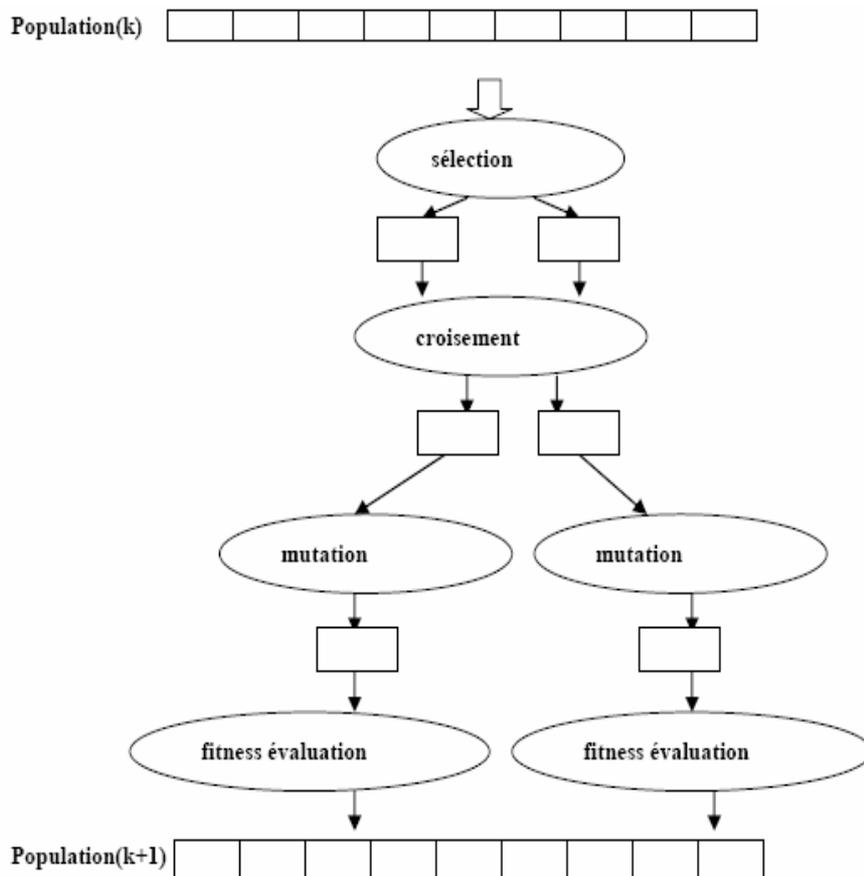


Figure3.14 Itération de base d'un algorithme génétique incrémental.

Dans le schéma d'algorithme génétique très simple illustré par la figure ci dessus, on part d'une population de départ qui est construite le mieux possible par rapport aux propriétés du problème considéré, ou bien complètement aléatoire, ou encore un mélange d'individus obtenus aléatoirement et d'autres obtenus par des heuristiques construisant des solutions

approchées, en essayant de tenir compte des caractéristiques du problème à résoudre. L'obtention d'une nouvelle population (ou nouvelle génération d'individus) est réalisée par l'application de croisements ou par modification de l'empreinte génétique (chromosome) via l'opérateur de mutation. Pour cela un opérateur de sélection choisit deux individus pour les accoupler (croiser) afin d'en obtenir deux enfants qui seront éventuellement mutés. Les nouveaux individus sont insérés dans la population, tandis que d'autres individus en sont supprimés pour toujours garder une taille constante (en général). Ce processus est répété jusqu'à atteindre un nombre maximum d'itérations.

Cela n'est pas dit explicitement dans la plupart des algorithmes présentés dans la littérature, mais il est évident que le résultat d'un algorithme génétique est toujours la meilleure solution trouvée, même si elle n'existe plus dans la population finale. Nous prendrons, bien sûr, toujours cette hypothèse.

L'algorithme 4.15 a présenté un schéma d'un algorithme génétique dit incrémental car il ne traite qu'un couple de parents à chaque itération.

```
Algorithme Génétique incrémental
Début
    Générer la population initiale
Répéter
    Sélection :
        sélectionner deux parents
    Croisement :
        obtenir des solutions enfants.
    Mutation :
        Modifier avec une faible probabilité les enfants.
    Remplacement
        Remplacer des solutions dans la population par les
        enfants
Jusqu'à un certain critère d'arrêt
Fin
```

Figure 3.15 Structure générale d'un algorithme génétique incrémental.

Le schéma le plus connu de l'application des algorithmes génétiques est le schéma simple de Goldberg (Goldberg, 1989). Il s'agit d'un schéma sans recouvrements ni chevauchements entre les populations de générations successives, avec tout de même une probabilité d'avoir des individus identiques dans des générations successives. Ce sont les individus qui ont « échappé » aux opérations de croisement et/ou de mutation.

Avant d'appliquer un tel algorithme illustré par la figure 3.16 , nous devons définir les paramètres suivants :

- La taille de la population qui reste constante au fil des générations.
- La probabilité de copier un individu.
- La probabilité de croiser un individu.
- La probabilité de mutation.
- Le nombre de générations avant de s'arrêter. Il s'agit du critère d'arrêt de l'algorithme.

Schéma simple de Goldberg**Début****Initialisation**

- Générer une population initiale P_0 de Q individus.

Evaluation :

- Evaluer la force (fitness) de chaque individu de la population P_{k-1} .

Sélection :

- Sélectionner $Q/2$ couples d'individus dans la population P_{k-1} .

Croisement :

- On applique l'opérateur de croisement à chaque couple d'individus avec la probabilité p_c . En cas de croisement, chaque couple est remplacé par ses enfants.
- Avec la probabilité $1-p_c$ on recopie le même couple.
- La probabilité de croisement est choisie plutôt grande.

Mutation

- On applique l'opérateur de mutation à chaque individu avec la probabilité p_m . L'individu est conservé à l'identique avec la probabilité $1-p_m$.
- La probabilité de mutation est choisie plutôt petite.

Remplacement :

- La population P_k obtenue remplace la population P_{k-1} .

Arrêt :

- Aller à *évaluation* tant que le nombre de générations à générer n'est pas atteint.

Fin

Figure 3.16. Schéma simple de Goldberg.

Les cinq éléments suivants sont essentiels lors de la conception d'un algorithme génétique pour un problème d'optimisation particulier :

- Le codage de chaque solution sous forme de chromosome. Le codage de la solution est un facteur très important dans l'efficacité d'un algorithme génétique, c'est la façon avec laquelle on représente un chromosome. Le codage le plus simple, toujours possible mais pas toujours efficace, est de représenter une solution par une chaîne de bits, une suite de '0' et de '1'.
- Un mécanisme pour la génération de la population initiale. Le choix de la population initiale est très important car il peut rendre plus ou moins rapide la convergence vers de très bonnes solutions proches de l'optimum global. Il est essentiel que la population de base soit bien dispersée dans l'espace de recherche pour bien balayer ce dernier (diversification). La population initiale peut être obtenue par génération aléatoire de solutions, ou par des heuristiques constructives spécifiques au problème étudié.
- Une mesure du fitness ou force de chaque individu (solution). La force d'un individu peut être (et elle est souvent) différente de la valeur de la fonction objective qui détermine le coût réel d'une solution, c'est obligatoire en particulier si on recherche

un minimum alors que la force doit être maximisée dans la plupart des schémas d'algorithmes génétiques.

- Les opérateurs génétiques de croisement, de mutation, de sélection et de remplacement. Le croisement est un opérateur agissant sur un couple d'individus, résultant en une combinaison qui tend à garder ou récupérer les meilleures propriétés des parents pour les transmettre à leurs enfants. La mutation modifie la structure d'un individu. Elle a pour but d'enrichir la diversification de la population pour explorer d'autres espaces non encore explorés. Cet opérateur évite d'avoir une population uniforme qui pourrait conduire vers un optimum local prématuré.
- Les paramètres de réglage, comme la taille de la population, les critères d'arrêt, les probabilités d'application des opérateurs.

Les algorithmes génétiques ont été appliqués avec succès à une multitude de problèmes d'optimisation combinatoire.

3.4.2.3 Opérateurs classiques

3.4.2.3.1 Sélection et remplacement

La procédure de sélection permet de choisir les chromosomes (solutions) géniteurs qu'on va donner en entrée pour les procédures de croisements. Nous présentons ici deux manières différentes pour effectuer la sélection :

- i. Le tournoi binaire : le principe est de tirer deux individus de la population d'une manière aléatoire, choisir le meilleur (celui dont la fonction objectif est minimale dans le cas d'une minimisation). Refaire cette opération une seconde fois. Les deux individus sélectionnés sont donnés comme parents au processus de croisement.

```

Sélection Tournoi binaire
Begin
  For k=1 to 2
  begin
    i=rand() ;
    j=rand()
    if (value(i)< value(j))
      select chromosome i
    else
      select chromosome j
    endfor
  End

```

Figure 3.17. Algorithme de la sélection par tournoi binaire.

- ii. La roulette (Goldberg, 1989) : Si l'on considère qu'on a une population $P = x_1, x_2, \dots, x_N$ de N individus, la technique de la roulette consiste à associer à chaque individu x_i une probabilité proportionnée avec laquelle il est sélectionné.

L'algorithme suivant décrit cette technique.

Sélection La roulette**Notations :**

N : la taille de la population.

X_i : l'individu numéroté i et/ou placé à la $i^{\text{ème}}$ place dans la population.

$F(X_i)$: la fitness de l'individu i , il s'agit d'une fonction à maximiser, plus la fitness est grande, plus l'individu doit avoir une forte probabilité d'être choisi.

Début

Calcul de la somme des fitness de toute la population :

$S=0$.

Pour $i=1$ à N **Faire**

$S=S+F(X_i)$

FinPour

Tirage aléatoire normalisé à la somme des fitness :

$H=S*\text{rand}(0,1)$;

Recherche de l'individu « res » sélectionné par la roulette :

$\text{res}=1$;

$SP=F(X_{\text{res}})$;

Tantque $H>SP$ **faire**

$\text{res}=\text{res}+1$;

$SP=SP+ F(X_{\text{res}})$;

fin tant que

Fin

Figure 3.18. Algorithme de la roulette.

Le remplacement est l'opération qui intègre les nouveaux individus issus de croisements et/ou de mutations dans la population. Cela implique, en général, de choisir certains individus pour les éliminer. Dans les schémas dits « élitistes » on choisit les individus les plus faibles (de mauvaise qualité). Il existe plusieurs schémas pour effectuer cette opération. On peut par exemple choisir d'éliminer les individus les plus faibles, ou bien choisir aléatoirement quelques individus aléatoirement dans une portion de mauvaise qualité, ou bien remplacer les parents (c'est ce qui est fait totalement dans le schéma simple de Goldberg).

Remplacement**Début**

▪ $i=\text{rand}(N/2,N)$.

▪ supprimer l'individu i .

▪ remplacer par un nouvel enfant.

Fin

Figure 3.19. Exemple d'une procédure de remplacement.

3.4.2.3.2 Les croisements

Il existe une multitude d'algorithmes de croisements dans la littérature, certains sont spécifiques à des problèmes particuliers, tandis que d'autres sont génériques à un grand ensemble de problèmes. Nous en présentons ici quelques uns des plus utilisés qui sont propres aux structures de chromosomes de permutations.

i. Le croisement d'ordre OX (Order Crossover):

C'est un croisement en deux points de coupure générés aléatoirement. Conçu par (Davis, 1985), il est réalisé comme suit :

- La première phase consiste à recopier dans le fils la partie centrale (entre les deux points de coupure) du père.
- La deuxième phase consiste à copier les éléments restants dans leur ordre dans la mère en commençant par remplir la partie droite, puis la partie gauche.
- Un deuxième fils est obtenu par l'inversion des rôles de la mère et du père.

Voici un exemple :

Coupure	p=4				q=7					
P1	2	9	4	7	6	5	8	3	0	1
P2	3	6	5	9	0	4	7	1	2	8
E1	9	0	4	7	6	5	8	1	2	3
E2	9	6	5	8	0	4	7	3	1	2

Figure 3.20. Exemple de croisement OX.

Il a été conçu pour les problèmes de voyageur de commerce.

ii. *Croisement d'ordre linéaire LOX (Linear Order Crossover) :*

LOX (Falkenauer et Bouffouix, 1991) est destiné à des chromosomes linéaires, c'est à dire codant des objets qui ont clairement un début et une fin. C'est une variante de IOX.

Pour deux parents P1 et P2 de longueur τ , LOX commence par tirer au sort deux positions p et q avec $1 \leq p < q \leq \tau$. Pour construire l'enfant C1, la portion de P1 entre p et q inclus est copiée dans C1 aux mêmes positions. Le parent P2 est ensuite balayé de 1 à τ , et les éléments non déjà présents dans le fils C1 remplissent de gauche à droite les positions libres. L'enfant C2 s'obtient de la même manière en permutant les rôles de P1 et P2.

Coupure	p=4				q=7					
P1	2	9	4	7	6	5	8	3	0	1
P2	3	6	5	9	0	4	7	1	2	8
E1	3	9	0	4	6	5	8	7	1	2
E2	2	9	6	5	0	4	7	8	3	1

Figure 3.21. Exemple de croisement LOX.

iii. *Le croisement d'ordre à un point 1X:*

Désigné par 1X pour (1 point Crossover), il a été présenté dans (Davis, 1985). C'est un croisement simple où l'on commence par déterminer aléatoirement un point de coupure qui correspond au rang associé à ce point. On procède ensuite, par recopie du père du début jusqu'au point de coupure, et de compléter par le reste des gènes dans l'ordre de la mère. Pour obtenir un deuxième enfant, on inverse le « début » et la « fin ». On peut encore obtenir deux autres fils en inversant les rôles des deux parents. En voici un exemple :

P1	2	9	4	7	6	5	8	3	0	1
P2	3	6	5	9	0	4	7	1	2	8
E1	2	9	4	7	3	6	5	0	1	8
E2	3	6	5	9	2	4	7	8	0	1
E3	9	2	4	7	6	5	8	3	0	1
E4	9	6	5	3	0	4	7	1	2	8

Figure 3.22. Croisement 1X.

iv. *Croisement 2X : Two point Crossover*

Ce croisement est une extension de 1X à deux points de coupure. Pour obtenir le premier fils, la partie centrale du père est recopiée aux mêmes positions dans le fils, puis la mère est balayée de gauche à droite. Un deuxième fils est obtenu en inversant les rôles de la mère et du père. Deux autres fils peuvent être obtenus en gardant les parties latérales plutôt que les parties centrales. En voici un exemple :

	Coupure				p=4			q=7		
P1	2	9	4	7	6	5	8	3	0	1
P2	3	6	5	9	0	4	7	1	2	8
E1	2	9	4	7	6	5	8	3	0	1
E2	3	6	5	9	4	7	0	1	2	8
E3	9	4	7	2	6	5	8	3	0	1
E4	9	6	5	3	0	4	7	2	8	1

Figure 3.23. Exemple de croisement 2X.

E1 est identique à P1 car son milieu correspond à des gènes qui ont le même ordre dans les deux chromosomes. E2 est très voisin de P2 car le milieu est petit et seul le gène 0 a changé de place par rapport aux autres gènes. Par contre, E3 et E4 sont sûrement très intéressants, on a réorganisé le sous-ensemble de gènes du début et de la fin de manière indépendante et on a conservé le milieu de l'ordonnancement.

Les enfants sont donc plus ou moins intéressants selon le degré de modification que l'on souhaite obtenir.

3.4.2.3.3 Les mutations

La mutation est appliquée avec une faible probabilité, par exemple de l'ordre de 0.2, pour apporter un facteur de diversification à la population (du sang neuf). Il existe une variété d'opérateurs de mutations dans la littérature. Le plus simple, dans le cas d'un chromosome binaire, est d'inverser la valeur d'un bit. Un autre opérateur très simple consiste en l'échange de deux travaux tirés aléatoirement. On pourrait aussi remplacer cet opérateur par une recherche locale, dans ce cas la mutation (opérateur unaire) joue un rôle améliorant (intensifiant de la recherche locale) et non plus diversifiant. C'est ce qui est fait dans les algorithmes mémétiques, extension des algorithmes génétiques.

Rang	:	1	2	3	4	5	6	7	8	9	10 = τ
Positions :		4,7.									
chromosome P1 :		2	9	4	7	6	5	8	3	0	1
après mutation											
chromosome P1 :		2	9	4	8	6	5	7	3	0	1

3.4.3 Recherche de solutions avec les AG

Nous décrivons dans cette section tous les éléments dont on a besoin pour l'algorithme génétique que nous proposons. Nous présentons d'abord le codage des chromosomes, les opérateurs de croisements et de mutation, et le schéma général de l'algorithme.

3.4.3.1 Le Codage

Le codage de la solution est un facteur très important dans l'efficacité d'un algorithme génétique, c'est la façon avec laquelle on représente un chromosome. Une solution du problème est simple : il s'agit de permutations de travaux, en voici un exemple :

3	4	1	8	9	2	10	7	6	5
---	---	---	---	---	---	----	---	---	---

Néanmoins, pour implémenter les opérateurs que nous proposons, il nous faut d'autres informations que nous conservons également dans le chromosome. Ces informations sont les dates de début d'exécution des jobs sur la première machine, et sur la dernière machine (car nous nous intéressons à des problèmes de juste à temps où les pénalités d'avance-retard sont associées à la première et à la dernière opération).

3.4.3.2 Les opérateurs

Nous présentons ici les opérateurs binaires (croisements) et unaires (mutations) que nous avons conçu ou adapté spécialement pour le problème étudié.

3.4.3.2.1 Les croisements

Nous avons développé des croisements basés sur un tri complet, ou partiel, et procédant par recopie de certaines parties des parents, le tout en utilisant les informations sur les dates de début d'exécution des jobs sur la première et la dernière machine selon l'algorithme suivant :

Algorithme

Début

- i. Nous disposons des parents P1 et P2.
- ii. Générer un point de coupure aléatoirement dans le parent P1. Soit L ce point.
- iii. Copier dans le fils F1 la partie de 1 à L (partie gauche) de P1.
- iv. Dans la deuxième partie du fils, les jobs non encore attribués sont mis dans l'ordre croissant de $\lambda_i^1 + \lambda_i^2 / 2$ tel que λ_i^1 (resp. λ_i^2) est la date de début d'exécution du job sur la première machine dans le parent P1 (resp. P2).

Fin

Il est à noter que les λ_i^k correspondent à la solution du Pert Coût pour les individus faisables et aux dates au plus tôt sur les dates de début impératives pour les individus non faisables.

En inversant la copie (copier la partie droite au lieu de la partie gauche) on obtient un autre enfant. En intégrant le classement aussi à la partie copiée, on obtient un autre enfant. En inversant les rôles de P1 et P2 on obtient un autre fils. Finalement, si on ne fait pas de copie, et qu'on fasse le tri sur tout le chromosome, on obtient un autre type d'enfants. On peut

prendre les λ_i de la première ou de la dernière machine. En définitif, on a 18 façons de générer des enfants.

La figure suivante montre la construction des enfants à partir des parents P1 et P2.

P1	$\lambda_i^1 + \lambda_i^2 / 2$
P2	
$\lambda_i^1 + \lambda_i^2 / 2$	P1
	P2
$\lambda_i^1 + \lambda_i^2 / 2$	

Figure 3.24 les croisements basés sur les dates d'exécutions.

Une autre manière de faire un croisement spécifique à notre problème de juste à temps qui n'est applicable que si les deux parents sont faisables, est de considérer le coût de chaque job (dans le parent considéré) à la position à laquelle il est affecté. La figure fig. décrit l'algorithme du croisement basé sur les coûts des jobs. Le principe est simple : on a deux parents, on parcourt le parent principal (soit le premier parent sélectionné) de gauche à droite, et on compare le coût de chaque job dans chacun des deux parents. Le job est affecté dans le fils, à la position dans le père ou la mère qui a le moindre coût (somme d'avance et retard par rapport aux dates souhaitées de livraison et de disponibilité). Si cette position est déjà affectée, alors on l'affecte à la première position libre qui succède. Le deuxième enfant est obtenu en se basant cette fois-ci sur les positions, et non sur les jobs. On parcourt les deux parents de gauche à droite, et pour chaque position est affecté le job dans le père ou la mère qui a le moindre coût. Si pour une position donnée, les deux jobs correspondant sont déjà affectés, alors on y affecte le premier job non affecté.

Algorithme Job Based Crossover

Début

- Soient P1 et P2 les deux parents de longueur N.

Pour i=1 à N **Faire**

- Soit J1 l'indice du job à la position i dans P1.
- Chercher la position de J1 dans P2. Soit j cette position.

Si (les positions i et j dans le fils F1 ne sont pas déjà remplies)

Alors

Si coût (P1, J1) < coût (P2, J1) **Alors**

- Le job J1 est affecté à la position i dans F1.

Si non J1 est affecté à la position j dans F1.

Si non

Si la position i est remplie alors affecter J1 à la position j.

Si la position j est remplie alors affecter J1 à la position i.

Si non

Affecter J1 à la première position libre après i ou j.

FinPour

Fin

Figure 3.25. Algorithme du croisement basé sur les coûts des jobs.

L'algorithme du croisement basé sur les positions est donné par la figure suivante :

Algorithme Position Based Crossover**Début**

- Soient P1 et P2 les deux parents de longueur N.

Pour i=1 à N **Faire**

- Soit J1, J2 les indices des jobs à la position i dans P1 et dans P2.
- **Si** coût(P1, J1) < coût(P2, J2) **Alors**
 - **Si** J1 non encore affecté **Alors**
 - Affecter J1 à la position i dans le fils F1.
 - **Sinon**
 - **Si** J2 non encore affecté **Alors**
 - Affecter J2 à la position i dans le fils F1.
- **Sinon**
 - **Si** J2 non encore affecté **Alors**
 - Affecter J2 à la position i dans le fils F1.
 - **Sinon**
 - **Si** J1 non encore affecté **Alors**
 - Affecter J1 à la position i dans le fils F1.

FinPour**FIN**

Figure 3.26 Algorithme du croisement basé sur les positons.

3.4.3.2 Les mutations

Nous avons utilisé la mutation simple basé sur l'échange de deux jobs tirés aléatoirement, ainsi que deux nouvelles mutations dans lesquelles nous avons essayé d'utiliser les propriétés du problème de juste à temps étudié. La première mutation consiste à avancer d'une position le job avec le plus grand retard, la deuxième consiste à reculer d'une position le job avec la plus grande avance.

En outre, pour les problèmes spécifiques de structure d'ateliers en flowshop de permutation, nous avons adapté la technique dite de *blocks de jobs* développée par (Nowicki et Smutnicki, 1996). Cette technique a été développée pour le problème de flowshop de permutation avec l'objectif de minimiser le makespan. Elle définissait une nouvelle méthode de recherche par voisinage utilisé dans un algorithme de recherche taboue (pour minimiser la durée totale).

Cette technique se base sur la définition de la notion de blocks de jobs. On commence par rechercher le chemin critique (le plus long chemin) dans le problème modélisé par un graphe potentiel. En utilisant certaines propriétés, les jobs sont regroupés en blocks. La principale utilité de cette méthode est qu'elle définit un ensemble de permutations (un voisinage) qui ne sont pas intéressantes (n'améliore pas le makespan minimum), et de ce fait on oriente la recherche vers un ensemble bien précis, ce qui peut améliorer et accélérer la convergence de l'algorithme de recherche de solutions faisables pour notre application. Les détails sur cette méthode sont présentés dans (Nowicki et Smutnicki, 1996) et aussi dans (Janiak et Portmann, 1998).

3.4.3.3 Schéma Elitiste**3.4.3.3.1 Les paramètres**

On doit aussi définir certains paramètres comme la taille de la population, les probabilités de croisement, et de mutation, le nombre de générations maximum, le critère d'arrêt. Pour faire le calibrage de l'algorithme génétique, on se base sur les expériences et on fait un compromis entre temps d'exécution, rapidité de convergence, et qualité de la population.

3.4.3.3.2 Choix des opérateurs

Dans la conception d'un algorithme génétique, il ne faut jamais perdre de vue qu'il n'y a aucun opérateur de croisement, mutation, sélection ou remplacement qui soit meilleur qu'un autre dans tous les cas et pour tous les problèmes. L'expérience montre qu'il faut faire des tests pour choisir les opérateurs adéquats pour un problème donné. C'est ainsi que sur la base des tests effectués nous avons choisis les opérateurs qui semblent les meilleurs parmi ceux testés pour notre problème.

3.4.3.3.3 Schéma

Nous proposons le schéma dit *élitiste* suivant :

Schéma élitiste
Début
Initialisation
<ul style="list-style-type: none"> ▪ Générer une population initiale P_0 de Q individus dont les $2/3$ sont faisables.
Evaluation :
<ul style="list-style-type: none"> ▪ Evaluer la force (fitness) de chaque individu de la population P_{k-1}. Trier la population dans l'ordre décroissant.
Sélection :
<ul style="list-style-type: none"> ▪ Sélectionner $2 \times Q/3$ couples d'individus dans la première partie de la population P_{k-1}.
Croisement :
<ul style="list-style-type: none"> ▪ On applique un opérateur de croisement orienté données à chaque couple d'individus. Si un des enfants est identique à l'un des parents, alors on génère des fils avec l'opérateur 2X. ▪ On garde et les parents, et les enfants dans la population.
Mutation
<ul style="list-style-type: none"> ▪ On applique l'opérateur de mutation à chaque individu avec la probabilité p_m. L'individu est conservé à l'identique avec la probabilité $1 - p_m$. ▪ La probabilité de mutation est choisie plutôt petite.
Remplacement :
<ul style="list-style-type: none"> ▪ On trie la population, et on ne garde que les Q meilleurs individus.
Arrêt :
<ul style="list-style-type: none"> ▪ Aller à <i>évaluation</i> tant que le nombre de générations à générer n'est pas atteint.
Fin

Figure 3.27 Schéma élitiste.

3.4.4 Le recuit simulé

3.4.4.1 Introduction

La méthode du recuit simulé pour les problèmes d'optimisation combinatoire s'appuie sur les travaux de Metropolis (Metropolis, 1953) qui a décrit l'évolution de l'équilibre thermodynamique d'un système physique. A température nulle, le système évolue à partir d'une configuration vers la, ou les, configurations d'énergie les plus basses par une méthode du type gradient. En revanche, à température non nulle, le système peut être tiré vers d'autres bassins plus intéressants. En bref, il faudrait que la température soit assez grande pour permettre au système de sauter les barrières, et assez faible pour lui permettre d'être attiré vers le fond. C'est une approche à dynamique double : recherche de minima à température fixée, plus une dynamique de diminution de la température. La recherche par un système physique

des états d'énergie les plus bas est transformée par analogie en un processus d'optimisation combinatoire.

Kirkpatrick (Kirkpatrick, 1983) a été le premier à introduire cette méthode pour les problèmes d'optimisation combinatoire.

3.4.4.2 Principe de fonctionnement

L'idée du recuit simulé en optimisation combinatoire est de simuler une opération de recuit thermique. Partant d'une solution (configuration) initiale qui peut être générée aléatoirement ou à l'aide d'une heuristique constructive, on génère une solution proche (dans le voisinage) d'une manière totalement aléatoire ou bien suivant une méthode spécifique au problème traité intégrant du hasard. Au départ, on configure la température à un niveau assez élevé pour éviter de tomber dès le départ dans un optimum local. Le voisin de la solution initiale peut soit améliorer, soit dégrader le critère à optimiser. Si le voisin améliore le critère, alors il est immédiatement accepté et devient la configuration courante. La réitération de ce processus conduit à trouver la meilleure solution dans un voisinage (méthode de la descente). Si le voisin dégrade la critère, alors il peut être accepté avec une certaine probabilité qui dépend de la dégradation et de la température. Ceci permet de sortir d'optima locaux, et d'explorer ainsi une grande partie de l'espace de recherche. La température diminue au fur et à mesure de l'avancement du processus d'optimisation, on tend ainsi à explorer le mieux possible une région de l'espace de recherche qu'on considère particulièrement intéressante. Il s'agit d'un processus d'intensification. L'algorithme du recuit simulé est donné par la figure suivante :

Algorithme Simulated Annealing

Début

```

Pour k de 1 à P Faire    % P essais

    % Initialisation de l'essai
    xo:=solution aléatoire; % Solution de départ
    fm:=f(xo) ; % Init. meilleure valeur
    J:=0; % Init. compteur arrêt
    T:=To; % Init. Température
    % Déroulement de l'essai %

    Tantque J≤Q Faire

        % Palier température
        I:=0; % Init. compteur pal. temp.
        % Boucle sur les essais élémentaires %
        % Fin palier température
        T:=T*a;

        FinTantque
        retenir fm , xm;

    FinPour    % Fin essai

Fin

% Boucle interne sur les essais élémentaires
Tantque I≤N & J≤Q Faire
x1:=VOISIN(xo); % 1 essai élémentaire
I:=I+1; (compteur pour palier de température)
J:=J+1; (compteur pour arrêt final)
Si f(x1) < f(xo) Alors
    xo:=x1;
    Si f(xo)<fm Alors
        xm:=xo; fm:=fo; J:=0
Si non
    Δf:=f(x1)-f(xo);

```

```
U:=Uniforme(0,1);  
Si U ≤ exp (- Δf / T) Alors  
  xo:=x1  
Finsi;  
FinTantque
```

Figure 3.28 Algorithme du recuit simulé.

Pour obtenir le voisin d'une solution, nous utilisons les opérateurs de mutation qu'on a défini pour l'algorithme génétique. Le reste des paramètres : température initiale, critère d'arrêt, probabilité de diminution, nombre de paliers, sont déterminés expérimentalement.

3.5 Conclusion

Nous avons présenté dans ce chapitre la méthode de résolution que nous développons pour les problèmes de juste à temps qui ont fait l'objet du deuxième chapitre. Nous avons combiné l'utilisation de méthodes exactes, et de méthodes approchées. Dans un premier lieu, nous donnons le schéma global de la solution. Celle-ci est basée sur une approche par décomposition qui utilise une métaheuristique pour parcourir l'espace de solutions, et utilise une méthode exacte pour évaluer la solution.

En premier temps, nous montrons l'équivalence entre notre problème de juste à temps pour les chaînes logistiques, et le problème connu de Pert coût. Nous avons plus particulièrement développé un algorithme pour minimiser le nombre d'itérations du Pert coût.

Dans un second temps, Nous présentons deux types de méta heuristiques : les algorithmes génétiques, et le recuit simulé. Nous faisons une présentation brève de la méthode du recuit simulé pour sa simplicité. Nous développons avec plus de précisions les algorithmes génétiques. Nous avons conçu de nouveaux opérateurs génétiques spécifiques au problème. Nous proposons en outre de les utiliser dans un schéma élitiste.

4 Chapitre 4 : Expériences numériques

Résumé

Ce chapitre est consacré à l'évaluation des algorithmes développés dans le chapitre précédent.

4.1 Introduction

Nous présentons dans ce chapitre les résultats des expérimentations que nous avons mené pour évaluer les performances des algorithmes basés sur le Pert Coût (évaluation d'une permutation), et sur les métaheuristiques. Nous présentons en premier lieu la méthode de génération de données. Comme il n'existe pas dans la littérature des benchmarks pour notre problème, nous avons développé une méthode pour les générer en essayant de tenir compte de certaines situations que l'on pourrait trouver dans un milieu industriel, comme par exemple l'hétérogénéité des temps d'exécution des opérations, et la présence de machines goulets.

4.2 Génération de données

Pour tester les performances des algorithmes développés dans cette thèse, nous avons besoin de générer un ensemble de données. Cet ensemble doit couvrir le plus largement possible l'ensemble des cas possibles en milieu réel. On a trois types de données à générer : les temps d'exécutions, les pénalités, et les dates souhaitées et les dates impératives.

4.2.1 Les temps d'exécutions

On distingue deux familles de temps d'exécution : Homogènes et hétérogènes.

a. Homogène : On considère trois familles :

1. H1 : génération aléatoire uniforme sur l'intervalle $[10,40]$ pour des temps d'exécutions relativement petits avec une moyenne de 25 sur une plage de 30. Le quotient de la division du maximum par le minimum est de 4.
2. H2 : génération aléatoire uniforme sur l'intervalle $[25,75]$ pour des temps d'exécutions d'une moyenne de 50, sur une plage de 50. Le quotient de la division du maximum par le minimum est de 3.
3. H3 : génération aléatoire uniforme sur l'intervalle $[1,100]$ pour des temps d'exécutions d'une moyenne de 50, sur une plage de 100. Le quotient de la division du maximum par le minimum est de 100.

b. Hétérogène : on distingue l'hétérogénéité sur les jobs, et sur les machines.

1. Sur les jobs : C'est un mélange des trois familles précédentes de temps d'exécutions pondérées par des probabilités : la famille $[10,40]$ avec une probabilité de 0,5 ; $[25,75]$ avec une probabilité de 0,25 ; $[1,100]$ avec une probabilité de 0,25.
2. Sur les machines : introduction de la notion de machines goulets. Une machine goulet est une machine dont les temps d'exécutions P_{ij} sont générés de manière identique au cas homogène avec les limites de l'intervalle de génération augmentées de 10. La moyenne des temps opératoires est donc augmentée de 10. La somme des durées opératoires sur une machine goulet est en moyenne plus grande de $10n$ que sur le reste des machines. Elle a donc, une charge supplémentaire de travail par rapport aux autres machines. On peut avoir une machine goulet au début (donc machine M_1), ou au milieu, ou bien deux machines goulets.

On a donc 12 familles d'exemples :

- i. Sans goulet H1 H2 H3 (1,2,3),

- ii. Goulet au début H1 H2 H3 (4,5,6),
- iii. Goulet au milieu H1 H2 H3 (7,8,9),
- iv. 2 machines goulets H1 H2 H3 (10,11,12).

4.2.2 Génération des pénalités d'avance et de retard

Nous avons généré six familles de pénalités d'avance et retards par rapports aux dates de disponibilités de composants, et de livraison des produits finis. Nous avons essayé de refléter ce qui se passe le plus souvent dans le milieu industriel, c'est-à-dire la pénalisation plus importante des retards des livraisons, et des avances des approvisionnements en composants. Le tableau suivant récapitule la façon avec laquelle on les génère :

	P1	P2	P3	P4	P5	P6
$\alpha_{ri,j}$	1	1	2	2	Rand (1,10)	Rand (1,10)
$\alpha_{di,j}$	0	0	1	1	0	0
$\beta_{ri,j}$	0	0	1	1	0	0
$\beta_{di,j}$	1	10	2	10	Rand (1,10)	Rand (1,100)

Figure 4.1 génération des pénalités d'avance et de retard.

4.2.3 Génération des dates souhaitées et des dates impératives

Pour être certain d'avoir au moins une solution faisable, nous sommes partis d'une permutation calée gauche et nous avons généré les dates souhaitées et impératives autour de cette solution. On distingue la marge (ou mou) par rapport aux dates souhaitées, et la marge par rapport aux dates impératives. La méthode de génération est décrite par l'algorithme suivant :

Début

Etape 0 :

1. générer une permutation aléatoire.
2. calculer les dates au plus tôt (λ_i) avec $Ri=RiBarre=0$:

$$Dmax = \left(\frac{100 + PourMou1}{100} \right) * \lambda_i \text{ avec } PourMou1 = \{10, 20, 40, 80\} \text{ le pourcentage de mou.}$$

3. calculer $\Delta = (Dmax - \lambda_i) / 2$.

Etape 1 :

4. décaler les λ_i de Δ :

$$\lambda_i^1 = \lambda_i + \Delta.$$

μ_i^1 Les dates au plus tard avec le Pert simple à partir de $Dmax - \Delta$.

5. $RiBarre = \text{random}(0, \lambda_i^1)$.

$$DiBarre = \text{random}(\mu_i^1 + P_{im}, Dmax)$$

Etape 2 :

6. $\text{Psi} = \text{DiBarre} - \text{RiBarre} - \sum_j P_{ij}$.
7. $\Delta RDi = (100 - \text{PourMou2}) * \text{Psi} / 100$.
8. $\alpha = \text{random}(0,1)$
 $\text{Ri} = \text{RiBarre} + \lceil \alpha \cdot \Delta RDi \rceil$
 $\text{Di} = \text{DiBarre} - \lceil (1 - \alpha) \cdot \Delta RDi \rceil$

Fin

4.3 Evaluation du Pert Coût

Nous avons implémenté la méthode du Pert coût en langage C++, et l'avons testé sur un ordinateur muni d'un processeur Pentium 4 à 1.7 Ghz et d'une mémoire de 1 Giga. Les tests ont été effectués sur des problèmes de flowshop de permutation dans lesquels on cherche à minimiser la somme des pénalités d'avance et de retard par rapport à la première opération, et à la dernière opération de chaque job.

On a généré des fichiers de données avec la première famille de temps d'exécution et la troisième famille des pénalités d'avance et de retard. Les deux facteurs de mou (par rapport aux dates souhaitées et aux dates impératives) sont à 80. C'est-à-dire assez grand, dans le but d'avoir assez rapidement des permutations faisables sur lesquelles on peut évaluer les performances. Le but ici n'étant pas d'évaluer le ratio de permutations faisables, mais d'évaluer les temps d'exécutions et le nombre d'itération nécessaires au Pert coût.

On a pris le nombre de machines dans l'ensemble (4, 5, 6, 8, 10) et le nombre de jobs dans l'ensemble (5, 10, 15, 20). Pour chaque couple (nombre de job, nombre de machines), on a généré instances différentes. Pour chaque instance on donne le nombre d'itérations ainsi que la durée d'exécution en secondes.

i. Nombre de travaux=5 :

Nombre de machines=4

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	5	4	3	3	5	7	3	3	3	5	4,1
Durée	0.047	0.046	0.047	0.032	0.062	0.094	0.031	0.031	0.031	0.062	0,0483

Nombre de machines=5

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	3	8	8	3	8	4	4	5	5	8	5,6
Durée	0.031	0.094	0.109	0.031	0.11	0.047	0.047	0.062	0.063	0.093	0,0687

Nombre de machines=6

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	4	4	5	6	5	3	6	4	5	3	4,5
Durée	0.047	0.047	0.047	0.078	0.062	0.031	0.063	0.047	0.063	0.031	0,0516

Nombre de machines=8

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	3	8	3	6	7	6	4	8	5	7	5,7
Durée	0,031	0,141	0,047	0,094	0,109	0,094	0,062	0,125	0,078	0,11	0,0891

Nombre de machines=10

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	4	5	3	3	8	7	3	3	3	7	4,6
Durée	0,062	0,079	0,046	0,047	0,125	0,109	0,046	0,047	0,047	0,125	0,0733

ii. Nombre de travaux=10 :

Nombre de machines=4

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	7	4	5	8	13	5	4	6	6	5	6,3
Durée	0,219	0,094	0,125	0,25	0,515	0,141	0,094	0,141	0,156	0,109	0,1844

Nombre de machines=5

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	9	3	7	7	9	12	5	4	4	10	7
Durée	0,296	0,063	0,172	0,235	0,25	0,422	0,125	0,125	0,093	0,391	0,2172

Nombre de machines=6

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	5	12	3	10	3	4	3	4	4	6	5,4
Durée	0,125	0,359	0,078	0,343	0,079	0,094	0,078	0,11	0,093	0,188	0,1547

Nombre de machines=8

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	3	3	3	3	10	12	3	6	3	5	5,1
Durée	0,078	0,094	0,094	0,094	0,422	0,562	0,078	0,219	0,094	0,172	0,1907

Nombre de machines=10

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	5	5	12	4	4	10	7	6	4	9	6,6
Durée	0,156	0,156	0,562	0,125	0,109	0,359	0,265	0,187	0,141	0,375	0,2435

iii. Nombre de travaux=15 :

Nombre de machines=4

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	6	5	5	7	3	8	9	3	5	5	5,6
Durée	0,39	0,266	0,235	0,281	0,109	0,375	0,453	0,11	0,203	0,203	0,2625

Nombre de machines=5

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	3	5	8	5	8	10	7	3	11	5	6,5
Durée	0,109	0,266	0,422	0,203	0,406	0,75	0,422	0,125	0,813	0,25	0,3766

Nombre de machines=6

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	6	3	8	12	5	4	7	3	4	7	5,9
Durée	0,297	0,125	0,391	0,828	0,234	0,156	0,344	0,125	0,172	0,328	0,3

Nombre de machines=8

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	6	22	3	4	3	3	8	4	5	4	6,2
Durée	0,313	2,797	0,14	0,188	0,157	0,14	0,468	0,187	0,281	0,25	0,4921

Nombre de machines=10

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	7	8	6	4	4	8	4	5	4	9	5,9
Durée	0,391	0,594	0,328	0,234	0,235	0,5	0,25	0,282	0,235	0,781	0,383

iv. Nombre de travaux=20 :

Nombre de machines=4

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	9	7	10	3	4	11	7	6	4	6	6,7
Durée	0,797	0,453	1,187	0,203	0,218	1,218	0,765	0,578	0,234	0,453	0,6106

Nombre de machines=5

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	11	5	13	11	12	10	6	10	19	16	11,3
Durée	1,156	0,344	1,922	0,938	1,625	0,828	0,421	0,828	3,719	1,875	1,3656

Nombre de machines=6

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	12	8	3	8	4	4	6	4	4	16	6,9
Durée	1,157	0,86	0,203	0,687	0,328	0,266	0,437	0,297	0,281	1,875	0,6391

Nombre de machines=8

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	9	5	15	5	14	6	7	7	12	6	8,6
Durée	0,812	0,36	2,094	0,406	1,547	0,5	0,562	0,609	1,718	0,469	0,9077

Nombre de machines=10

	Inst1	Inst2	Inst3	Inst4	Inst5	Inst6	Inst7	Inst8	Inst9	Inst10	moyenne
Nombre Itérations	4	4	8	12	4	6	8	4	15	3	6,8
Durée	0,328	0,312	0,703	1,297	0,328	0,531	0,765	0,238	0,375	0,25	0,5127

Les expériences montrent que les temps d'exécution sont relativement courts pour les problèmes générés, et que le nombre d'itérations reste assez petit. Le temps d'exécution ne dépend pas du nombre de machines. Il dépend du nombre de jobs, une comparaison rapide présentée sur le schéma suivant montre que l'allure est approximativement quadratique (la « exp » contient les durées expérimentales pour chaque nombre de jobs, la courbe « lin » est la droite passant par 0 au point 0 et par la durée maximale au point 20, la courbe « car » est la courbe en n^2 passant par 0 au point 0 et par la durée maximale au point 20).

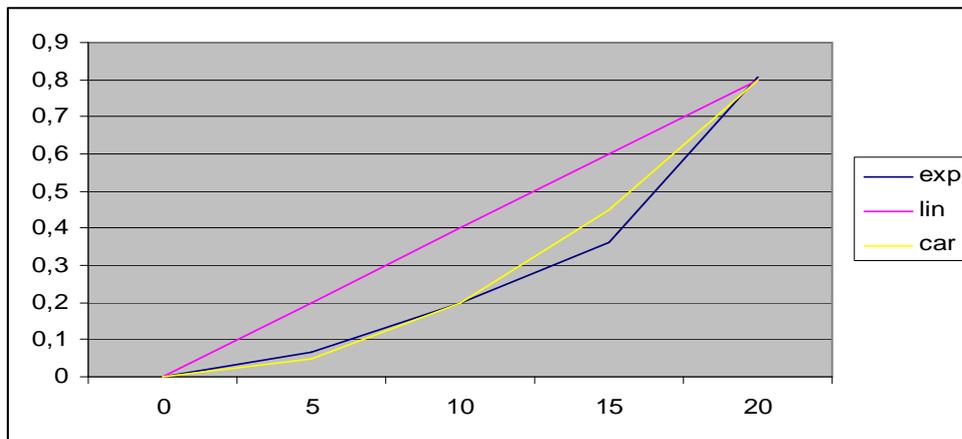


Figure 4.2 Courbe des durées opératoires par nombre de jobs.

4.4 Evaluation des métaheuristiques

Dans cette section, nous présentons les résultats des évaluations des métaheuristiques développées dans le chapitre précédent. Nous comparons quatre méthodes : génération complètement aléatoire, suite à la suggestion d'un des rapporteurs du papier (Portmann et Mouloua, 2007), mais ceci uniquement pour les tests de recherche de solutions faisables ; recuit simulé, algorithme génétique simple avec l'opérateur 2X, et enfin, l'algorithme

génétique élitiste avec un tirage aléatoire uniforme dans un ensemble d'opérateurs conçus et présentés au chapitre 3 pour 2/3 des croisements et avec 2X pour 1/3 des croisements et chaque fois que les nouveaux croisements ne sont pas applicables (un des parents non faisables).

Nous avons programmé en langage C++, et nous avons réalisé les expériences sur un ordinateur Pentium 4 à 1.7 Ghz, et une mémoire de 1 Giga. Comme pour l'évaluation du Pert coût, les expériences ont été effectuées sur des problèmes de flowshop de permutation dans lesquels on cherche à minimiser la somme des pénalités d'avance et de retard par rapport à la première opération et à la dernière opération de chaque job.

Dans une première partie, nous évaluons le pourcentage de solutions faisables créées par chaque méthode par rapport au nombre total de solutions générées. Pour cela, nous générons des fichiers de données avec plus ou moins de marge (coefficient de mou). Plus le mou est grand, plus il y a de chances de trouver des solutions faisables.

Dans la deuxième partie, nous comparons la qualité des meilleures solutions faisables générées par les différentes méta-heuristiques. Sauf si une méthode n'arrive pas à trouver le nombre demandé de solutions faisables au bout d'un temps limité, on génère le même nombre de solutions faisables pour chaque méthode.

4.4.1 Pourcentage de faisabilité

Dans cette première partie, nous évaluons le pourcentage de permutations faisables générées par chacune des quatre méthodes au bout de 1000 solutions générées. Nous utilisons des jeux de données avec des temps opératoires de la première famille, et des pénalités de la troisième famille. Selon le mou utilisé, nous distinguons quatre catégories de fichiers de données : (10, 20, 40, 50, 80). Pour tous les fichiers de données utilisés, on a le nombre de job égal à dix, et le nombre de machines égal à cinq. On exécute les programmes trois fois pour chaque fichier et on donne le résultat moyen. Dans les deux types d'algorithmes génétiques testés, nous avons pris une taille de population égale à 30, et une probabilité de mutation de 0,2. Concernant l'algorithme génétique simple, la probabilité de recopier un parent est de 0,33. Les résultats sont donnés par les tableaux suivants :

Mou	Random	SA	AGE	AGS
10	0,93 %	2,43 %	17,9%	45,63 %
20	18,93 %	34,86 %	69,53%	69,63 %
40	8,5 %	20,13 %	53,83%	72,23 %
50	9,26 %	20,8 %	61,76%	71 %
80	61,03%	83,4 %	89,9%	91,03 %

Figure 4.3 degré de faisabilité des solutions.

Dans le tableau précédent, *Random* désigne la génération aléatoire, *SA* le recuit simulé, *AGS* l'algorithme génétique simple, et *AGE* pour l'algorithme génétique élitiste. Nous pouvons constater que l'algorithme génétique est clairement meilleur que le recuit simulé et la génération aléatoire du point de vue du pourcentage de création de solutions faisables. L'algorithme génétique parcourt, donc, l'espace de solution d'une manière plus intelligente que ne le fait le recuit simulé (en outre, on a vérifié de visu sur les déroulements numériques que les solutions générées par *AGS* et de l'*AGE* étaient diversifiées). On a donc plus de

chances d'explorer des solutions intéressantes avec l'AGS qu'avec le recuit simulé ou avec l'AGE.

4.4.2 Qualité des solutions trouvées

Dans cette deuxième partie des tests sur les métaheuristiques, nous évaluons la qualité des solutions trouvées par chacune des métaheuristiques. Pour cela, nous générons quatre fichiers de données tel que les durées d'exécutions des jobs sur les machines appartiennent à la première famille (pour les expériences faites pour le mémoire provisoire), et les pénalités appartiennent à la troisième famille (idem). Le nombre de jobs est dix, et le nombre de machines est cinq. Nous arrêtons l'exécution lorsqu'on a créé 300 solutions faisables, ou bien si l'on arrive à une certaine limite de temps (génération de 30000 permutations, faisables et infaisables confondues). Nous répétons l'exécution de chaque méthode cinq fois pour chaque instance, et nous retenons à chaque fois, et à chaque instant, la meilleure solution.

Les résultats sont résumés sur les figures suivantes.

Instance 1 : les meilleures solutions sont trouvées par l'AGE et SA pour une somme de pénalité d'avance retard SPAR égale à 3819, ce qui donne un gain de 0.01% par rapport à la solution trouvée par l'AGS (3891). Il faut noter toutefois que SA ne trouve pas 300 individus faisables.

Instance 2 : la meilleure solution est trouvée par SA (SPAR = 1395), mais celle trouvée par AGE est très proche (1396). Le gain par rapport à AGS est de 0.01% (SPAR = 1418).

Instance 3 : la meilleure solution est trouvée par AGE (SPAR = 1559), avec un gain de 0.01% par rapport à SA, et de 0.05% par rapport à AGS.

Instance 4 : la meilleure solution est trouvée par AGS (SPAR = 2064), avec un gain de 0.01% par rapport à SA, et de 0.05% par rapport à AGE.

Les figures suivantes illustrent les résultats obtenus.

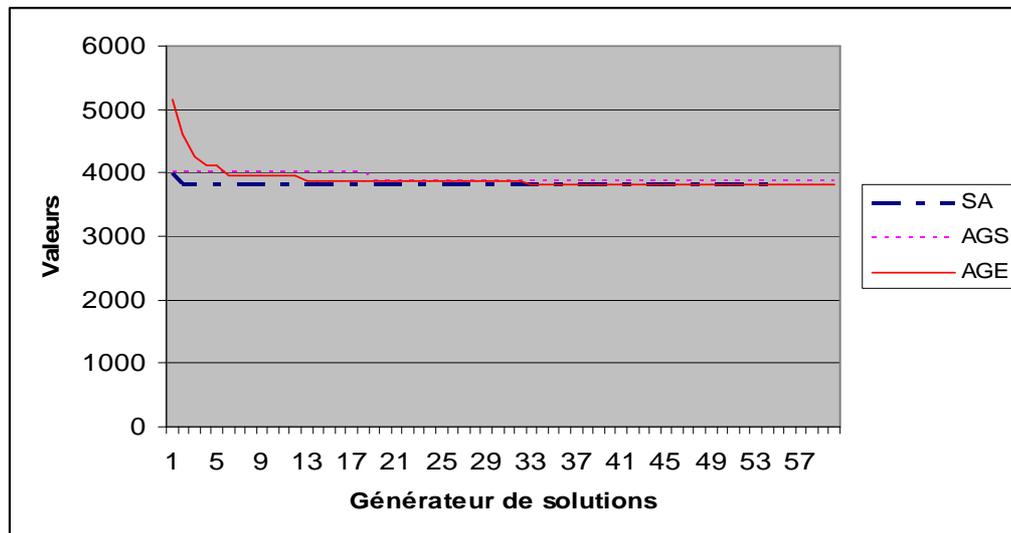


Figure 4.4 SA Vs AGS Vs AGE ; instance 1.

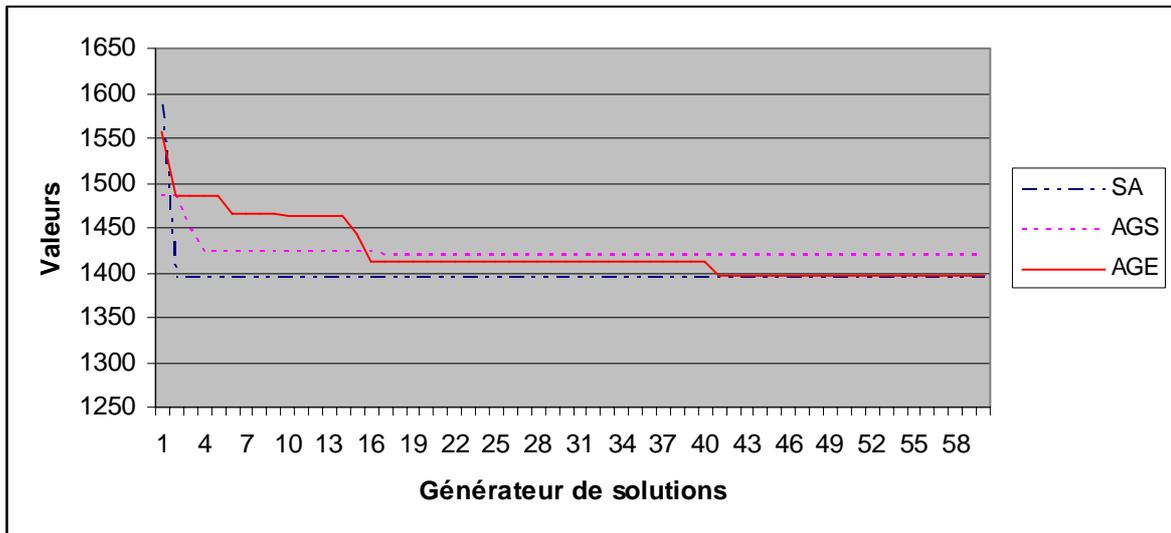


Figure 4.5 SA Vs AGS Vs AGE ; instance 2.

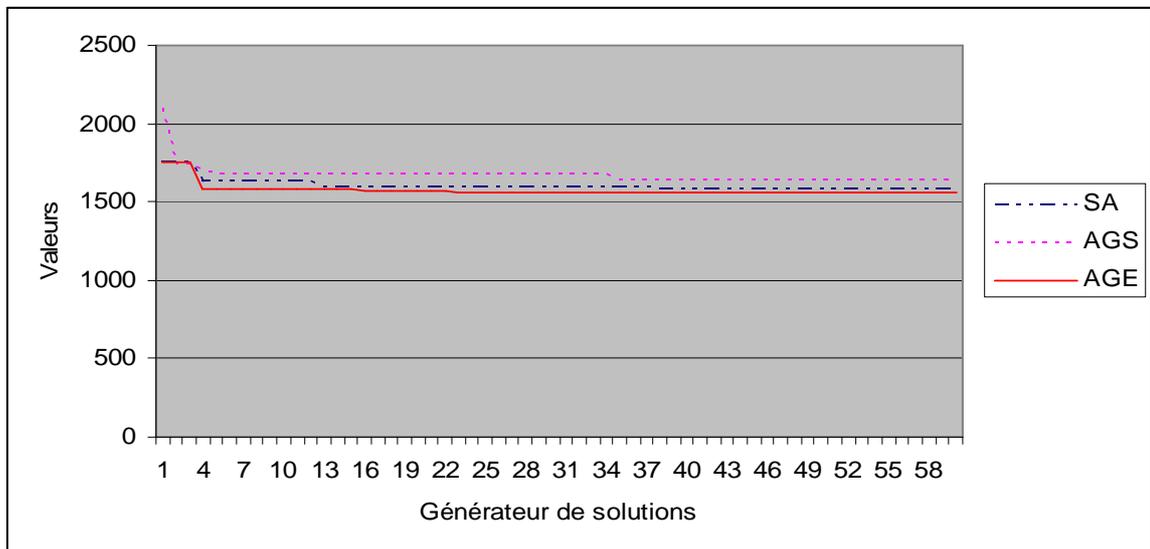


Figure 4.6 SA Vs AGS Vs AGE ; instance 3.

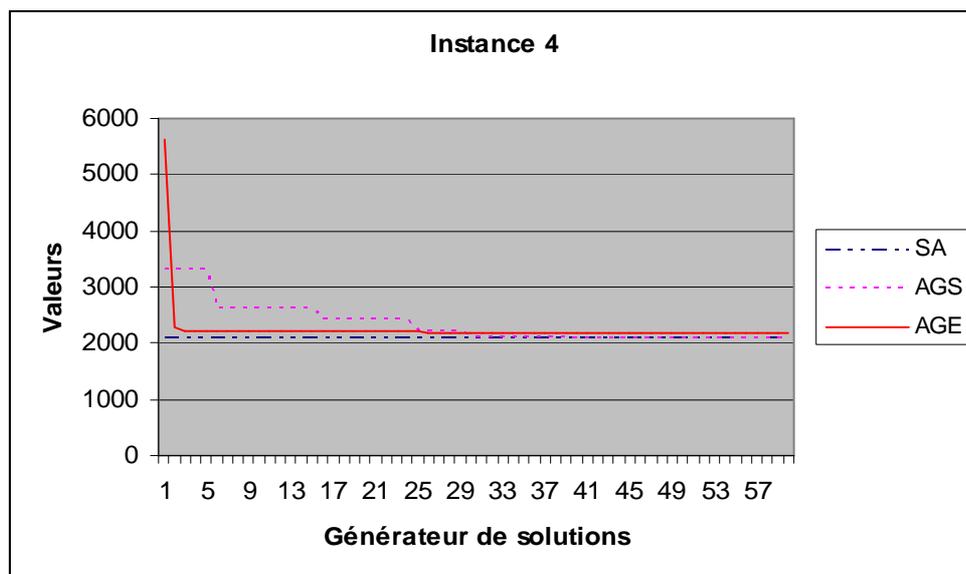


Figure 4.7 A Vs AGS Vs AGE ; instance 4.

Les expériences menées nous montrent que les trois métaheuristiques se valent. Il n'y a pas une qui est franchement meilleure qu'une autre. Toutefois, nous n'avons pas menés assez d'expériences (contrainte de temps) pour affirmer une telle conclusion. Nous pensons que si on mène des expériences plus longtemps, nous pourrions probablement les départager. Néanmoins l'objectif de la thèse était de montrer que l'on peut traiter, en un temps raisonnable, des problèmes d'ordonnement pour organiser des négociations dans la chaîne logistique en croisant une métaheuristique et une technique de PERT COUT et cet objectif est atteint.

4.4.3 Evaluation des opérateurs génétiques

Dans cette section, nous mettrons les résultats de la comparaison des opérateurs développés pour le schéma élitiste de l'algorithme génétique (nouvelle expérience à venir).

4.5 Conclusion

Nous avons menés dans ce chapitre des expériences pour évaluer les méthodes de résolution proposées pour les problèmes de juste à temps définis dans les chapitres précédents. Nous n'avons pas pu mener toutes les expériences nécessaires pour pouvoir juger d'une manière irréfutable la qualité de chaque méthode. Des expériences qui ont pu être menées, il en ressort que :

1. L'évaluation du Pert coût nous permet d'avoir l'assurance de trouver les solutions optimales par rapport à une séquence donnée de traitement des travaux. Ceci est fait en un temps relativement court, ce qui est pratique pour une utilisation en milieu réel. En outre, notre méthode de calcul du pas de compression a permis d'avoir un nombre d'itérations du Pert coût relativement petit et très loin de la borne supérieure définie la section 3.3.4.
2. L'utilisation des métaheuristiques nous permet de parcourir l'espace de solutions d'une manière intelligente. Les deux algorithmes génétiques utilisés donnent plus de chances de trouver des solutions faisables par rapport au recuit simulé, mais ceci n'assure pas de trouver de meilleures solutions. En effet, les trois méthodes testées se valent.

Il serait très intéressant de poursuivre la série des expériences sur les métaheuristiques avec des temps d'exécutions plus longs, et sur les différents types de données que l'on peut générer grâce à notre méthode de génération de données décrite dans la première section du présent chapitre.

5 Chapitre 5 : problèmes de transports dans chaînes logistiques

Résumé

Dans ce chapitre, nous étudions un ensemble de problèmes d'ordonnements dans des chaînes logistiques spécifiques. Nous étudions trois grands problèmes pour lesquels nous déduisons des cas spécifiques. Nous nous intéressons ici principalement à l'optimisation des coûts de transports, ainsi qu'aux moyens de mettre en œuvre la coopération entre partenaires, ou bien entre deux différentes fonctions de la chaîne. Nous proposons des méthodes de résolutions exactes pour la plupart des problèmes considérés.

5.1 Introduction

Comme nous l'avons présenté au chapitre consacré à l'état de l'art, le traitement des décisions opérationnelles au niveau des chaînes logistiques est très peu étudié dans la littérature. Beaucoup plus d'importance est accordée aux problèmes des niveaux stratégiques et tactiques. Néanmoins, dans l'industrie le lot de problèmes quotidiens rencontrés au niveau opérationnel est très grand, et peu de solutions sont proposées à ce niveau de décision. L'absence d'optimalité des décisions opérationnelles peut avoir des conséquences néfastes sur la performance globale de la chaîne. Par exemple, une mauvaise exploitation des ressources humaines et matérielles au jour le jour peut engendrer des coûts de fonctionnements énormes, allonger les délais de livraison, amoindrir la qualité des produits fabriqués, et ainsi abaisser la compétitivité de l'entreprise. C'est dans ce contexte que nous nous sommes intéressés à la résolution de certains problèmes à au niveau opérationnel de la chaîne logistique.

Dans ce chapitre, nous développons des techniques d'ordonnancement pour différents problèmes pouvant arriver dans la chaîne logistique afin d'optimiser différents critères. Nous étudions trois grands problèmes.

En premier, nous étudions un problème dans la chaîne logistique qui met en œuvre la coopération entre fournisseurs pour satisfaire les demandes d'un client. Les fournisseurs coopèrent pour optimiser le coût total de possession de stocks chez l'ensemble des fournisseurs. Nous commençons d'abord par un problème simplifié avec un nombre réduit de fournisseurs et de types de produits fabriqués, pour ensuite généraliser à un nombre quelconque de fournisseurs et de produits fabriqués. L'objectif est de calculer les quantités de chaque produit que doit fabriquer chacun des fournisseurs (Mouloua et Oulamara, 2006).

Dans le deuxième problème, nous étudions les problèmes de transport dans une chaîne logistique élémentaire à deux niveaux. Le premier niveau est un système de production simplifié, et le deuxième niveau est le système de transport. En amont de cette chaîne on a des courbes cumulées d'arrivées des composants des fournisseurs, et en aval on a des courbes cumulées des demandes de livraisons des clients. Dans un premier temps nous proposons un algorithme pour décider de la faisabilité ou pas d'un problème donné. Nous ajoutons ensuite des contraintes sur les capacités des transports. Nous proposons des algorithmes polynomiaux pour l'optimisation des coûts de transport en minimisant le nombre total de trajets effectués entre le producteur et ses clients (Mouloua et Portmann, 2006).

Dans le troisième problème nous traitons trois cas d'optimisation dans une chaîne logistique à deux niveaux (production et transport) avec des clients en aval et des courbes cumulées de demandes de livraisons, et des fournisseurs en amont avec des courbes cumulées d'arrivées des matières premières. Nous supposons dans les deux premiers cas que le planning de production est fixé à l'avance et est optimal. Relativement à ce planning de production, nous proposons un algorithme de programmation dynamique qui minimise les coûts de transports en minimisant le nombre de trajets effectués entre le producteur et ses clients (Mouloua et Oulamara, 2007). Dans le premier cas on ne considère qu'un seul type de produit, une généralisation à plusieurs types de produits est étudiée dans le deuxième cas. Dans le troisième cas on traite un problème de coopération entre les fonctions production et transport de la chaîne logistique. En effet, nous développons un algorithme à base de programmation dynamique pour l'optimisation conjointe et simultanée des coûts de production et de transport.

5.2 Problème 1 : La coopération entre fournisseurs pour minimiser les coûts de possession des stocks

5.2.1 Description du problème

Nous traitons un problème d'ordonnancement dans une chaîne logistique où plusieurs fournisseurs (ou sites de production) coopèrent pour assurer une production, d'un ou de plusieurs types de produits, satisfaisant les besoins d'un client. Le client demande une certaine quantité pour chaque type de produits dont il a besoin, les fournisseurs du client coopèrent entre eux pour décider qui produit combien en faisant en sorte de livrer le client en quantité demandées et dans les plus brefs délais. Les produits sont regroupés en batches et transportés au client. L'objectif est de minimiser le coût total de possession des produits chez l'ensemble des fournisseurs ainsi que les coûts de livraisons. Les batches ne contiennent que des produits (items) du même type. Le temps de fin d'exécution d'un item est le temps de fin d'exécution du dernier item appartenant au batch. Le temps d'exécution d'un batch est proportionnel au nombre d'items le composant. On associe un coût de possession à chaque item depuis le temps de traitement du premier item du même produit et jusqu'à la livraison de l'item en question. Cette dernière hypothèse implique que l'ordre dans lequel on traite les produits n'influe pas sur la fonction objective, donc les produits sont indépendants les uns des autres.

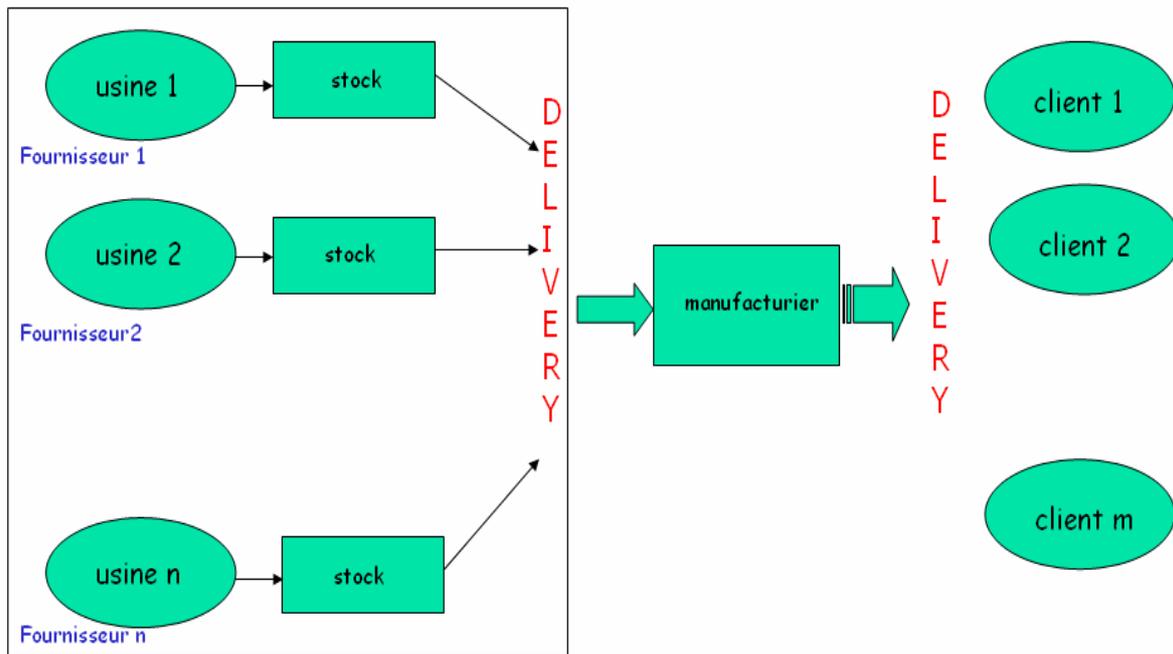


Figure 5.1 chaîne logistique considérée.

5.2.2 Hypothèses et Notations

5.2.2.1 Hypothèses :

Pour ce problème, nous posons les hypothèses suivantes :

- i. Tous les items des matières premières sont disponibles à l'instant $t = 0$ pour le cas mono produit.

- ii. Le temps d'exécution d'un batch dépend de sa taille (du nombre d'items qu'il contient).
- iii. Un batch contient uniquement des items du même produit. Séries de productions ininterrompues et pas de temps morts internes aux batches.
- iv. Les livraisons au client se font en juste à temps, un trajet est effectué pour chaque batch.
- v. Un coût de possession est associé à chaque item depuis le temps de production du premier item du produit, et jusqu'à la livraison du batch auquel appartient l'item. On suppose donc, qu'on dispose de toutes les matières premières nécessaires juste avant le début de production de chaque produit. La matière première arrive donc en juste à temps.
- vi. Le client ne paye les produits qu'à la date convenue, si les produits arrivent plutôt que prévu, alors les fournisseurs doivent attendre avant d'être payés.
- vii. Chaque fournisseur ne peut exécuter qu'une seule tâche à la fois.
- viii. La production se fait sur un horizon de temps H .

5.2.2.2 Notations

Nous utilisons les données et variables suivantes :

Données

N : Le nombre de fournisseurs.

M : Le nombre de produits.

W : Quantité totale demandée par le client.

p_i^j : Temps d'exécution chez le fournisseur i du produit j .

d_{ij} : Coût de livraison d'un batch du produit j du fournisseur i .

n_i^j : Nombre de batches du fournisseur i du produit j .

h_i : Coût de possession d'un item chez le fournisseur i pendant une unité de temps.

Variables

W_i^j : Quantité à produire par le fournisseur i du produit j .

b_{ij}^k : Taille du $k^{\text{ème}}$ batch du produit j du fournisseur i .

C_{ij}^k : Temps de fin d'exécution du $k^{\text{ème}}$ batch du produit j du fournisseur i .

La fonction objective exprime la somme des coûts de possession des stocks et des coûts de livraisons. Comme un coût de possession est associé à chaque item tout au long de son traitement (le traitement du batch auquel il appartient), alors la fonction objective est définie par :

TC : Coût total de possession et de livraison chez l'ensemble des fournisseurs

$$\sum_{j=1}^M \sum_{i=1}^N \sum_{k=1}^{n_{ij}} h_i b_{ij}^k C_{ij}^k + \sum_{j=1}^M \sum_{i=1}^N n_i^j . d_{ij}$$

Les deux figures suivantes montrent deux ordonnancements arbitraires chez deux fournisseurs.

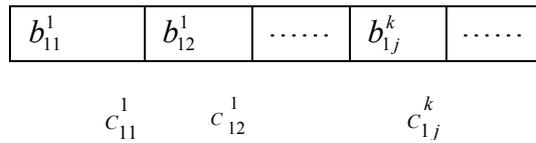


Figure 5.2 ordonnancement du fournisseur 1.

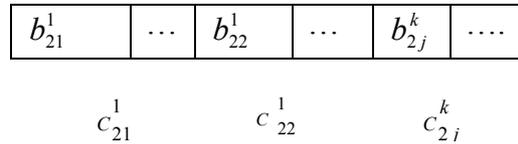


Figure 5.3. Ordonnancement du fournisseur 2.

Comme le nombre de batches est fixé et connu à l’avance, la partie relative aux coûts de livraisons dans la fonction objective est constante. Elle sera donc omise dans les calculs qui suivent. Pour des raisons de clarté et de simplicité, nous n’utiliserons pas l’indice j quand il s’agit de problèmes à un seul produit.

5.2.3 Le problème Mono Produit

Dans cette section, nous considérons d’abord le problème à un seul type de produit avec deux fournisseurs, ensuite, nous généralisons au cas de plusieurs fournisseurs. Pour chaque problème, nous recherchons la valeur de W_i , le nombre d’items que chaque fournisseur doit fabriquer, dans le but de minimiser le coût total. Il est évident que pour le cas d’un seul fournisseur, on a $W_1 = W$. Dans ce cas, le seul problème est de trouver la taille b_1^k du $k^{\text{ème}}$ batch. Pour déterminer cette quantité, Selvarajah et Steiner (Selvarajah et Steiner, 2006) utilisent la “almost-equal-batch-size policy” (politique de taille de batches Presque égales), i.e. si W items sont groupés en n batches de taille entière égale à b^k tel que $b^k \in \{x, x-1\}$ pour $k = 1, \dots, n$, alors $x = \left\lceil \frac{W}{n} \right\rceil$. La preuve de l’optimalité de cette politique est donnée par (Selvarajah et Steiner, 2006).

5.2.3.1 Le problème à 2 fournisseurs

Données

W : Quantité totale demandée par le client

$N = 2$,

$M = 1$,

p_i : Temps d’exécution chez le fournisseur i ,

d_i : Coût de livraison d’un batch par le fournisseur i ,

n_i : Nombre de batchs du fournisseur i ,

h_i : Coût de possession d’un item pendant une unité de temps par le fournisseur i ,

Variables

W_i : Quantité à produire par le fournisseur i ,

b_i^k : Taille du $k^{\text{ème}}$ batch du fournisseur i ,

C_i^k : Temps de fin d'exécution du $k^{\text{ème}}$ batch du fournisseur i ,

La fonction objective est donnée par :

$$TC = \sum_{k=1}^{n_1} h_1 \cdot b_1^k \cdot C_1^k + \sum_{k=1}^{n_2} h_2 \cdot b_2^k \cdot C_2^k$$

La figure suivante illustre ce cas :

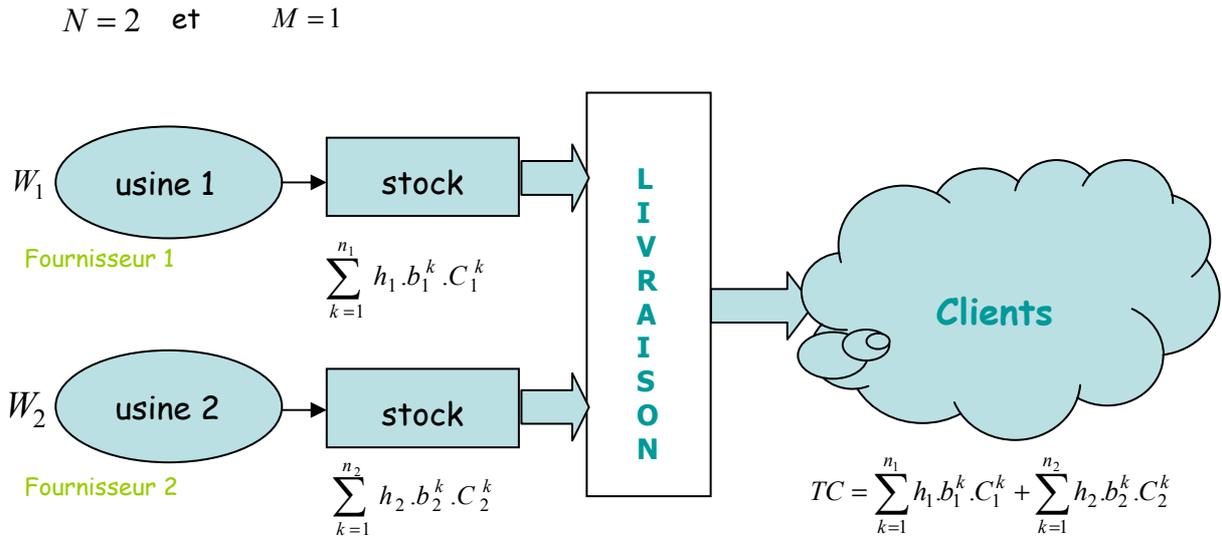


Figure 5.4 cas de deux fournisseurs.

Sans perdre en généralité, nous supposons que $h_1 = h_2 = h$, alors

$$TC = h \cdot \left(\sum_{k=1}^{n_1} b_1^k \cdot C_1^k + \sum_{k=1}^{n_2} b_2^k \cdot C_2^k \right)$$

Lemme 1. Les quantités optimales que chaque fournisseur doit produire pour minimiser le coût total, pour le problème 2 fournisseurs- mono produit sont données par :

$$W_1 = \frac{h_2 p_2 \cdot W \cdot (n_2 + 1) \cdot n_1}{h_1 p_1 \cdot n_2 \cdot (n_1 + 1) + h_2 p_2 \cdot n_1 \cdot (n_2 + 1)}$$

$$W_2 = \frac{h_1 p_1 \cdot W \cdot (n_1 + 1) \cdot n_2}{h_1 p_1 \cdot n_2 \cdot (n_1 + 1) + h_2 p_2 \cdot n_1 \cdot (n_2 + 1)}$$

Preuve. Nous remarquons dans la fonction objective que h est en facteur, de ce fait, nous pouvons l'omettre dans les calculs. Nous avons donc :

$$\sum_{k=1}^{n_1} b_1^k \cdot C_1^k = b_1^1 \cdot C_1^1 + b_1^2 \cdot C_1^2 + b_1^3 \cdot C_1^3 + \dots + b_1^{n_1} \cdot C_1^{n_1}$$

$$\sum_{k=1}^{n_1} b_1^k \cdot C_1^k = b_1^1 \cdot b_1^1 \cdot p_1 + b_1^2 \cdot (b_1^1 + b_1^2) \cdot p_1 + b_1^3 \cdot (b_1^1 + b_1^2 + b_1^3) \cdot p_1 + \dots + b_1^{n_1} \cdot (b_1^1 + b_1^2 + b_1^3 + \dots + b_1^{n_1}) \cdot p_1$$

$$\begin{aligned} \sum_{k=1}^{n_1} b_1^k \cdot C_1^k &= [(b_1^1)^2 + (b_1^2)^2 + (b_1^3)^2 + \dots + (b_1^{n_1})^2 + b_1^2 b_1^1 + b_1^3 (b_1^1 + b_1^2) + b_1^4 (b_1^1 + b_1^2 + b_1^3) + \dots \\ &\quad + b_1^{n_1} (b_1^1 + b_1^2 + b_1^3 + \dots + b_1^{n_1-1})] \cdot p_1 \\ &= p_1 \left(\frac{\left(\sum_{k=1}^{n_1} b_1^k \right)^2}{2} + \frac{\sum_{k=1}^{n_1} (b_1^k)^2}{2} \right) \end{aligned}$$

Alors

$$\begin{aligned} TC &= p_1 \left(\frac{\left(\sum_{k=1}^{n_1} b_1^k \right)^2}{2} + \frac{\sum_{k=1}^{n_1} (b_1^k)^2}{2} \right) + p_2 \left(\frac{\left(\sum_{k=1}^{n_2} b_2^k \right)^2}{2} + \frac{\sum_{k=1}^{n_2} (b_2^k)^2}{2} \right) \\ TC &= p_1 \left(\frac{W_1^2}{2} + \frac{\sum_{k=1}^{n_1} (b_1^k)^2}{2} \right) + p_2 \left(\frac{W_2^2}{2} + \frac{\sum_{k=1}^{n_2} (b_2^k)^2}{2} \right) \\ TC &= \frac{p_1 W_1^2 + p_2 W_2^2}{2} + p_1 \frac{\sum_{k=1}^{n_1} (b_1^k)^2}{2} + p_2 \frac{\sum_{k=1}^{n_2} (b_2^k)^2}{2} \end{aligned} \quad (5.1)$$

En suivant la technique d'équilibrage de batches, on aura :

$$b_1^k = x = \left\lceil \frac{W_1}{n_1} \right\rceil$$

$$b_2^k = y = \left\lceil \frac{W_2}{n_2} \right\rceil$$

Chez le premier fournisseur on aura m_1 batches de taille x et $(n_1 - m_1)$ batches de taille $(x-1)$

Chez le second fournisseur on aura m_2 batches de taille y et $(n_2 - m_2)$ batches de taille $(y-1)$

Ce qui donne la fonction objective suivante

$$TC = \frac{p_1 \cdot W_1^2 + p_2 \cdot W_2^2}{2} + p_1 \left[\frac{m_1 \cdot x^2 + (n_1 - m_1) \cdot (x-1)^2}{2} \right] + p_2 \left[\frac{m_2 \cdot y^2 + (n_2 - m_2) \cdot (y-1)^2}{2} \right]$$

On peut écrire W_1 et W_2 sous la forme

$$W_1 = a.n_1 + b$$

$$W_2 = c.n_2 + d$$

$$tq : a \geq 1 \text{ et } c \geq 1 \text{ et } 0 \leq b < n_1 \text{ et } 0 \leq d < n_2$$

On peut remarquer que : $b = m_1$ et $d = m_2$, nous pouvons donc calculer x et y .

$$x = \left\lceil \frac{W_1}{n_1} \right\rceil = \left\lceil \frac{a.n_1 + b}{n_1} \right\rceil = \left\lceil a + \frac{b}{n_1} \right\rceil = a + \left\lceil \frac{b}{n_1} \right\rceil = a + 1 \quad \text{car } b < n_1. \quad (5.2)$$

$$y = \left\lceil \frac{W_2}{n_2} \right\rceil = \left\lceil \frac{c.n_2 + d}{n_2} \right\rceil = \left\lceil c + \frac{d}{n_2} \right\rceil = c + \left\lceil \frac{d}{n_2} \right\rceil = c + 1 \quad \text{car } d < n_2. \quad (5.3)$$

En remplaçant (5.2) et (5.3) dans (5.1) nous obtenons :

$$TC = \frac{p_1.W_1^2 + p_2.W_2^2}{2} + p_1 \left[\frac{n_1.x^2}{2} \right] + p_2 \left[\frac{n_2.y^2}{2} \right]$$

$$TC = \frac{p_1.W_1^2 + p_2.W_2^2}{2} + p_1 \left[\frac{m_1.(a+1)^2 + (n_1 - m_1).a^2}{2} \right] + p_2 \left[\frac{m_2.(c+1)^2 + (n_2 - m_2).c^2}{2} \right]$$

$$TC = \frac{1}{2} \left[p_1.a^2.n_1.(n_1+1) + p_1.2.a.b.(n_1+1) + p_1.b^2 + p_1.b + p_2.c^2.n_2.(n_2+1) \right. \\ \left. + p_2.2.c.d.(n_2+1) + p_2.d^2 + p_2.d \right]$$

Nous utilisons la méthode de Lagrange pour trouver le minimum de la fonction TC sous la contrainte : $C(a,b,c,d) = a.n_1 + b + c.n_2 + d - W = 0$, la fonction à minimiser est $g = TC + \lambda.(a.n_1 + b + c.n_2 + d - W)$.

Alors

$$\begin{cases} \frac{\partial g}{\partial a} = p_1.a.n_1.(n_1+1) + p_1.b.(n_1+1) + \lambda.n_1 = 0 \\ \frac{\partial g}{\partial b} = p_1.a.(n_1+1) + \frac{p_1}{2} + p_1.b + \lambda = 0 \\ \frac{\partial g}{\partial c} = p_2.c.n_2.(n_2+1) + p_2.d.(n_2+1) + \lambda.n_2 = 0 \\ \frac{\partial g}{\partial d} = p_2.c.(n_2+1) + \frac{p_2}{2} + p_2.d + \lambda = 0 \\ \frac{\partial g}{\partial \lambda} = a.n_1 + b + c.n_2 + d - W = 0 \end{cases}$$

C'est un système d'équations avec 5 variables et 5 équations. La résolution du système d'équations donne les résultats suivants :

$$\begin{cases} a = \frac{p_2 \cdot W \cdot (n_2 + 1) - p_1 \cdot n_2 \cdot (n_1 + 1) - p_2 \cdot n_1 \cdot (n_2 + 1)}{p_1 \cdot n_2 \cdot (n_1 + 1) + p_2 \cdot n_1 \cdot (n_2 + 1)} \\ b = n_1 \\ c = \frac{1}{n_2} \left(\frac{p_1 \cdot W \cdot (n_1 + 1) \cdot n_2}{p_1 \cdot n_2 \cdot (n_1 + 1) + p_2 \cdot n_1 \cdot (n_2 + 1)} - \frac{n_2}{2} \right) \\ d = \frac{n_2}{2} \end{cases}$$

Ce qui donne:

$$W_1 = a \cdot n_1 + b = \frac{p_2 \cdot W \cdot (n_2 + 1) \cdot n_1}{p_1 \cdot n_2 \cdot (n_1 + 1) + p_2 \cdot n_1 \cdot (n_2 + 1)}$$

$$W_2 = c \cdot n_2 + d = \frac{p_1 \cdot W \cdot (n_1 + 1) \cdot n_2}{p_1 \cdot n_2 \cdot (n_1 + 1) + p_2 \cdot n_1 \cdot (n_2 + 1)}$$

Dans le cas où h_1 et h_2 sont quelconques, par le changement de variables suivant $h_1 p_1 = p_1$, on retombe sur exactement le même système que dans le cas où $h_1 = h_2$. On aura donc les formules suivantes :

$$W_1 = a \cdot n_1 + b = \frac{h_2 p_2 \cdot W \cdot (n_2 + 1) \cdot n_1}{h_1 p_1 \cdot n_2 \cdot (n_1 + 1) + h_2 p_2 \cdot n_1 \cdot (n_2 + 1)}$$

$$W_2 = c \cdot n_2 + d = \frac{h_1 p_1 \cdot W \cdot (n_1 + 1) \cdot n_2}{h_1 p_1 \cdot n_2 \cdot (n_1 + 1) + h_2 p_2 \cdot n_1 \cdot (n_2 + 1)}$$

5.2.3.2 Le problème à N fournisseurs

Dans ce cas, nous considérons le problème où N fournisseurs coopèrent pour satisfaire la demande du client pour un seul type de produits, (i.e. $N = n, M = 1$). La fonction objective est donnée par :

$$TC = \frac{p_1 W_1^2 + p_2 W_2^2 + \dots + p_n W_n^2}{2} + p_1 \frac{\sum_{k=1}^{n_1} (b_1^k)^2}{2} + p_2 \frac{\sum_{k=1}^{n_2} (b_2^k)^2}{2} + \dots + p_n \frac{\sum_{k=1}^{n_n} (b_n^k)^2}{2}$$

Elle exprime la somme des coûts de possession de stocks (encours, matières premières, et produits finis) chez tous les fournisseurs. Une telle approche serait envisageable si les fournisseurs appartiennent à la même entreprise, ou centre de décision, ou bien qu'il y ait un contrat entre eux (stratégie gagnant gagnant pour tout le monde).

Lemme 2. Les quantités optimales que chaque fournisseur doit produire pour le problème N fournisseurs- mono produit sont données par :

$$W_i = \frac{W \cdot n_i \cdot \prod_{j=1 \neq i}^n h_j p_j \cdot (n_j + 1)}{\sum_{s=1}^n n_s \cdot \prod_{j=1 \neq s}^n h_j p_j \cdot (n_j + 1)}$$

Preuve. La preuve est similaire à celle du Lemme 1. On aura un système linéaire à $(2N + 1)$ variables, et $(2N + 1)$ équations qu'on résout de la même manière en utilisant la méthode de Lagrange.

5.2.3.3 Heuristique pour rendre les W_i entiers

Pour chaque fournisseur, le nombre d'items à fabriquer, donné par le lemme 2, est une valeur réelle, et donc ne peut pas être appliqué directement car on ne peut pas diviser un item. Nous devons donc trouver les valeurs entières optimales pour les W_i . Ceci est fait en appliquant l'algorithme qui suit.

Soit $F(W_i, S_i) = h_i p_i (b_{n_i} + 1 + \lfloor W_i \rfloor)$ la fonction qui permet de calculer le nouveau coût de possession chez le fournisseur S_i , si un nouvel item est produit par S_i . Initialement, le nombre d'items à produire par chaque fournisseur est donné par la partie entière inférieure de la quantité calculée par le lemme 2, c'est à dire $\lfloor W_i \rfloor$. Dans l'algorithme A ci-dessous, nous calculons $F(W_i, S_i)$ pour chaque fournisseur, et nous trions les fournisseurs dans l'ordre croissant de $F(W_i, S_i)$ dans la liste L . On ajoute un item pour les $W - \sum_{i=1}^n \lfloor W_i \rfloor$ premiers fournisseurs dans la liste triée L , tel que $W - \sum_{i=1}^n \lfloor W_i \rfloor \leq n$.

Algorithme A

Début

étape 1 : trier les fournisseurs dans l'ordre croissant de $F(W_i, S_i)$.

étape 2 : Init $y_i^ = \lfloor W_i \rfloor$ pour $i = 1, \dots, n$.*

$$\text{Calculer } y = \sum_{i=1}^n y_i^*$$

étape 3 : Pour $i = 1 \dots W - y$ Faire

$$y_{s[i]}^* = y_{s[i]}^* + 1$$

Fin

Dans la première étape de l'algorithme A, L'opération de tri se fait en $O(n \log n)$. Dans les étapes 2 et 3 Nous accomplissons $W - y$ opérations, La complexité de ces deux étapes est donc de $O(n)$. La complexité totale de l'algorithme est de $O(n \log n)$.

5.2.4 Le problème Multi Produits

Dans cette section, nous traitons le problème avec plusieurs types de produits à fabriquer. Nous commençons par résoudre le problème à 2 fournisseurs- 2 types de produits, ensuite, 2 fournisseurs- 3 produits, ensuite nous traitons le cas le plus général à N fournisseurs- M produits. Pour chacun de ces problèmes, nous rappelons les données et les variables.

5.2.4.1 Le problème à 2 fournisseurs- 2 produits

Données

$$N = 2$$

$$M = 2$$

p_i^j : Temps d'exécution chez le fournisseur i d'un item du produit j .

n_i^j : Nombre de batches du fournisseur i du produit j .

h_i : Coût de possession d'un item pendant une unité de temps chez le fournisseur i .

W^j : Quantité totale demandée du produit j .

Variables

W_i^j : Quantité à produire du produit j par le fournisseur i .

b_{ij}^k : Taille du $k^{\text{ème}}$ batch du produit j du fournisseur i .

C_{ij}^k : Temps de fin d'exécution du $k^{\text{ème}}$ batch du produit j du fournisseur i .

La fonction objective est donnée par :

$$TC = \sum_{k=1}^{n_1^1} h_1 b_{1,1}^k C_{1,1}^k + \sum_{k=1}^{n_1^2} h_1 b_{1,2}^k C_{1,2}^k + \sum_{k=1}^{n_2^1} h_2 b_{2,1}^k C_{2,1}^k + \sum_{k=1}^{n_2^2} h_2 b_{2,2}^k C_{2,2}^k$$

$$TC = \sum_{k=1}^{n_1^1} h_1 b_{1,1}^k C_{1,1}^k + \sum_{k=1}^{n_1^2} h_2 b_{2,1}^k C_{2,1}^k + \sum_{k=1}^{n_2^1} h_1 b_{1,2}^k C_{1,2}^k + \sum_{k=1}^{n_2^2} h_2 b_{2,2}^k C_{2,2}^k$$

$$TC = h_1 \frac{p_1^1 W_1^{12}}{2} + h_1 p_1^1 \frac{\sum_{k=1}^{n_1^1} (b_{1,1}^k)^2}{2} + h_2 \frac{p_2^1 W_2^{12}}{2} + h_2 p_2^1 \frac{\sum_{k=1}^{n_2^1} (b_{2,1}^k)^2}{2}$$

$$h_1 \frac{p_1^2 W_1^{22}}{2} + h_1 p_1^2 \frac{\sum_{k=1}^{n_1^2} (b_{1,2}^k)^2}{2} + h_2 \frac{p_2^2 W_2^{22}}{2} + h_2 p_2^2 \frac{\sum_{k=1}^{n_2^2} (b_{2,2}^k)^2}{2}$$

$$TC = TC_1 + TC_2$$

tg

$$TC_1 = h_1 \frac{p_1^1 W_1^{12}}{2} + h_1 p_1^1 \frac{\sum_{k=1}^{n_1^1} (b_{1,1}^k)^2}{2} + h_2 \frac{p_2^1 W_2^{12}}{2} + h_2 p_2^1 \frac{\sum_{k=1}^{n_2^1} (b_{2,1}^k)^2}{2}$$

$$TC_2 = h_1 \frac{p_1^2 W_1^{22}}{2} + h_1 p_1^2 \frac{\sum_{k=1}^{n_1^2} (b_{1,2}^k)^2}{2} + h_2 \frac{p_2^2 W_2^{22}}{2} + h_2 p_2^2 \frac{\sum_{k=1}^{n_2^2} (b_{2,2}^k)^2}{2}$$

On remarque que TC_1 et TC_2 sont les mêmes que la fonction objective dans le cas de deux fournisseurs et un seul produit. Trouver le minimum de TC revient à trouver les minimums de TC_1 et TC_2 . Ceci se fait de la même manière que dans le cas de 2 fournisseurs et mono produit. On aura donc pour chaque TC_j un système d'équations de 5 variables et 5 équations qu'on va résoudre en utilisant la méthode de Lagrange comme dans le cas mono produit.

Lemme 3. Les quantités optimales que chaque fournisseur doit produire pour le problème 2 fournisseurs- 2 produits sont données par :

Pour le produit 1

$$W_1^1 = \frac{h_2 p_2^1 W_1 n_1^1 (n_2^1 + 1)}{h_1 p_1^1 (n_1^1 + 1) n_2^1 + h_2 p_2^1 n_1^1 (n_2^1 + 1)}$$

$$W_2^1 = \frac{h_1 p_1^1 W_1 n_2^1 (n_1^1 + 1)}{h_1 p_1^1 (n_1^1 + 1) n_2^1 + h_2 p_2^1 n_1^1 (n_2^1 + 1)}$$

Pour le produit 2

$$W_1^2 = \frac{h_2 p_2^2 W_2 n_1^2 (n_2^2 + 1)}{h_1 p_1^2 (n_1^2 + 1) n_2^2 + h_2 p_2^2 n_1^2 (n_2^2 + 1)}$$

$$W_2^2 = \frac{h_1 p_1^2 W_2 n_2^2 (n_1^2 + 1)}{h_1 p_1^2 (n_1^2 + 1) n_2^2 + h_2 p_2^2 n_1^2 (n_2^2 + 1)}$$

Preuve. La preuve est similaire à celle du Lemme 1.

5.2.4.2 Le problème à N fournisseurs- M produits

Nous considérons ici le cas le plus général de N fournisseurs et M produits ($N = n$ et $M = m$). La fonction objective est définie par

$$TC = \sum_{j=1}^m TC_j \text{ tel que } TC_j = \sum_{i=1}^n \left(h_i \frac{p_i^j W_i^{j^2}}{2} + h_i p_i^j \frac{\sum_{k=1}^{n_i^j} (b_{i,j}^k)^2}{2} \right)$$

Lemme 5. Les quantités optimales que chaque fournisseur doit produire pour le problème N fournisseurs- M produits sont données par :

$$W_i^j = \frac{W^j n_i^j \prod_{k=1 \neq i}^n h_k p_k^j (n_k^j + 1)}{\sum_{s=1}^n n_s^j \prod_{k=1 \neq s}^n h_k p_k^j (n_k^j + 1)}$$

Preuve. A chaque TC_j correspond un système d'équation de $(2n + 1)$ variables et $(2n + 1)$ équations. En utilisant la même procédure de résolution que dans les cas précédents, on obtient la formule des quantités de chaque produit à produire par chaque fournisseur.

5.2.4.3 Algorithme pour rendre les W_i^j entiers

De la même façon que dans le cas mono produit, les résultats donnés par le lemme 5 sont des nombres réels, et donc pas directement applicables dans un environnement qui n'autorise pas de diviser les items de produits. Nous devons donc calculer les valeurs entières optimales pour les W_i^j .

Nous proposons un algorithme qui est une extension ou une généralisation de l'algorithme A. En effet, nous appliquons l'algorithme A pour chaque type de produits indépendamment parce que le nombre d'items que chaque fournisseur doit fabriquer de chaque produit ne

dépend pas du nombre d'items qu'il fabrique des autres types de produits. Comme la complexité du cas mono produit est de $O(n \log n)$, et que le nombre de types de produits est de m , alors le nouvel algorithme a une complexité de $O(m \times n \times \log n)$ (on exécute la cas mono produit m fois).

Algorithme B

Step 1 : Soit $F(W_i^j, S_i) = h_i p_i^j (b_{n_i^j} + 1 + \lfloor W_i^j \rfloor)$ la fonction qui définit le coût de l'ajout d'un item du produit j au fournisseur S_i .

Step 2 : Pour $j = 1, \dots, m$ faire

$$W^j = \sum_{i=1}^n W_i^j$$

Trier les fournisseurs dans l'ordre croissant de $F(W_i^j, S_i)$ dans le tableau S .

Init $y_i^{j*} = \lfloor W_i^j \rfloor$ pour $i = 1, \dots, n$

$$y^j = \sum_{i=1}^n y_i^{j*}$$

pour $i = 1$ *to* $W^j - y^j$ *faire*

$$y_{s[i]}^{j*} = y_{s[i]}^{j*} + 1$$

finPour

FinPour

End

5.3 Problème 2 : ordonnancement des transports dans une supply chain élémentaire

Nous traitons dans cette section différents problèmes simplifiés liés à la minimisation des coûts de transport dans les chaînes logistiques. Le schéma général de la chaîne logistique étudié est illustré par la figure 5.5

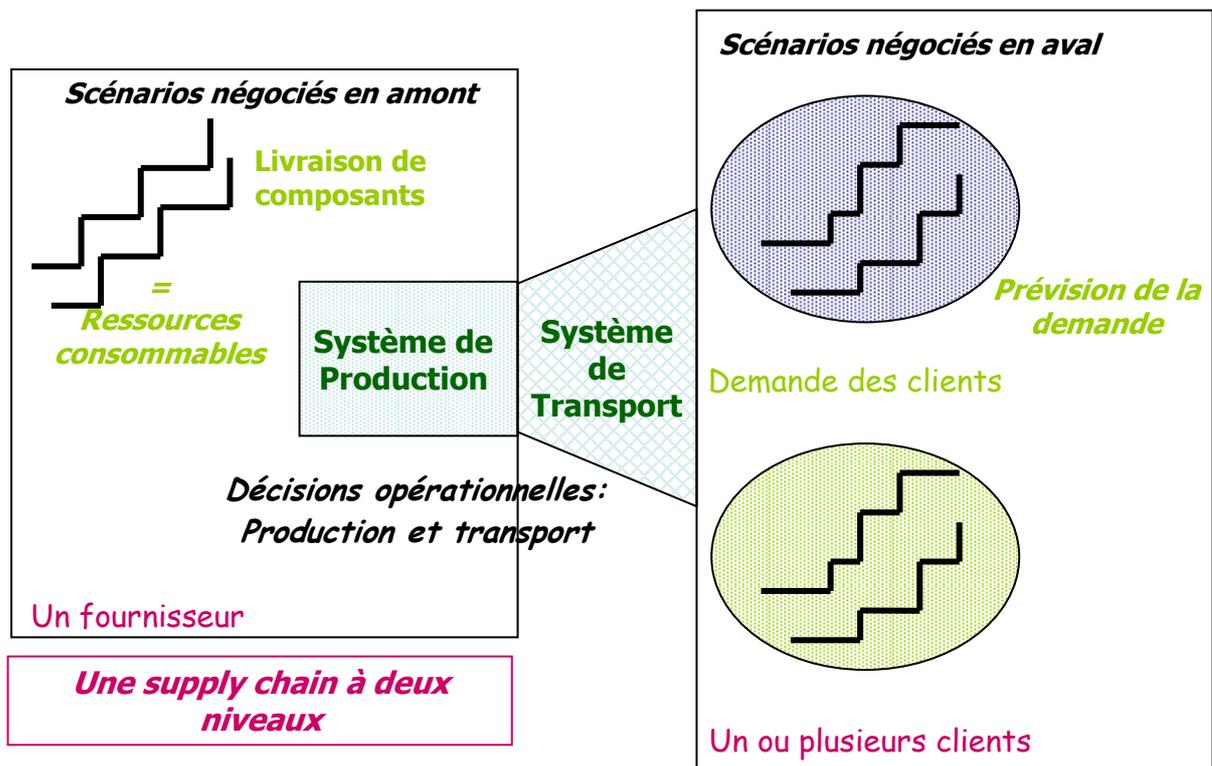


Figure 5.5 schéma général du système étudié.

Nous pouvons identifier quatre composantes de la chaîne logistique illustrée ci-dessus : les fournisseurs en composants, le système de production, le système des transport, et la prévision de la demande. Les clients effectuent des commandes avec des quantités précises et pour des dates de livraisons souhaitées. Ces commandes pouvant intervenir d'une manière continue dans le temps forment une courbe cumulée de la demande. Pour satisfaire ces commandes, le producteur effectue des demandes de livraisons de composants. De la même façon, ces commandes ont des quantités précises et arrivent à des dates convenues et forment une courbe cumulée d'arrivées de composants. Les composants doivent arriver suffisamment en avance et en quantités suffisantes pour permettre au producteur d'honorer les commandes des clients. Les produits finis sont transportés aux clients. Les coûts de transport sont généralement une partie importante des coûts de fonctionnement d'une chaîne logistique, nous nous intéressons ici à la minimisation du nombre de trajets entre le producteur et ses clients.

5.3.1 Description du problème mono fournisseur et mono produit

Nous traitons un problème de décision dans une chaîne logistique élémentaire constituée d'un fournisseur de composants et d'un producteur qui doit livrer un type de produits à ses clients. Sur un horizon H de temps, les composants dont a besoin le producteur arrivent à des dates connues en quantités connues, et le producteur doit livrer à des dates convenues en

quantités connues le produit au(x) client(s). Les arrivées de composants cumulées dans le temps constituent une courbe cumulée d'arrivées des composants, et les demandes de livraisons constituent une courbe cumulée de demandes de clients. La chaîne logistique simplifiée est illustrée par la figure 5.6.

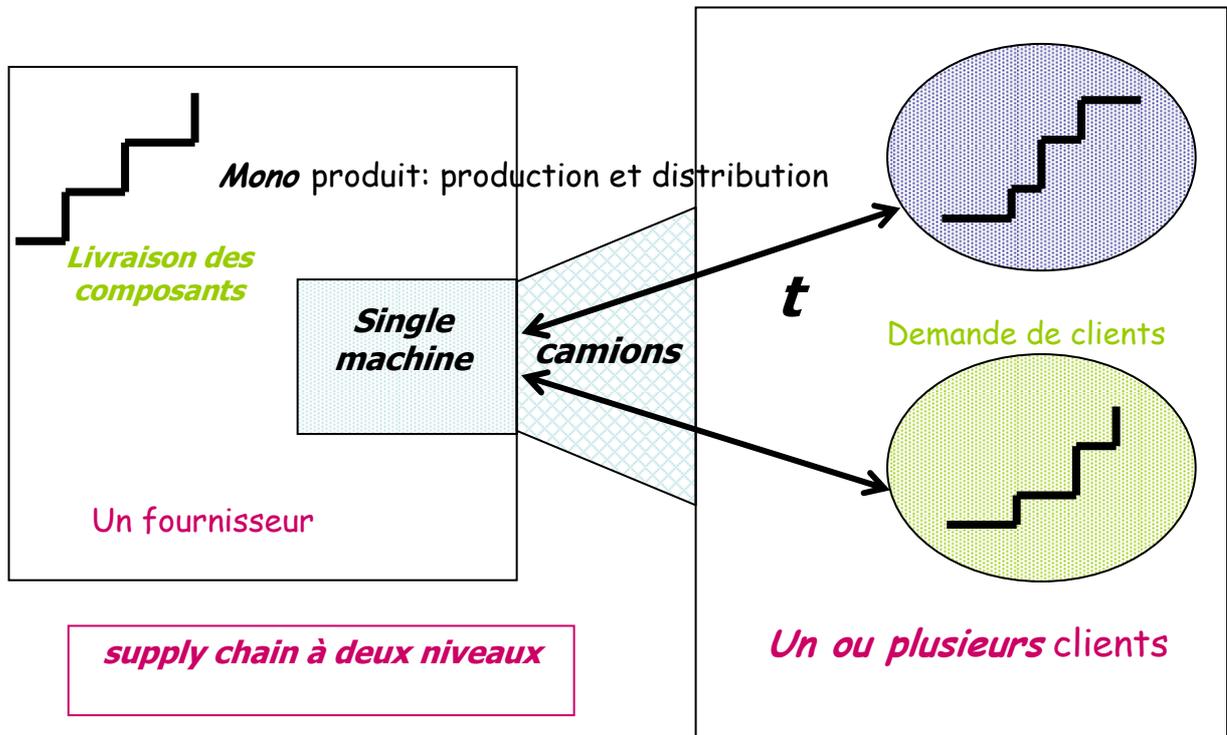


Figure 5.6 schéma simplifié mono produit.

Nous supposons qu'il n'y a qu'un seul type de produits finis, et qu'on a besoin de P unités de temps pour fabriquer un item du produit fini sur une seule machine. Cette machine est continuellement disponible (nous ne considérons pas ici les problèmes de maintenance) et ne peut traiter qu'un seul travail à la fois. Dans ce modèle, nous ne considérons pas les coûts de possession de stocks, en conséquence, un ordonnancement de la production calé gauche (produire au plus tôt, dès qu'il y a des composants disponibles) est optimal et est utilisé par le producteur. Notons que dans ce cas particulier, l'ordonnancement calé gauche est aussi optimal et polynomial dans le cas de plusieurs machines identiques qui fonctionnent en parallèle. Pour des raisons de simplicité et de clarté, nous ne présentons que le cas d'une seule machine. Il est évident qu'on arrête la production dès qu'on a atteint la quantité maximale demandée par les clients, c'est à dire qu'on ne fabrique pas plus que la demande.

Nous considérons un temps de transport constant et donné entre le producteur et le client (ou plusieurs clients ayant approximativement la même localisation géographique) égal à t unités de temps. Ce temps de transport est soustrait des dates de livraisons. Les quantités des produits finis peuvent être livrées en avance, mais le producteur ne sera payé qu'aux dates convenues, et c'est strictement interdit de livrer en retard. Les dates de livraisons convenues sont considérées comme des contraintes impératives.

Ayant les données des deux courbes cumulées (d'arrivées de composants, et de demandes de livraisons), le premier problème est de décider si le producteur peut satisfaire ces clients. Ceci revient à vérifier qu'à chaque date de départ de livraison, le producteur peut satisfaire la quantité associée à cette date. Il suffit que le producteur ne puisse satisfaire son client au moins une fois tout au long de l'horizon H pour que le problème soit non réalisable, sinon il est réalisable. Dans le cas où le problème est réalisable, nous proposons des algorithmes pour

minimiser les coûts de transports. Nous considérons le cas où on dispose d'un nombre illimité de camions avec une capacité illimitée pour effectuer les transports, et le cas où le nombre de ces camions est illimité avec une capacité limitée.

5.3.2 Notations

Pour ce problème, nous utilisons les notations suivantes :

NC : Nombre d'arrivées de composants.

QC_j : Quantité associée à la $j^{\text{ème}}$ arrivée de composants.

DC_j : Date d'arrivée associée à QC_j .

ND : Nombre de livraisons des produits finis.

QD_i : La quantité associée à la $i^{\text{ème}}$ livraison.

DD_i : La date de départ de livraison associée à QD_i (les t unités de temps nécessaires au transport sont soustraits à la date de livraison convenue chez le client).

CQP_i : La quantité cumulée produite à partir de l'instant 0 et jusqu'à la $i^{\text{ème}}$ date de livraison DD_i incluse.

CQD_i : La quantité cumulée demandée à partir de l'instant 0 et jusqu'à la $i^{\text{ème}}$ date de livraison DD_i incluse.

ΔPQ_i : Quantité produite entre DD_{i-1} et DD_i .

ND , NC , QD_i , DD_i , QC_j et DC_j sont les données du problème. CQP_i , CQD_i et ΔPQ_i sont calculés par l'algorithme 1 décrit ci-dessous.

5.3.3 Problème de faisabilité

En supposant que les capacités de transport sont illimitées, le problème est faisable si, et seulement si, à chaque date de livraison la quantité cumulée produite jusqu'à cette date est supérieure ou égale à la quantité cumulée demandée jusqu'à cette date. Plus formellement, cette condition peut être formulée comme suit :

$$CQP_i \geq CQD_i \text{ pour } i = 1 \text{ à } ND.$$

L'algorithme 1 calcule les valeurs des CQP_i et aussi celles des ΔPQ_i en produisant au plus tôt.

Algorithme 1

Début

$j=T=stock=Qtity=0$

Pour $i = 1, \dots, ND$ Faire

$Qtity=stock$

Tant que $DC_j \leq DD_i$ et $T \leq DD_i$ Faire

$Qtity= Qtity+ QC_j$;

$T=\max (T, DC_j)+ QC_j *P$;

$j=j+1$

FinTQ

Si $T > DD_i$ Alors

$$\text{stock} = \left\lceil \frac{T - DD_i}{P} \right\rceil ;$$

Qty = Qty - stock ;
 FinSi
 $\Delta PQ_i = Qty$
 fin de l'itération
 FinPour

Fin

A capacité illimitée, la production et le transport peuvent être organisées de la manière suivante. Entre chaque deux dates de livraison on produit au maximum et on transporte t unités de temps avant la date de livraison. A chaque date de livraison on vérifie si les quantités transportées sont supérieures ou égales au cumul des quantités demandées à cet instant là, si c'est le cas on poursuit la procédure, sinon, le problème est non réalisable. Si à la fin de la procédure on a satisfait toutes les demandes, alors le problème est réalisable. La complexité de l'algorithme 1 dépend des dates auxquelles sont programmées une livraison ou une arrivée de composants, sa complexité est donc de $O(ND+NC)$.

La figure suivante montre un exemple d'un plan de livraisons réalisable, et la figure montre un plan non réalisable.

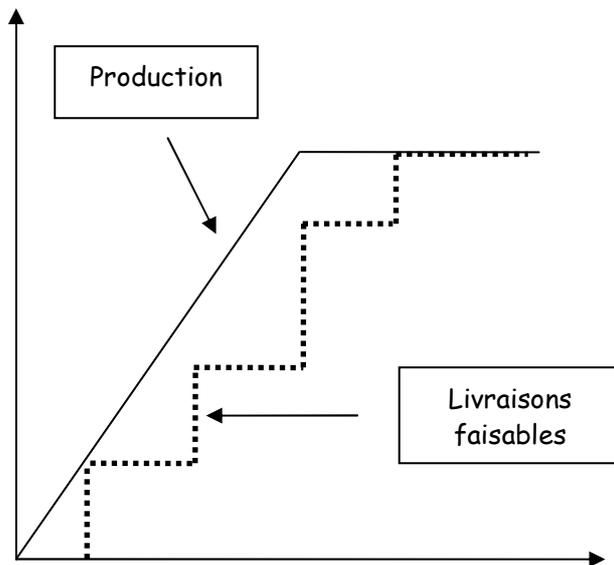


Figure 5.7 scénario faisable.

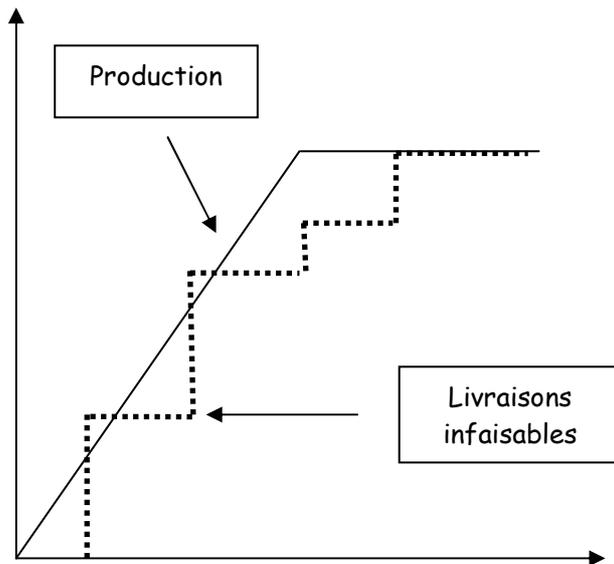


Figure 5.8 Scénario infaisable.

5.3.4 Minimisation des transports : capacité illimitée

En supposant que le problème est faisable à la fin de l'application de l'algorithme 1, on cherche ici à minimiser les coûts de transport. On suppose que l'activité de transport a été déléguée à une tierce partie, de ce fait, on dispose d'une flotte d'un nombre illimité de camions. La capacité de ces camions est illimitée. Cette hypothèse peut arriver dans le cas où les produits finis sont des objets de taille très petites (objets technologiques ou composant électroniques). Un coût de transport est associé à chaque trajet effectué entre le producteur et le client. La minimisation des coûts de transport revient donc à minimiser ce nombre de trajets. La politique de transport calée droite consistant à effectuer des trajets le plus tard possible (t unités de temps avant la date de livraison) est optimale dans notre cas.

Ce problème de transport peut être modélisé par les équations suivantes.

On définit d'abord la variable X_i tel que :

$$X_i = \begin{cases} 1 & \text{si un trajet est effectué à la } i^{\text{ème}} \text{ date de livraison.} \\ 0 & \text{sinon} \end{cases}$$

Soit CQT_i la quantité cumulée transportée entre l'instant 0 et la $i^{\text{ème}}$ date de livraison DD_i incluse.

$$CQT_i = \sum_{j=1}^i \Delta PQ_j \cdot X_j \text{ pour } i = 1 \text{ à } ND.$$

$$CQT_i \geq CQD_i \text{ pour } i = 1 \text{ à } ND \text{ (satisfaction de la demande du client).}$$

$$\text{Minimiser } \sum_{i=1}^{ND} X_i$$

Propriété 1. Le nombre optimal de trajets entre le producteur et le client est donné par :

$$X = \sum_{i=1}^{ND} X_i \text{ tel que si } CQT_{i-1} \geq CQD_i \text{ alors } X_i = 0 \text{ sinon } X_i = 1.$$

Preuve. On suppose que notre solution n'est pas optimale. C'est à dire qu'on peut faire au moins un trajet en moins sans que le problème ne devienne non faisable. Soit le $i^{\text{ème}}$ trajet celui qu'on peut supprimer. On aura donc $X_i = 0$ ce qui implique que $CQT_i = CQT_{i-1}$ et comme $CQT_{i-1} < CQD_i$ on aura $CQT_i < CQD_i$ ce qui rend le problème non faisable.

L'algorithme 2 suivant permet de calculer le nombre minimum de trajets à effectuer. En effet, l'Algorithme 2 est similaire à l'algorithme 1. Après vérification de la faisabilité du problème, on ajoute les opérations suivantes à la place de **fin de l'itération** :

Algorithme 2

Appliquer l'algorithme 1.

Si $CQT_{i-1} \geq CQD_i$ alors $X_i = 0$

Sinon $X_i = 1$ FinSi

$$CQT_i = CQT_{i-1} + \Delta PQ_i$$

Fin

Comme ces instructions n'affectent pas le temps d'exécution de l'algorithme, l'algorithme 2 aura la même complexité que l'algorithme 1, c'est à dire $O(ND+NC)$.

5.3.5 Minimisation des transports : capacité limitée

C'est un problème similaire à celui de la section précédente où on rajoute une nouvelle contrainte sur la capacité des camions. Nous supposons ici que les camions ont une capacité limitée à C items des produits finis. Nous disposons toujours d'un nombre illimité de camions. Pour minimiser le nombre de trajets, nous proposons l'algorithme 3 qui est une extension de l'algorithme 1. En effet, on rajoute les instructions suivantes à la place de **fin de l'itération** :

Algorithme 3

Appliquer l'algorithme 1.

$$X_i = \left\lfloor \frac{\Delta PQ_i}{C} \right\rfloor;$$

$$CQT_i = CQT_{i-1} + X_i \cdot C$$

Si $CQT_i < CQD_i$ alors

$$X_i = X_i + 1;$$

$$CQT_i = CQT_i + \Delta PQ_i \text{ mod } C$$

Sinon

$$\text{inventory} = \Delta PQ_i \text{ mod } C$$

FinSi

Fin

On transporte à chaque fois que la production atteint la quantité C . A la fin de chaque période, il reste une quantité Q (éventuellement nulle et obligatoirement inférieure à C). Si la

somme des quantités transportées jusque là est supérieure ou égale à la somme des quantités demandées à cet instant, alors Q reste en attente sinon, elle est transportée.

Théorème 1. *Le nombre optimal de trajet pour le problème de minimisation des coûts de transports avec capacité limitée des camions est donné par l'algorithme 3.*

Preuve. On suppose que notre solution n'est pas optimale. Ceci implique que l'on peut supprimer au moins l'un des trajets sans pour autant rendre le problème non faisable. Soit ce trajet, le trajet de la $i^{\text{ème}}$ itération. On aura à considérer deux cas :

I. On supprime un camion qui a été totalement chargé : Dans ce cas, et comme la somme des quantités transportées est égale à la somme des quantités demandées, ce trajet peut être effectué à une autre itération k , tel que $i < k \leq ND$ et dans ce cas le nombre total de trajets va rester constant.

II. On supprime un trajet avec un camion non totalement chargé : nous sommes alors dans le cas où $CQT_i < CQD_i$ et on choisit de ne pas transporter, ce qui implique que CQT_i reste inchangée et on aura donc $CQT_i < CQD_i$ ce qui rend le problème infaisable.

L'algorithme 3 a la même complexité que les deux algorithmes précédents, pour les mêmes raisons que pour l'algorithme 2.

5.4 Problème 3: Optimisation des transports dans une supply chain spécifique

Nous étudions dans cette section trois nouveaux problèmes d'optimisation des transports qui sont une extension des problèmes étudiés dans la section précédente (problème 2) mais avec l'addition de nouvelles contraintes. Dans le premier problème de cette section nous nous intéressons à l'optimisation des coûts de transport dans une chaîne logistique mono produits, ayant plusieurs clients dispersés géographiquement, et avec un seul producteur. Dans le deuxième problème de cette section on considère la même chaîne logistique mais avec la production de plusieurs types de produits finis. Dans ces deux problèmes, le planning de production est supposé fixé et connu, et donc on optimise les coûts de transport relativement à une politique de production optimale calculée en avance. Dans le dernier et troisième problème, on considère l'optimisation simultanée des coûts de production et de transport. Pour ces différents problèmes nous proposons une modélisation en graphes et une approche de résolution basée sur la programmation dynamique.

5.4.1 Optimisation des transports dans une supply chain mono-produit multi-clients avec planning de production fixé

On considère une chaîne logistique composée d'un ensemble de fournisseurs, un producteur et plusieurs clients. Un ou plusieurs fournisseurs approvisionnent le producteur avec la matière première nécessaire à la production d'un seul type de produits. La matière première arrive en quantités et à des dates connues. Le système de production est vu comme un système à une seule machine qui a besoin d'une certaine durée de temps pour produire un item du produit fini. Le producteur doit satisfaire les demandes de ses clients. Les clients effectuent des demandes de livraisons en quantités connues pour des dates bien déterminées. Plusieurs clients peuvent émettre des demandes de livraisons à des dates simultanées. Les clients sont dispersés géographiquement et donc les temps de trajets du producteur vers les clients sont différents d'un client à un autre. Pour effectuer les livraisons de produits finis vers les clients on dispose d'une flotte de véhicules à capacité limitée à un certain nombre d'items. La flotte est homogène, donc tous les véhicules ont la même capacité. On suppose que l'activité de transport est déléguée à une tierce partie, et que, donc, on a autant de véhicules que l'on veut, et on ne gère ni les problèmes de transport à vide, ni l'organisation d'éventuelles tournées de camions.

Sachant que le planning et l'ordonnancement de la production sont déjà établis et connus, on cherche à trouver l'ordonnancement des transports dont le coût est minimum, cela revient ici à trouver le nombre de trajets minimum entre le producteur et les clients.

5.4.1.1 Notations

Données

n : Le nombre de périodes de l'horizon d'ordonnancement qui correspond au nombre de fois où une demande de livraison est émise.

m : Le nombre de clients.

C : Capacité de chaque véhicule.

d_{ij} : Quantité demandée par le $j^{\text{ème}}$ client à la $i^{\text{ème}}$ période. Cette date prend en compte les délais de transport vers les clients, il s'agit des départs souhaités au plus tard des livraisons vers les clients.

dd_i : date correspondant à la $i^{\text{ème}}$ demande de livraison ou période.

CQD_i : La demande cumulée de tous les clients depuis la première période jusqu'à la période i incluse.

CQD_{ij} : La demande cumulée du client j depuis la première période jusqu'à la période i incluse.

Comme on ne considère pas de contraintes de stock, la politique de production optimale (du point de vue de la production) consiste à produire au plus tôt, c'est-à-dire une production calée à gauche. On suppose dans ce premier problème que l'ordonnancement de la production est fixé et connu à l'avance, on ne se préoccupe donc que de la politique de transport dont on veut minimiser les coûts en minimisant le nombre de trajets de livraison effectués.

ΔPQ_i : Quantité produite pendant la période i , c'est-à-dire entre dd_{i-1} et dd_i

Variables

X_{ij} : Quantité totale transportée pour le client j pendant la période i .

S_i : Quantité en stock au début de la période i .

Le tableau tab montre un exemple d'un problème à deux clients :

	1	2	3
demande client 1	3	5	6
demande client 2	4	6	7
Production	8	12	11

Figure 5.9 exemple de scénario de production.

On a un horizon de 3 périodes et on a 2 clients, la capacité des véhicules C est égale à 5.

5.4.1.2 Représentation graphique

On ramène le problème étudié à un problème de recherche du plus court chemin dans un graphe orienté acyclique. Pour modéliser ce problème par un graphe, nous devons définir l'ensemble des états (sommets), l'ensemble des arcs, comment on les génère, et on doit définir la fonction coût sur les arcs.

Ensemble des états

Pour chaque période i on construit un ensemble d'états (sommets) fini, éventuellement vide. Un état est représenté par un vecteur contenant les quantités transportées pour chaque client pendant la période considérée. Pour cela on définit les variables suivantes :

X_{ij}^l : Quantité transporté pendant la période i pour le client j dans le $l^{\text{ème}}$ état.

S_i^l : La quantité en stock correspondant au $l^{\text{ème}}$ état de la période i .

S_{ij}^l : Quantité cumulée jusqu'à la période i en surplus de la demande et transportée au client j .

$X_i^l = (X_{i1}^l, X_{i2}^l, \dots, X_{im}^l, S_{i1}^l, S_{i2}^l, \dots, S_{im}^l, S_i^l)$ vecteur définissant le $l^{\text{ème}}$ état de la période i .

e_i : est le nombre d'états dans la période i .

Le calcul des quantités en stock se fait comme suit :

$$S_i^l = \Delta PQ_{i-1} - \sum_{j=1}^m X_{i-1j}^e + S_{i-1}^e \text{ Avec la condition qu'il existe un arc de } X_{i-1}^e \text{ vers } X_i^l.$$

On rajoute deux états (deux sommets), un sommet de départ appelé S et un sommet d'arrivée appelé F .

Les arcs

Un arc part d'un état X_i^l vers un état X_{i+1}^e , si et seulement si, les contraintes suivantes sont respectées :

$$X_{i+1j}^e \geq d_{i+1j} - S_{ij}^l \quad \forall j \quad (5.4)$$

$$\sum_{j=1}^m X_{i+1j}^e \leq \Delta PQ_{i+1} + S_i^l \quad (5.5)$$

La contrainte (5.4) permet de satisfaire la demande de chaque client. La contrainte (5.5) assure que l'on ne peut pas transporter plus que la quantité effectivement produite. Ces deux contraintes assurent une borne inférieure et une borne supérieure pour les valeurs que peuvent prendre les X_{ij}^l . Le nombre de permutations (valeurs que peuvent prendre les X_{ij}^l) possibles est donc fini, ce qui implique que le nombre d'états est fini, éventuellement nul. Dans le cas où le nombre d'états pour une période donnée est nul, alors le problème est non faisable (i.e. $e_i = 0$ alors le problème est non faisable).

Il peut y avoir plusieurs arcs partants du même état (plusieurs arcs sortants) mais il n'y a qu'un seul arc pour atteindre un état (un seul arc entrant) sauf pour les états de la dernière période n . Il ne peut pas y avoir des arcs partant d'un état vers un état de la même période. Un arc doit obligatoirement partir d'un état d'une période donnée (soit i), vers un des états de la période qui la suit immédiatement (soit $i+1$). C'est donc un graphe orienté qui ne comporte pas de cycles.

A partir du sommet de départ S , part un arc pour chacun des états de la première période ($i=1$) avec un coût qui dépend des états d'arrivées. De chaque état de la dernière période ($i=n$) part un arc d'un coût nul vers l'état d'arrivée F .

La fonction coût sur les arcs

Comme notre objectif est de minimiser les coûts de transport en trouvant la séquence de livraisons de coût minimum, c'est à dire celle avec le nombre minimum de trajets entre le producteur et les clients, alors la fonction coût de transition d'un état X_i^l de la période i vers l'état X_{i+1}^e de la période $i+1$ est défini par le nombre de trajets qu'une telle transition implique, cela dépend donc des valeurs du vecteur X_{i+1}^e . La fonction coût de transition est donnée par :

$$fct(X_i^l, X_{i+1}^e) = fct(X_{i+1}^e) = \sum_{j=1}^m \left[\frac{X_{i+1j}^e}{C} \right] \quad (5.6)$$

5.4.1.3 Approche de résolution

L'algorithme du programme dynamique pour le calcul du plus court chemin dans un graphe orienté ne comportant pas de cycle est très connu, on va l'adapter à notre problème. L'idée générale de l'algorithme est de dire que le plus court chemin jusqu'à la période i est obtenu par le calcul du plus court chemin (donc le chemin de coût minimum) jusqu'à la période $i-1$, puis on rajoute le coût de l'arc de coût minimum entre les périodes i et $i-1$. La fonction de notre programme dynamique donne le nombre de trajets minimum jusqu'à une certaine période :

$$\begin{cases} F(0) = 0 \\ F(i) = \underset{l=1, e_i}{\text{Min}}(F(i-1) + fct(X_i^l)) \end{cases} \quad (5.7)$$

Remarque : il peut y avoir plusieurs solutions optimales, donc plusieurs chemins de même coût.

La figure suivante illustre le graphe correspondant aux données du tableau.

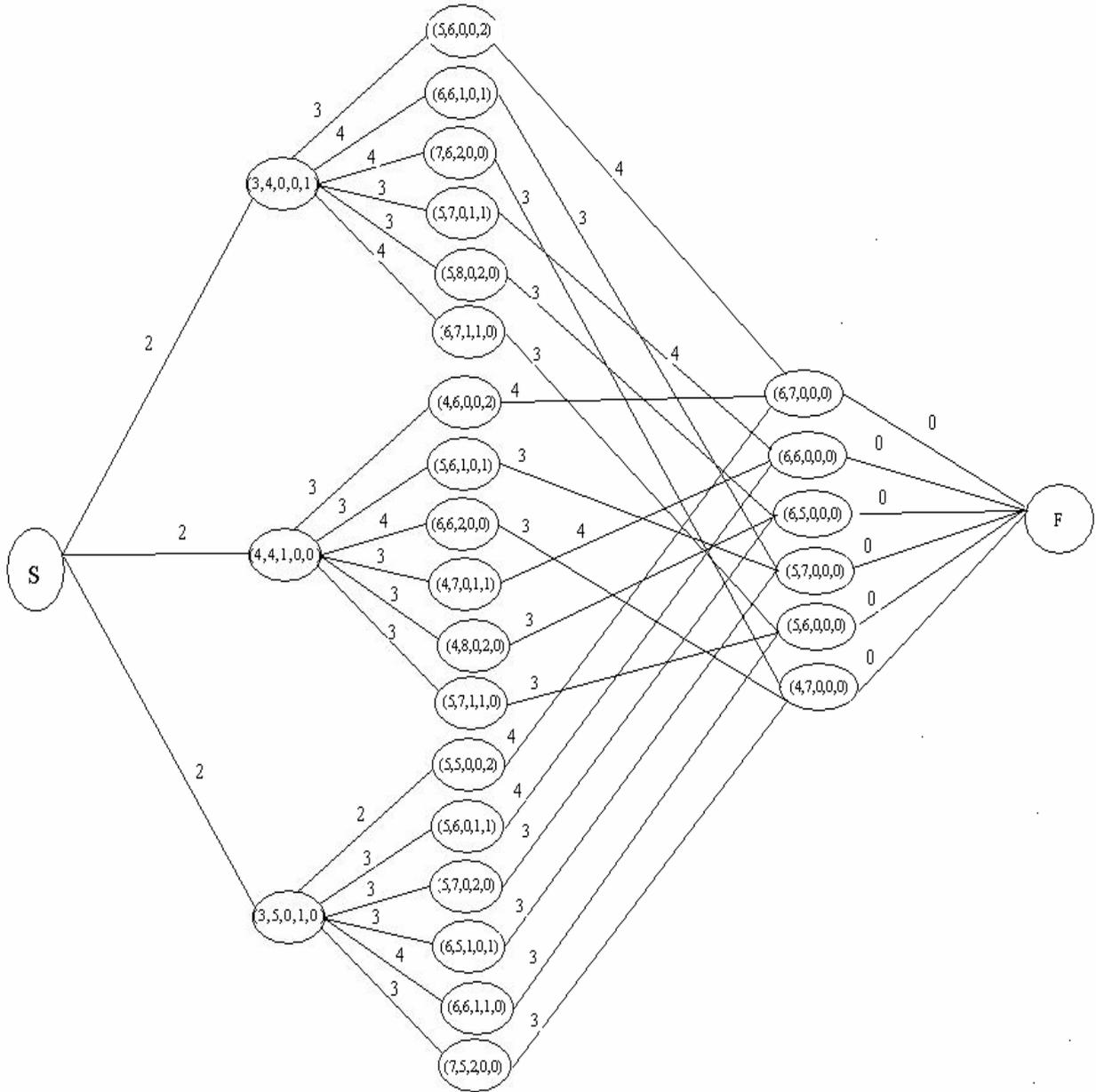


Figure 5.10 Graphe correspondant au problème du tableau 5.1

5.4.2 Optimisation des transports dans une supply chain multi-produit multi-clients avec planning de production fixé

On généralise maintenant le problème étudié dans la section 5.4.1 au cas où le producteur doit fournir plusieurs types de produits. Comme dans le cas précédent, les clients émettent des commandes avec des quantités de chaque type de produit connues, et des dates de livraisons souhaitées. Le producteur doit satisfaire ces demandes dans les délais déterminés. On doit donc déterminer la quantité à transporter pour chaque client, de chaque produit, pendant chaque période de telle sorte à minimiser les coûts de transports, donc minimiser le nombre de trajets effectués. On suppose que les produits ont le même volume, et que différents produits vers le même client peuvent être mis dans le même véhicule.

5.4.2.1 Notations

Données

n : Le nombre de périodes de l'horizon d'ordonnancement qui correspond au nombre de fois où une demande de livraison est émise.

m : Le nombre de clients.

C : Capacité de chaque véhicule.

d_{ijk} : Quantité demandée par le $j^{\text{ème}}$ client à la $i^{\text{ème}}$ période.

dd_i : Date correspondant à la $i^{\text{ème}}$ demande de livraison ou période.

P_k : Temps d'exécution d'un item du produit k .

R : Nombre de produits.

ΔPQ_i^k : Quantité du produit k produite durant la période i , c'est à dire entre dd_{i-1} et dd_i .

St_k : Temps de réglage nécessaire pour préparer la machine à produire un produit de type k .

Variables

CQT_{ij}^k : Cumul des quantités du produit k transportées jusqu'à la période i pour le client j .

X_{ijk} : Quantité du produit k transportée durant la période i pour le client j .

Y_{ij} : Nombre de trajets effectués pour le client j durant la période i .

Rs_i^k : Quantité en stock du produit k à la fin de la période i .

Contraintes

1. *Contrainte de transport* : la quantité transportée doit être supérieure ou égale à la quantité demandée de chaque produit, pour chaque client et à chaque période.
2. *Contrainte de production* : le cumul des quantités produites doit être supérieure ou égale au cumul des quantités demandées à chaque période et pour chaque produit.

Objectif

Minimiser les coûts de transport en minimisant le nombre de trajets.

Ce problème peut être modélisé par le programme mathématique suivant :

$$\text{Min } \sum_{i=1}^n \sum_{j=1}^m Y_{ij} \quad (5.14)$$

sous les contraintes :

$$\sum_{j=1}^m X_{ijk} \leq \Delta PQ_i^k \quad \forall i, k \quad (5.8)$$

$$\Delta PQ_i^k = \text{Re } s_{i-1}^k + \Delta PQ_i^k \quad \forall i, k \quad (5.9)$$

$$\text{Re } s_i^k = \Delta PQ_i^k - \sum_{j=1}^m X_{ijk} \quad \forall i, k \quad (5.10)$$

$$CQT_{ij}^k = \sum_{s=1}^i X_{sjk} \quad \forall i, j, k \quad (5.11)$$

$$CQT_{ij}^k \geq CQD_{ij}^k \quad \forall i, j, k \quad (5.12)$$

$$Y_{ij} = \left\lceil \frac{\sum_{k=1}^R X_{ijk}}{C} \right\rceil \quad \forall i, j \quad (5.13)$$

La contrainte (5.8) exprime le fait que la somme des quantités transportées doit être inférieure ou égale aux quantités dont on dispose pour chaque produit. On calcule les quantités dont on dispose de chaque produit et à chaque période par la contrainte (5.9), et les quantités en stock de chaque produit et à chaque période par la contrainte (5.10). La somme des quantités transportées par produit et par client, jusqu'à une période donnée est calculée à l'aide de la contrainte (5.11). La contrainte de transport est exprimée par la contrainte (5.12). La contrainte (5.13) permet de calculer le nombre de trajets à effectuer lors de chaque période et pour chaque client. La fonction objective qui consiste à minimiser le nombre total de trajets est donnée par la contrainte (5.14).

5.4.2.2 Représentation graphique

Ensemble des états (sommets)

Comme pour le premier problème étudié, à chaque période correspond un ensemble d'états fini éventuellement vide. L'ensemble des états correspond à toutes les combinaisons des X_{ijk} possibles respectant les contraintes de transport et de production. Dans le cas d'un seul produit, un état était représenté par un vecteur correspondant aux quantités qu'on pouvait transporter pour chaque client. Dans le cas de différents produits, on aura le même vecteur pour chaque type de produit, un état sera représenté donc par une matrice de dimension $(R \times n)$. Pour cela, nous définissons les variables suivantes :

X_{ijk}^l : Quantité du produit k transportée pendant la période i pour le client j dans le l^{eme} état.

S_{ik}^l : la quantité en stock du produit k correspondant au l^{eme} état de la période i .

S_{ijk}^l : Quantité du produit k cumulée jusqu'à la période i en surplus de la demande et transportée au client j .

$X_{ik}^l = (X_{i1k}^l, X_{i2k}^l, \dots, X_{imk}^l, S_{i1k}^l, S_{i2k}^l, \dots, S_{imk}^l, S_{ik}^l)$ vecteur définissant le l^{eme} état de la période i pour le produit k .

e_i : est le nombre d'états dans la période i .

Voici un exemple d'un ensemble d'états de la période i .

$$\begin{array}{l}
 \text{état } 1 \\
 \dots\dots\dots \\
 \text{état}
 \end{array}
 \left(
 \begin{array}{l}
 X_{i11}^1, X_{i21}^1, \dots, X_{im1}^1, S_{i11}^1, S_{i21}^1, \dots, S_{im1}^1, S_{i1}^1 \\
 X_{i12}^1, X_{i22}^1, \dots, X_{im2}^1, S_{i12}^1, S_{i22}^1, \dots, S_{im2}^1, S_{i2}^1 \\
 \dots\dots\dots \\
 X_{i1k}^1, X_{i2k}^1, \dots, X_{imk}^1, S_{i1k}^1, S_{i2k}^1, \dots, S_{imk}^1, S_{ik}^1
 \end{array}
 \right)$$

$$\left(
 \begin{array}{l}
 X_{i11}^{e_i}, X_{i21}^{e_i}, \dots, X_{im1}^{e_i}, S_{i11}^{e_i}, S_{i21}^{e_i}, \dots, S_{im1}^{e_i}, S_{i1}^{e_i} \\
 X_{i12}^{e_i}, X_{i22}^{e_i}, \dots, X_{im2}^{e_i}, S_{i12}^{e_i}, S_{i22}^{e_i}, \dots, S_{im2}^{e_i}, S_{i2}^{e_i} \\
 \dots\dots\dots \\
 X_{i1k}^{e_i}, X_{i2k}^{e_i}, \dots, X_{imk}^{e_i}, S_{i1k}^{e_i}, S_{i2k}^{e_i}, \dots, S_{imk}^{e_i}, S_{ik}^{e_i}
 \end{array}
 \right)$$

Soit E_i^l la matrice définissant le l^{eme} état de la période i . On aura donc :

$$E_i^l = \left(
 \begin{array}{l}
 X_{i11}^l, X_{i21}^l, \dots, X_{im1}^l, S_{i11}^l, S_{i21}^l, \dots, S_{im1}^l, S_{i1}^l \\
 X_{i12}^l, X_{i22}^l, \dots, X_{im2}^l, S_{i12}^l, S_{i22}^l, \dots, S_{im2}^l, S_{i2}^l \\
 \dots\dots\dots \\
 X_{i1k}^l, X_{i2k}^l, \dots, X_{imk}^l, S_{i1k}^l, S_{i2k}^l, \dots, S_{imk}^l, S_{ik}^l
 \end{array}
 \right)$$

Si le nombre d'états dans une période donnée est nul, alors le problème n'est pas faisable.

Les arcs

Un arc part d'un état X_i^l vers un état X_{i+1}^e , si et seulement si, les contraintes suivantes sont respectées :

$$X_{i+1jk}^e \geq d_{i+1jk} - S_{ijk}^l \quad \forall j, k, e, i \quad (5.15)$$

$$\sum_{j=1}^m X_{i+1jk}^e \leq \Delta PQ_{i+1}^k + S_{ik}^l \quad \forall j, k, e, i \quad (5.16)$$

Comme pour le problème avec un seul produit, on a un état de départ et un état d'arrivée avec les mêmes propriétés que dans le premier problème. Notre méthode de construction du graphe nous assure de n'avoir que des solutions faisables et que le nombre d'états est fini (éventuellement nul).

Fonction coût

La fonction coût d'un arc exprime le nombre de trajets qui correspond à un état donné. Le coût de transition d'un état E_i^l de la période i vers l'état E_{i+1}^e de la période $i + 1$ est défini par le nombre de trajets qu'une telle transition implique, cela dépend donc des valeurs de la matrice E_{i+1}^e . La fonction coût de transition est donnée par :

$$C(E_i^l, E_{i+1}^e) = C(E_{i+1}^e) = \sum_{j=1}^m \left[\frac{\sum_{k=1}^R X_{i+1jk}^e}{C} \right]$$

5.4.2.3 Approche de résolution

L'approche de programmation dynamique que nous proposons pour le problème avec des produits différents est une généralisation de l'approche proposée pour le cas d'un seul produit. En effet, nous modélisons le problème par un graphe orienté sans cycles dans lequel nous recherchons le plus court chemin, c'est-à-dire le chemin de coût minimum représentant les quantités qu'on doit transporter de chaque produit pour chaque client et à chacun des périodes de l'horizon en respectant les contraintes de production et de transport.

La fonction de récurrence du programme dynamique s'écrit comme suit :

$$\begin{cases} F(0) = 0 \\ F(i) = \underset{l=1, e_i}{\text{Min}}(F(i-1) + C(E_i^l)) \end{cases} \quad (5.17)$$

5.4.3 Optimisation conjointe de la production et des transports dans une supply chain multi-produit multi-clients

Dans ce nouveau problème, on intègre l'optimisation de la production sur une seule machine et des transports dans le même modèle. Ce qui change par rapport au problème précédent, c'est que la planification de la production n'est pas connue à l'avance et n'est donc pas fixée. L'objectif est de trouver la combinaison d'un ordonnancement de la production et de l'ordonnancement des transports qui permette de minimiser l'ensemble (c'est-à-dire la somme) des coûts de production et de transports. Pour cela, on propose un programme dynamique dans l'esprit est le même que celui proposé pour le problème précédent. On définit et prouve certaines propriétés du modèle sur la politique d'ordonnancement de la production et des transports qui nous permettent de réduire l'espace de recherche de solutions sans perdre en optimalité.

5.4.3.1 Notations

On va considérer donc de nouvelles données et de nouvelles variables.

Données

Cst_k : Coût de préparation de la machine à produire du type k .

Cp_k : Coût de production par unité de temps pour produire du type k .

Ctr : Coût de transport associé à un trajet.

Variables

ΔPQ_i^k : Quantité du produit k produite durant la période i , c'est à dire entre dd_{i-1} et dd_i .

Fonction Objective

La fonction objective à minimiser est la somme de deux composantes. La première désigne les coûts de production, et la deuxième désigne les coûts de transports :

$$F_{obj} = F_{prod} + F_{tr}$$

1) Coût de production : Cette fonction exprime les coûts de production qui sont la somme des coûts de setups d'un type de produits à un autre, et les coûts unitaires de production. Dans ce cadre, on intègre la notion de lot qui définit une suite de produits du même type exécutés successivement (i.e. les items qui sont entre deux setups appartiennent au même lot). On définira les deux variables suivantes :

L_k : Le nombre de lots de produits k .

L_k^b : Nombre d'items (taille) du b^{eme} lot de produits k .

L_{kq} : Désigne le q^{eme} lot de produits de type k .

On aura donc la fonction coût de production suivante :

$$F_{prod} = \sum_{k=1}^R \left(Cst_k \times L_k + \sum_{b=1}^{L_k} (Cp_k \times L_k^b) \right).$$

2) Coût de transport : Cette fonction exprime les coûts de transports qui ne dépendent que du nombre de trajets effectués.

$$F_{tr} = \sum_{i=1}^n \sum_{j=1}^m \left(\left[\frac{\sum_{k=1}^R X_{ijk}}{C} \right] \times Ctr \right)$$

La fonction objective est donc :

$$F_{obj} = \sum_{k=1}^R \left(Cst_k \times L_k + \sum_{b=1}^{L_k} (Cp_k \times L_k^b) \right) + \sum_{i=1}^n \sum_{j=1}^m \left(\left[\frac{\sum_{k=1}^R X_{ijk}}{C} \right] \times Ctr \right)$$

5.4.3.2 Approche de résolution

Propriété 2

Dans un ordonnancement optimal, tous les items du même produit doivent se suivre. Donc, il ne doit y avoir qu'un seul lot de chaque produit.

Preuve. Considérons un ordonnancement optimal dans lequel il existe plusieurs lots d'un même produit, il existe donc des items du même produit qui ne sont pas exécutés successivement. Considérons une partie de cet ordonnancement dans laquelle le $q^{ème}$ lot du produit k , donc L_{kq} commence à l'instant t et est suivi de r lots de produits différents de k , puis vient le $q+1^{ème}$ lot de produit de type k , comme le montre la figure suivante :

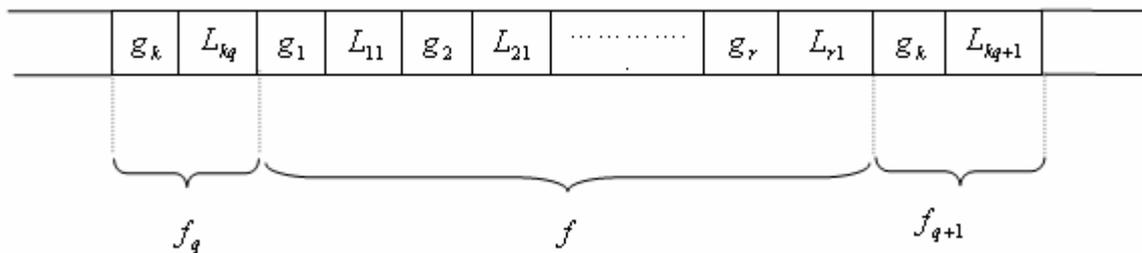


Figure 5.11 ordonnancement initial.

Comme le nombre de trajets pour une période donnée ne dépend que des quantités transportées pendant cette période, et que celles-ci dépendent bien sûr des quantités produites, alors la position dans laquelle sont exécutés les items de chaque produit n'influe pas sur le nombre de trajets car quelque soit la position dans laquelle les items sont exécutés, cela ne changerait pas la quantité totale produite. On va donc ignorer la partie relative aux coûts de transport dans les calculs qui suivent.

Soient $f_q, f,$ et f_{q+1} les coûts de productions associés aux parties de l'ordonnancement montrés par la figure. On a :

$$f_q = Cst_k + Cp_k \times L_k^q$$

$$f = Cst_1 + Cp_1 \times L_1^1 + Cst_2 + Cp_k \times L_2^1 + \dots + Cst_r + Cp_r \times L_r^1$$

$$f_{q+1} = Cst_k + Cp_k \times L_k^{q+1}$$

Soit f_1 le coût de production de la partie de l'ordonnancement montrée par la figure 5.13 :
 $f_1 = f_q + f + f_{q+1}$

- Considérons maintenant un ordonnancement dérivé du précédent, dans lequel le lot L_{kq+1} est placé juste après le lot L_{kq} . La figure suivante montre un tel ordonnancement.

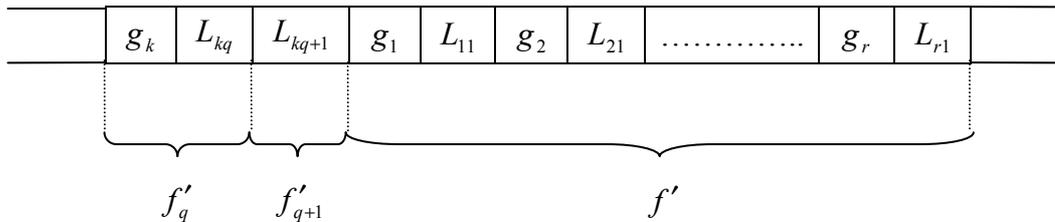


Figure 5.12 ordonnancement 1 dérivé de l'ordonnancement original.

$$f'_q = Cst_k + Cp_k \times L_k^q$$

$$f' = Cst_1 + Cp_1 \times L_1^1 + Cst_2 + Cp_k \times L_2^1 + \dots + Cst_r + Cp_r \times L_r^1$$

$$f'_{q+1} = Cp_k \times L_k^{q+1}$$

Comme L_{kq+1} suit immédiatement L_{kq} , le temps de setup entre ces deux lots est nul car il s'agit du même produit. Soit f'_1 le coût associé à l'ordonnancement de la figure, On a :

$$f'_1 = f'_q + f' + f'_{q+1}$$

- Considérons maintenant un ordonnancement dérivé de l'ordonnancement original dans lequel le lot L_{kq+1} est placé juste avant le lot L_{kq} . La figure suivante montre un tel ordonnancement

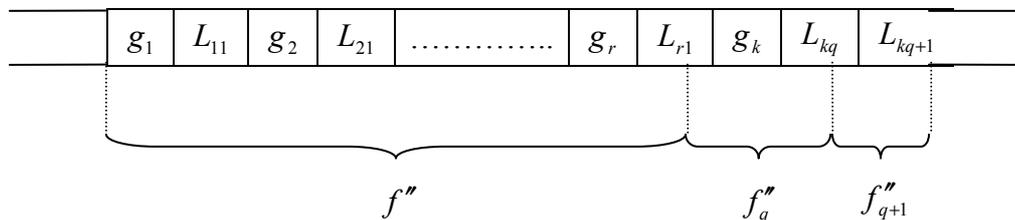


Figure 5.13 ordonnancement 2 dérivé de l'ordonnancement original.

Comme pour les cas précédents, on définit :

$$f''_q = Cst_k + Cp_k \times L_k^q$$

$$f'' = Cst_1 + Cp_1 \times L_1^1 + Cst_2 + Cp_k \times L_2^1 + \dots + Cst_r + Cp_r \times L_r^1$$

$$f''_{q+1} = Cp_k \times L_k^{q+1}$$

Comme pour le cas précédent, il n'y a de temps de réglage (setup) entre les lots $L_{k,q+1}$ et $L_{k,q}$ car il s'agit du même produit. Soit f_1'' le coût associé à l'ordonnancement de la figure, On a :

$$f_1'' = f_q'' + f'' + f_{q+1}''$$

▪ Comme l'ordonnancement original est optimal, et que les deux autres sont obtenus en faisant des modifications sur ce dernier, on a donc :

$$f_1' \geq f_1 \text{ et } f_1'' \geq f_1 \Leftrightarrow f_1' - f_1 \geq 0 \text{ et } f_1'' - f_1 \geq 0 \quad (5.18)$$

➤ Pour le cas du premier ordonnancement dérivé :

$f_1' - f_1 = f_{q+1}' - f_{q+1} = Cp_k \times L_k^{q+1} - Cst_k - Cp_k \times L_k^{q+1} = -Cst_k < 0 \Rightarrow f_1' - f_1 < 0$ ce qui est en contradiction avec (5.18)

➤ Pour le cas du deuxième ordonnancement dérivé :

$f_1'' - f_1 = f_{q+1}'' - f_{q+1} = Cp_k \times L_k^{q+1} - Cst_k - Cp_k \times L_k^{q+1} = -Cst_k < 0 \Rightarrow f_1'' - f_1 < 0$ ce qui est en contradiction avec (5.18).

En répétant cette preuve pour tous les produits et pour tous les lots on arrive à un ordonnancement optimal dans lequel tous les items du même produit se suivent.

Propriété 3. Dans un ordonnancement optimal, l'ordre de séquencement des lots de produits n'influe pas sur la fonction objective.

Preuve. Considérons un ordonnancement optimal dans lequel la propriété 1 est vérifiée. Le séquencement des lots est effectué selon leurs indices numériques du plus petit vers le plus grand comme montré sur la figure suivante :

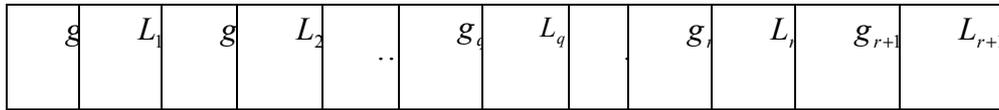


Figure 5.14 ordonnancement optimal vérifiant la propriété 1.

Comme le nombre de trajets ne dépend pas de l'ordre d'exécution des lots (car on a toujours les mêmes quantités quelque soit l'ordre), la partie correspondante aux coûts de transports dans la fonction objective restera donc constante quelque soit l'ordre, elle sera donc négligée dans les calculs.

Soit f_2 la fonction exprimant les coûts de production. Elle est définie par :

$$f_2 = Cst_1 + Cp_1 \times L_1 + Cst_2 + Cp_2 \times L_2 + \dots + Cst_k + Cp_k \times L_k + \dots + Cst_r + Cp_r \times L_r + Cst_{r+1} + Cp_{r+1} \times L_{r+1}$$

▪ Considérons maintenant un ordonnancement dérivé de l'ordonnancement optimal dans lequel on déplace le lot L_{q+1} juste après le lot L_k . On aura :

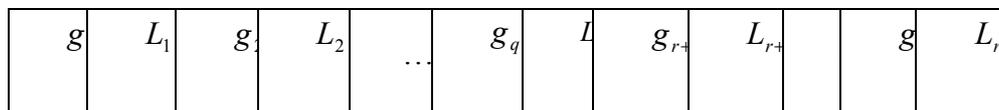


Figure 5.15 ordonnancement dérivé de celui de la figure 5.13

Soit la nouvelle fonction coût f'_2 :

$$f'_2 = Cst_1 + Cp_1 \times L_1 + Cst_2 + Cp_2 \times L_2 + \dots + Cst_k + Cp_k \times L_k + Cst_{r+1} + \dots + Cp_{r+1} \times L_{r+1} \dots + Cst_r + Cp_r \times L_r$$

On peut facilement vérifier que $f_2 = f'_2$. On peut répéter cette opération sur n'importe quelle position et n'importe quel produit, la propriété restera toujours vraie.

L'approche de résolution par programmation dynamique que nous proposons ici est une généralisation et adaptation de celle proposée pour le problème avec l'ordonnancement de la production fixée. Le problème est modélisé par un graphe acyclique dans lequel on cherche le plus court chemin, c'est à dire le chemin de coût minimum qui représente la combinaison entre ordonnancement de la production et la politique de transport résultant dont la somme des coûts est minimum.

Représentation Graphique

Ensemble des Etats (Sommets)

A chaque période i correspond un ensemble d'états e_i fini, et éventuellement vide. Un état est représenté par un couple (VP_i^l, E_i^l) tel que :

$VP_i^l = (\Delta PQ_{i1}^l, \Delta PQ_{i2}^l, \dots, \Delta PQ_{im}^l)$ avec ΔPQ_{ik}^l représente la quantité produite de type k à la $i^{ème}$ période dans le $l^{ème}$ état.

Le seconde composante du couple E_i^l est similaire à celles des états dans le cas « multi produits avec ordonnancement de la production fixé ». C'est donc une matrice de dimension $(m \times n)$, on a alors une ligne par type de produits avec les quantités transportées pour chaque client.

E_i^l : représente la deuxième composante du $l^{ème}$ état de la période i

$$E_i^l = \begin{pmatrix} X_{i11}^l, X_{i21}^l, \dots, X_{im1}^l, S_{i11}^l, S_{i21}^l, \dots, S_{im1}^l, S_{i1}^l \\ X_{i12}^l, X_{i22}^l, \dots, X_{im2}^l, S_{i12}^l, S_{i22}^l, \dots, S_{im2}^l, S_{i2}^l \\ \dots\dots\dots \\ \dots\dots\dots \\ X_{i1k}^l, X_{i2k}^l, \dots, X_{imk}^l, S_{i1k}^l, S_{i2k}^l, \dots, S_{imk}^l, S_{ik}^l \end{pmatrix}$$

tel que –comme pour le problème précédent- X_{ijk}^l exprime la quantité du produit k livrée pour le client j à le période i dans l'état l . Voici l'exemple du $l^{ème}$ état de la période i :

$$(VP_i^l, E_i^l) = \left((\Delta PQ_{i1}^l, \Delta PQ_{i2}^l, \dots, \Delta PQ_{im}^l), \begin{pmatrix} X_{i11}^l, X_{i21}^l, \dots, X_{im1}^l, S_{i11}^l, S_{i21}^l, \dots, S_{im1}^l, S_{i1}^l \\ X_{i12}^l, X_{i22}^l, \dots, X_{im2}^l, S_{i12}^l, S_{i22}^l, \dots, S_{im2}^l, S_{i2}^l \\ \dots\dots\dots \\ \dots\dots\dots \\ X_{i1k}^l, X_{i2k}^l, \dots, X_{imk}^l, S_{i1k}^l, S_{i2k}^l, \dots, S_{imk}^l, S_{ik}^l \end{pmatrix} \right)$$

Et voici un exemple des états de la période i :

$$\begin{array}{c}
 \text{état } 1 \left(\begin{array}{c} \left(\Delta PQ_{11}^l, \Delta PQ_{12}^l, \dots, \Delta PQ_{1m}^l \right), \\ \left(\begin{array}{c} X_{111}^l, X_{121}^l, \dots, X_{1m1}^l, S_{111}^l, S_{121}^l, \dots, S_{1m1}^l, S_{11}^l \\ X_{112}^l, X_{122}^l, \dots, X_{1m2}^l, S_{112}^l, S_{122}^l, \dots, S_{1m2}^l, S_{12}^l \\ \dots \\ \dots \\ X_{11k}^l, X_{12k}^l, \dots, X_{1mk}^l, S_{11k}^l, S_{12k}^l, \dots, S_{1mk}^l, S_{1k}^l \end{array} \right) \end{array} \right) \\
 \\
 \text{état } l \left(\begin{array}{c} \left(\Delta PQ_{i1}^l, \Delta PQ_{i2}^l, \dots, \Delta PQ_{im}^l \right), \\ \left(\begin{array}{c} X_{i11}^l, X_{i21}^l, \dots, X_{im1}^l, S_{i11}^l, S_{i21}^l, \dots, S_{im1}^l, S_{i1}^l \\ X_{i12}^l, X_{i22}^l, \dots, X_{im2}^l, S_{i12}^l, S_{i22}^l, \dots, S_{im2}^l, S_{i2}^l \\ \dots \\ \dots \\ X_{i1k}^l, X_{i2k}^l, \dots, X_{imk}^l, S_{i1k}^l, S_{i2k}^l, \dots, S_{imk}^l, S_{ik}^l \end{array} \right) \end{array} \right) \\
 \\
 \dots \\
 \\
 \text{dernier état } e_i \left(\begin{array}{c} \left(\Delta PQ_{i1}^{e_i}, \Delta PQ_{i2}^{e_i}, \dots, \Delta PQ_{im}^{e_i} \right), \\ \left(\begin{array}{c} X_{i11}^{e_i}, X_{i21}^{e_i}, \dots, X_{im1}^{e_i}, S_{i11}^{e_i}, S_{i21}^{e_i}, \dots, S_{im1}^{e_i}, S_{i1}^{e_i} \\ X_{i12}^{e_i}, X_{i22}^{e_i}, \dots, X_{im2}^{e_i}, S_{i12}^{e_i}, S_{i22}^{e_i}, \dots, S_{im2}^{e_i}, S_{i2}^{e_i} \\ \dots \\ \dots \\ X_{i1k}^{e_i}, X_{i2k}^{e_i}, \dots, X_{imk}^{e_i}, S_{i1k}^{e_i}, S_{i2k}^{e_i}, \dots, S_{imk}^{e_i}, S_{ik}^{e_i} \end{array} \right) \end{array} \right)
 \end{array}$$

Le nombre d'états dans la période i est donnée par e_i . Si $e_i = 0$ alors le problème n'a pas de solutions faisables.

Ensemble des Arcs et Pondération des Arcs

Le nombre d'états dans une période donnée dépend des états de la période qui la précède immédiatement. Les couples de vecteurs de production et de matrices de transports constituant les états sont calculés avec des contraintes permettant d'engendrer un nouvel ensemble d'états à partir d'un état donné (qui sont les mêmes que celles du problème précédent s'agissant des contraintes de transports, en ajoutant une contrainte sur la production).

Un arc part d'un état X_i^l vers un état X_{i+1}^e , si et seulement si, les contraintes suivantes sont respectées :

$$X_{i+1,jk}^e \geq d_{i+1,jk} - S_{ijk}^l \quad \forall j, k, e, i$$

$$\sum_{j=1}^m X_{i+1,jk}^e \leq \Delta PQ_{i+1}^k + S_{ik}^l \quad \forall j, k, e, i$$

Comme pour les problèmes précédents, il n'y a pas d'arcs vers des états de la même période, ni vers des états des périodes suivantes, il y'a des arcs juste vers les états de la période qui suit immédiatement et qui vérifient les contraintes de production et de transport ci-dessus.

Le coût d'un arc dépend de l'état d'origine, il intègre les coûts de production relatifs au vecteur de production, ainsi que les coûts de transports relatifs à la matrice de transport.

La fonction coût sur un arc est définie par :

$$C((VP_i^l, E_i^l), (VP_{i+1}^e, E_{i+1}^e)) = C(VP_i^l, E_i^l) = \sum_{k=1}^R \left(Cst_k \times L_k + \sum_{b=1}^{L_k} (Cp_k \times L_k^b) \right) + \sum_{i=1}^n \sum_{j=1}^m \left(\frac{\sum_{k=1}^R X_{ijk}}{C} \right) \times Ctr$$

Les propriétés 1 et 2 définies précédemment nous permettent de réduire l'espace de recherche de solutions, car on n'a plus besoin d'explorer les différentes solutions résultantes des différentes combinaisons d'un même vecteur de production qui peuvent être engendrées par un ordre différent d'exécutions des produits (sachant qu'il s'agit des mêmes quantités).

Lemme 6. Soit le vecteur de production $VP_i^l = (\Delta PQ_{i1}^l, \Delta PQ_{i2}^l, \dots, \Delta PQ_{im}^l)$. Les différentes combinaisons obtenues en changeant l'ordre des produits n'influent pas sur la fonction coût.

Preuve. La preuve est déduite directement des propriétés 1 et 2.

La formule de récurrence pour le programme dynamique est donnée par la formule suivante :

$$F(0) = 0$$

$$F(i) = \underset{l=1, e_i}{\text{Min}} (F(i-1) + C(VP_i^l, E_i^l))$$

5.5 Conclusion

Nous avons proposé dans ce chapitre un panel d'outils et de techniques pour résoudre certains problèmes qui peuvent arriver dans un contexte industrialisé de chaînes logistiques. Nous nous sommes notamment intéressés aux problèmes de coopération entre fournisseurs, coopération entre production et transport, réduction des coûts de possession de stock, et de minimisation des coûts de transports. Nous avons proposé des algorithmes polynomiaux pour les deux premiers problèmes étudiés, et des algorithmes pseudo polynomiaux à base de programmation dynamique pour les sous problèmes du troisième problème.

Les problèmes d'ordonnement au niveau opérationnel de décision des chaînes logistiques étant peu étudiés dans la littérature, les opportunités de recherches dans ce domaine sont vastes. Il existe un nombre considérable de problèmes dépendant de l'activité, la structure, et la particularité de la chaîne logistique considérée. De ce fait, on ne peut pas faire une recherche exhaustive dans ce domaine.

6 Conclusions et perspectives

Conclusion générale et perspectives

6.1 Conclusions

Dans cette thèse, nous avons étudié un ensemble de problèmes d'ordonnancement pour les chaînes logistiques. Nous nous sommes intéressés plus particulièrement au développement d'outils d'aide à la décision pour l'implémentation des systèmes du juste à temps dans un environnement semi distribué (plusieurs partenaires indépendants, appartenant néanmoins à la même chaîne logistique avec l'intention de collaborer pour obtenir les meilleurs résultats). Dans le problème étudié dans la première partie de ce mémoire, nous avons considéré des problèmes d'ordonnancement dont l'objectif étaient de minimiser les avances et/ou retards par rapport aux dates souhaitées de disponibilité des matières premières (relation fournisseur-système de production), et aux dates de livraison des produits finis aux clients finaux (relation système de production-clients). Dans la deuxième partie de ce mémoire, nous nous sommes intéressés notamment à certains problèmes d'optimisation des transports dans des chaînes logistiques spécifiques.

Ces études répondent à un besoin réel issu du monde des entreprises, que ce soit dans le milieu industriel de fabrication de biens physiques, ou bien dans le milieu des services. Beaucoup de ces entreprises font face quotidiennement à des problèmes pour satisfaire leurs clients, qui sont de plus en plus demandeurs de meilleures performances. Livrer à temps, réduire les coûts de stockage, utiliser au mieux la capacité de production, et répondre rapidement à la demande du marché (besoin de réactivité pour être meilleur que la concurrence), tels sont les bénéfices (non exhaustifs) de la mise en œuvre d'un bon système de juste à temps. Seulement, dans la littérature dédiée aux chaînes logistiques, on se focalise plutôt sur les problèmes de conception (niveau stratégique) et de planification (niveau tactique), mais de plus en plus d'études concernant les aspects opérationnels voient le jour. Notre travail en fait partie, et nous espérons qu'il contribue à ouvrir de nouvelles voies quant aux possibilités d'améliorer le rendement global d'une chaîne logistique se basant sur des outils de recherche opérationnelle.

Dans le premier chapitre, nous avons mené une étude bibliographique qui touchent aux différents domaines liés aux travaux de cette thèse, c'est-à-dire : les chaînes logistiques, l'ordonnancement et le juste à temps, et l'ordonnancement et la coopération dans les chaînes logistiques. Bien sûr, notre recherche n'est pas exhaustive, mais nous avons essayé néanmoins d'être le plus complet possible pour permettre à des lecteurs non spécialistes ni en chaînes logistiques, ni en ordonnancement de pouvoir comprendre l'environnement global de l'étude. Parfois, nous avons détaillé des notions pas en liens directs ou apparents avec le problème proprement étudié, je pense notamment aux mesures de performances et aux systèmes d'aides à la décision (sections 1.2.7, 1.2.8 et 1.2.10). Nous avons jugé utile d'introduire ces notions pour élargir le champs de vision du lecteur non spécialiste afin d'avoir une idée globale sur les enjeux et les buts finaux du développement d'outils d'optimisation au niveau opérationnel. En clair, c'est pour ne pas dissocier les objectifs des niveaux tactique et stratégique, avec ceux au niveau opérationnel. En outre, notre étude montre la relative pauvreté de la littérature scientifique quant aux études prônant la coopération et la coordination des activités, ou entre les partenaires de la chaîne logistique, à un niveau opérationnel. On trouvera généralement un ensemble de règles pour cadrer cette coopération. Néanmoins, pour rendre cette belle idée de

collaboration effective, on a besoin d'outils puissants pour la mettre en œuvre dans les différentes opérations quotidiennes.

Après avoir introduit les notions liées aux problématiques de la thèse, nous introduisons dans le chapitre 2 le problème constituant la majeure partie de la thèse. A notre connaissance, un tel problème n'a pas été étudié dans la littérature concernant les chaînes logistiques. Dans ce chapitre, nous définissons d'abord l'architecture globale de la chaîne logistique étudiée. Les partenaires, leurs rôles, les relations entre eux, et leurs relations avec des entités externes à la chaîne y sont exposés. Pour bien le comprendre, nous l'avons illustré sur un exemple d'un système de production à deux étages où les opérations sont traitées dans un système de flowshop. Nous avons donné ensuite les différents problèmes de juste à temps qu'une telle problématique induit. A ce niveau de l'étude, nous avons proposé une approche de négociation entre les partenaires de la chaîne logistique basée sur la notion de fenêtres de temps virtuelles, permettant d'organiser le fonctionnement du système de production en tenant compte des interactions, et des règles de précédences entre les opérations, ainsi qu'établir des contraintes quant aux dates livraisons et de disponibilité des différents composants et/ou produits finis. Le processus de négociation proposé ne peut être validé que par des simulations qui ne font pas l'objet de cette thèse.

Dans le troisième chapitre nous avons proposé une approche de résolutions pour les problèmes de juste à temps définis dans le précédent chapitre. Notre méthode est générique, et est valable pour le cas le plus général de problèmes de juste à temps. Elle est basée sur une approche par décomposition, et par l'utilisation de deux types de méthodes : une méthode exacte, et une méthode approchée. En effet, nous parcourons l'espace des solutions avec une méta heuristique, et nous évaluons la solution choisie à un moment donnée par une méthode exacte, ce qui permet de donner la meilleure solution possible par rapport à un ensemble déterminé (ou fixé) de données. Quand à la méthode exacte, il s'agit du Pert coût. Nous l'avons adapté à notre problème, tout en proposant un nouvel algorithme pour accélérer son déroulement basé sur les caractéristiques du problème étudié. Pour les métaheuristiques, nous avons choisis d'en utiliser deux : le recuit simulé et les algorithmes génétiques. Nous avons utilisé les algorithmes génétiques d'une façon plus importante, tandis que le recuit simulé nous a servi de repère pour faire une évaluation des performances de l'algorithme génétique. Au lieu d'essayer d'imaginer de nouveaux schémas d'algorithmes génétiques, nous nous sommes intéressés à développer de nouveaux opérateurs plus spécifiques et plus adaptés à notre problème que nous avons utilisé dans un schéma élitiste. Les performances de la méthode du Pert Coût, ainsi que les performances des métaheuristique sont évaluées dans le quatrième chapitre sur des instances de données que nous avons générés nous mêmes d'une manière qui puisse permettre de plus ou moins faire une comparaison honnête entre la génération de solutions aléatoires, et l'utilisation des métaheuristiques.

Enfin, dans le cinquième chapitre nous avons étudié un panel de problèmes d'ordonnancement des transports et de la production pour des chaînes logistiques spécifiques, et pour certains de ces problèmes, nous avons proposé des méthodes de résolutions exactes. Ces outils peuvent être intégrés dans des outils plus généralistes afin de proposer des optimisations locales là où cela peut donner de bons résultats sans pour autant influencer négativement d'autres mesures de performances à d'autres niveaux de la chaîne. En outre, nous avons mis l'accent sur la nécessité de développer des outils pour mettre en œuvre la coordination entre les différents maillons de la chaîne, ou du moins entre les maillons qui se suivent, et qui ont donc des relations très fortes.

6.2 Perspectives

Comme nous l'avons dit plus haut, nous espérons que notre travail ouvrira de nouvelles voies quant à l'étude et la recherche pour le développement d'outils puissants pour la mise en œuvre de la coordination et la coopération au niveau opérationnel de la chaîne logistique. Les modèles présentés et étudiés dans cette thèse peuvent servir de base pour de futures études qui s'intéresseraient au développement d'algorithmes et outils pour le juste à temps, et pour coordonner au niveau opérationnel les efforts entre fonctions de la chaîne logistiques, ou bien entre partenaires.

Comme perspectives immédiates, il serait très intéressant d'approfondir les expériences concernant l'utilisation des algorithmes génétiques afin de déterminer les meilleurs opérateurs. Mesurer les performances de chaque opérateur relativement à la structure de données à laquelle il est appliqué, peut nous mener vers des schémas d'algorithmes génétiques dynamiques où l'utilisation des opérateurs dépendra de la nature des données, et donc permettra de maximiser à chaque instant l'utilisation du meilleur opérateur.

Après l'attribution des fenêtres virtuelles (soft et hard) initiales, les ordonnancements des différents centres de décisions deviennent indépendants, car les dépendances entre eux ont été prises en compte à travers les fenêtres de temps. Chaque centre de production peut utiliser une méthode de résolution exacte ou approchée pour calculer l'ordonnement en juste à temps pour minimiser la somme des pénalités d'avances et de retards. Nous avons traité dans cette thèse la première partie du processus global de négociation.

Une autre perspective immédiate serait d'analyser le système global de négociations. Après chaque étape (itération) du processus itératif (de négociation), une analyse des solutions calculées par les différents centres de production s'avère indispensable pour prendre des décisions pour orienter le système global vers la cohérence et vers des solutions de meilleures qualité. Nous pouvons distinguer quatre cas :

- 1) **Premier cas** : si les dates de disponibilité et de livraisons souhaitées (promises et agréées avec les fournisseurs et clients externes) sont respectées, et que toutes les contraintes de précédences entre les différents centres de production sont respectées - compatibilité des ordonnancements- (i.e. quand une opération est en retard par rapport à ses dates virtuelles alors les opérations qui la suivent sont au minimum aussi en retard qu'elle, ou quand elle est en avance, alors les opérations qui la précèdent sont au moins aussi en avance qu'elle), alors l'ordonnement global est admissible et on peut arrêter le processus de négociation. Dans ce cas, on est dans le cas idéal.
- 2) **Deuxième cas** : si les dates de disponibilité et de livraisons impératives (promises et agréées avec les fournisseurs et clients externes) sont respectées, et que les ordonnancements des différents centres de production sont compatibles (vérifiant les contraintes de précédence), mais que les dates souhaitées de livraison ou de disponibilité ne sont pas respectées, alors on peut soit i) renégocier de nouvelles dates souhaitées, soit ii) augmenter les pénalités associées aux avances et retards par rapport aux partenaires externes dans l'espoir de trouver de meilleures solutions admissibles.
- 3) **Troisième cas** : si les dates de disponibilité et de livraisons impératives (promises et agréées avec les fournisseurs et clients externes) sont respectées, et que les ordonnancements induisent des incompatibilités entre les différents centres de production, alors il faut modifier les fenêtres de temps virtuelles internes au système pour garantir la convergence vers des solutions admissibles. Dans une approche de pilotage semi-décentralisé, l'ordre dans lequel les fenêtres de temps sont négociées

peut être organisé soit dans le sens direct depuis les fournisseurs vers les clients, ou bien dans le sens rétrograde. Cette organisation est un moyen de garantir la convergence du processus global.

- 4) **Quatrième cas** : si les dates impératives de disponibilité ou bien de livraisons ne sont pas respectées, alors ces contraintes doivent être relaxées légèrement et un nouveau cycle de négociation est relancé. Ce cas peut être dû au fait qu'il n'existe pas de solutions admissibles pour le problème global.

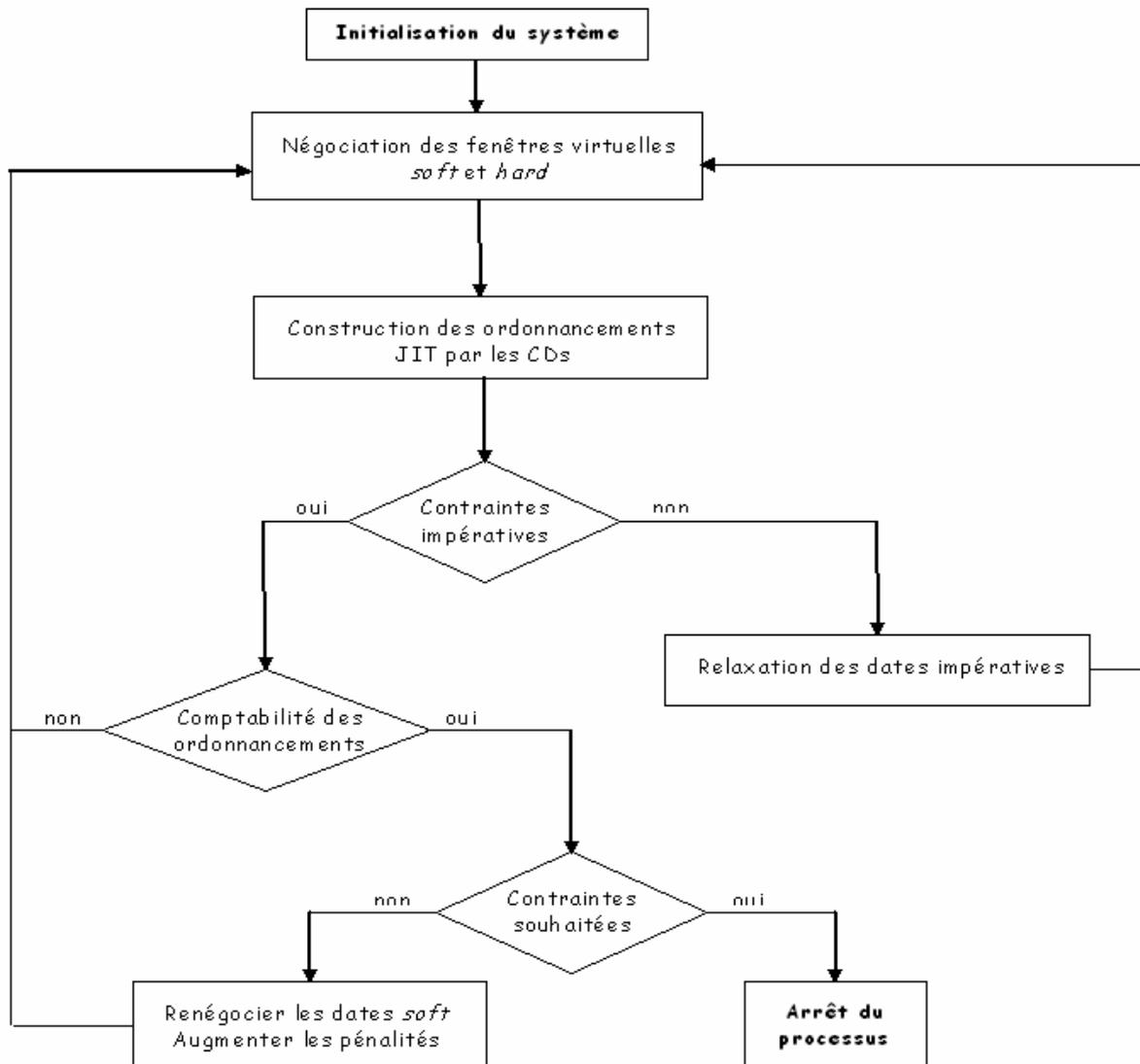


Figure 6.1 fonctionnement du processus de négociation.

La figure précédente montre un schéma possible du processus de négociations. Ce schéma ne peut être validé et analysé qu'à partir de simulations. L'utilisation de la simulation est très fréquente dans le milieu des entreprises, ce qui en fait un outil puissant capable d'être intégré et utilisé dans le monde réel sans rencontrer beaucoup de réticences de la part des industriels et commerciaux.

En outre, comme perspectives futures pour la poursuite de notre travail, nous pouvons suggérer l'ajout de nouvelles contraintes, comme par exemple tenir compte des aléas qui peuvent apparaître au cours de la production tel que les pannes. Tenir compte de la maintenance peut s'avérer très intéressant surtout lorsqu'on sait que ce genre d'aléas est fréquent dans le milieu industriel.

Aussi, nous pouvons ajouter d'autres objectifs dans le calcul des ordonnancements en juste à temps pour tenir compte d'autres contraintes réalistes, comme le niveau de stock (dans chaque entrepôt ou atelier) et d'établir simultanément les plannings de livraisons. Cette approche globale, également appelée multi échelons, serait difficile à mettre en œuvre surtout lorsqu'il s'agit de chaînes logistiques de très grandes tailles, mais c'est ce qui rend le challenge très intéressant, car les bénéfices d'un tel système capable de prendre en considération la majeure partie des contraintes réalistes et de les intégrer dans le calcul d'ordonnancements au niveau opérationnel permettant d'optimiser la totalité de la chaîne serait une avancée importante qu'utiliseraient les progiciels inclus dans les ERP et les APS.

Annexe 1

Annexe 1

1. Illustration du Pert Simple

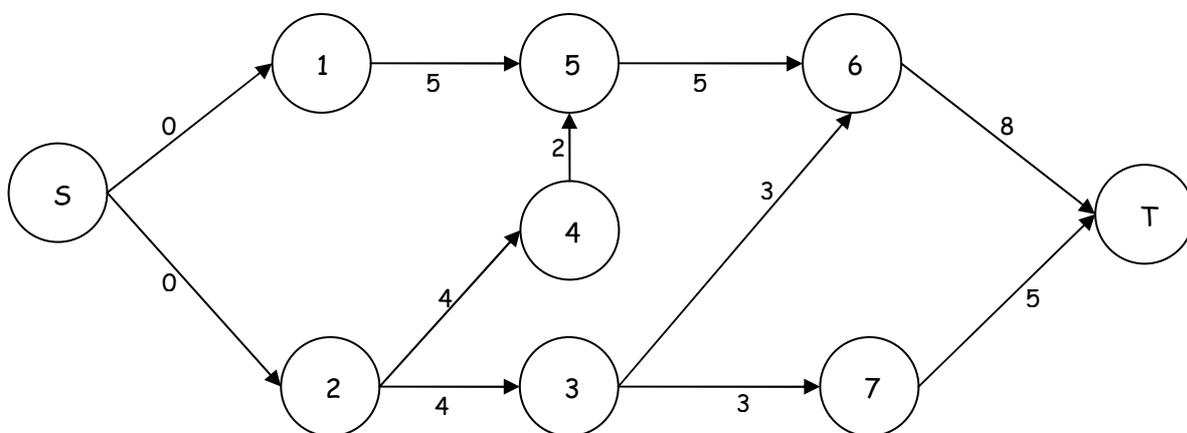
Nous allons illustrer les différentes étapes de cette méthode avec un exemple. Considérons les données suivantes :

Tâches	Durée	Tâches qui précèdent	Tâches qui suivent
1	5	-	5
2	4	-	3,4
3	3	2	6,7
4	2	2	5
5	5	1,4	6
6	8	3,5	-
7	5	3	-

Pour clarifier les données des problèmes d'ordonnancement, la théorie des graphes est un support des plus puissants. Dans la méthode du Pert, on rajoute deux tâches fictives :

- La tâche du début de projet (source du graphe).
- La tâche de fin du projet (puit du graphe).

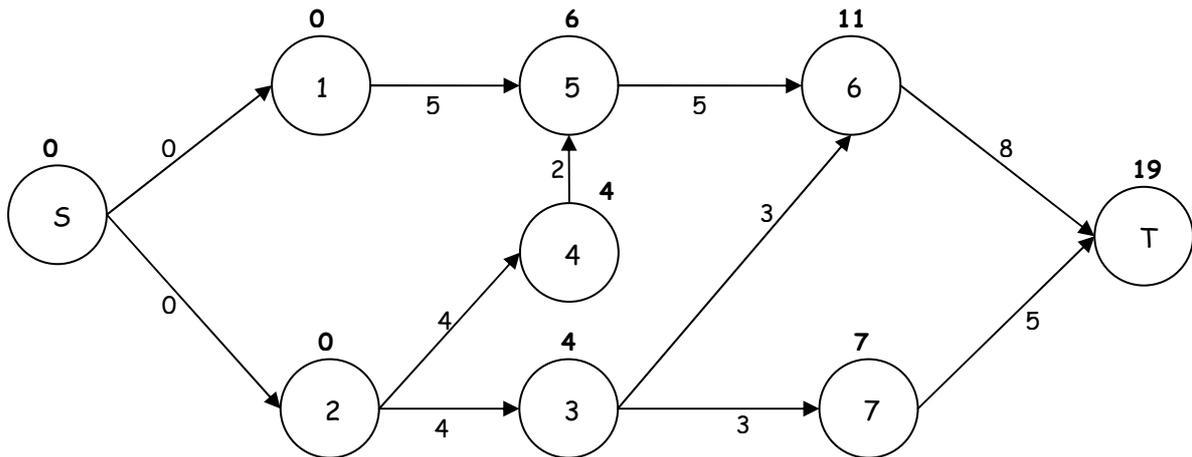
Chaque nœud du graphe représente une activité (tâche) du projet, et est atteignable par un arc venant d'un nœud qui le précède. Chaque arc a une valeur égale à la durée de la tâche (nœud) dont il est originaire. La figure suivante montre le graphe correspondant au tableau précédent.



L'étape suivante est d'établir l'ordonnancement au plus tôt (calé gauche). On calcule la date de début au plus tôt de chaque tâche. C'est à dire la date à partir de laquelle on peut

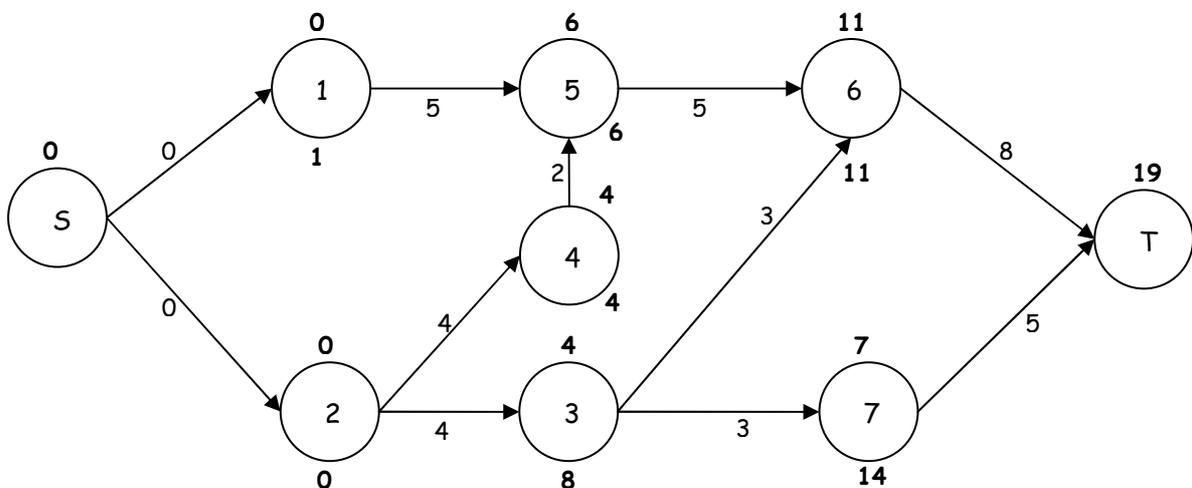
commencer l'exécution de cette tâche en tenant en compte les tâches qui la précèdent. Pour cela, on parcourt le graphe de gauche à droite et on calcule le chemin le plus long depuis le début du projet jusqu'à cette tâche. La date au plus tôt à un nœud donné est égale au maximum entre les dates au plus tôt parmi tous ses prédécesseurs en ajoutant la durée de chaque tâche. Il s'agit du calcul du plus long chemin du début du projet jusqu'à chacun des nœuds.

Les dates au plus tôt pour notre exemple sont données par la figure suivante (les valeurs et en gras et en haut de chaque nœud).



La durée minimum de réalisation du projet est donnée par la date de début au plus tôt sur la tâche de fin. Dans notre exemple, il s'agit de 19.

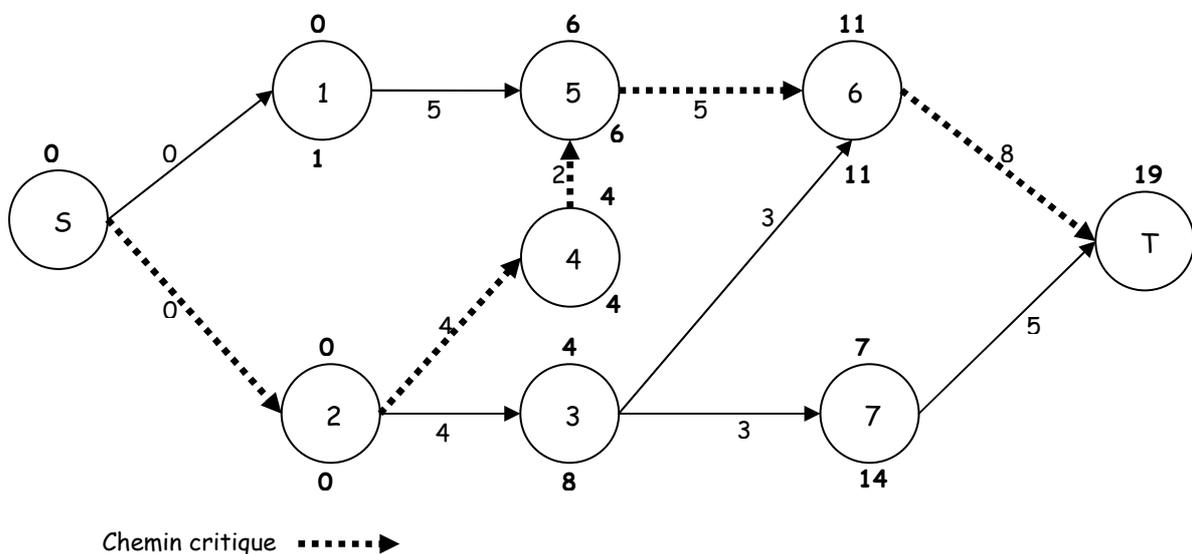
La troisième étape est de calculer l'ordonnancement au plus tard (calé droit). Il s'agit de calculer la date à laquelle la tâche doit impérativement commencer pour ne pas retarder tout le projet. A partir de la date de fin au plus tôt, et en parcourant le graphe de droite à gauche, on calcule la date de début au plus tard pour chaque nœud qui est égale au minimum entre les dates au plus tard des nœuds successeurs dont on soustrait la durée de la tâche dont on calcule la date au plus tard. Les dates au plus tard pour notre exemple sont montrées par la figure suivante (en gras et en bas de chaque nœud).



La différence entre la date de début au plus tôt et la date de début au plus tard est appelée *marge totale*. Elle « mesure le degré de liberté dont on dispose pour programmer cette tâche sans remettre en cause la durée de l'exécution minimale du projet » (Giard, 2003).

La différence entre la plus petite des dates au plus tôt des tâches suivantes et la date au plus tôt de la tâche courante ajoutée à sa durée, est appelée *marge libre*. Elle représente le délai de retard maximum que l'on peut se permettre pour l'exécution d'une tâche sans pour autant retarder les tâches suivantes.

Les tâches dont la date de début au plus tôt est égale à la date de début au plus tard sont appelées *tâches critiques*. Elles n'ont aucune marge de manœuvre concernant leur ordonnancement. Un chemin du début du projet jusqu'à sa fin et ne passant que par des tâches critiques est dit *chemin critique*. Le retard pris sur la réalisation des tâches sur ce chemin entraînera obligatoirement un retard sur la fin de l'ensemble du projet. Sur un chemin critique, la différence entre les dates au plus tôt de deux tâches successives doit être égale à la durée de la tâche qui précède l'autre. Il peut exister plusieurs chemins critiques. Le chemin critique, est en fait le plus long chemin dans le graphe.



À la fin de cette étape, on a toutes les informations nécessaires pour ordonnancer le projet.

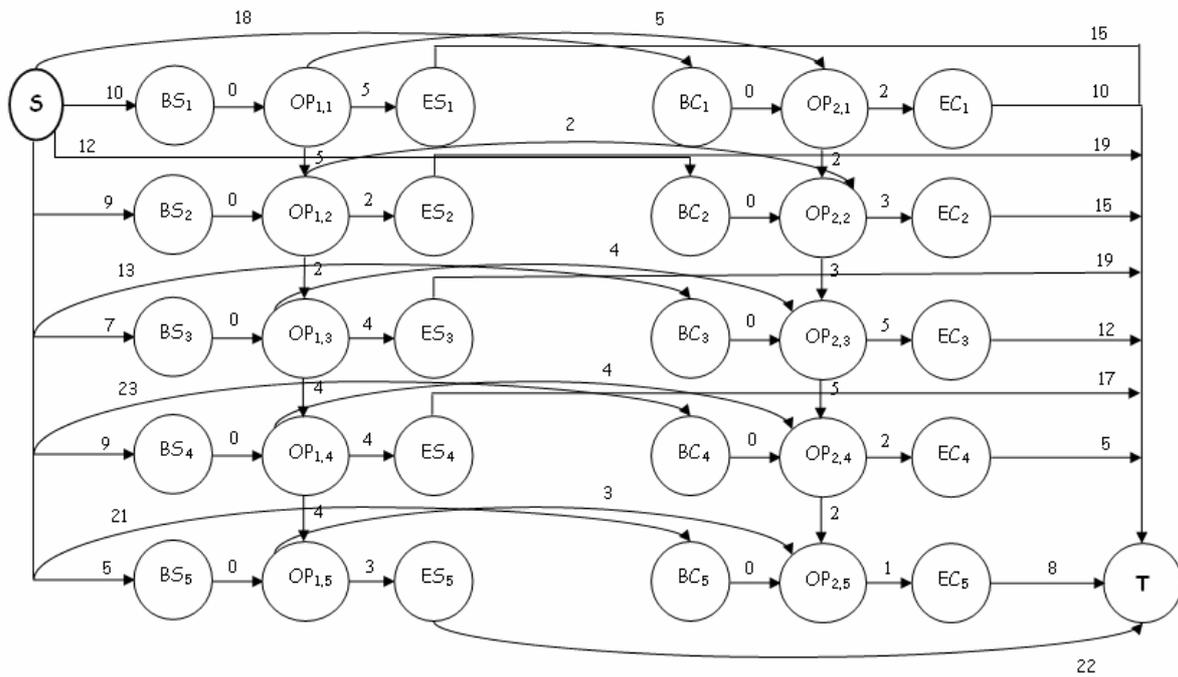
2. Exemple

Dans cette section, nous allons dérouler notre exemple jusqu'à obtenir la solution de l'ordonnancement final. Nous nous sommes arrêtés à la fin de la première itération pour laquelle nous avons calculé les ordonnancements au plus tôt et au plus tard, les chemins critiques, et la coupe minimale. Dans ce qui suit nous ne donnons que le résultat du pas de compression pour raison de lisibilité de l'exemple (mettre à chaque fois les six graphes et les calculs pour chacun de leurs nœuds peut être source de confusion et d'ambiguïté pour le lecteur).

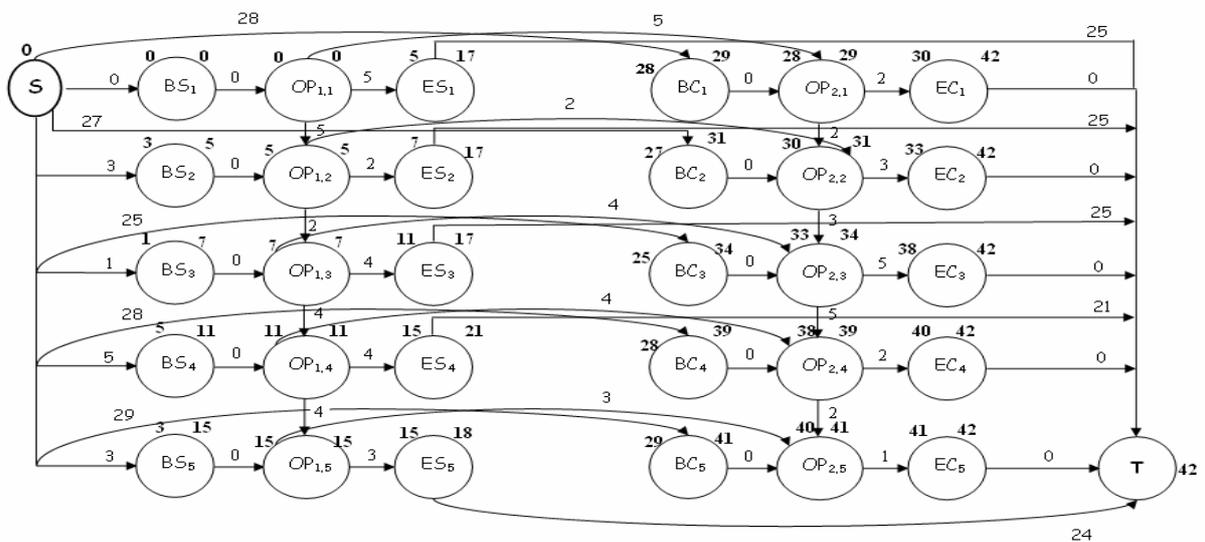
Nous illustrons la construction du graphe du Pert coût par l'exemple suivant. Les données sont décrites par le tableau suivant.

i	$P_{i,1}$	$P_{i,2}$	\bar{r}_i	r_i	α_{ri}	β_{ri}	α_{di}	β_{di}	d_i	\bar{d}_i
1	5	2	0	10	2	1	1	2	20	30
2	2	3	3	9	3	2	1	3	15	30
3	4	5	1	7	3	0	0	5	18	30
4	4	2	5	9	4	3	2	3	25	30
5	3	1	3	5	2	1	1	2	22	30

C'est un exemple avec deux machines et cinq jobs. Nous considérons la permutation suivante : 1, 2, 3, 4, 5. Le graphe Pert correspondant est donné par la figure suivante :

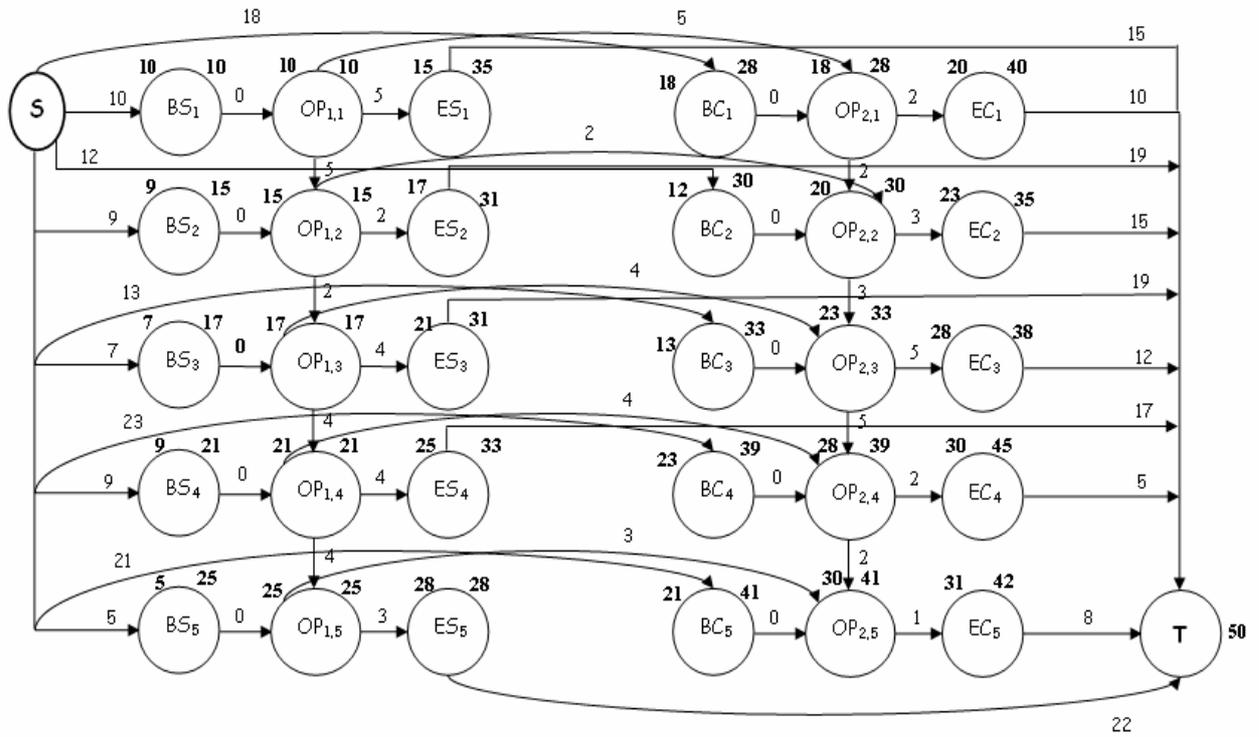


Initialisation :

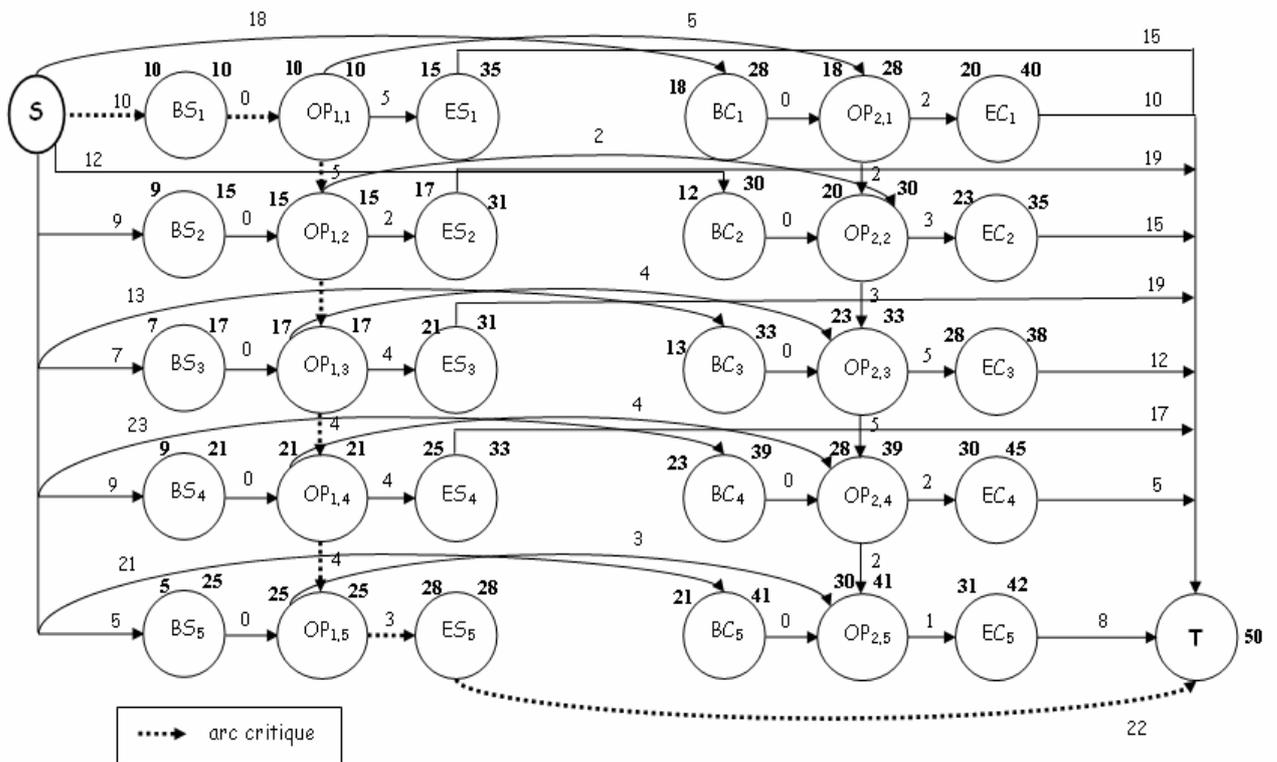


Graphe correspondant au Pert simple.

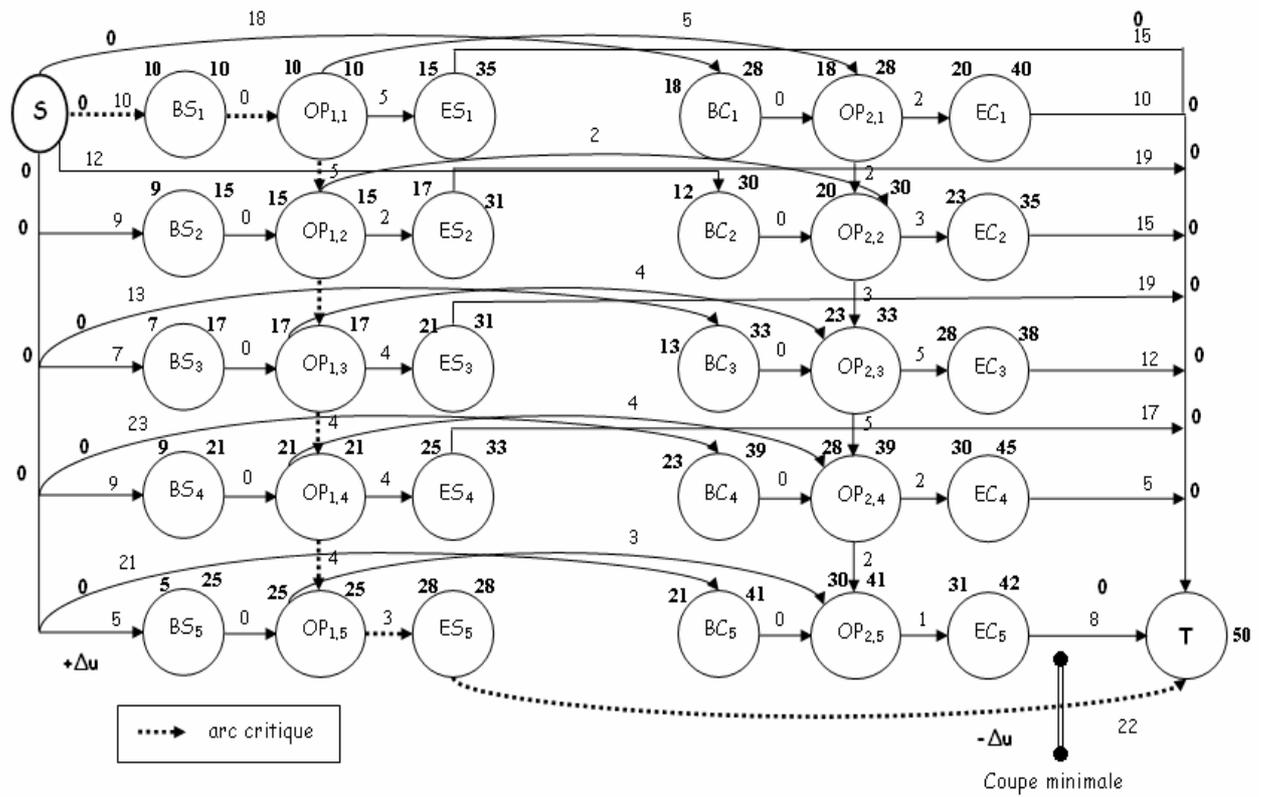
Itération 1 :



Les dates au plus tôt et les dates au plus tard.



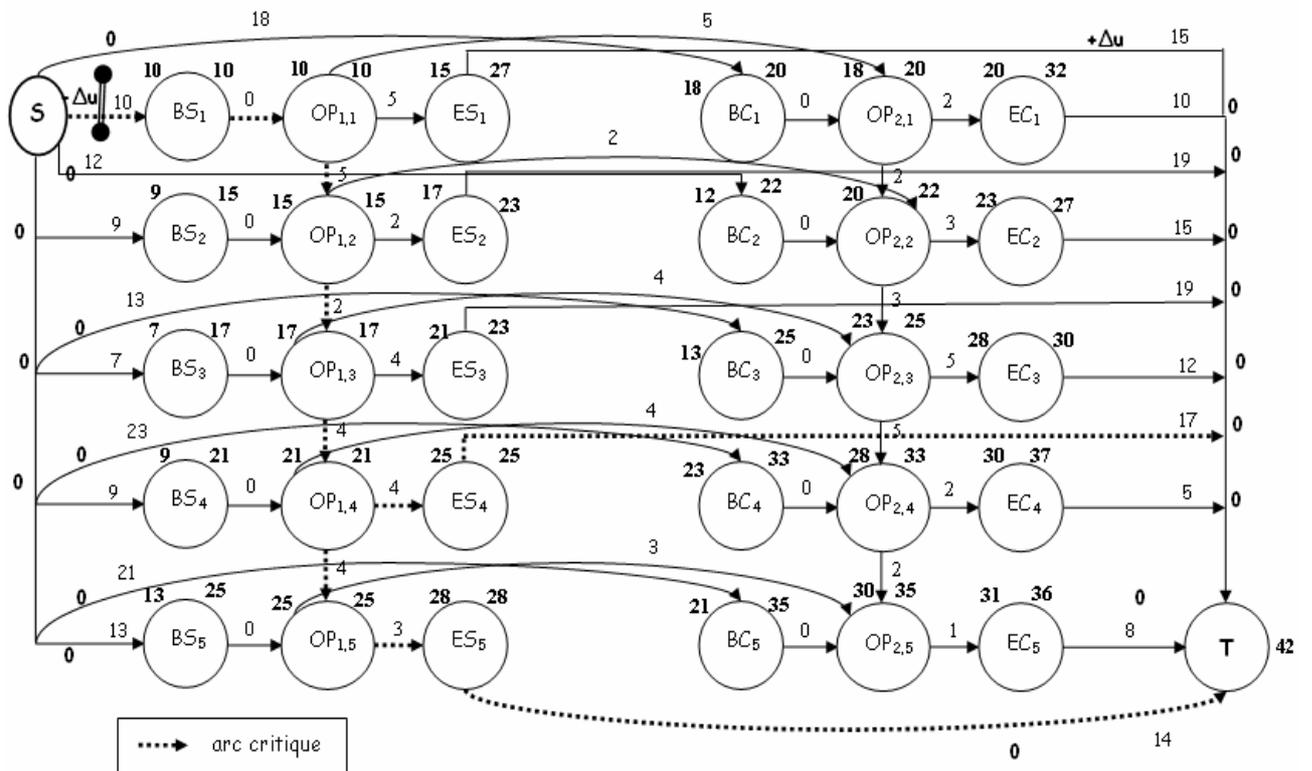
Chemins critiques.



Coupe minimale

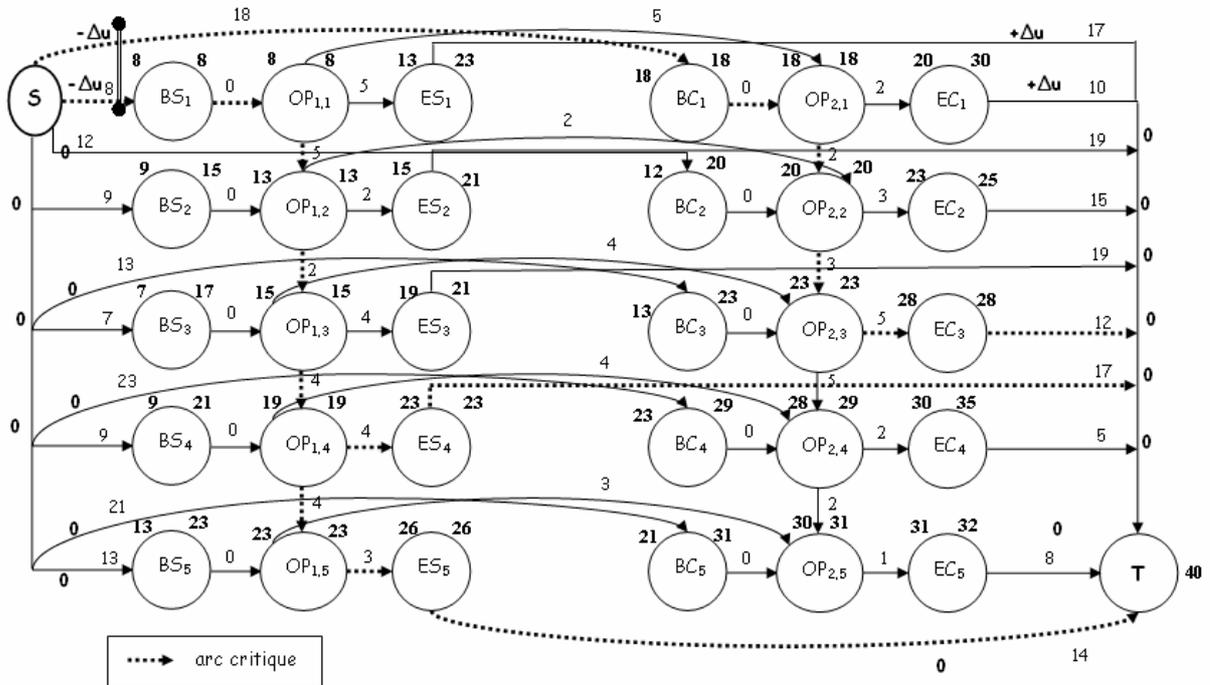
Le pas de compression pour cette itération est : $\Delta u = 8$.

Itération 2 : la figure suivante montre l'état du graphe après la deuxième itération.



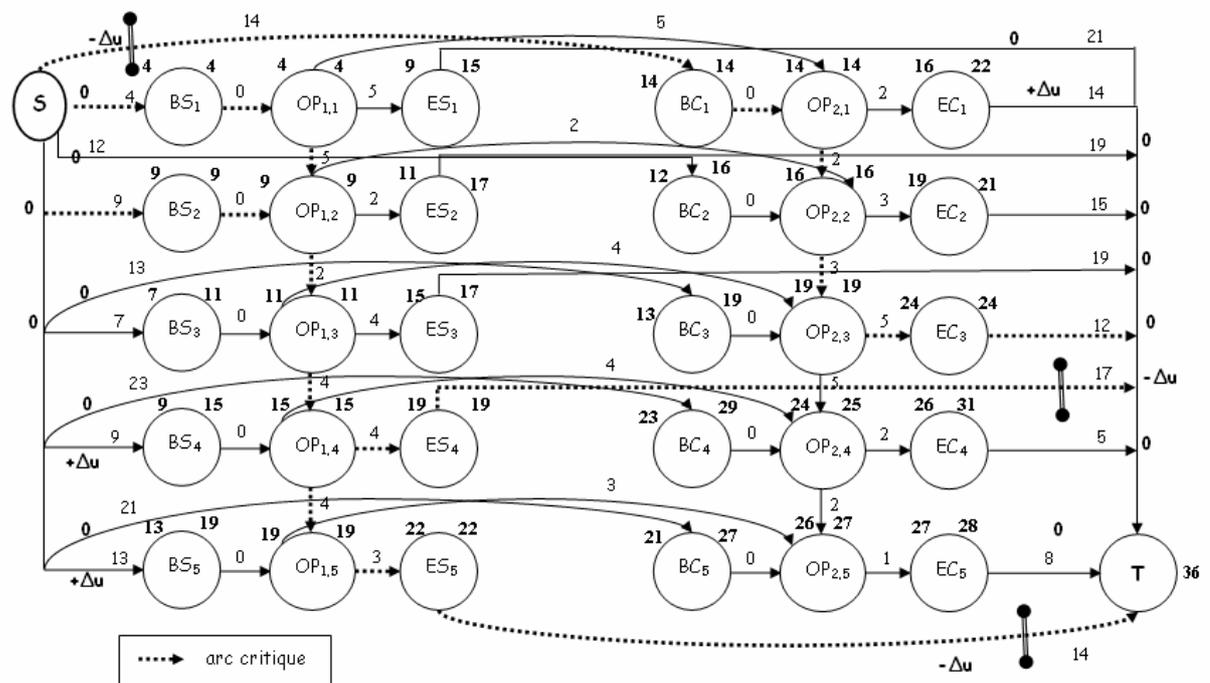
La coupe minimale consiste à avancer le début du travail 1. Le pas de compression est égal à 2.

Itération 3 :



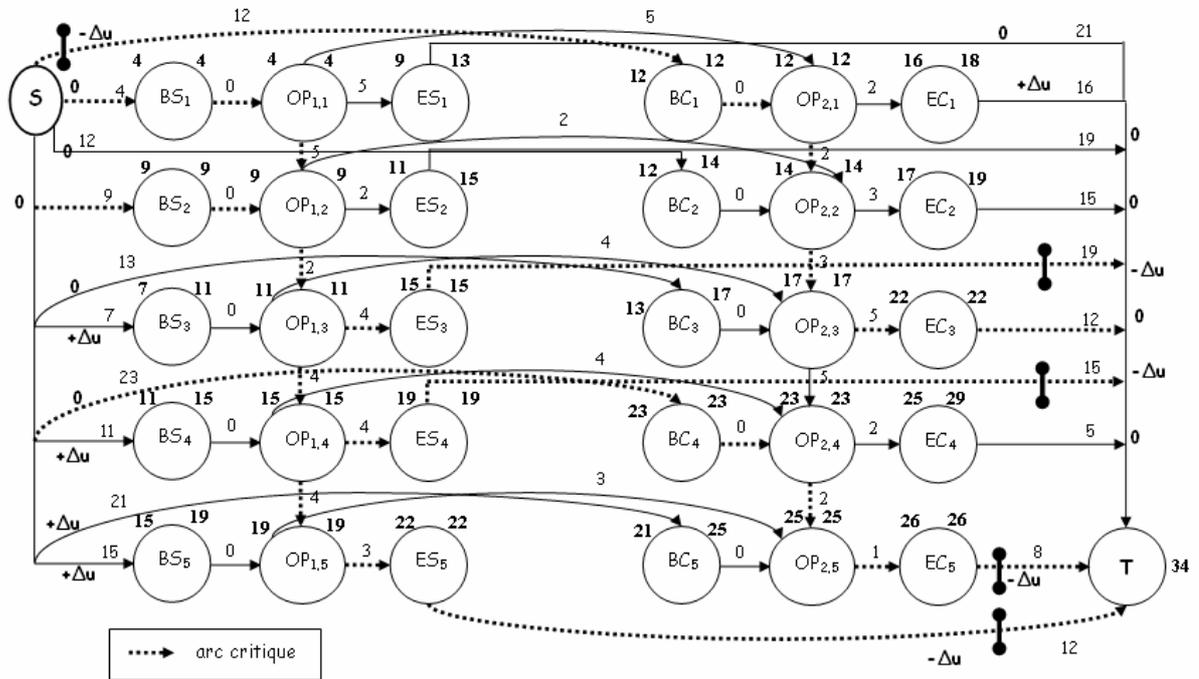
La coupe minimale consiste à avancer le début du travail 1, et d'avancer la fin du travail 1. Le pas de compression est égal à 4.

Itération 4 :



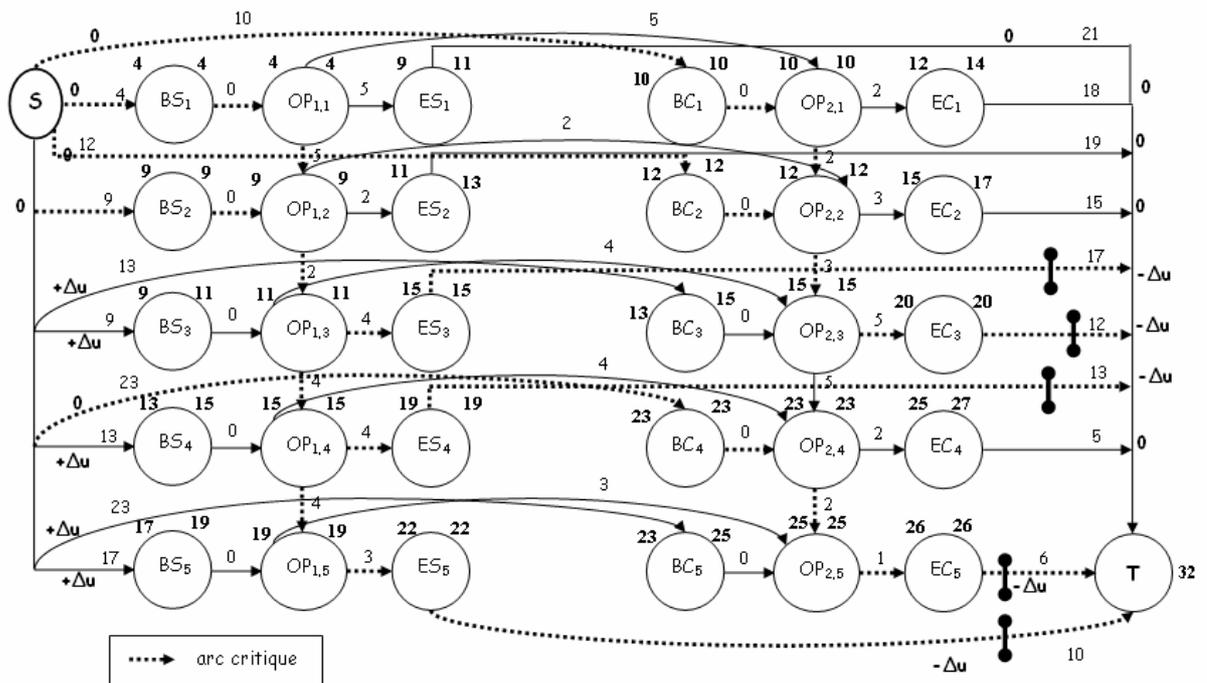
La coupe minimale consiste à avancer la fin du travail 1, et à reculer le début des travaux 4 et 5. Le pas de compression pour cette itération est égal à 2.

Itération 5 :



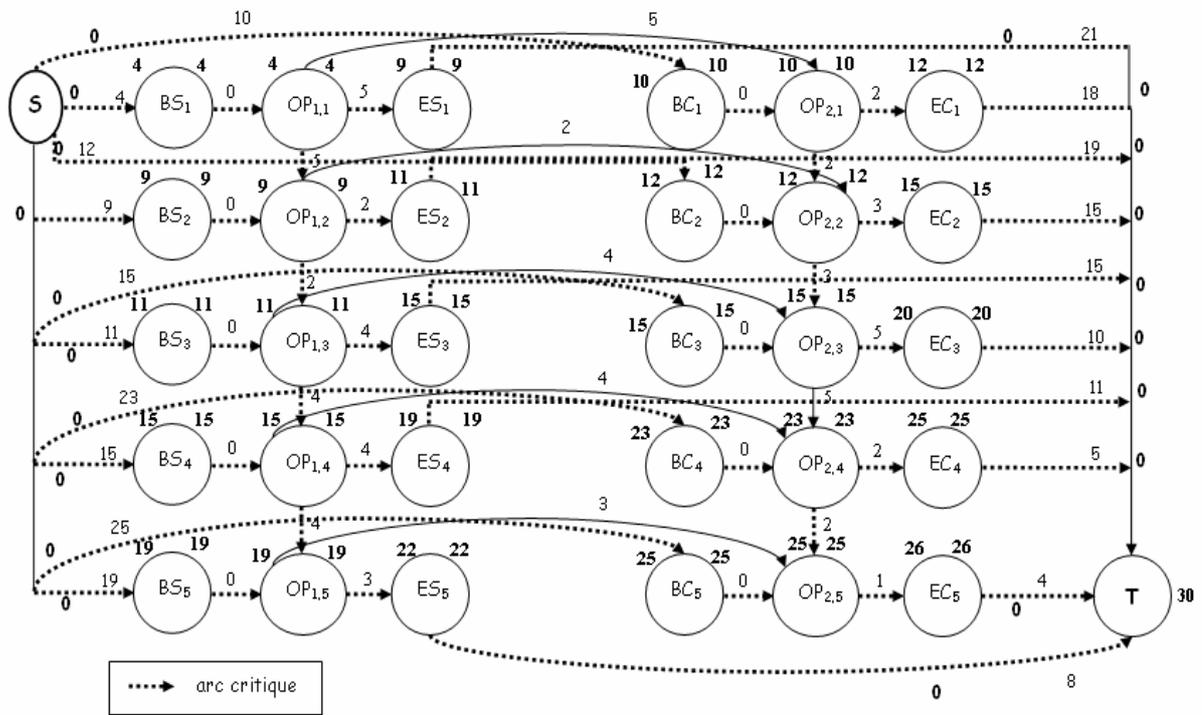
La coupe minimale consiste à avancer la fin du travail 1, et à reculer le début des travaux 3, 4, et 5, et à reculer la fin du travail 5. Le pas de compression de cette itération est égal à 2.

Itération 6 :



La coupe minimale consiste à reculer le début des travaux 3, 4, et 5, et à reculer la fin des travaux 3 et 5. Le pas de compression pour cette itération est égal à 2.

Itération 7



La longueur totale de l'ordonnancement est égale à 30 qui est égale à la longueur de l'horizon (d_{max}). Le pas de compression devient nul et l'algorithme s'arrête.

Bibliographie

Aggoune R., (2002). Ordonnancement d'Ateliers sous Contraintes de Disponibilité des Machines. Thèse de doctorat de l'université de Metz.

Aggoune R., (2004). Minimizing the makespan for the flow shop scheduling problem with availability constraints. *European Journal of Operational Research* (153) 534–543.

Agnētis A, Hall N.G and Pacciarelli D, (2006) Supply chain scheduling: Sequence coordination. *Discrete Applied Mathematics*, 154-15, 2044-2063

Barbarosoglu G., et Yazgac T., (1997). An application of the analytical hierarchy process to the supplier selection problem. *Production and Inventory Management Journal*, vol 38 (1), 14-21.

BÄulbÄul, K., Kaminsky, P., and Yano, C. (2004). Flow Shop Scheduling with Earliness, Tar-diness and Intermediate Inventory Holding Costs. *Naval Research Logistics*, 51(3):407-445.

Bhasharan S., (1998). Simulation Analysis of a Manufacturing Supply Chain. *Decision Sciences*, vol(29), 633-657.

Bourland K.E., Pwell S.G., Pyke D.F., (1996). Exploiting timely demand information to reduce inventories, *European Journal of Operational research* (92) 239-243.

Browne J., Sackett P., Wortmann J.C., (1995). Futur manufacturing systems- toward the extended entreprise, *Computer in Industry* (25).

Camirenelli E., et Cantu A., (2006). Measuring the Value of the Supply Chain: A Framework. *Supply Chain Practice* Vol. 8. No. 2.

Carlier J., et Chrétienne P., (1988). Problèmes d'ordonnancement (modélisation/complexité/ algorithmes). Collection Etudes et Recherches En Informatique, Masson Editions.

Chang Y., and Lee C.,(2004) Machine scheduling with job delivery coordination. *European Journal of Operational Research*, 158, 470-487.

Chen Z.L., and N.G. Hall (2001) Supply chain scheduling: Assembly systems. Working Paper, Department of Systems Engineering, University of Pennsylvania.

Chopra S., et Meindil P., 2007. Supply chain management: Strategy, planning, and Operations, third edition, Pearson Education, Inc. New Jersey.

Christopher, M.L., (1992). Logistics and supply chain management. Pitman Publishing, London.

Chu C., Portmann M.C., Proth J.M. (1992) A splitting up approach to simplify job-shop scheduling problems. *International Journal of Production Research*, 30-4, 859-870.

Cooper, M.C., D.M. Lambert and J.D. Pagh (1997). Supply chain management more than a new name for Logistics. *The International Journal of Logistics Management*, 8, No.1, 1-14.

Courtois A., Martin-Bonnefous C., Pillet M., (1996). « gestion de production », les éditions d'organisation, troisième édition, Paris.

D'amours S., Montreuil B., Lefrançois P., Soumis F., (1999). Networked Manufacturing : The impact of information sharing, *International Journal of Production Economics*, (58) 63-79.

Davis L., (1985). A Generalized permutation approach to job shop scheduling with genetic algorithms. *OR Spektrum*, 17, 87-92.

Dawande, M., Neil Geismar, H. and Hall N.G., (2006). Supply chain scheduling: distribution systems, *Production and Operations Management*, (15)2, 243-261.

Desrosiers P., (2006). Gestion des opérations et de la production : Planification et contrôle de la production – l'approche juste à temps, cours de HEC Montréal.

Ding H., (2004). Une Approche d'Optimisation Basée sur la Simulation pour la Conception de Chaînes Logistiques : Application dans les Industries Automobiles et Textiles. Thèse de doctorat, Université de Metz.

Dudek, G. and H. Stadtler (2005) Negotiation-based collaborative planning between supply chains partners. *European Journal of Operational Research*, 163, 668–687.

Engles D.W., (1978). Alexander the great and the logistics of the Macedonian Army, Los Angeles, CA, University of California Press.

Ertogral K., Darwish M., Ben-Daya M., (2007). Production and shipment lot sizing in a vendor-buyer supply chain with transportation cost, *European Journal of Operational Research*, vol. 176, 1592-1606.

Falkenauer E., Bouffouix S., (1991). A genetic algorithm for job shop. *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, 1, 824-829.

Fondrevelle J., (2005). Résolution exacte de problèmes d'ordonnancement de type flowshops de permutation en présence de contraintes d'écarts temporels entre opérations. Thèse de doctorat de l'institut National Polytechnique de Lorraine, Nancy.

Ganeshan, Ram, and Terry P. Harrison, (1995). An Introduction to Supply Chain management. Department of Management Sciences and Information Systems, 303 Beam Business Building, Penn State University, University Park, PA, USA.

Giard, V. (2003). Gestion de la production et des flux. 3^{ème} édition, édition Economica.

Glover F., (1989). Tabu Search – Part I, *Journal on Computing* 1, 190–206.

Glover F., (1990). Tabu Search – Part II, *Journal on Computing* 2, 4–32.

Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley.

Goldratt E., (2007). *Le But*, Editions Eyrolles, 3^{ème} édition.

Govil M., et Proth J.M., (2002). Supply chain design and management: strategic and tactical perspectives. Academic Press, San Diego, California, USA.

Graham R.L., Lawler E.L., Lenstra J.K., et Rinooy Kan A.H.G., (1979). Optimization and approximation in deterministic sequencing and scheduling theory: a survey. *Annals of Discrete Mathematics*, (5) 287-326.

Gunasekaran A., Patel C., McGaughey R.E., (2004). A framework for supply chain performance measurement. *International Journal of Production Economics* (87) 333–347.

Gupta D., and Weerawat W., (2006). Supplier manufacturer coordination in capacitated two stage supply chains, *European Journal of Operational research* (175), 67-89

- Hadfield R.B., Bechtel C., (2002). The role of trust and relationship structure in improving supply chain responsiveness, *Industrial marketing Management* (27) 1219-1227.
- Hall, N.G., and C.N. Potts, (2005). The coordination of scheduling and batch deliveries, *Annals of operations research* (135) 41-64.
- Hall, N.G. and C.N. Potts (2003) Supply Chain Scheduling : Batching and Delivery. *Operations Research*, 51, 566-584.
- Harmon R.L., (1992). Reinventing the factory 2. The free Press.,
- Hassin R., et Shani M., (2005). Machine scheduling with earliness, tardiness and non execution penalties, *Computers and Operations Research* (32), 683-705.
- Hendel, Y., Sourd, F.,(2006a). An improved earliness–tardiness timing algorithm. *Computers & Operations Research*, In Press.
- Hendel, Y., Sourd, F.,(2006b). Efficient neighbourhood search for the one machine earliness-tardiness scheduling problem, *European Journal of Operational Research* (173) 108-119.
- Hermann J.W., Lin E., Pundoor G., (2003). Supply Chain Simulation Modeling Using The Supply Chain Reference Model. *Proceedings of DETC'03*, 1-9, Chicago, USA.
- Holland J.H.,(1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.
- Huang B., Iravani S.M.R., (2007). Optimal production and rationing decisions in supply chains with information sharing, *Operations research Letters* (35)5, 669-676.
- Hugos M., (2003). Essentials of supply chain management. John Wiley and sons, Inc, New Jersey, USA.
- Jain S., Workman R.W., Collins L.M, Ervin E.C., (2001). Development of a High-Level Supply Chain Simulation Model. *Proceedings of the 2001 Winter Simulation Conference*, 1129-1137.
- Jiao J.R., You X., Kumar A., (2006). An agent-based network for collaborative negotiation in the global manufacturing supply chain network, *Robotics and Computer-Integrated Manufacturing* (22) 239-255.
- Jung H., Jeong B., Lee C.G., (2006). An order quantity negotiation model for distributor-driven supply chains, *Intelligent Journal Production Economics*, à paraître.
- Kearney. A.T., (1994). Management approach to Supply Chain Integration, Rapport aux membres de l'équipe de recherche , A.T. Kearney, Chicago.
- Kovacs G.L., Paganelli P., (2003). A planning and management infrastructure for large, complex, distributed projects- beyond ERP and SCM, *Computers in Industry* (51) 165-183.
- Kreipl S., and M. Pinedo (2004) Planning and scheduling in supply chains: An overview of issues in practice. *Production and Operations management*, 13-1, 77-92.
- Lambert, D.M., J.R. Stock, et L.M. Ellram, (1998), *Fundamentals of Logistics Management*, Boston, MA: Irwin/McGraw-Hill, Chapitre 14.
- Lamming R., (1996). Squaring lean supply with supply chain management. *International Journal of Operations and Production Management* (16)2, 183-1996.
- Lee H.L. et Billington C, (1993), Material management in decentralized supply chain", *Operations Research*, vol 41, n°5, p. 835-847.

- Lee, Hau L; Padmanabhan, V. and Whang, Seungjin (1997). "The Bullwhip Effect in Supply Chains". *Sloan Management Review* 38 (3): 93-102.
- Lee Y. L., Jung J. W., Lee K. M. (2006) Vehicle routing scheduling for cross-docking in the supply chain . *Computers & Industrial Engineering*, 51-2, 247-256.
- Li C, Vairaktarakis G., Lee C. (2005). Machine scheduling with deliveries to multiple customer locations. *European Journal of Operational Research*, 164, 39-51.
- Li C.L., and Xiao W.Q., (2004). Lot streaming with supplier-manufacturer coordination. *Naval research Logistics*, (51), 522-542.
- Lin B.M.T., Cheng T.C.E., Chou A.S.C., (2007). Scheduling in an assembly type production chain with batch transfer, *Omega* (35) 143-151.
- Lin F., Lin Y., (2006). Integrating multi-agent negotiation to resolve constraints in fulfilling supply chain orders, *Electronic Commerce Research and Applications* (5) 313- 322.
- Lopez P., et Roubellat F., (2001). Ordonnancement de la production. Hermès science publications, Paris, France.
- Makastoris C., Leach N.P., Richards H.D, Ristic M., Besant C.B., (1996). Addressing the planning and control gaps in semiconductor virtual enterprises, *Proceedings of the conference on Integration in Manufacturing*, 117-129, Galway, Ireland.
- Meijboom B., and Obel B., (2007). Tactical coordination in a multi-location and a multi-stage operations structure : A model and a pharmaceutical company case, *Omega* (35).258-273.
- Ménard J.P., (2005). Concept: le Juste à Temps. *Le Journal Industriel du Québec*.
- Mentzer J.T., Dewitt W., Keebler J.S., Min S., Nix N.W., Smith C.D., Zacharia Z.G., (2001). *Journal of business logistics Management*, vol 22 (2).
- Metropolis, W., A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller (1953), Equation of the State Calculations by Fast Computing Machines, *Journal of Chemical Physics* 21, 1087–1092.
- Moller C., (1995). *Toward design effective logistics systems*, Thèse de doctorat, Aalborg University, Danemark.
- Monteiro T., (2001). Conduite distribuée d'une coopération entre entreprises: le cas de la relation donneurs d'ordres – fournisseurs, Thèse de l'Institut National Polytechnique de Grenoble.
- MonteiroT., Roy D., Anciaux D., (2007). Multi-site coordination using a multi-agent system, *Computers in Industry*, (58) 367-377.
- Mouloua Z., and Oulamara A., (2006). Cooperation in supply chain scheduling: minimizing the inventory holding cost. *The International conference on Information systems, Logistics and Supply chain*. (1), pp 564-573, Lyon.
- Mouloua Z., and Portmann M.C., (2006). Single product scheduling and transportation optimization in a supply chain, *International conference on project management and scheduling*, pp Poznan, Poland.
- Mouloua Z., and Oulamara A., (2007). A dynamic programming approach for minimizing the transportation costs in a supply chain, *The 4th International Federation of Automatic Control Conference on Management and Control of Production and Logistics*, Sibiu, Romania.

- Mouloua Z., and Oulamara A., (2007). Joint optimization of production and transportation in a multiple products, multiple customers supply chain. Papier à soumettre.
- Naso D., Surico M., Turchiano B. and Kaymak U. (2007) Genetic algorithms for supply-chain scheduling: A case study in the distribution of ready-mixed concrete. *European Journal of Operational Research*, 177-3, 2069-2099.
- Nearchou A.C., (2004). Flowshop sequencing using hybrid simulated annealing, *Journal of Intelligent manufacturing* (15) 317-328.
- Neubert R., Gorlitz O., Teich T., (2004). Automated negotiations of supply contracts for flexible production networks, *Int. J. Production Economics* (89) 175-187.
- New S.J. et P. Payne, (1995), "Research framework in logistics : three models, seven dinners and a survey", *International Journal of Physical Distribution and logistics management*, 25 (10).
- Noorul Haq A., Ravindran D., Aruna V., Nithiya S., (2004). A hybridisation of metaheuristics for flowshop scheduling, *International Journal of Advanced Manufacturing Technologies* (24) 376-380.
- Osman I.H., et Laporte G., (1996). Metaheuristics : a bibliography. *Annals of Operations Research*, (63), 513-623.
- Ould Louly M.A., (2001). Optimisation des stocks des composants et planification de leurs réapprovisionnements pour les systèmes d'assemblage soumis à des aléas. Thèse de doctorat de l'université de technologie de Troyes.
- Ouzizi L., (2005). Planification de la production par co-décision et négociation de l'entreprise virtuelle, Thèse de doctorat de l'université de Metz.
- Ouzizi L., Anciaux D., Portmann M-C., Vernadat F. (2006) A model for co-operative planning within a virtual enterprise. *International journal of Computer Integrated Manufacturing*. 19-3, 248-263.
- Paché J., (2004). Proximité spatiale et gestion des supply chains : une approche critique à partir des logiques de radialisation, Quatrième journées de la proximité, juin 17-18.
- Pons J., Chevalier P., (1996). *La logistique intégrée*, P. 34-35, Hermes.
- Portmann M.C., (2006). Chaînes logistiques et gestion de la production, ISDP 32, cours à l'école des mines de Nancy.
- Portmann M.C. (1988) Méthodes de décompositions spatiales et temporelles en ordonnancement de la production. *RAIRO-APII*, 22-5, 439-451.
- Portmann M.C., and Mouloua Z., (2007). A window time negotiation approach at the scheduling level inside supply chains, *3rd multidisciplinary International Scheduling Conference: Theory and Applications*, Paris.
- Pundoor G., Chen Z.-L. (2005) Scheduling a production-distribution system to optimize the tradeoff between delivery tardiness and distribution cost. *Naval research Logistics*, 52-6, 571-589.
- Rajendran C., Ziegler H., (2005). Two ant-colony algorithms for minimizing total flowtime in permutation flowshop, *Computers and Industrial Engineering* (48) 789-797.
- Rota K., Thierry C. et Bel G., (2001). *La maîtrise des flux* (coordonateur : J.P. Campagne) : Chapitre 5 "Gestion des flux dans les chaînes logistiques (Supply Chain Management)", Ouvrage Hermès : Traité Systèmes pour l'ingénieur, Ed. Hermès.

- Ruiz R., Maroto C., Alcaraz J., (2006). Two new robust genetic algorithms for the flowshop scheduling problem, *Omega* (34), 461-476.
- Selvarajah E., Steiner G. (2006). Batch scheduling in a two-level supply chain—a focus on the supplier, *European Journal of Operational Research*, 173-1, 226-240.
- Sérgio M. S. Neiro M., Pinto J.M. (2004), A general modeling framework for the operational planning of petroleum supply chains. *Computers & Chemical Engineering*, 28(6-7), 871-896.
- Shabtay D., and Steiner S., (2006). Two due date assignment problems in scheduling a single machine, *Operations research letters* (34) 6, 683-691.
- Sourd F., (2003). The one machine problem with earliness and tardiness penalties, *Journal of scheduling* (6) 533-549.
- Sourd, F.,(2005). Punctuality and idleness in just-in-time scheduling. *European Journal of Operational Research*, Volume 167, Issue 3, 16 December 2005, Pages 739-751.
- Sourd F., (2006). Dynasearch for the earliness-tardiness scheduling problem with release dates and setup constraints, *Operations Research letters* (34) 591-598.
- Tan K.C., (2001). A framework of supply chain management literature, *European Journal of Purchasing and Supply Management* (7) 39-48.
- Taylor D.A., (2003). *Supply Chains: A Manager's Guide*. Addison Wesley, USA.
- Tayur S., Ganeshan R., M. Magazine. (1999), *Quantitative models for supply chain management*, Kluwer Academic Publishers.
- Thierry C., J. Lamothe et V. Galvagnon, (2004). A re-planning support system to small batch production with booked resources", *International Journal of Production Research*, 42(23), 4993-5008.
- Thonemann U.W., (2002). Production, manufacturing and logistics improving supply chain performance by sharing advance demand information, *European Journal of Operational Research*, (142) 81-107.
- Vidal C.J., et Goetschalckx M., (1997). Strategic production-distribution models : a critical review with emphasis on global supply chain models. *European Journal of Operational Research*, vol (98), 1-18.
- Voss S., Martello S., Osman I.H., Roucairol C., (1999). *Meta-Heuristics – Advance and trends in local search paradigms for optimization*. Kluwer Academic Publishers, Dordrecht, the Netherlands.
- Wang H., Lee C.Y., (2005). Production and transport logistics scheduling with two transport mode choices *Naval Research Logistics*. 52-8, 796-809.
- Wang L., Zheng D.Z., (2003). An effective hybrid heuristic for flowshop scheduling, *International Journal of Advanced Manufacturing Technologies* (21) 38-44.
- Yang Y., Kreipl S., Pinedo M., (2000). Heuristics of minimizing total weighted tardiness in flexible flowshop, *Journal of Scheduling* (3) 89-108.
- Zhao X., Xie J., Leung J., (2002). The impact of forecasting model selection on the value of information sharing in a supply chain, *European Journal of Operational Research*, (142) 321-344.

Liste des tableaux et figures

Figure 1.1 modèle de chaîne logistique selon (new et al, 1995).	9
Figure 1.2 modèle de chaîne logistique (Kearney, 1994).	9
Figure 1.3 chaîne logistique globale.	10
Figure 1.4 chaîne logistique interne. Source (Portmann, ISDP32).	11
Figure 1.5 entreprise étendue. Source (Portmann, ISDP32).	11
Figure 1.6 entreprise virtuelle. Source (Portmann, ISDP32).	12
Figure 1.7 pyramide des niveaux de décisions.	15
Figure 1.8 différence entre les niveaux de décisions. Source (Portmann, ISDP32).	16
Figure 1.9 différence entre les types de modélisations des chaînes logistiques (Taylor, 2003).	23
Figure 1.10 modules d'un ERP (Taylor, 2003).	24
Figure 1.11 modules d'un APS (Taylor, 2003).	25
Figure 1.12 différences entre APS et ERP.	25
Figure 1.13 Chaîne logistique de DELL.	26
Figure 1.14 flux tendus.	38
Figure 2.1 Schéma général de la chaîne logistique.	49
Figure 2.2 les éléments de la supply chain.	50
Figure 2.3 Interactions entre les différents éléments de la supply chain.	50
Figure 2.4 schéma du processus de pilotage semi-décentralisé.	51
Figure 2.5 premier étage du système de production.	52
Figure 2.6 fonctionnement du deuxième étage du système de production.	52
Figure 2.7 interactions entre partenaires et pilotage semi-décentralisé.	55
Figure 2.8 schéma global du processus de négociation.	61
Figure 2.9 dépendances entre les centres de décisions (Portmann et Mouloua, 2007).	62
Figure 2.10 association des fenêtres de temps virtuelles aux jobs (Portmann et Mouloua, 2007).	63
Figure 3.1 Schéma général de la méthode de résolution.	68
Figure 3.2. Schéma général de l'évaluation d'une permutation.	68
Figure 3.3 Algorithme d'évaluation d'une permutation.	72
Figure. 3.4. Modèle du graphe pour chaque opération.	73
Figure. 3.5. Les arcs sortants d'une opération.	73
Figure 3.6. Valeurs sur les arcs.	74
Figure 3.7 Arcs sortants.	75
Figure 3.8. Graphe G^1 .	75
Figure 3.9. Graphe G^2 .	76

Figure. 3.10. Graphe G^3 .	76
Figure. 3.11. Graphe G^4 .	76
Figure 3.12. Graphe G^5 .	77
Figure 3.13. Graphe G^6 .	77
Figure3.14 Itération de base d'un algorithme génétique incrémental.	83
Figure 3.15 Structure générale d'un algorithme génétique incrémental.	84
Figure 3.16. Schéma simple de Goldberg.	85
Figure 3.17. Algorithme de la sélection par tournoi binaire.	86
Figure 3.18. Algorithme de la roulette.	87
Figure 3.19. Exemple d'une procédure de remplacement.	87
Figure 3.20. Exemple de croisement OX.	88
Figure 3.21. Exemple de croisement LOX.	88
Figure 3.22. Croisement 1X.	89
Figure 3.23. Exemple de croisement 2X.	89
Figure 3.24 les croisements basés sur les dates d'exécutions.	91
Figure 3.25. Algorithme du croisement basé sur les coûts des jobs.	91
Figure 3.26 Algorithme du croisement basé sur les positons.	92
Figure 3.27 Schéma élitiste.	93
Figure 3.28 Algorithme du recuit simulé.	94
Figure 4.1 génération des pénalités d'avance et de retard.	99
Figure 4.2 Courbe des durées opératoires par nombre de jobs	103
Figure 4.3 degré de faisabilité des solutions.	104
Figure 4.4 SA Vs AGS Vs AGE ; instance 1.	105
Figure 4.5 SA Vs AGS Vs AGE ; instance 2.	106
Figure 4.6 SA Vs AGS Vs AGE ; instance 3.	106
Figure 4.7 SA Vs AGS Vs AGE ; instance 4.	106
Figure 5.1 chaîne logistique considérée.	110
Figure 5.2 ordonnancement du fournisseur 1.	112
Figure 5.3. Ordonnancement du fournisseur 2.	112
Figure 5.4 cas de deux fournisseurs.	113
Figure 5.5 schéma général du système étudié.	121
Figure 5.6 schéma simplifié mono produit.	122
Figure 5.7 scénario faisable.	124
Figure 5.8 Scénario infaisable.	125
Figure 5.9 exemple de scénario de production.	129
Figure 5.10 Graphe correspondant au problème du tableau 5.9.	132
Figure 5.11 ordonnancement initial.	138

Figure 5.12 ordonnancement 1 dérivé de l'ordonnancement original.	139
Figure 5.13 ordonnancement 2 dérivé de l'ordonnancement original.	139
Figure 5.14 ordonnancement optimal vérifiant la propriété 1.	140
Figure 5.15 ordonnancement dérivé de celui de la figure 5.13.	140
Figure 6.1 fonctionnement du processus de négociation.	150