



**HAL**  
open science

# Chaîne d’approvisionnement: approches stratégique et tactique

Satyaveer Singh Chauhan

► **To cite this version:**

Satyaveer Singh Chauhan. Chaîne d’approvisionnement: approches stratégique et tactique. Sciences de l’ingénieur [physics]. Université de Metz, 2003. Français. NNT : . tel-00007076

**HAL Id: tel-00007076**

**<https://theses.hal.science/tel-00007076>**

Submitted on 11 Oct 2004

**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.

année : 2003

N° attribué par la bibliothèque

--	--	--	--	--	--	--	--	--	--

# THESE

présentée à  
**UNIVERSITE DE METZ**

**Mathématiques, Informatique, Mécanique**

pour obtenir le titre de  
**DOCTEUR**

Spécialité  
Automatique

soutenue par

**Satyaveer Singh Chauhan**

le 17 Septembre 2003

**Chaînes d'approvisionnement :  
approches stratégique et tactique**

Directeur de thèse : Jean-Marie PROTH

Jury

Président

Prof. Jean-Pierre CAMPAGNE

Rapporteurs

Prof. Jean-Pierre CAMPAGNE

Prof. Yves DALLERY

Prof. Gerd FINKE

Examineurs

Prof. Bernard MUTEL

Prof. Jean-Marie PROTH

# Remerciements

Je ne saurais exprimer l'étendue de mon endettement à l'égard de Monsieur Jean-Marie Proth pour l'aide inestimable, les conseils pertinents et les encouragements qu'il m'a donné tout au long de ce travail. Je lui adresse ma plus vive gratitude pour la confiance qu'il m'a montré, et pour la liberté qu'il m'a laissé dans l'accomplissement de ce travail.

Je souhaite exprimer ma gratitude à Monsieur Rakesh Nagi, Professeur à l'Université de Buffalo, pour son suivi de mes recherches. Je tiens aussi à remercier Monsieur Jean-Pierre CAMPAGNE, Professeur à l'INSA de Lyon, Monsieur Gerd FINKE, Professeur à l'Université Joseph Fourier de Grenoble et enfin Monsieur Yves DALLERY, Professeur à l'École Centrale de Paris, de m'avoir fait l'honneur d'accepter la charge de rapporteur de ma thèse, ainsi que d'avoir participé à mon jury. Je les remercie également pour les pistes de recherches futures qu'ils m'ont suggérées.

Je ne devrais pas oublier de citer Monsieur Cyril Duron pour sa coopération, son aide, ses suggestions, et pour avoir créé une atmosphère constructive et propice à la recherche dans le laboratoire, ainsi que pour m'avoir offert une compagnie appréciable dès l'instant de mon arrivée en France.

C'est une étape décisive de ma vie pour laquelle je remercie mes parents et ma femme qui ont su m'entourer tout au long de ces années de leur amour inconditionnel et d'un support constant. Je les remercie également d'avoir eu foi en moi et mes capacités, et pour les conseils qu'ils m'ont prodigué.

Enfin, que les personnes qui m'ont accompagné tout au long de cette thèse - mes amis, les membres de l'INRIA et les nombreux chercheurs avec qui j'ai eu la chance de travailler reçoivent toute ma reconnaissance et ma sympathie.



*à mes parents et maîtres*



# Résumé

Directeur de Thèse : Prof. Jean-Marie PROTH

Cette thèse se divise en trois parties. La première est consacrée à la conception d'une chaîne d'approvisionnement au niveau stratégique. Une approche systématique est proposée ; elle est construite à partir de cinq macro-activités. Un logiciel a été développé à partir de cette approche. Il est présenté en détail.

Dans la seconde partie de la thèse, des problèmes de recherche opérationnelle liés aux chaînes d'approvisionnement vues au niveau stratégique sont exposés. Il s'agit en particulier de problèmes relatifs à la sélection des partenaires d'une chaîne d'approvisionnement.

Enfin, la dernière partie de la thèse est dédiée à des problèmes sensibles au niveau tactique dans l'environnement des chaînes d'approvisionnement. Il s'agit d'ordonnancement en temps réel d'assemblage lorsque les composants sont sous-traités et de la conception d'un processus de partage des bénéfices et des pertes entre partenaires.



# Abstract

Ph. D. Supervisor : Prof. Jean-Marie PROTH

This thesis is divided in three parts : The first part is devoted to the design of a supply chain at the strategic level. A systematic approach is proposed based on five macro-activities. A software has been developed to support this approach. It is presented in detail. In the second part of the thesis, solutions of O.R. problems related to supply chains at the strategic level are proposed. In particular, models related to the selection of partners are developed. The last part of the thesis is dedicated to problems that are the most sensitive in a supply chain environment : real time scheduling, assembly problem when components are outsourced and definition of a sharing process in a specific example.



# Table des matières

<b>Remerciements</b>	<b>i</b>
<b>Dédicace</b>	<b>iii</b>
<b>Résumé</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Table des Matières</b>	<b>viii</b>
<b>Liste des Tables</b>	<b>xiv</b>
<b>Liste des Figures</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Un peu d'histoire . . . . .	1
1.2 Evolution structurelle des systèmes de production . . . . .	3
1.3 Conception d'une chaîne d'approvisionnement . . . . .	5
1.4 Etude bibliographique . . . . .	6
1.4.1 Etude bibliographique générale . . . . .	7
1.4.2 Etude bibliographique technique . . . . .	10
1.5 Les outils . . . . .	17
1.6 Présentation de la thèse . . . . .	18

---

<b>I</b>	<b>Conception au niveau stratégique</b>	<b>21</b>
<b>2</b>	<b>Conception au niveau stratégique</b>	<b>23</b>
2.1	Concepts de base . . . . .	23
2.2	Décisions au niveau stratégique . . . . .	25
2.2.1	Décisions primaires . . . . .	25
2.2.2	Décisions secondaires . . . . .	26
2.3	Evaluation économique au niveau stratégique . . . . .	28
2.3.1	Les coûts liés aux décisions secondaires . . . . .	28
2.3.2	Définition des coûts incrémentaux . . . . .	29
2.3.3	Unification des coûts incrémentaux . . . . .	30
2.3.4	Coûts et chiffre d'affaire d'une chaîne d'approvisionnement	32
2.4	Le logiciel CAD-SC-SL . . . . .	35
2.4.1	Démarrage du logiciel . . . . .	35
2.4.2	Utilisation des décisions existantes . . . . .	37
2.4.3	Introduction de nouvelles décisions stratégiques . . . . .	41
2.4.4	Optimisation de la production . . . . .	46
2.4.5	Conclusion . . . . .	47
<b>II</b>	<b>Quelques problèmes du niveau stratégique</b>	<b>49</b>
<b>3</b>	<b>Choix des fournisseurs</b>	<b>51</b>
3.1	Introduction . . . . .	51
3.2	Le cas d'une unité de fabrication unique . . . . .	52
3.2.1	Position du problème . . . . .	52
3.2.2	Deux résultats de base . . . . .	53
3.2.3	Un algorithme heuristique . . . . .	56
3.3	Un exemple numérique . . . . .	60
3.3.1	Une approche basée sur la programmation dynamique . . . . .	61
3.4	Le cas de plusieurs unités de fabrication . . . . .	64
3.4.1	Position du problème . . . . .	64

3.4.2	Deux résultats de base . . . . .	65
3.4.3	Un algorithme heuristique. . . . .	74
3.4.4	Applications numériques . . . . .	76
<b>4</b>	<b>Planification des capacités lors de la conception d'une chaîne d'approvisionnement</b>	<b>79</b>
4.1	Introduction . . . . .	79
4.2	Présentation du problème . . . . .	80
4.2.1	Description du problème . . . . .	80
4.2.2	Notations et formalisation du problème . . . . .	81
4.2.3	Taille du problème . . . . .	88
4.3	Propriétés du problème . . . . .	90
4.4	Calcul d'une solution admissible proche de l'optimum . . . . .	96
4.5	Application numérique . . . . .	97
4.6	Conclusion . . . . .	98
<b>5</b>	<b>Conception d'une chaîne d'approvisionnement multi-niveaux</b>	<b>101</b>
5.1	Introduction . . . . .	101
5.2	Présentation du modèle . . . . .	102
5.2.1	Description . . . . .	102
5.2.2	Formulation mathématique . . . . .	103
5.3	Une approche par relaxation des chemins . . . . .	110
5.3.1	Remarques . . . . .	110
5.3.2	Présentation de la méthode . . . . .	111
5.3.3	Calcul du coût associé au couple virtuel . . . . .	112
5.3.4	Propriétés du problème $(P_1^{u,i})$ . . . . .	114
5.3.5	Les algorithmes . . . . .	118
5.3.6	Propriété d'optimalité . . . . .	119
5.4	Exemple numérique et évaluation numérique . . . . .	120
5.4.1	Un exemple numérique . . . . .	120
5.4.2	Evaluation numérique de l'algorithme 2 . . . . .	123
5.5	Conclusion . . . . .	124

<b>III</b>	<b>Quelques problèmes du niveau tactique</b>	<b>127</b>
<b>6</b>	<b>Ordonnancement en temps réel</b>	<b>129</b>
6.1	Introduction . . . . .	129
6.2	Le problème "no-wait" et "limited controlable processing times" original . . . . .	131
6.2.1	Introduction . . . . .	131
6.2.2	Notations et classement des fenêtres . . . . .	131
6.2.3	Le problème . . . . .	134
6.2.4	Un algorithme optimal . . . . .	135
6.2.5	Des exemples d'application . . . . .	136
6.2.6	Complexité de l'algorithme . . . . .	142
6.2.7	Gestion des en-cours . . . . .	142
6.2.8	Un exemple de système non automatisé . . . . .	144
6.2.9	Un exemple de système automatisé . . . . .	151
6.2.10	En Conclusion . . . . .	152
6.3	Ordonnancement temps réel des processus d'assemblage . . . . .	152
6.3.1	Ordonnancement d'un processus d'assemblage simple . . . . .	152
6.3.2	Ordonnancement d'un processus d'assemblage complexe . . . . .	168
6.4	Conclusion . . . . .	175
<b>7</b>	<b>Assemblage avec incertitude sur la livraison des composants</b>	<b>177</b>
7.1	Introduction . . . . .	177
7.2	Formulation du problème . . . . .	178
7.3	Densité de probabilité associée à $T$ . . . . .	180
7.4	Les coûts qui composent le coût total . . . . .	181
7.4.1	Coût de stockage jusqu'à l'instant $T$ . . . . .	181
7.4.2	Coût de stockage moyen entre $T$ et $Z$ . . . . .	182
7.4.3	Coût moyen associé à la rupture de stock . . . . .	182
7.5	Analyse du problème . . . . .	183
7.5.1	Un cas particulier . . . . .	183
7.5.2	Propriété liée au cas général . . . . .	185

---

7.5.3	Etude de $W(Z)$ . . . . .	187
7.5.4	Un autre cas particulier : les composants ont tous le même comportement . . . . .	189
7.5.5	Présentation de l'algorithme général . . . . .	190
7.6	Exemple numérique . . . . .	193
7.7	Conclusion . . . . .	195
<b>8</b>	<b>Partage des risques et des bénéfices dans une chaîne d'approvisionnement : un modèle</b>	<b>197</b>
8.1	Introduction . . . . .	197
8.2	Formulation du problème . . . . .	198
8.3	Résultats fondamentaux . . . . .	202
8.4	Solution du problème . . . . .	205
8.4.1	Maximisation du profit total . . . . .	205
8.4.2	Partage équitable du profit . . . . .	206
8.5	Illustration numérique . . . . .	207
<b>9</b>	<b>Conclusion</b>	<b>209</b>
<b>IV</b>	<b>Bibliographie</b>	<b>213</b>
	Références bibliographiques	214

# Liste des tableaux

2.1	Augmenter la capacité de transport . . . . .	27
2.2	Améliorer la qualité . . . . .	27
2.3	Evolution des bénéfices . . . . .	35
3.1	Exemple d'un système de production livré par six fournisseurs .	63
3.2	Besoins des unités de fabrication . . . . .	77
3.3	Exemple d'un système de production livré par six fournisseurs .	78
4.1	Solutions heuristiques et optimales . . . . .	98
5.1	Coûts fixes et coûts de transport : étapes 1 à 2 . . . . .	121
5.2	Coûts fixes et coûts de transport : étapes 2 à 3 . . . . .	122
5.3	Coûts fixes et coûts de transport : étapes 3 à 4 . . . . .	122
5.4	Valeur des paramètres . . . . .	123
5.5	Résultats fournis par l'algorithme 2 . . . . .	124
5.6	Résultats de l'évaluation numérique . . . . .	124
7.1	Données de l'exemple 1 . . . . .	194
7.2	Evolution des instants de commande avec BL . . . . .	194
8.1	Suite des itérations . . . . .	208
8.2	Evolution de solution par rapport à $f$ . . . . .	208

# Table des figures

1.1	Gestion par départements et gestion par projets . . . . .	4
1.2	Présentation de la thèse . . . . .	18
2.1	Formes des coûts proposés . . . . .	31
2.2	Entrée du logiciel CAD-SC-SL . . . . .	36
2.3	Introduction des caractéristiques du project . . . . .	38
2.4	Sélection des décisions . . . . .	40
2.5	Relation des variables utilisées avec la variable de référence . . . . .	42
2.6	Introduction d'une décision stratégique . . . . .	43
2.7	Choix de la forme des coûts . . . . .	44
2.8	Fonction linéaire . . . . .	44
2.9	Fonction linéaire par morceaux . . . . .	45
2.10	Fonction concave . . . . .	45
2.11	Introduction des paramètres du recuit simulé . . . . .	46
2.12	Solution optimale . . . . .	47
4.1	Schématisation de l'algorithme . . . . .	97
5.1	Le modèle . . . . .	103
5.2	Le graphe partiel du problème . . . . .	121
6.1	Occupation des ressources . . . . .	137
6.2	Ordonnancement du produit . . . . .	140
6.3	Ordonnancement du produit . . . . .	141
6.4	Occupation des ressources . . . . .	144

---

6.5	Ordonnancement du premier produit P1 . . . . .	145
6.6	Ordonnancement du produit P2 . . . . .	146
6.7	Occupation des ressources après l'ordonnancement de P2 . . . . .	147
6.8	Occupation des ressources après mise à jour . . . . .	148
6.9	Ordonnancement du produit P3 . . . . .	148
6.10	Ordonnancement du produit P4 . . . . .	149
6.11	Ordonnancement du produit P5 . . . . .	150
6.12	Occupation des ressources après mise à jour . . . . .	150
6.13	Ordonnancement du produit P6 . . . . .	151
6.14	Processus d'assemblage simple . . . . .	153
6.15	Processus d'assemblage simple . . . . .	164
6.16	Occupation des ressources . . . . .	166
6.17	Ordonnancement du produit . . . . .	167
6.18	Processus d'assemblage complexe . . . . .	170
6.19	Décomposition d'un processus d'assemblage complexe . . . . .	170
6.20	Ordonnancement d'un produit pour deux différents niveaux d'en- cours permis lors d'un processus d'assemblage . . . . .	174
7.1	Une situation d'assemblage . . . . .	180
7.2	Fonction $W(Z)$ pour $h = 0$ . . . . .	188
7.3	Evolution du coût associé au retard . . . . .	188
7.4	Forme générale de $W(Z)$ . . . . .	189

# Chapitre 1

## Introduction

### 1.1 Un peu d'histoire

Le monde de la production a beaucoup évolué depuis la seconde guerre mondiale. L'utilisation de l'ordinateur, au cours des années soixante, a incité les responsables à organiser leur production de manière plus rationnelle dans un univers structuré en départements. L'organisation en départements était, il faut le rappeler, la conséquence du fait que le suivi des tâches et leur gestion se faisait jusque là " à la main " avec l'aide, au mieux, de machines à calculer. L'arrivée de l'ordinateur n'a pas modifié l'organisation dans un premier temps. Elle a simplement permis de gérer plus facilement chaque département de l'entreprise. Les gestionnaires s'occupaient alors de gestion des stocks, des approvisionnements, de la fabrication, du personnel, pour ne citer que les plus courants.

A partir des années 80, plusieurs facteurs se sont conjugués pour inciter les gestionnaires à s'interroger sur la structure même de leurs systèmes de production. Ce fut d'abord un accroissement considérable de la complexité des produits qui a conduit à un accroissement tout aussi considérable des problèmes à traiter. Ce fut ensuite la mondialisation qui a conduit à une compétition toujours plus intense, donc à un resserrement des contraintes temporelles. Ce dernier facteur a entraîné dans un premier temps, l'approche

"Juste-À-Temps" (JAT) , plus connue sous la dénomination de (JIT) "Just-In-Time".

Il y eut aussi l'explosion du nombre de logiciels de gestion de production, souvent basés sur les mêmes principes, mais écrits pour des ordinateurs différents. Ce fut l'époque des générations de MRP (MRP I, MRPII), puis des logiciels KANBAN, auxquels vinrent ensuite s'ajouter les logiciels de gestion de la qualité (TQM=Total Quality Management). Le développement anarchique de ces outils a rendu toute normalisation impossible et a conduit, progressivement, à admettre une évidence : il convenait de " remettre à plat " l'organisation des systèmes de production en même temps que leur gestion. Cette remise en cause a par ailleurs été favorisée par différents facteurs :

1. Les progrès énormes des ordinateurs et une réduction drastique de leur prix. En 2002, un ordinateur de bureau, équipé d'un processeur cadencé à 2Ghz, d'une mémoire de 256 Mo et d'une mémoire de masse de 40Go coûte moins de 1000 euros.
2. Les réseaux de communication, avec au particulier INTERNET, effacent les distances et permettent aux ordinateurs de coopérer. Rappelons que les réseaux locaux (LAN) transportent l'information à la vitesse de 50 G bit/s, les réseaux métropolitains (MAN) à 600 M bit/s et les réseaux à grande distance (WAN) à 600 M bit/s.
3. Les systèmes d'exploitation (OS), tels UNIX, LINUX, NT, offrent désormais de nombreux outils et favorisent grandement les échanges entre systèmes.
4. Enfin, les langages se sont libérés des contraintes matérielles.

## 1.2 Evolution structurelle des systèmes de production

Les facteurs que nous venons de rappeler ont conduit les systèmes de production à modifier à la fois leur structure et leur méthode de gestion pour passer de l'organisation par départements à une organisation par projets, caractéristique des chaînes d'approvisionnement.

Les défauts d'une gestion par départements sont bien connus. Ils proviennent du fait qu'une telle gestion incite les responsables à optimiser leur propre département, sans toujours examiner les conséquences d'une telle optimisation sur les autres départements. En effet il est bien connu, en particulier en mathématiques, que l'optimum global d'un système n'est pas la concaténation d'optima locaux.

La différence entre une gestion classique par départements et une gestion par projets est schématisée par la figure 1.1.

La gestion par projets permet, dans la mesure où chaque projet est rendu indépendant des autres par affectation des ressources à priori, d'assurer un flux physique continu, et donc de réduire le cycle de production et les en-cours. Nous voyons cependant que la gestion par projets nécessite que les décisions d'ordonnancement englobent l'ensemble de la production, de la matière première à la livraison de la commande. Nous pouvons dire que, d'une part, la difficulté du problème décisionnel gagne en complexité car les décisions englobent désormais tout le cycle de production mais que, d'autre part, la complexité du problème décisionnel est réduite du fait de la diversité moindre des entités auxquelles s'applique la décision. En effet, un projet se réduit typiquement à une famille de produits, ou à la mise en place d'un service. Nous verrons dans cette thèse, comment ordonnancer en "temps réel" l'ensemble des tâches à effectuer dans un projet. Bien entendu, le concept "temps réel" doit être compris dans le contexte de la gestion : savoir ordonnancer les tâches qui permettent une

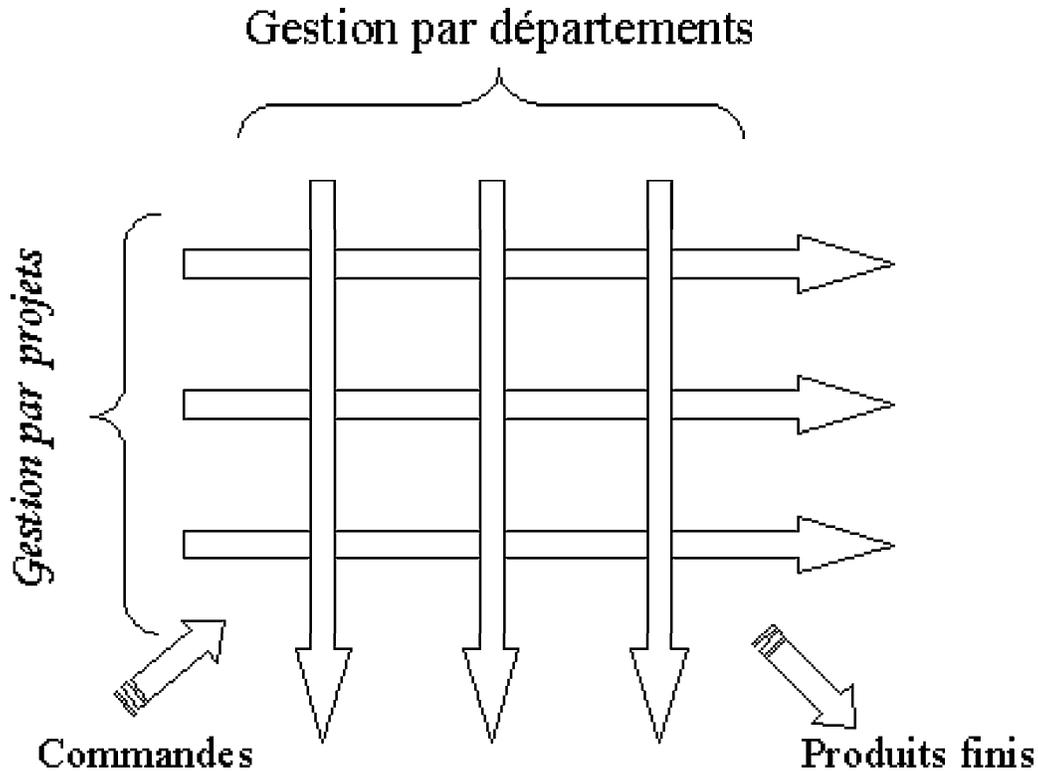


FIG. 1.1 – Gestion par départements et gestion par projets

production donnée en moins de 2 minutes est du temps réel. Plus généralement, nous disons qu'une décision est prise en temps réel lorsque le processus décisionnel ne ralentit pas le processus physique auquel il s'applique. La notion de temps réel est donc relative au temps d'horloge du système physique.

La gestion par projets est en cohérence avec les chaînes d'approvisionnement qui représentent une évolution structurelle majeure. Les définitions de chaînes d'approvisionnement sont diverses.

Nous retenons la suivante, donnée dans [CDP03] : "Une chaîne d'approvisionnement est un réseau d'organisations qui coopèrent pour optimiser le flux de matériel entre fournisseurs et clients, l'optimum étant des flux aussi rapides

que possible à des coûts minimaux. L'objectif d'une chaîne d'approvisionnement est la satisfaction du client "

Remarquons qu'une telle définition n'exclut pas la présence, dans une même chaîne d'approvisionnement, de partenaires appartenant à des sociétés différentes et éventuellement concurrentes. Cette situation exige que soit établie une règle de partage des pertes et des bénéfices dès la conception d'une chaîne d'approvisionnement, comme indiqué dans [SSL95]. Nous retrouvons le même concept dans [GP02] et [CDP03] sous le nom de "sharing process".

### 1.3 Conception d'une chaîne d'approvisionnement

Comme tout système, une chaîne d'approvisionnement se conçoit d'abord au niveau stratégique. Les décisions prises à ce niveau se déclinent ensuite au niveau tactique. Le contenu de cette thèse concerne ces deux niveaux.

Nous nous sommes d'abord intéressés à l'interdépendance des activités, laquelle a pour conséquence qu'une décision stratégique primaire se décline en décisions stratégiques secondaires qui concernent plusieurs des activités de la chaîne d'approvisionnement. Cette étude nous a conduit à développer un logiciel d'aide à la conception de chaînes d'approvisionnement au niveau stratégique. Ce logiciel se nomme CAD-SC-SL pour " Computer Aided Design of a Supply Chain at the Strategic Level ". Il a été le premier travail dans la préparation de cette thèse, travail qui a été mené conjointement avec le travail bibliographique dont nous parlons plus loin. L'analyse d'une chaîne d'approvisionnement et la présentation du logiciel CAD-SC-SL constitueront la première partie de cette thèse.

Il faut souligner que le logiciel CAD-SC-SL a pour objectif l'évaluation rapide d'une chaîne d'approvisionnement dans un cadre structurant. Ce cadre aide l'utilisateur à donner les informations dont il dispose. La difficulté de cette première partie a été de trouver un cadre commun à toutes les chaînes

d'approvisionnement qui peuvent se présenter.

Toujours au niveau stratégique se posent un certain nombre de problèmes qui sont du domaine de la recherche opérationnelle. Il s'agit de problèmes d'affectation essentiellement liés au choix des participants à une chaîne d'approvisionnement, mais aussi de problèmes de structuration des chaînes d'approvisionnement. Leur exposé et leur solution constituent la seconde partie de cette thèse.

Le niveau tactique pose d'autres problèmes. Le plus important d'entre eux concerne l'ordonnancement dont la nature a changé fondamentalement. On retiendra également, dans cette troisième partie, la présentation d'un modèle qui s'intéresse au partage des pertes et des bénéfices. Cet aspect constitue la troisième partie de notre travail.

En résumé :

- la première partie de la thèse dessine un cadre structurant pour notre approche des chaînes d'approvisionnement et propose l'outil logiciel correspondant,
- la seconde partie traite de problèmes spécifiques aux chaînes d'approvisionnement qui se posent au niveau stratégique,
- enfin, la troisième partie prolonge le travail précédent au niveau tactique.

## 1.4 Etude bibliographique

La bibliographie relative aux chaînes d'approvisionnement est vaste. Nous la divisons en deux parties. La première, qualifiée de générale, regroupe les ouvrages qui inscrivent les chaînes d'approvisionnement dans l'histoire de la gestion de la production et développent les grands principes qui gouvernent ce nouveau type d'approche. La seconde partie, que nous qualifions de technique, concerne les problèmes d'optimisation posés par ce nouveau concept. A l'intérieur de cette seconde partie, nous distinguerons les problèmes qui se posent

au niveau stratégique de ceux qui se posent au niveau tactique.

### 1.4.1 Etude bibliographique générale

Nous passons ici en revue un certain nombre d'ouvrages généraux dont nous avons fait usage, en particulier, pour la conception du logiciel CAD-SC-SL qui fait l'objet de la première partie de la thèse.

L'ouvrage de [Chr98] est le plus connu. Il marie les chaînes d'approvisionnement et la logistique pour en faire les deux composantes majeures d'une gestion par projets. Ce livre est très riche en consignes pratiques pour aboutir à une amélioration de la qualité et à la continuité entre les étapes de la production qui mènent de la matière première au produit fini en minimisant les temps d'attente. Nous remarquons cependant que, dans l'approche présentée par Christopher, chaîne d'approvisionnement et logistique sont deux concepts différents qui, associés, conduisent à une gestion efficace. Pour nous, la logistique fait partie des chaînes d'approvisionnement. A ceci près, le livre de Christopher est particulièrement riche et orienté vers les applications.

Le livre de [Cop97] est celui d'un historien qui décrit l'évolution de la gestion des systèmes de production du milieu des années 80 à la fin des années 90. Moins riche que le précédent, il permet cependant de comprendre la genèse des chaînes d'approvisionnement et doit être conseillé à celles et à ceux qui pénètrent le sujet.

Nous pourrions caractériser l'ouvrage de [Gat98] par la place qu'il donne à l'homme dans le fonctionnement d'une chaîne d'approvisionnement. De son point de vue, l'homme est le facteur qui rapproche les exigences du marché et la démarche stratégique mise en oeuvre par les ressources disponibles dans l'entreprise. Gattorna qualifie de " Strategic alignment model " l'aboutissement de la mise en oeuvre du facteur humain. De notre point de vue, Gattorna donne peut-être un peu trop d'importance à la réactivité et la flexibilité naturelle de

l'homme pour mettre en oeuvre les ressources disponibles afin d'atteindre les objectifs classiques des chaînes d'approvisionnement.

Le livre de [Fin98] présente une démarche tout à fait originale. Ayant constaté que les biologistes observent plusieurs générations successives d'insectes à durée de vie brève pour étudier l'évolution dans le temps de certains facteurs génétiques, il transfère la démarche en observant les sociétés à évolution rapide pour étudier comment la fonction dominante de la chaîne d'approvisionnement évolue en son sein. Son objectif est également de dégager les caractéristiques communes des chaînes d'approvisionnement. Cette démarche est celle d'un biologiste : elle se contente d'observer l'existant sans faire de nouvelles propositions. On peut se demander si cette manière de procéder conduit à l'innovation.

Dans leur livre, [SSL95] prétendent étudier essentiellement les chaînes d'approvisionnement dans le contexte européen. Leurs exemples sont cependant, majoritairement, des sociétés U.S. Le concept de chaîne d'approvisionnement qu'ils développent reste vague. Ils soulignent cependant un point important en insistant sur la nécessité d'accords préalables précis entre les sociétés qui participent à la même chaîne d'approvisionnement. Nous y voyons un précurseur du "sharing process" (partage des bénéfices et des pertes) qui nous semble être un facteur fondamental dans la coopération entre sociétés différentes. Nous nous attarderons sur le livre de [GP02], ainsi que sur celui de [CDP03], tous deux partiellement issus des travaux du laboratoire dans lequel j'ai préparé ma thèse. Govil et Proth considèrent successivement les chaînes d'approvisionnement au niveau stratégique et au niveau tactique.

Au niveau stratégique, ils considèrent qu'une chaîne d'approvisionnement est toujours constituée de cinq macro-activités :

- la macro-activité "acheter" qui inclut les tâches d'achat de matières premières, de composants, de ressources et de services,

- la macro-activité "produire" qui inclut la création de services, la fabrication de produits, la maintenance, la formation, etc. En résumé, "produire" inclut toutes les tâches nécessaires pour produire,
- la macro-activité "transporter" inclut les activités de transport des produits et du personnel à l'intérieur et à l'extérieur de la chaîne d'approvisionnement,
- la macro-activité "stocker" concerne le stockage de la matière première des en-cours et des produits finis, ainsi que la manutention associée,
- la macro-activité "distribuer" concerne toutes les activités tournées vers le marché comme le marketing, la vente, les promotions, le conditionnement, etc.

Bien entendu, ces macro-activités se déclinent différemment suivant le type de système auquel elles s'appliquent, mais l'utilisation de cette structure générale permet de standardiser l'approche de conception ou de transformation d'une chaîne d'approvisionnement au niveau stratégique.

Partant de cette structure générale, le livre précise les objectifs à atteindre pour chacune des macro-activités. Il tient compte de l'interdépendance entre activités, introduit la notion de partage des bénéfices et des pertes et développe le processus d'évaluation économique basé sur la notion de coût incrémental.

Au niveau tactique, les auteurs développent les règles à appliquer pour aboutir aux propriétés souhaitées pour la chaîne d'approvisionnement. Les méthodes d'évaluation économique et opérationnelle sont données.

Un chapitre est consacré au développement des produits dans une chaîne d'approvisionnement. L'analyse des étapes du cycle de vie d'un produit est menée dans le détail.

Une étude des technologies modernes qui facilitent le développement des chaînes d'approvisionnement est proposée et la liste des sociétés qui fournissent des logiciels efficaces est donnée.

[CDP03] retiennent également les cinq activités de base au niveau stratégique, comme nous le faisons dans cette thèse.

Les décisions stratégiques sont appelées décisions primaires. Les décisions se décomposent en décisions secondaires, plus précises. Le logiciel CAD-SC-SL, que j'ai développé sur ces bases et qui a été décrit dans cet ouvrage, fait l'objet de la première partie de cette thèse.

L'ouvrage explique également la nouvelle approche qui s'impose à l'ordonnement et insiste sur les procédures d'évaluation économique et d'optimisation au niveau stratégique.

Le niveau tactique est analysé dans le détail. Nous y trouvons les procédures à adopter pour aboutir à une chaîne d'approvisionnement qui correspond à la définition donnée plus haut. Des procédures détaillées pour l'évaluation opérationnelle et économique sont développées dans cet ouvrage. Il comporte en outre un chapitre entièrement dédié à l'informatique et à la communication. On y trouve un descriptif détaillé de l'évolution des ordinateurs, des systèmes de communication et des OS (Operating systems).

## 1.4.2 Etude bibliographique technique

### Le niveau stratégique

Nous nous limitons aux études relatives aux chaînes d'approvisionnement. Comme nous l'avons souligné précédemment, ces travaux concernent essentiellement la sélection des participants à une chaîne d'approvisionnement. Le critère est évidemment l'adéquation des partenaires à la chaîne d'approvisionnement envisagée et le coût, qui inclut les coûts d'investissement et les coûts d'approvisionnement sur un horizon donné.

Une chaîne d'approvisionnement est généralement structurée en trois niveaux :

- Le niveau des fournisseurs, dont le rôle sera de fournir la matière première, les composants et les services aux unités de fabrication.
- Le niveau des unités de fabrication (que nous appellerons également niveau

des producteurs) qui fournissent les produits ou les services à partir des éléments obtenus des fournisseurs.

- Le niveau des distributeurs dont le rôle est d'assurer l'écoulement des produits (ou des services) vers la clientèle.

Certains auteurs ont un point de vue plus limité et considèrent qu'une chaîne d'approvisionnement se borne au réseau des fournisseurs. C'est le cas de [CHH96] et de [LB93a] qui limitent leur étude stratégique au réseau des fournisseurs de Hewlett-Packard. Les raisons qui expliquent cette vue relativement étroite des chaînes d'approvisionnement tiennent essentiellement au fait que la localisation d'une chaîne d'approvisionnement dépend beaucoup, chez HP et dans de nombreux autres domaines, de la proximité des fournisseurs et de la nécessaire implication de ceux-ci dans le processus général, d'où l'intérêt qu'ils suscitent.

Un autre facteur qui explique l'intérêt porté aux fournisseurs est souligné dans [KR96] : on y lit qu'une unité de production consacre 60% de ses ventes à l'achat de matières premières, de composants, de pièces, et plus généralement de services. On comprend que les échanges des producteurs avec les fournisseurs sont, pour les producteurs, une source intéressante d'économies. De plus, comme le soulignent [WCB91], des échanges positifs avec les fournisseurs sont indispensables si l'on veut pouvoir bénéficier de livraisons dans les temps et d'une qualité en constante progression.

Une étude par simulation faite dans [BK90] montre l'importance de fournisseurs de qualité pour l'ensemble de la chaîne d'approvisionnement. Dans leurs conclusions, les auteurs soulignent que (i) la présence de stocks intermédiaires rend le système moins sensible aux variations des délais de livraison, (ii) une qualité insuffisante a peu d'effet sur les stocks, mais affecte fortement les retours de produits défectueux, (iii) une conception qui favorise des composants communs à plusieurs produits permet d'éviter plus facilement les

ruptures de stocks et, enfin, (iv) une conception qui favorise les composants communs réduit l'effet des incertitudes sur les délais de livraison, mais amplifie les problèmes liés à la qualité. Nous pouvions nous attendre à ces conclusions. L'intérêt de l'étude est qu'elles sont chiffrées pour les systèmes étudiés.

Il faut savoir que la tendance des unités de fabrication est de réduire le nombre de leurs fournisseurs mais, simultanément, de s'en rapprocher et de les associer à leurs efforts de qualité et de conception. Les fournisseurs deviennent des partenaires. Cette tendance est étudiée, en particulier, dans [KL94]. Le rapprochement des fournisseurs permet de mettre en place des livraisons "Just-A-temps" ("Just-In-time", JIT). On pourra consulter, à ce sujet [AM88] et [O'N87].

Considérons maintenant plus spécifiquement le niveau des unités de fabrication (producteurs). Les travaux effectués à ce niveau sont nombreux, et nous n'en citerons que quelques-uns. Certains modèles optimisent un ensemble d'unités de fabrication entre lesquelles circulent les produits. De tels systèmes comportent des stocks. Les modèles dans lesquels la demande varie avec le temps ont été traités en utilisant la programmation dynamique comme dans [CWW73] ou la relaxation lagrangienne associée à des méthodes de type " Séparation et Evaluation " (Branch-and-Bound) comme dans [AK84] ou [AG86].

Les modèles que nous venons de présenter supposent des capacités de production infinies. Le cas des capacités limitées est étudié dans [MK93], [BSSW94] ou [GQSL96], pour n'en citer que quelques uns. L'effet du temps de préparation (set-up time) dans ce type de modèle est étudié par [TTM89]. De nombreuses études ont également été menées pour évaluer l'effet de l'utilisation des unités de fabrication réparties sur plusieurs continents en fonction de l'évolution des conditions économiques (voir [KK94], par exemple). Nous sommes, dans cette étude, proches des préoccupations de flexibilité des chaînes d'approvisionnement. Toujours dans le cadre des préoccupations habituelles des

chaînes d'approvisionnement, [LB92] et [LB93b] s'intéressent à l'effet d'une mauvaise coordination des décisions, alors que [Fos96] s'intéresse à la coordination logistique.

Bien entendu, les stocks générés par le passage des produits d'une unité de fabrication à une autre ont donné naissance à une abondante littérature. Nous nous contentons de citer une étude qui compare la planification centralisée et la planification décentralisée exposée dans [MDE93].

Considérons maintenant le niveau des distributeurs. Trois types de modèles sont étudiés à ce niveau. Le premier type concerne le cas où chaque unité de fabrication dispose de son propre distributeur qui expédie les produits aux consommateurs. Le second type de modèle considère des unités de production qui approvisionnent différents distributeurs. Enfin, le troisième type de modèle est similaire au précédent avec un intermédiaire supplémentaire qui est le grossiste.

Une série d'études a pour objectif de localiser les distributeurs dans le cas d'une ou plusieurs unités de fabrication. On lira, en particulier, l'étude de [GG74]. Dans le modèle présenté, la capacité de production est connue pour chaque unité de fabrication et chaque produit. Certains autres modèles s'intéressent à l'équipement des distributeurs, comme par exemple [MN98].

Dans leur communication, [CL88] s'intéressent à la production discrète par lots qui peut être organisée en processus linéaires composés de plusieurs étapes ou en réseau de distribution arborescent divergent auquel s'ajoute un réseau de fournisseurs arborescent convergent. L'objectif est l'analyse des stocks dans les chaînes d'approvisionnement. L'importance de ce papier réside dans l'approche globale des décisions et des mesures de performances. [NDZ02] présentent un modèle de type réseau dont ils analysent l'équilibre et les propriétés qualitatives. Dans [VG97], nous avons trouvé un état de l'art détaillé des modèles de production-distribution dans un environnement global de réapprovisionne-

ment.

[Han59] a produit ce qui a probablement été la première étude sur les liens entre fournisseurs, producteurs et distributeurs. Il a défini le niveau de stock optimal aux différents niveaux du processus. De nombreuses simplifications ont été nécessaires pour obtenir des résultats intéressants. Parmi ces simplifications nous notons :

- des temps de production constants et indépendants du nombre de produits identiques fabriqués en séquence,
- l'existence de produits différents candidats à l'utilisation d'une même ressource à un instant donné n'est pas considéré.

[GG80] se sont intéressés au couple (production + stock de produits finis) avec demandes aléatoires.

[Wil81] a effectué une étude complétée des heuristiques disponibles en 1981 pour résoudre les problèmes d'ordonnement pour les structures multi-étapes du type de celles que nous traitons dans les chapitres 4 et 5. Malheureusement les hypothèses simplificatrices retenues (production et demandes déterministes et constantes) ne permettent pas d'utiliser les résultats. [Wil84] et [Ber85] ont travaillé à étendre les résultats proposés par [GG80] au cas de plusieurs produits.

Dans cette thèse, nous nous proposons de considérer simultanément les trois niveaux et de sélectionner, pour chacun de ces niveaux, les acteurs qui nous permettront de rassembler les capacités voulues au moindre coût. Il s'agit donc d'une vue plus globale que celles qui se traduisent par les modèles que nous venons de présenter.

Le papier [CNP02] fait partie du travail mené dans le cadre de cette thèse. Il traite le cas de candidats potentiels à une chaîne d'approvisionnement organisée en trois niveaux : réapprovisionnement, production et distribution.

L'objectif est de choisir les participants qui conduisent aux coûts minima tout en respectant les capacités (y compris les capacités de transport). Nous développons cette approche dans la seconde partie de cette thèse. Nous avons également développé un algorithme destiné à sélectionner des fournisseurs dans le cas de coûts concaves avec discontinuité à l'origine. Ce travail fait également partie de cette thèse et a été présenté dans [CP03a].

### **Le niveau tactique**

De nombreux problèmes originaux se posent au niveau tactique lorsqu'on traite des chaînes d'approvisionnement. Le plus important, de mon point de vue, est la nécessité d'ordonnancer les demandes des clients en temps réel en prenant des engagements sur les délais de livraison. L'approche que nous proposons dans ce mémoire a été initialisée par [CLP00]. Dans ces travaux, les auteurs proposent un algorithme temps réel pour l'ordonnancement de commandes de type linéaire, c'est-à-dire de commandes qui s'exécutent par une séquence d'opérations. J'ai prolongé ce travail aux systèmes d'assemblage. Nous sommes donc en mesure d'ordonnancer " à la volée " des commandes dès qu'elles se présentent à la condition de ne pas remettre en question les commandes précédemment ordonnancées.

Les problèmes d'ordonnancement en temps réel concernent les systèmes sensibles (centrales nucléaires), les systèmes de surveillance (radars de combat), les processeurs et, bien entendu, les systèmes de production modernes. Les radars de combat ont été traités par [OSM98] et, surtout, par [Dur02]. Les radars de combat sont multifonctions. Ils effectuent des tâches de surveillance qui sont répétitives, mais lancent des tâches de confirmation lorsqu'ils ont détecté un objet volant. Les tâches de confirmation doivent être intercalées entre les tâches répétitives, et cela doit se faire en temps réel. Dans sa thèse, [Dur02] propose des heuristiques qui tiennent compte de la période d'inactivité

qui sépare l'émission des ondes de leur réception après réflexion sur l'objet volant.

L'ordonnancement en temps réel dans la gestion des processeurs a été étudié, en particulier, par [CCLL95].

Dans les chaînes d'approvisionnement qui nous intéressent, nous considérons des problèmes d'ordonnancement sans délai facteur indispensable à la réactivité. Un ordonnancement sans délai est un ordonnancement dans lequel une tâche débute aussitôt que la précédente se termine.

Les problèmes d'ordonnancement sans délai peuvent être divisés en trois classes :

- les problèmes à temps opératoires déterministes dans lesquels chaque tâche a un temps opératoire fixé. Ces problèmes sont qualifiés de "no-wait" dans la littérature anglo-saxonne. [HS96], [GLLR79] et [Bla87] traitent de ces problèmes,
- les problèmes à temps opératoires illimités dans lesquels chaque tâche a un temps opératoire minimal (temps opératoire réel) mais qui peut se prolonger indéfiniment. Bien entendu, la ressource sur laquelle s'effectue l'opération ne peut entreprendre une autre opération tant qu'elle n'est pas libérée. Ces problèmes sont qualifiés de "blocking " dans la littérature anglo-saxonne. Ces problèmes sont également traités par [HS96],
- les problèmes à temps opératoires limités dans lesquels chaque tâche a un temps opératoire situé entre deux limites. Ces problèmes sont qualifiés de "no-wait" et "limited controllable processing time". Ce problème a été examiné dans la thèse de [Cha99] essentiellement pour les problèmes linéaires et pour les gammes d'assemblage et de désassemblage simples. On trouve une étude détaillée du problème linéaire dans [CPW01].

Bien entendu, certains problèmes peuvent appartenir à plusieurs des catégories ci-dessus.

Revenons aux classes que nous venons de déterminer. Les problèmes à

temps opératoires limités se posent soit du fait de la nature du processus de fabrication, soit du fait que l'espace disponible interdit le stockage entre opérations. En sidérurgie, par exemple, la coulée et le laminage à chaud s'enchaînent sans interruption [Cal71]. Nous trouvons le même type de problème en pétro-chimie [Sal73], en plasturgie, en chimie [ER81], [Rek82], dans l'industrie pharmaceutique, l'industrie alimentaire, pour certains produits [RR92], ainsi que dans l'industrie informatique [RR72].

## 1.5 Les outils

La complexité des problèmes traités nous a conduit, dans certains cas, à développer des heuristiques.

Le recuit simulé a été utilisé dans le programme CAD-SC-SL introduit précédemment et que nous développons dans le détail dans la première partie de notre travail. L'objectif est de définir la quantité optimale à produire pour maximiser le bénéfice unitaire. De nombreuses publications ont été consacrées au sujet. Nous citons en particulier [GS86], [HA92], [CDP03].

Soulignons que les métaheuristiques ont été, et sont encore, utilisées par de nombreux chercheurs pour résoudre des problèmes combinatoires similaires aux problèmes d'affectation auxquels nous nous intéressons. [LMR94] ainsi que [YL99] ont utilisé des métaheuristiques telles que le recuit simulé [KGV83], acceptation du seuil (threshold accepting, en anglais) [DS90], ou encore les algorithmes génétiques [Gol89]. La méthode lagrangienne, qui permet d'intégrer une partie des contraintes dans le critère à optimiser, connaît un regain d'intérêt [LMR94],[CcP98].

Les autres heuristiques utilisées sont spécifiques. Une méthode par approximations successives sera proposée dans cette thèse pour approcher la solution optimale dans le problème de choix des fournisseurs.

## 1.6 Présentation de la thèse

La figure 1.2 schématise la structure de cette thèse.

La première partie de la thèse propose une approche systématique de conception de chaîne d'approvisionnement au niveau stratégique. L'objectif est d'aider le concepteur à recenser les décisions à prendre et à les chiffrer lors de la mise en place ou la modification d'une chaîne d'approvisionnement. Une couche "intelligente" permet d'optimiser la quantité à produire dans les limites données par l'étude du marché.

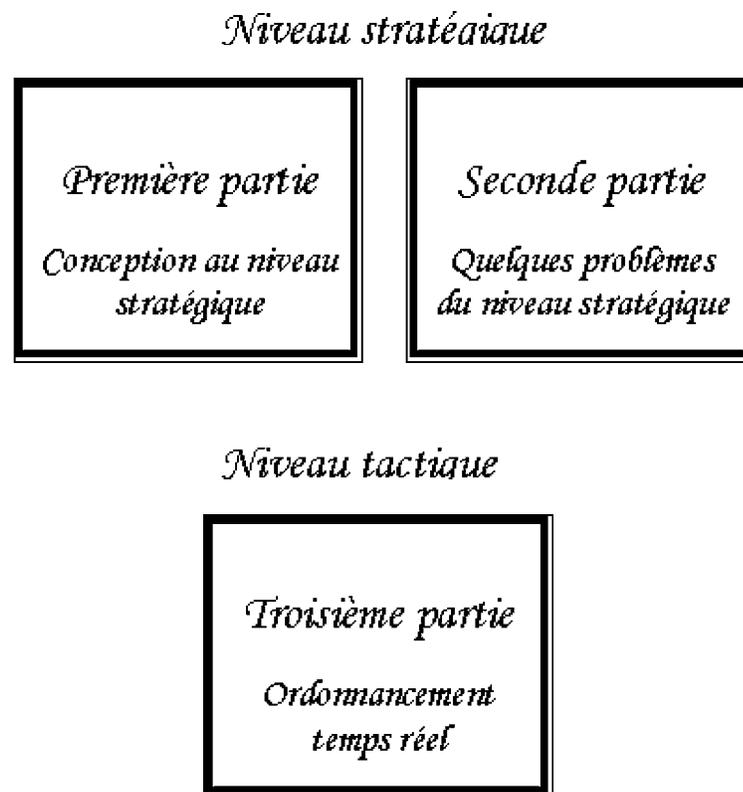


FIG. 1.2 – Présentation de la thèse

La seconde partie concerne quelques problèmes d'optimisation que posent les chaînes d'approvisionnement. Cette partie n'a pas la prétention d'être exhaustive, mais se concentre sur quelques problèmes fondamentaux.

---

Enfin, la troisième partie concerne le niveau tactique. Il s'agit de l'ordonnement temps réel des problèmes d'assemblage. Cette partie est à la base de la réactivité des chaînes d'approvisionnement.



## Première partie

### Conception au niveau stratégique



# Chapitre 2

## Conception au niveau stratégique

### 2.1 Concepts de base

Comme nous l'avons souligné en introduction, nous considérons que toute chaîne d'approvisionnement est composée de cinq macroactivités (Acheter, Transporter, Produire, Stocker et Distribuer). Nous détaillerons le contenu de ces macroactivités dans les paragraphes suivants. Diverses décisions peuvent être prises à l'intérieur de chacune de ces macroactivités. Chacune de ces décisions primaires représente un objectif qui se traduit par des décisions secondaires dans chacune des cinq macroactivités. Par exemple, la décision primaire " Augmenter la production de 50% ", qui concerne la macroactivité "Produire", va nécessiter les décisions secondaires suivantes :

-dans la macroactivité "Acheter"

1. acheter de nouvelles machines,
2. acheter de nouvelles ressources de transport,
3. acheter de nouveaux ordinateurs,
4. etc.

- Dans la macroactivité "Transporter"

1. réorganiser le transport,

- 2. embaucher,
  - 3. etc.
- Dans la macroactivité "Produire" :
- 1. réorganiser les ateliers,
  - 2. embaucher des ouvriers,
  - 3. introduire un nouveau système de gestion,
  - 4. etc.
- Dans la macroactivité "Stocker"
- 1. embaucher des manutentionnaires,
  - 2. réorganiser le processus de stockage,
  - 3. réorganiser les aires de stockage,
  - 4. etc.
- Dans le macroactivité "Distribuer"
- 1. décider de nouvelles campagnes de promotion,
  - 2. augmenter la capacité du service marketing,
  - 3. réorganiser les circuits de distribution,
  - 4. etc.

Nous reviendrons sur les décisions secondaires dans le détail dans les paragraphes suivants.

Nous considérons ensuite le problème de l'évaluation d'une chaîne d'approvisionnement au niveau stratégique. Enfin, nous présentons le logiciel CAD-SC-SL (Computer Aided Design of a Supply Chain at the Strategic Level) développé à partir de l'approche décrite précédemment.

## 2.2 Décisions au niveau stratégique

### 2.2.1 Décisions primaires

Au niveau stratégique, une décision primaire est un objectif global à atteindre. Nous donnons ci-dessous quelques exemples en les classant par rapport aux macroactivités auxquelles ils appartiennent.

#### a. Macroactivité "Acheter".

Nous nous limitons aux décisions primaires les plus fréquentes :

- choisir des fournisseurs pour un produit donné,
- améliorer les tests de qualité sur les composants achetés à l'extérieur,
- réduire le nombre de fournisseurs,
- former le personnel chargé des achats,
- réduire les délais de réapprovisionnement,
- améliorer les contacts avec les fournisseurs (commandes électroniques,...),
- réduire les périodes de recouvrement.

#### b. Macroactivité "Transport".

- augmenter la capacité de transport,
- changer de type de transport afin de réduire les délais,
- améliorer la communication avec les transporteurs,
- former aux nouveaux modes de distribution,
- améliorer les communications avec les chauffeurs.

La liste n'est évidemment pas exhaustive.

#### c. Macroactivité "Produire".

- lancer une nouvelle ligne de fabrication,
- automatiser une ligne de fabrication,
- modifier la conception d'un produit,
- faire appel à la sous-traitance,
- améliorer le contrôle de la qualité.

d. Macroactivité "Stocker".

- changer de mode de stockage,
- améliorer la gestion des stocks,
- réorganiser l'aire de stockage,
- améliorer la communication avec les transporteurs,
- former le personnel à la qualité.

e. Macroactivité "Distribuer".

- relancer la publicité autour d'un type de produits,
- repenser la distribution,
- introduire un nouveau mode d'échange avec les clients.

Bien entendu, cette liste peut être largement étendue.

Chaque décision primaire se décline en décisions secondaires, plus précises, et qui peuvent concerner toutes les macroactivités. Nous donnons deux exemples dans le paragraphe suivant.

### 2.2.2 Décisions secondaires

#### Exemple 1

Considérons la décision primaire "Augmenter la capacité de transport". Le tableau 2.1 donne une liste non exhaustive des décisions secondaires à appliquer pour réaliser cette décision primaire.

#### Exemple 2

Considérons maintenant la décisions primaire : "Améliorer la qualité". Les décisions primaires associées sont données dans la tableau 2.2.

Les décisions secondaires sont celles que nous pouvons identifier et affecter à des équipes. Elles définissent les actions à mener pour atteindre l'objectif fixé par la décision primaire à laquelle elles se rattachent. Lorsqu'une décision primaire est prise au niveau stratégique, tout le problème est de recenser les décisions secondaires, c'est-à-dire les actions à mener, pour atteindre l'objec-

TAB. 2.1 – Augmenter la capacité de transport

Macroactivités	Décisions secondaires
Acheter	- Acquérir des ressources de transport, - acquérir des ressources de manutention.
Transporter	- Embaucher des chauffeurs, - repenser les circuits de distribution.
Produire	- Travailler en 3 postes, - modifier la planification de la production, - réorganiser l'agencement d'un atelier.
Stocker	- Repenser les ressources de manutention, - embaucher des manutentionnaires, - former à la qualité.
Distribuer	- Modifier le conditionnement, - introduire la communication radio avec les transporteurs.

TAB. 2.2 – Améliorer la qualité

Macroactivités	Décision secondaires
Acheter	- Acheter des formations à la qualité, - acheter les ressources de transport et de manutention.
Transporter	- Former à la qualité du transport, - modifier des ressources de transport.
Produire	- Décider éventuellement de faire appel à la sous-traitance, - automatiser certaines opérations (robot), - optimiser les tests de qualité.
Stocker	- Automatiser le stockage et destockage, - améliorer les systèmes de localisation des produits, - améliorer la communication.
Distribuer	- Optimiser le conditionnement, - optimiser les circuits de distribution.

tif fixé par la décision primaire. Notons que la liste des décisions primaires donnée n'est pas forcément la même suivant la chaîne d'approvisionnement à laquelle la décision primaire s'applique. C'est la raison pour laquelle le logiciel CAD-SC-SL que nous présentons plus loin et dont l'un des objectifs est d'aider le décideur à recenser les décisions secondaires attachées à une décision primaire est un logiciel ouvert qui permet d'ajouter ou de retrancher des décisions secondaires d'une liste. Il permet aussi d'ajouter des décisions primaires.

## 2.3 Evaluation économique au niveau stratégique

### 2.3.1 Les coûts liés aux décisions secondaires

Nous distinguons deux types de décisions secondaires :

- Les décisions *secondaires quantifiables* dont un coût peut être relié à un paramètre caractéristique de la décision. Par exemple, la décision secondaire "Augmenter la capacité de stockage" a un coût fonction de l'accroissement de capacité recherché. De même, la décision secondaire "Embaucher des ouvriers de production" implique un coût fonction du nombre d'ouvriers embauchés, ou de l'augmentation de production attendue.
- Les décisions *secondaires non quantifiables* sont des décisions dont aucun coût n'est fonction d'une variable. Par exemple, la décision secondaire "Former le personnel de manutention à la qualité" implique un coût global qui n'est généralement pas fonction du nombre d'ouvriers formés. De même la décision secondaire "Réorganiser l'atelier" ne dépend ni du nombre de machines contenues dans l'atelier, ni d'aucun des paramètres de la décision primaire : il s'agit, ici encore, d'un coût global attaché à l'étude demandée. Il en serait de même dans le cas d'une étude de marché.

Dans tous les cas, nous aurons à associer deux coûts à chaque décision secondaire : *le coût d'investissement* et *le coût de fonctionnement*. Une décision secondaire quantifiable peut n'avoir qu'un de ses coûts fonction d'une

variable.

Avant de développer la procédure d'affectation des coûts, il faut introduire un concept essentiel pour l'évaluation des coûts dans les chaînes d'approvisionnement : il s'agit du concept de *coût incrémental*. Ce concept nécessite quelques explications. Lorsqu'une chaîne d'approvisionnement est conçue, elle fait appel à des partenaires qui appartiennent éventuellement à des entreprises différentes. Chaque partenaire est déjà en activité pour le compte de sa propre entreprise ou d'autres chaînes d'approvisionnement, mais il dispose souvent d'une capacité non utilisée plus ou moins importante qu'il peut mettre à la disposition de la nouvelle chaîne d'approvisionnement. Par exemple, ses machines ou ses ressources de transport peuvent ne pas être utilisées 24 heures sur 24. Les capacités non utilisées peuvent alors être mises à la disposition de la nouvelle chaîne d'approvisionnement. Le coût d'investissement imputé à la décision primaire n'inclut pas la capacité disponible. De même, le coût de fonctionnement affecté à la décision primaire se limite au coût engagé spécifiquement pour la nouvelle chaîne d'approvisionnement. Nous sommes donc en face de coûts incrémentaux.

Bien entendu, cette manière de procéder ne conduit pas à une évaluation exacte du coût de la chaîne d'approvisionnement, mais le concept de coût incrémental permet d'éviter l'utilisation des principes de la comptabilité analytique qui conduisent à des affectations des coûts qui ne sont pas justifiées. Le concept de coût incrémental permet une évaluation relative du coût de la chaîne d'approvisionnement et permet de prendre des décisions qui réduisent les coûts.

### 2.3.2 Définition des coûts incrémentaux

Les coûts d'investissement et de fonctionnement attachés à une décision secondaire dépendent de la chaîne d'approvisionnement considérée. Seuls les

responsables de la conception peuvent les définir. Un concepteur est généralement en mesure de proposer la forme du coût et quelques valeurs du coût en fonction de la valeur du paramètre caractéristique lorsque la décision secondaire est quantifiable. Par exemple, le coût d'investissement attaché à la décision secondaire " Augmenter la capacité de stockage" peut être linéaire sur  $]0, +\infty)$  et nulle lorsqu'on n'augmente pas la capacité. Dans ce cas, il suffit que le concepteur évalue deux paires  $(x_1, y_1)$  et  $(x_2, y_2)$ , où  $x_1$  (respectivement  $x_2$ ) est l'augmentation de capacité et  $y_1$  (respectivement  $y_2$ ) est le coût associé.

Nous considérons que les formes des coûts nécessaires au niveau stratégique se résument aux formes données par la figure 2.1.

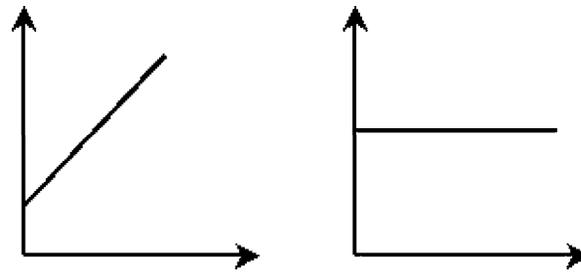
Le concepteur qui aura à définir un coût incrémental d'investissement ou de fonctionnement choisira donc une des formes de la figure 2.1, puis fournira les informations nécessaires pour définir la fonction correspondante :

- pour une forme linéaire, il fournira deux points,
- pour une forme linéaire par morceaux, il fournira deux points pour chaque intervalle ou un seul point si la fonction est constante sur l'intervalle,
- pour une forme exponentielle, il fournira deux points et la valeur limite, mais il faudra alors vérifier la cohérence de ces informations.

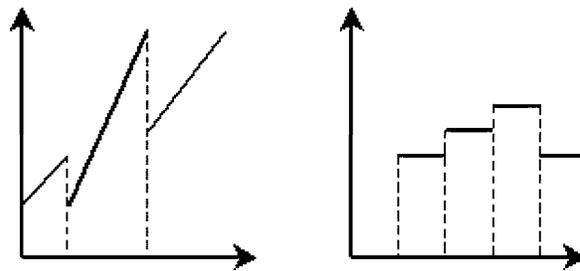
Lorsqu'un coût n'est fonction d'aucune variable, il appartient au concepteur de le fournir.

### 2.3.3 Unification des coûts incrémentaux

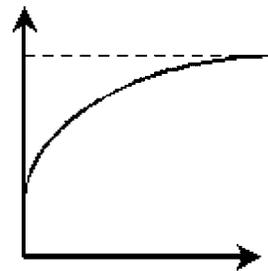
Nous venons de voir que certains coûts incrémentaux, qu'ils soient d'investissement ou de fonctionnement, sont fonction d'une variable (quantité produite, nombre d'employés, capacité, distance, ...). Le problème est que si deux de ces coûts sont fonctions de variables différentes, ils ne sont comparables que si l'on sait relier ces variables. Il est donc nécessaire d'exprimer chacune de ces variables en fonction d'une variable unique qui peut être, par exemple, le ni-



**a. Formes linéaires**



**b. Formes linéaires par morceaux**



**c. Forme exponentielle**

FIG. 2.1 – Formes des coûts proposés

veau de production de la chaîne d'approvisionnement visée, ou l'augmentation du niveau de production si l'action menée consiste à augmenter la production.

Pour relier les variables, nous utilisons les formes de fonctions de la figure 2.1 et nous procédons comme précédemment. L'objectif est de définir les variables utilisées en fonction des quantités produites.

Nous disposons donc désormais de coûts d'investissement ou de fonctionnement qui sont soit fonctions des quantités produites, soit constants. Dans le premier cas, nous connaissons les fonctions. Nous connaissons les coûts dans le second cas.

### 2.3.4 Coûts et chiffre d'affaire d'une chaîne d'approvisionnement

Ce paragraphe est consacré à l'évaluation d'une chaîne d'approvisionnement au niveau stratégique. Nous supposons connues les dépenses d'investissement fixes, que nous notons :

$$If_k, \quad k = 1, 2, \dots, I_1$$

où  $I_1$  est le nombre d'investissements non quantifiables. Nous connaissons également les dépenses d'investissement fonctions de la quantité  $x$  à produire et que nous notons :

$$Ig_k(x), k = 1, 2, \dots, I_2$$

où  $I_2$  est le nombre d'investissements quantifiables. Les dépenses de fonctionnement sont connues sur l'horizon de l'évaluation  $H$ .  $H$  est le nombre de périodes élémentaires (une période élémentaire est le trimestre, l'année ou plusieurs années, suivant le type de système) sur lesquelles se fait l'évaluation.

Nous désignons les dépenses de fonctionnement fixes par :

$$Ff_{k,i}, k = 1, \dots, U_1 \text{ et } i = 1, 2, \dots, H$$

$U_1$  est le nombre de coûts de fonctionnement non quantifiables. Les dépenses de fonctionnement fonction de la quantité  $x$  à produire s'écrivent :

$F_{g_{k,i}}(x), k = 1, \dots, U_2$  et  $i = 1, 2, \dots, H$

$U_2$  est le nombre de coûts de fonctionnement quantifiables. Le lecteur notera que notre présentation suppose que nous avons ramené toutes les décisions quantifiables à des fonctions des quantités à produire.

Enfin, le chiffre d'affaires s'écrit

$Ca_i(x), i = 1, 2, \dots, H$

A ce niveau, nous introduisons le taux d'actualisation, que nous désignons par  $\alpha$ , et le taux de dépréciation, que nous désignons par  $\beta$ . Plusieurs explications sont avancées pour justifier le taux d'actualisation. Nous retiendrons celle-ci : le taux d'actualisation est l'intérêt qui serait servi pour une unité monétaire qui serait placée durant une période élémentaire. Nous considérons que  $\alpha$  est constant sur l'horizon  $H$ .

Le taux de dépréciation est le pourcentage de perte du capital investi par période élémentaire. Cette dépréciation correspond à l'usure, à la casse, au fait que le matériel devient obsolète, par exemple. Nous supposons que  $\beta$  est constant pour toutes les sommes investies.

Nous considérons que les coûts de fonctionnement sont engagés en début de période élémentaire, et que les chiffres d'affaires sont encaissés en fin de période.

Les chiffres d'affaire récoltés à la fin de chaque période élémentaires conduisent, à l'horizon  $H$ , à un chiffre d'affaire total :

$$CA_H = \sum_{i=1}^H Ca_i(x) \cdot (1 + \alpha)^{H-i} \quad (2.1)$$

Nous désignons par  $IT$  l'investissement total :

$$IT(x) = \sum_{k=1}^{I_1} If_k + \sum_{k=1}^{I_2} Ig_k(x) \quad (2.2)$$

La valeur résiduelle de l'investissement à l'horizon  $H$  est, en supposant que l'investissement est fait en début de première période élémentaire :

$$VR = IT(x) \cdot (1 - \beta)^H \quad (2.3)$$

La valeur de l'investissement serait devenu, s'il avait été placé :

$$VD = IT(x).(1 + \alpha)^H \quad (2.4)$$

La dépréciation de l'investissement à l'horizon  $H$  est donc :

$$DI_H(x) = IT(x)[(1 + \alpha)^H - (1 - \beta)^H] \quad (2.5)$$

Considérons maintenant le coût de fonctionnement. La dépense totale en fonctionnement au début de la  $i$ -ème période élémentaire est :

$$IV_i(x) = \sum_{k=1}^{U_1} Ff_{k,i} + \sum_{k=1}^{U_2} Fg_{k,i}(x) \quad (2.6)$$

Les sommes dépensées en fonctionnement sont, à l'horizon  $H$  :

$$IV_H(x) = \sum_{i=1}^H IV_i(x).(1 + \alpha)^{H-i+1} \quad (2.7)$$

En résumé :

Le bénéfice  $B_H(x)$  à l'horizon  $H$  lorsque la production est  $x$  par période élémentaire est :

$$\begin{aligned} B_H(x) &= CA_H - DI_H(x) - IV_H(x) \\ &= \sum_{i=1}^H \{Ca_i(x).(1 + \alpha)^{H-i} - IV_i(x).(1 + \alpha)^{H-i+1}\} \\ &\quad - IT(x).[ (1 + \alpha)^H - (1 - \beta)^H ] \\ &= \sum_{i=1}^H \{ (1 + \alpha)^{H-i} [Ca_i(x) - IV_i(x).(1 + \alpha)] \} \\ &\quad - IT(x).[ (1 + \alpha)^H - (1 - \beta)^H ] \end{aligned}$$

Si les chiffres d'affaires et les coûts d'investissement ne dépendent pas du temps, la formule précédente devient :

$$B_H(x) = [Ca(x) + IV(x).(1 + \alpha)].\frac{(1+\alpha)^H - 1}{\alpha} - IT(x).[ (1 + \alpha)^H - (1 - \beta)^H ]$$

### Exemple

Considérons l'exemple suivant. Nous supposons que les coûts ne sont pas fonction du temps et que :

$Ca(x) = 5x$  ( Chiffre d'affaire)

$IV(x) = 0.22 + 0.5x$  (Coût de fonctionnement)

$IT(x) = 30 + 14x$  (Coût d'investissement).

$\alpha = 0.4, \beta = 0.01$

La table 2.3 donnent  $B_h(x)$  pour  $H = 110$  et pour  $x = 1000, 1500$  et  $2000$  :

TAB. 2.3 – Evolution des bénéfices

H	1	2	3	4	5	6	7	8	9	10
Bénéfice pour x=1000	-52	-67	-32	75	279	620	1153	1953	3127	4824
Bénéfice pour x=1500	-72	-87	-22	154	483	1027	1870	3132	4979	7645
Bénéfice pour x=2000	-92	-106	-11	233	687	1434	2588	4311	6838	10466

## 2.4 Le logiciel CAD-SC-SL

### 2.4.1 Démarrage du logiciel

Ce paragraphe est consacré au logiciel CAD-SC-SL développé dans le cadre de cette thèse. Les principes de base ont été exposés ci-dessus. Nous ne décrivons ici que son utilisation. Lorsque le logiciel est lancé, la figure 2.2 apparaît à l'écran.

Nous voyons que la fenêtre est partagée verticalement en deux parties. La partie de gauche contient des informations déjà introduites dans le système. Nous y voyons des décisions primaires :

- Improving quality (Améliorer la qualité).
- Launching a new type of product (Lancer un nouveau type de produit).
- Improving productivity (Améliorer la productivité).
- Improving transportation (Améliorer le transport).
- Selecting new providers (Choisir de nouveaux fournisseurs).
- Improving the storage system (Améliorer le système de stockage).
- Reorganizing production (Réorganiser la production).

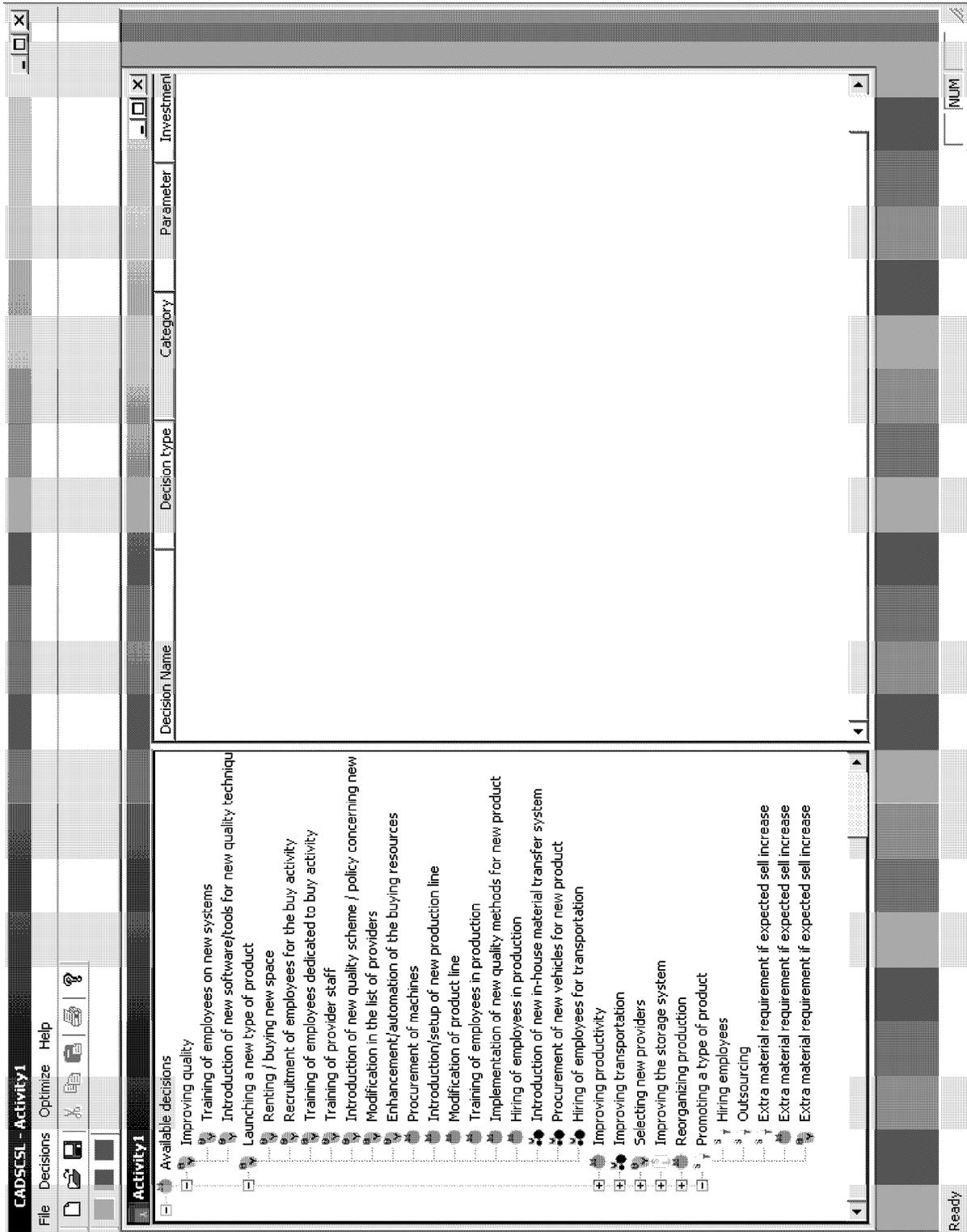


FIG. 2.2 – Entrée du logiciel CAD-SC-SL

Sous chaque décision primaire se trouvent, en retrait, les décisions secondaires qui s'y rattachent. Par exemple, les décisions secondaires suivantes sont rattachées à la décision primaire "Improving quality" :

- Training of employees on new systems (Formation des employés aux nouveaux systèmes).
- Introduction of new software tools for new quality techniques (Introduction de nouveaux logiciels pour les nouvelles techniques de qualité).

Cette fenêtre permet :

- De visualiser les explications correspondant aux décisions primaires et secondaires affichées. En cliquant deux fois sur le bouton de gauche de la souris, les explications apparaissent sur la partie droite de l'écran. Rappelons que le logiciel est implanté sur PC.
- D'introduire de nouvelles décisions stratégiques primaires ou secondaires. Nous verrons plus loin comment introduire de nouvelles décisions.
- Définir ou modifier des coûts correspondant à des décisions. Cet aspect sera traité plus loin. Pour l'instant, nous expliquons comment utiliser les informations déjà introduites dans le système.

## 2.4.2 Utilisation des décisions existantes

En cliquant sur l'étiquette "Decision" située en haut et à gauche de la fenêtre précédente, nous voyons apparaître la fenêtre représentée par la figure 2.3 qui concerne l'utilisation des décisions stratégiques déjà introduites dans le système. Bien entendu, si certaines décisions stratégiques nécessaires au traitement du problème ne sont pas enregistrées, on utilisera les indications données plus loin avant d'aborder l'étape décrite dans ce paragraphe.

La fenêtre représentée dans la figure 2.3 permet d'introduire :

- Le nom du projet. Ici, nous l'appelons "New project".
- La variable qui servira de base à tous les coûts quantifiables. Dans ce projet,

**Selection Wizard** [X]

Project Name: New Project

Project variable: NB

Time horizon: 10

Discount rate: 0.15

Depreciation rate: 0.11

Revenue per unit: 55

Target range (quantity): 20 20000

< Précédent Suivant > Annuler

FIG. 2.3 – Introduction des caractéristiques du project

cette variable est le nombre d'unités produites, et nous la désignons par "NB".

- La ligne désignée par "Time horizon" demande que soit fourni le nombre de périodes élémentaires sur lesquelles se déroulera l'étude.
- La ligne référencée "Discount rate" attend que soit fourni le taux d'actualisation.
- La ligne référencée "Depriciation rate" attend que soit fourni le taux de dépréciation.
- "Revenue per unit" est, dans notre cas, le prix de vente unitaire. Cette ligne se rapporte toujours à la variable de base.
- Enfin, la ligne référencée "Target range" demande que soient fournis la borne inférieure et la borne supérieure entre lesquelles l'utilisateur accepte qu'évolue la variable de base. L'optimisation contenue dans le logiciel et dont nous parlerons plus loin donnera la valeur optimale de la variable de base, c'est-à-dire la valeur qui maximise le bénéfice unitaire.

Dans la pratique, les bornes que nous venons d'introduire sont le résultat d'études de marché. Dans le cas d'une production, elles représenteront les quantités maximale et minimale que peut absorber le marché.

Bien entendu, si les deux bornes sont égales, aucune optimisation n'est réalisée et le système se contentera de fournir le bénéfice unitaire relatif à la quantité commune.

En cliquant sur le bouton "Suivant" situé au bas de la fenêtre, nous passons à la fenêtre suivante représentée dans la figure 2.4

Cette fenêtre permet de sélectionner les décisions stratégiques nécessaires au projet parmi les décisions stratégiques disponibles. Une décision listée dans la partie gauche de l'écran est sélectionnée en :

- Cliquant sur la décision,
- Cliquant sur le signe ">" situé à mi-hauteur, entre les deux parties de l'écran.

La décision sélectionnée apparaît alors dans la partie droite de l'écran.

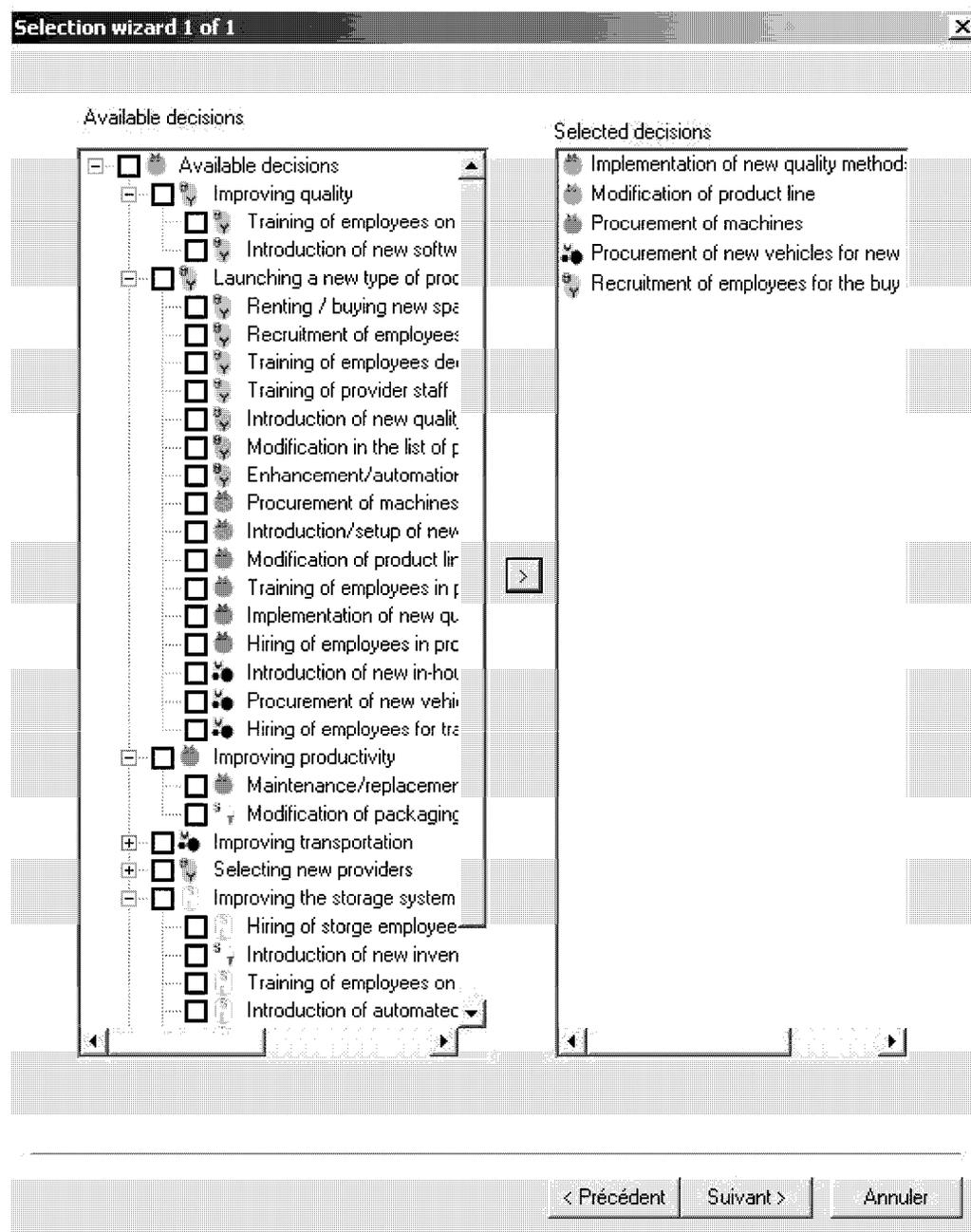


FIG. 2.4 – Sélection des décisions

Le figure ?? représente le cas où des décisions stratégiques secondaires ont été sélectionnées parmi les décisions qui détaillent la décision primaire intitulée "Launching a new type of product".

Un coût d'investissement et un coût de fonctionnement vont être fournis pour chacune des décisions stratégiques sélectionnées. Pour introduire ces coûts, nous passons à la fenêtre suivante en cliquant sur le bouton "Suivant". Cette fenêtre est représentée dans la figure 2.5.

Elle permet de relier les variables utilisées à la variable de référence qui est "NB" dans notre cas. La sélection d'une décision stratégique fait apparaître la variable utilisée ("Level", dans le cas de la figure). Il est alors possible de choisir le type de fonction qui relie ces deux variables parmi quatre formes possibles :

- Linéaire.
- Linéaire par morceaux.
- Constante par morceaux.
- Concave.

Nous allons montrer comment introduire de nouvelles décisions stratégiques et les coûts correspondants.

### 2.4.3 Introduction de nouvelles décisions stratégiques

L'introduction de nouvelles décisions se fait également à partir de la fenêtre initiale représentée dans la figure 2.2. Il suffit de cliquer sur le bouton "File" situé en haut et à gauche de la fenêtre. La fenêtre représentée dans la figure 2.6 apparaît alors.

Cette fenêtre permet :

- D'introduire le nom de la décision stratégique concernée.
- De rattacher cette décision à l'une des cinq macro-activités définies au niveau stratégique : Acheter, Transporter, Produire, Stocker, Distribuer.

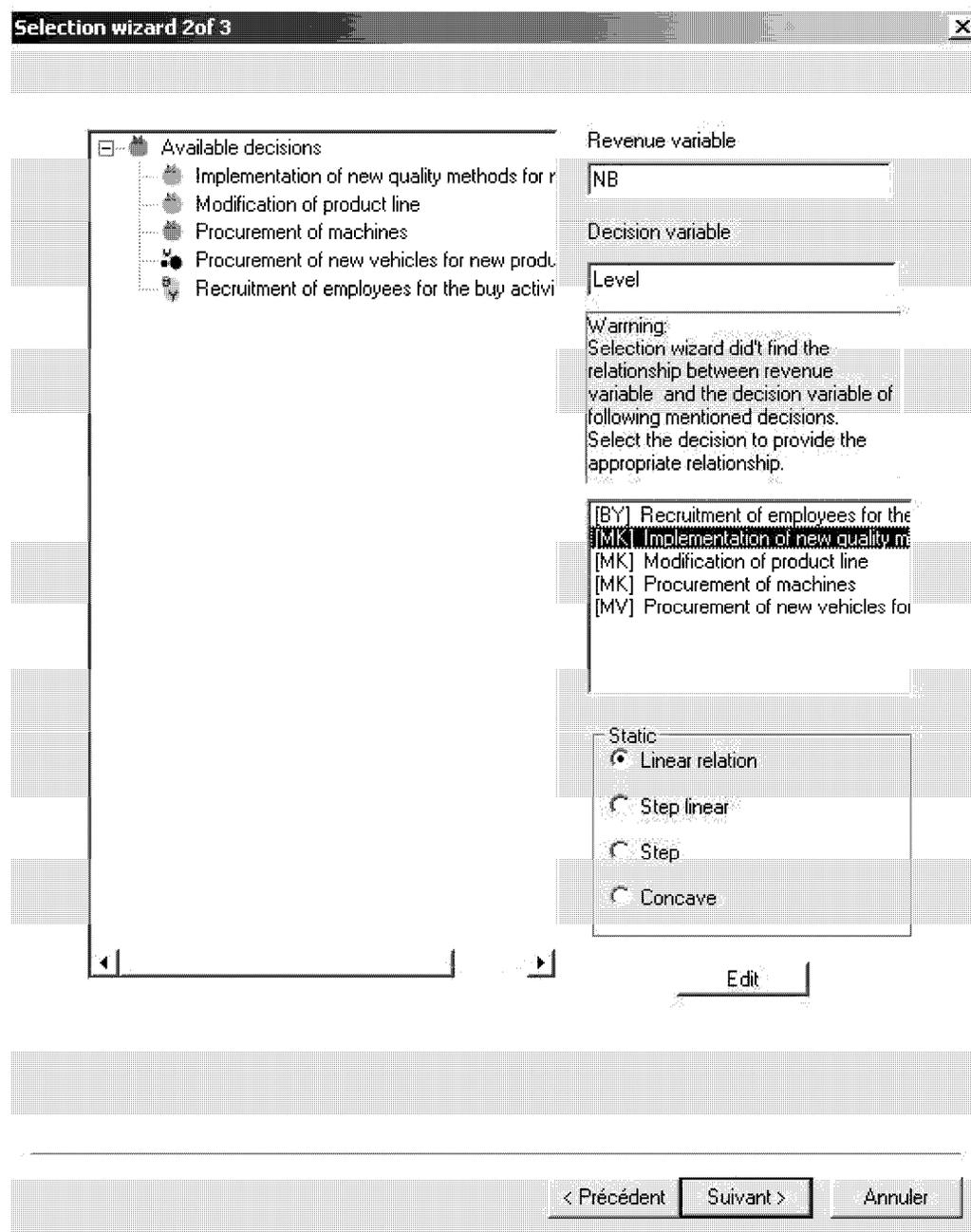


FIG. 2.5 – Relation des variables utilisées avec la variable de référence

- De fournir la variable par rapport à laquelle seront définis les coûts. Cette variable peut être prélevée dans la liste située sous la ligne " Decision parameter ". Dans l'exemple présenté dans la figure, cette variable est " NLine ".
- De choisir la catégorie dans laquelle se range cette décision stratégique ("qualitative" ou "quantitative").

Note : Lorsque la décision est qualitative, la variable n'est pas prise en compte.

En cliquant sur le bouton "Coût" la fenêtre 2.7, qui permet d'introduire les formes des coûts d'investissement et de fonctionnement, est affichée.

Les boutons "Parameter" et "Plot" permettent respectivement d'appeler la fenêtre qui permettra d'introduire l'information et la forme de cette fonction.

Nous donnons dans les figures 2.8, 2.9 et 2.10 les fenêtres qui montrent comment est introduite l'information dans le cas d'une fonction linéaire, d'une fonction linéaire par morceaux et d'une fonction concave.

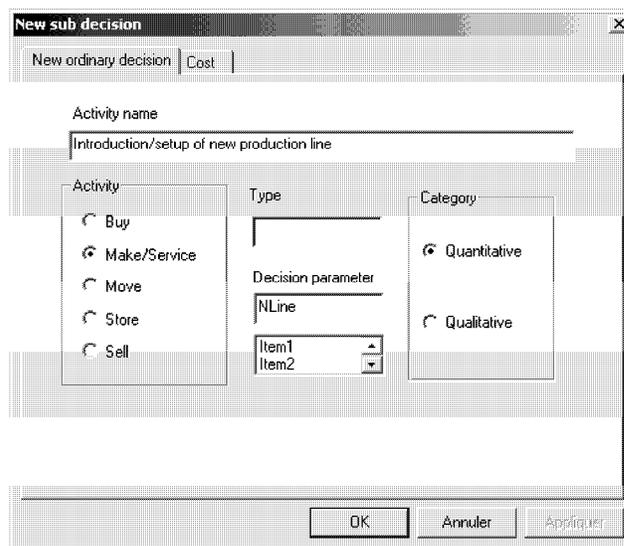


FIG. 2.6 – Introduction d'une décision stratégique

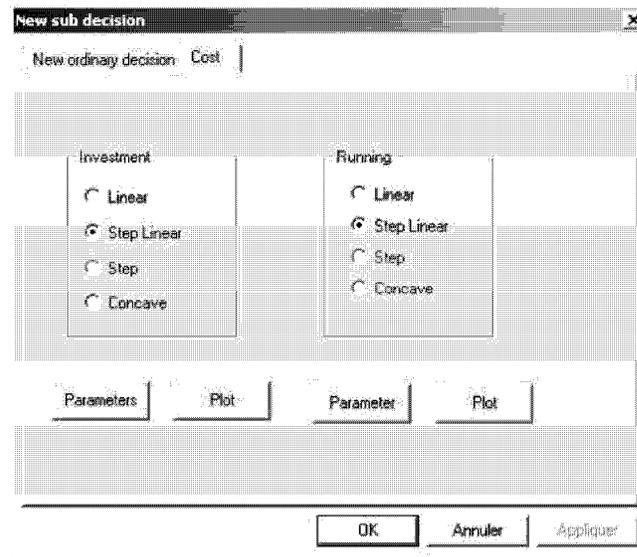


FIG. 2.7 – Choix de la forme des coûts

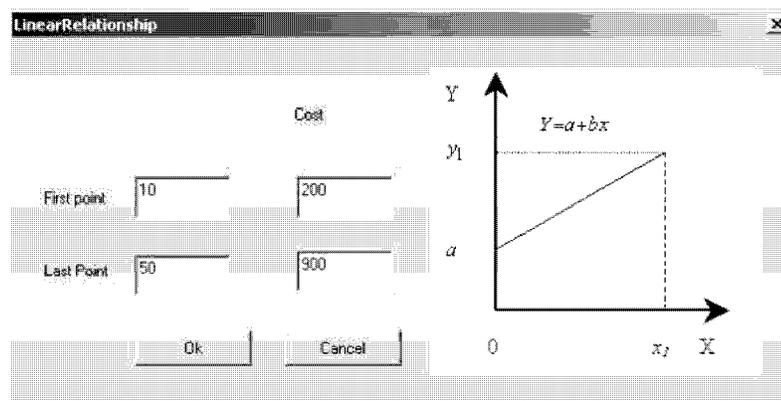


FIG. 2.8 – Fonction linéaire

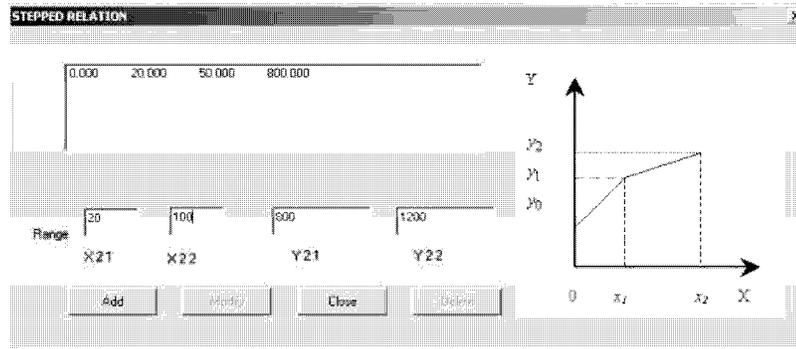


FIG. 2.9 – Fonction linéaire par morceaux

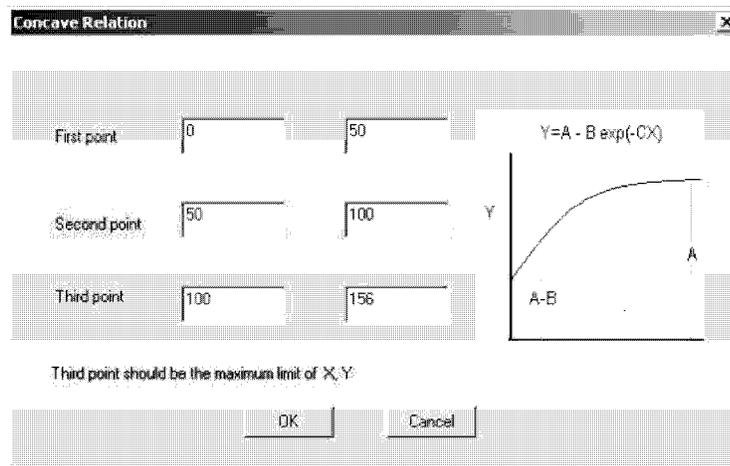


FIG. 2.10 – Fonction concave

### 2.4.4 Optimisation de la production

Revenons à la fenêtre initiale. En cliquant sur le bouton " Optimize ", la fenêtre donnée par la figure 2.11 apparaît. Elle permet d'introduire les paramètres qui vont permettre de mettre en oeuvre le recuit simulé chargé de l'optimisation. L'utilisateur est invité à donner :

- La température initiale (Initial temperature).
- La température finale à laquelle le processus de recherche est interrompu (Frozen temperature).
- Le nombre d'itérations effectuées à température constante (Maximum move).
- Le taux de diminution de la température (Cooling rate).

Le résultat est donné comme l'indique la figure 2.12. Pour chaque décision stratégique, le système :

- Rappelle la variable utilisée.
- Donne la valeur de la variable à l'optimum.
- Donne le coût d'investissement à l'optimum.
- Donne le coût de fonctionnement à l'optimum.

Les coûts totaux d'investissement et de fonctionnement sont donnés au bas de

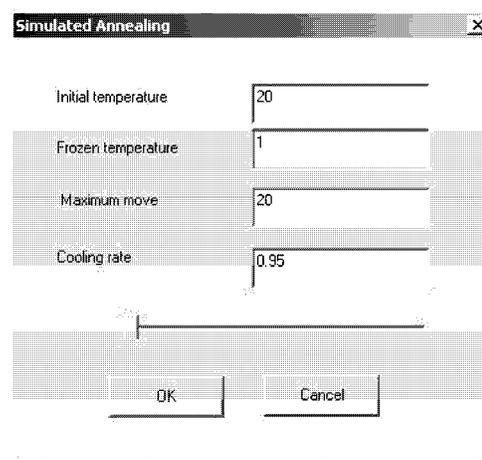
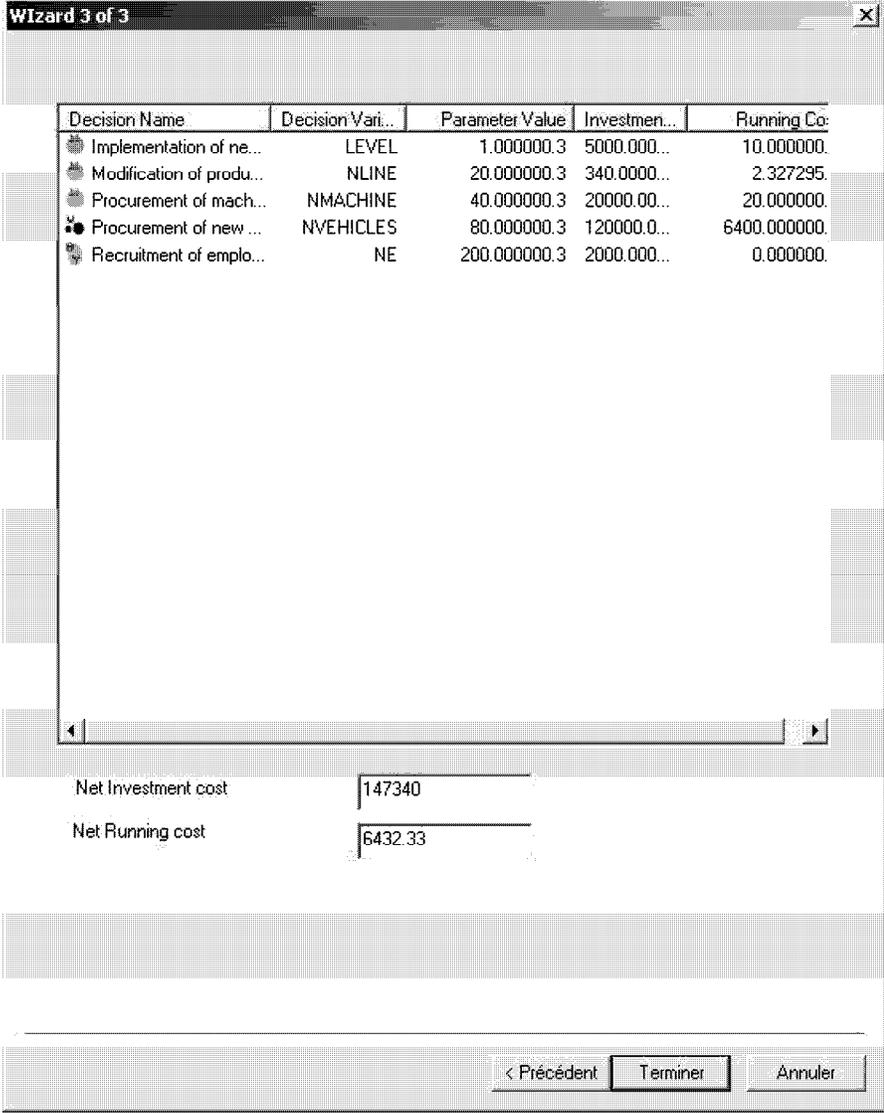


FIG. 2.11 – Introduction des paramètres du recuit simulé

la fenêtre.

### 2.4.5 Conclusion

Le logiciel CAD-SC-SL est un logiciel d'aide à la prise de décision au niveau stratégique. Il guide l'utilisateur dans l'expression de son expertise tout en lui évitant d'oublier les effets de bord de ses décisions. Il propose également une



Decision Name	Decision Vari...	Parameter Value	Investmen...	Running Co:
Implementation of ne...	LEVEL	1.000000.3	5000.000...	10.000000.
Modification of produ...	NLINE	20.000000.3	340.0000...	2.327295.
Procurement of mach...	NMACHINE	40.000000.3	20000.00...	20.000000.
● Procurement of new ...	NVEHICLES	80.000000.3	120000.0...	6400.000000.
Recruitment of emplo...	NE	200.000000.3	2000.000...	0.000000.

Net Investment cost: 147340

Net Running cost: 6432.33

< Précédent   Terminer   Annuler

FIG. 2.12 – Solution optimale

couche intelligente qui permet d'optimiser la décision prise dans le cadre des contraintes fournies par le service commercial.

## Deuxième partie

### Quelques problèmes du niveau stratégique

Dans cette seconde partie de la thèse, nous traitons des problèmes qui se sont posés au niveau stratégique et dont les solutions sont complémentaires du logiciel CAD-SC-SL que nous venons de présenter. Il s'agit essentiellement de problèmes de choix des fournisseurs et de constitution des chaînes d'approvisionnement.

# Chapitre 3

## Choix des fournisseurs

### 3.1 Introduction

Ce chapitre concerne le choix des fournisseurs d'une ou plusieurs unités de fabrication.

Nous considérons d'abord le cas d'une seule unité de fabrication. Dans cette première étude, nous supposons :

- que la quantité que peut fournir un fournisseur est située entre un minimum et un maximum. Le minimum est fixé pour des raisons économiques : le fournisseur considère qu'il n'est pas compétitif s'il produit moins que cette quantité par période élémentaire. Le maximum représente la capacité du fournisseur.
- que le coût associé à une livraison est composé d'un coût fixe et d'un coût concave et croissant.

L'algorithme que nous proposons pour résoudre ce premier problème sélectionne les fournisseurs et détermine la quantité que chacun d'eux fournira au cours de chaque période élémentaire.

Ce problème, bien que proche du problème de transport à coûts concaves (voir [YL99]) s'en écarte du fait que les quantités fournies sont bornées supérieurement et inférieurement. Une autre différence est la discontinuité des coûts à l'origine.

Nous mettons en évidence des propriétés des optima locaux dont nous nous

servons ensuite pour développer une heuristique dont l'efficacité est vérifiée par comparaison avec l'optimum sur une série d'exemples numériques.

Nous considérons ensuite le cas de la sélection de fournisseurs pour plusieurs unités de production. Ici encore, nous utilisons des propriétés des optima locaux pour développer une heuristique. Cette heuristique est ensuite appliquée à des exemples numériques.

## 3.2 Le cas d'une unité de fabrication unique

### 3.2.1 Position du problème

Nous désignons par  $N = \{1, 2, \dots, n\}$  l'ensemble des fournisseurs candidats pour approvisionner une unité de fabrication. Les besoins de l'unité de fabrication au cours d'une période élémentaire sont constants et égaux à  $A$ . Nous rappelons que nous sommes au niveau stratégique et que l'hypothèse qui consiste à dire que la demande sur chaque période est constante est défendable du fait de l'importance des périodes et du fait que la capacité du système de fabrication à utiliser est une décision stratégique.

Chaque fournisseur  $i \in N$  est caractérisé par :

- la quantité minimale  $m_i$  qu'il est disposé à fournir.
- la quantité maximale  $M_i$  qu'il peut fournir.
- le coût  $k_i(x_i)$  d'une livraison  $x_i$

$$k_i(x_i) = \begin{cases} 0 & \text{si } x_i = 0 \\ a_i + g_i(x_i) & \text{si } x_i > 0 \end{cases} \quad (3.1)$$

Dans cette formulation,  $a_i \geq 0$ ,  $g_i$  est concave, continue, continuellement dérivable et croissante et  $\lim_{x_i \rightarrow 0^+} g_i(x_i) \geq 0$ .

Rappelons que, du fait de l'existence d'une quantité minimale et d'une quantité maximale, l'optimum  $x_i^*$  est tel que :

$$x_i^* \in \{0\} \cup [m_i, M_i]$$

Le problème dont nous cherchons la solution s'écrit finalement :

$$\text{Min} \sum_{i=1}^n k_i(x_i) \quad (3.2)$$

sous les contraintes :

$$\sum_{i=1}^n x_i = A \quad (3.3)$$

$$x_i \in \{0\} \cup [m_i, M_i], \text{ pour } i = 1, 2, \dots, n. \quad (3.4)$$

Dans le paragraphe suivant, nous introduisons deux résultats qui vont être utilisés pour construire l'heuristique.

### 3.2.2 Deux résultats de base

Le premier résultat donne les caractéristiques d'une solution optimale. Il indique qu'à l'optimum tous les réapprovisionnements sont extrémaux, sauf un au maximum.

#### Résultat 1 :

Il existe au moins une solution optimale  $X^* = \{x_1^*, x_2^*, \dots, x_n^*\}$  telle que  $x_i^* = m_i$  ou  $x_i^* = 0$  ou  $x_i^* = M_i$  pour  $i = 1, 2, \dots, n$  excepté, peut-être, pour un réapprovisionnement  $x_j^*$  qui vérifie  $m_j < x_j^* < M_j$ .

#### Démonstration

Cette démonstration est basée sur la remarque suivante : si  $f(x)$  et  $h(x)$  sont deux fonctions concaves, croissantes et continuellement dérivables, et si  $\frac{df}{dx}(x_1) > \frac{dh}{dx}(x_2)$ , alors  $\frac{df}{dx}(a) > \frac{dh}{dx}(b)$  pour tout  $x_1 \geq a$  et  $b \geq x_2$  et donc :

$$f(x_1) - f(a) > h(b) - h(x_2)$$

si  $x_1 - a = b - x_2$ . Supposons que  $X^1 = \{x_1^1, x_2^1, \dots, x_n^1\}$  satisfasse les contraintes (3.3) et (3.4), c'est-à-dire que  $X^1$  soit une solution admissible. Supposons éga-

lement que pour deux fournisseurs.  $u, v \in \{1, 2, \dots, n\}$  nous avons :

$$m_u < x_u^1 < M_u$$

et

$$m_v < x_v^1 < M_v$$

et que :

$$\frac{dk_u}{dx_u}(x_u^1) \leq \frac{dk_v}{dx_v}(x_v^1) \quad (3.5)$$

Soit  $\delta = \text{Min}(M_u - x_u^1, x_v^1 - m_v)$

Posons :

$$X^2 = \{x_1^2, x_2^2, \dots, x_n^2\}$$

avec :

$$\left\{ \begin{array}{l} x_u^2 = x_u^1 + \delta \\ x_v^2 = x_v^1 - \delta \\ x_i^2 = x_i^1 \quad \text{si } i \in \{1, 2, \dots, n\} \setminus \{u, v\} \end{array} \right\} \quad (3.6)$$

Comme  $x_u^2 + x_v^2 = x_u^1 + x_v^1$ ,  $X^2$  est encore admissible. De plus, compte tenu de la relation (3.5) et du fait que  $g_i$  est concave et croissante, nous pouvons écrire, en nous souvenant de la remarque faite en début de démonstration :

$$k_u(x_u^2) - k_u(x_u^1) \leq k_v(x_v^1) - k_v(x_v^2)$$

soit :

$$k_u(x_u^2) + k_v(x_v^2) \leq k_u(x_u^1) + k_v(x_v^1).$$

Comme d'autre part

$$k_i(x_i^2) = k_i(x_i^1) \quad \text{pour } i \in \{1, 2, \dots, n\} \setminus \{u, v\},$$

nous pouvons en conclure que :

$$\sum_{i=1}^n k_i(x_i^2) \leq \sum_{i=1}^n k_i(x_i^1).$$

De plus nous voyons que, du fait de la définition de  $\delta$ , soit  $x_u^2 = M_u$ , soit  $x_v^2 = m_v$ . Finalement, un réapprovisionnement au plus, parmi les deux réapprovisionnements considérés, n'est pas extrémal.

Si plus de deux réapprovisionnements ne sont pas extrémaux, nous recommençons le processus. A chaque itération, une solution non extrémale disparaît. Par conséquent, quelle que soit la solution admissible qui ne vérifie pas les conditions du résultat 1, il est toujours possible de trouver une solution admissible qui conduit à un coût inférieur ou égal au précédent et qui vérifie les conditions du résultat 1. Cela achève la démonstration. ■

Le résultat 2 que nous allons présenter maintenant fournit une condition nécessaire pour qu'une solution qui vérifie les conditions du résultat 1 soit optimale.

### Résultat 2

Si une solution admissible  $X^1 = \{x_1^1, x_2^1, \dots, x_n^1\}$  vérifie les conditions du résultat 1 et si il existe  $u, v \in \{1, 2, \dots, n\}$  tels que :

$$m_u < x_u^1 < M_u \text{ et } x_v^1 = m_v^1 \text{ avec } \frac{dk_u}{dx_u}(x_u^1) > \frac{dk_v}{dx_v}(x_v^1) \quad (3.7)$$

ou :

$$x_u^1 = M_u \text{ et } m_v < x_v^1 < M_v \text{ avec } \frac{dk_u}{dx_v}(x_u^1) > \frac{dk_v}{dx_v}(x_v^1) \quad (3.8)$$

alors  $X^1$  n'est pas optimale.

### Démonstration

a. Supposons que (3.7) soit vérifiée. Posons  $\delta = \text{Min}(x_u^1 - m_u, M_v - m_v)$  et considérons  $X^2 = \{x_1^2, x_2^2, \dots, x_n^2\}$  définie comme suit :

$$\left\{ \begin{array}{l} x_u^2 = x_u^1 - \delta \\ x_v^2 = x_v^1 + \delta \\ x_i^2 = x_i^1 \quad \text{si } i \in \{1, 2, \dots, n\} \setminus \{u, v\} \end{array} \right\} \quad (3.9)$$

Alors :

$$\sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i^1 = A$$

et

$$k_u(x_u^1) - k_u(x_u^2) > k_v(x_v^2) - k_v(x_v^1)$$

compte tenu de la remarque précédente.

Donc :

$$k_u(x_u^1) + k_v(x_v^1) > k_u(x_u^2) + k_v(x_v^2)$$

et :

$$\sum_{i=1}^n k_i(x_i^1) > \sum_{i=1}^n k_i(x_i^2)$$

La solution admissible  $X^2$  conduit à un coût inférieur au coût correspondant à  $X^1$  :  $X^1$  n'est donc pas optimal.

b. Supposons que (3.8) soit vérifiée. Posons  $\delta = \text{Min}(M_u - m_u, x_v^1 - m_v)$ .

Nous considérons la solution  $X^2 = \{x_1^2, x_2^2, \dots, x_n^2\}$  définie comme dans la partie

a. Alors, de la même manière que dans a, on montre que la solution admissible  $X^2$  conduit à un coût inférieur au coût correspondant à  $X^1$  :  $X^1$  n'est donc pas optimal. ■

### 3.2.3 Un algorithme heuristique

L'heuristique que nous proposons se déroule en deux étapes. Ici, nous supposons que :

$$\sum_{i=1}^n M_i \geq A$$

Dans le cas contraire, le problème n'a pas de solution.

La première étape peut se résumer comme suit :

- nous ordonnons les fournisseurs dans l'ordre croissant des coûts unitaires sous l'hypothèse que la quantité maximale est livrée ou, si elle est supérieure à  $A$ , sous l'hypothèse que  $A$  est livrée.

- nous affectons les fournisseurs dans cet ordre et arrêtons l'affectation lorsque la quantité maximale livrée dépasse la demande (notée  $A$  dans ce travail).

Par conséquent, si  $1, 2, \dots, n$  sont les fournisseurs classés dans l'ordre croissant des  $k_i(Z)/Z$  où  $Z = \text{Min}(M_i, A)$ , alors le dernier fournisseur affecté est le fournisseur  $j$  tel que :

$$\sum_{i=1}^j M_i \leq A < \sum_{i=1}^{j+1} M_i$$

Dans le cas particulier où  $\sum_{i=1}^j M_i = A$ , l'algorithme s'arrête.

Sinon, nous abordons la seconde étape. Considérons tous les fournisseurs non encore affectés c'est-à-dire, avec les notations ci-dessus, les fournisseurs  $j+1, j+2, \dots, n$ . Pour chacun de ces fournisseurs nous sommes dans l'un des deux cas suivants :

1. La quantité restant à livrer, c'est-à-dire la quantité  $q_j = A - \sum_{i=1}^j M_i$ , est comprise entre la livraison minimale et la livraison maximale du fournisseur suivant. En d'autres termes :

$$m_{j+1} \leq q_j < M_{j+1}.$$

Dans ce cas, il est demandé au fournisseur  $j+1$  de livrer  $q_j$ .

2. La quantité restant à livrer est inférieure à la quantité minimale que le fournisseur suivant est disposé à livrer, c'est-à-dire :  $q_j < m_{j+1}$ .

Dans ce cas, nous demandons au fournisseur  $j+1$  de livrer  $m_{j+1}$  et nous réduisons la livraisons d'un des fournisseurs  $\{1, 2, \dots, j\}$  de la quantité  $v = m_{j+1} - q_j$  de façon à ne pas modifier la quantité totale. Le fournisseur  $i_1 \in \{1, 2, \dots, j\}$  auprès duquel s'effectuera la réduction est tel que :

$$k_{i_1}(M_{i_1}) - k_{i_1}(M_{i_1} - v) = \text{Max}_{i=1,2,\dots,j} \{k_i(M_i) - k_i(M_i - v)\}.$$

Il se peut que la quantité  $v$  à soustraire d'une des livraisons déjà effectuées soit supérieure à l'une de ces livraisons. En d'autres termes, il se peut que

$$v > \text{Min}_{i=1,2,\dots,j} M_i.$$

Dans ce cas, nous sélectionnons d'abord le fournisseur auquel nous allons soustraire la quantité

$$v_1 = \text{Min}_{i=1,2,\dots,j} M_i.$$

Nous recommençons ensuite le processus de sélection pour la quantité  $v_2 = v - v_1$ , et ainsi de suite.

Nous donnons maintenant l'algorithme d'une manière plus formelle.

### **Algorithme 1 : Système de fabrication unique**

#### **A. Entrée des données et tests**

1. Nous introduisons  $m_i, M_i$  et  $k_i$  pour  $i = 1, 2, \dots, n$ .
2. Nous introduisons  $A$ , quantité à livrer.
3. Nous calculons  $h = \sum_{i=1}^n M_i$ .
4. Tests :
  - (a) Si  $A > h$ , le problème n'a pas de solution. Stop.
  - (b) Si  $A < \text{Min}_{i \in \{1,2,\dots,n\}} m_i$ , le problème n'a pas de solution. Stop.

#### **B. Première étape.**

5. Classement des fournisseurs dans l'ordre croissant des quantités  $k_i(a_i)/a_i$  avec  $a_i = \text{Min}(A, M_i)$  pour  $i = 1, 2, \dots, n$ . Afin de ne pas compliquer les notations, nous désignons encore par  $1, 2, \dots, n$  les fournisseurs après classement.
6. Faire  $x = 0$  et  $j = 0$ .
7. Faire  $y = x + M_{j+1}$ .
8. Si  $y < A$  :
  - (a) Faire  $x = y$ .
  - (b) Faire  $j = j + 1$ .
  - (c) Aller en 7.

9. Si  $y = A$ , les fournisseurs  $1, 2, \dots, j + 1$  livreront leur quantité maximale.  
Stop.

**C. Seconde étape.**

10. Poser  $E = \{1, 2, \dots, j\}$ .
11. Calculer  $q = A - \sum_{i \in E} M_j$   
*q est la quantité restant à livrer*
12. Si  $m_{j+1} \leq q < M_{j+1}$ , le fournisseur  $j + 1$  livrera la quantité  $q$ . Stop.
13. Si  $q < m_{j+1}$ , alors :
- (a) Calculer  $v = m_{j+1} - q$ .
  - (b) Le fournisseur  $j + 1$  livrera la quantité  $m_{j+1}$ .
  - (c) Pour tout  $i \in E$ , calculer  $d_i = \text{Min}(M_i - m_i, v)$   
 *$d_i$  est la quantité maximale qui peut être déduite de la livraison de  $i \in E$*
  - (d) Calculer  $d^* = \text{Min}_{i \in E} d_i$   
 *$d^*$  est la quantité maximale qui peut être déduite de la livraison de tout  $i \in E$ .*
  - (e) Calculer  $i_1$  tel que :
 
$$k_{i_1}(M_{i_1}) - k_{i_1}(M_{i_1} - d^*) = \text{Min}_{i \in E} \{k_i(M_i) - k_i(M_i - d^*)\}$$
  - (f) La livraison de  $i_1$  devient  $M_{i_1} - d^*$ .
  - (g) Si  $d^* = v$ , stop.
  - (h) Faire  $v = v - d^*$ .
  - (i) Faire  $E = E \setminus \{i_1\}$
  - (j) Aller en 13 (c)

### 3.3 Un exemple numérique

Nous avons publié l'exemple qui suit dans [CP03a]. Cet exemple comporte 6 fournisseurs. D'après le résultat 1, nous savons que cinq de ces fournisseurs livrent à l'optimum une quantité extrême qui est, pour le fournisseur  $i$ , soit 0, soit  $m_i$ , soit  $M_i$ , le dernier fournisseur étant l'un des six fournisseurs disponibles. La recherche d'une solution optimale conduit donc à examiner  $6 \times 3^5 = 1458$  possibilités. C'est ce que nous avons fait. Nous avons ensuite comparé les résultats obtenus en appliquant l'heuristique aux résultats optimaux.

Les données du problème pour chacun des six fournisseurs sont les suivantes.

Fournisseur 1 :

$$k_1(x) = \left\{ \begin{array}{ll} 0 & \text{si } x = 0 \\ p_1 \cdot x + 2 & \text{si } x > 0 \end{array} \right\}, m_1 = 20, M_1 = 100$$

Fournisseur 2 :

$$k_2(x) = \left\{ \begin{array}{ll} 0 & \text{si } x = 0 \\ 4 - \exp(-p_2 \cdot x) & \text{si } x > 0 \end{array} \right\}, m_2 = 30, M_2 = 80$$

Fournisseur 3 :

$$k_3(x) = \left\{ \begin{array}{ll} 0 & \text{si } x = 0 \\ 7 - \exp(-p_3 \cdot x) & \text{si } x > 0 \end{array} \right\}, m_3 = 10, M_3 = 50$$

Fournisseur 4 :

$$k_4(x) = \left\{ \begin{array}{ll} 0 & \text{si } x = 0 \\ \frac{p_4 \cdot x + 2}{x+1} & \text{si } x > 0 \end{array} \right\}, m_4 = 40, M_4 = 120$$

Fournisseur 5 :

$$k_5(x) = \left\{ \begin{array}{ll} 0 & \text{si } x = 0 \\ 1 + p_5 \cdot x & \text{si } x > 0 \end{array} \right\}, m_5 = 15, M_5 = 70$$

Fournisseur 6 :

$$k_6(x) = \left\{ \begin{array}{ll} 0 & \text{si } x = 0 \\ 5 - \exp(-p_6 \cdot x) & \text{si } x > 0 \end{array} \right\}, m_6 = 5, M_6 = 60$$

De plus,  $A = 340$

Nous donnons ici 20 exemples numériques avec des valeurs différentes des paramètres  $p_1, p_2, p_3, p_4, p_5$  et  $p_6$ .

Dans tous ces cas (voir tableau 3.1) nous voyons que l'heuristique conduit à la solution optimale. C'est le cas pour l'écrasante majorité des exemples traités.

### 3.3.1 Une approche basée sur la programmation dynamique

Dans ce paragraphe, nous supposons que les réapprovisionnements sont des quantités entières. Avec cette hypothèse, il est possible de résoudre notre problème à l'aide de la programmation dynamique.

Posons :

$$C(P, a) = \text{Min}_{x_1, x_2, \dots, x_P} \left( \sum_{i=1}^P k_i(x_i) \right)$$

sous les contraintes :

$$\sum_{i=1}^P x_i = a$$

et

$$x_i = \{0\} \cup [m_i, M_i] \text{ pour } i = 1, \dots, P$$

Nous calculons d'abord  $C(1, a)$  pour  $a = 0, 1, 2, \dots, A$ . Bien entendu, certaines de ces solutions ne sont pas admissibles : dans ce cas, nous considérons que le coût est infini. Nous procédons ensuite par itérations en écrivant :

$$C(P+1, a) = \text{Min}\{C(P, a), \text{Min}_{m_{P+1} \leq x_{P+1} \leq M_{P+1}} [C(P, a - x_{P+1} + k_{P+1}(x_{P+1}))]\}$$

Nous calculons  $C(P+1, a)$  pour  $a = 0, 1, \dots, A$  et nous poursuivons jusqu'à obtenir  $C(n, A)$  qui est la quantité cherchée. Cet algorithme est pseudo polynomial. Cependant, il ne s'applique que lorsque les livraisons sont des quantités entières et il conduit évidemment à l'optimum.

Nous considérons un exemple simple pour illustrer cette approche. Soit  $A = 5$  et trois fournisseurs dont les coûts sont définis comme suit :

$$k_1(x) = \left\{ \begin{array}{ll} 0 & \text{si } x = 0 \\ x + 2 & \text{si } x > 0 \end{array} \right\}, m_1 = 2, M_1 = 3$$

$$k_2(x) = \left\{ \begin{array}{ll} 0 & \text{si } x = 0 \\ 1 + 2.x & \text{si } x > 0 \end{array} \right\}, m_2 = 2, M_2 = 4$$

$$k_3(x) = \left\{ \begin{array}{ll} 0 & \text{si } x = 0 \\ 3.x & \text{si } x > 0 \end{array} \right\}, m_3 = 20, M_3 = 100$$

Nous calculons d'abord  $C(1, a)$  pour  $a = 0, 1, 2, 3, 4$  et  $5$ .

$$C(1, 0) = k_1(0) = 0.$$

$$C(1, 1) = k_1(1) = +\infty \text{ car la solution n'est pas admissible.}$$

$$C(1, 2) = k_1(2) = 4.$$

$$C(1, 3) = k_1(3) = 5.$$

$$C(1, 4) = k_1(4) = +\infty \text{ car la solution n'est pas admissible.}$$

$$C(1, 5) = k_1(5) = +\infty \text{ car la solution n'est pas admissible.}$$

Nous calculons ensuite  $C(2, a)$  pour  $a = 0, 1, 2, 3, 4$  et  $5$  à partir de la formule :

$$C(2, a) = \text{Min}\{C(1, a), \text{Min}_{m_2 \leq x_2 \leq M_2}[C(1, a - x_2) + k_2(x_2)]\}.$$

Nous obtenons :

$$C(2, 0) = \text{Min}\{C(1, 0), \text{Min}_{2 \leq x_2 \leq 4}[C(1, -x_2) + k_2(x_2)]\} = 0 \text{ et } x_2 = 0.$$

$$C(2, 1) = \text{Min}\{C(1, 1), \text{Min}_{2 \leq x_2 \leq 4}[C(1, 1 - x_2) + k_2(x_2)]\} = +\infty.$$

$$C(2, 2) = \text{Min}\{C(1, 2), \text{Min}_{2 \leq x_2 \leq 4}[C(1, 2 - x_2) + k_2(x_2)]\}$$

TAB. 3.1 – Exemple d'un système de production livré par six fournisseurs

Ex.	Valeurs des paramètres						Livraisons						Coût total
	$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$p_6$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	
1.	0.1	0.1	0.3	5	0.3	0.2	80(80)	80(80)	0(0)	120(120)	0(0)	60(60)	23.98(23.98)
2.	3	0.1	0.3	5	0.3	0.2	0(0)	80(80)	50(50)	120(120)	30(30)	60(60)	30.99 (30.99)
3.	0.1	0.1	0.3	50	0.3	0.2	100(100)	80(80)	50(50)	0(0)	50(50)	60(60)	44.01(44.01)
4.	0.1	0.1	0.3	5	0.1	0.2	80(80)	80(80)	0(0)	120(120)	0(0)	60(60)	23.98(23.98)
5.	0.1	0.1	0.2	5	0.1	0.3	80(80)	80(80)	0(0)	120(120)	0(0)	60(60)	23.98(23.98)
6.	0.1	0.1	0.3	5	0.2	0.3	80(80)	80(80)	0(0)	120(120)	0(0)	60(60)	23.98(23.98)
7.	0.1	0.5	0.3	5	0.2	0.3	80(80)	80(80)	0(0)	120(120)	0(0)	60(60)	23.99(23.99)
8.	0.1	0.5	0.3	5	0.2	0.1	80(80)	80(80)	0(0)	120(120)	0(0)	60(60)	23.98 (23.98)
9.	0.5	0.5	0.3	5	0.2	0.1	0(0)	80(80)	50(50)	120(120)	30(30)	60(60)	27.99(27.99)
10.	0.5	0.5	0.3	50	0.2	0.1	0(0)	80(80)	50(50)	120(120)	30(30)	60(60)	72.62(72.62)
11.	0.5	0.8	0.3	5	0.2	0.1	0(0)	80(80)	50(50)	120(120)	30(30)	60(60)	27.993(27.993)
12.	0.5	0.8	0.3	5	0.2	0.5	0(0)	80(80)	50(50)	120(120)	30(30)	60(60)	27.999(27.999)
13.	0.5	0.5	0.3	5	0.2	0.5	0(0)	80(80)	50(50)	120(120)	30(30)	60(60)	27.996(27.996)
14.	0.5	0.1	0.3	5	0.2	0.5	0(0)	80(80)	50(50)	120(120)	30(30)	60(60)	27.99(27.99)
15.	0.1	0.1	0.3	5	0.2	0.5	80(80)	80(80)	0(0)	120 (120)	0(0)	60(60)	23.98(23.98)
16.	0.1	0.1	0.3	5	0.2	0.9	80(80)	80(80)	0(0)	120(120)	0(0)	60(60)	23.99(23.99)
17.	0.1	0.1	0.3	50	0.2	0.9	100(100)	80(80)	50(50)	0(0)	50 (50)	60(60)	30.2(30.2)
18.	3	0.1	0.3	50	0.2	0.9	0(0)	80(80)	50(50)	120(120)	30 (30)	60(60)	72.62(72.62)
19.	3	0.1	0.3	50	0.2	0.1	0(0)	80(80)	50 (50)	120(120)	30 (30)	60(60)	72.61(72.61)
20.	3	0.1	0.1	50	0.2	0.1	0 (0)	80(80)	50 (50)	120 (120)	30 (30)	60 (60)	72.60(72.60)

\* Les valeurs entre parenthèses sont les valeurs optimales.

$$\begin{aligned}
&= \text{Min}\{4, C(1, 0) + k_2(2), C(1, -1) + k_2(3), C(1, -2) + k_2(4)\} \\
&= \text{Min}\{4, 5, +\infty, +\infty\} = 4 \text{ et } x_2 = 0.
\end{aligned}$$

$$\begin{aligned}
C(2, 3) &= \text{Min}\{C(1, 3), \text{Min}_{2 \leq x_2 \leq 4}[C(1, 3 - x_2) + k_2(x_2)]\} \\
&= \text{Min}\{5, C(1, 1) + k_2(2), C(1, 0) + k_2(3), C(1, -1) + k_2(4)\} \\
&= \text{Min}\{5, +\infty, 7, +\infty\} = 5 \text{ et } x_2 = 0.
\end{aligned}$$

$$\begin{aligned}
C(2, 4) &= \text{Min}\{C(1, 4), \text{Min}_{2 \leq x_2 \leq 4}[C(1, 4 - x_2) + k_2(x_2)]\} \\
&= \text{Min}\{+\infty, 4 + 5, +\infty, 9\} = 9 \text{ avec } x_2 = 2 \text{ ou } 4.
\end{aligned}$$

$$\begin{aligned}
C(2, 5) &= \text{Min}\{C(1, 5), \text{Min}_{2 \leq x_2 \leq 4}[C(1, 5 - x_2) + k_2(x_2)]\} \\
&= \text{Min}\{+\infty, C(1, 3) + k_2(2), C(1, 2) + k_2(3), C(1, 1) + k_2(4)\} \\
&= \text{Min}\{+\infty, 5 + 5, 4 + 7, +\infty\} = 10 \text{ avec } x_2 = 2.
\end{aligned}$$

Enfin, nous calculons  $C(3, 5)$

$$\begin{aligned}
C(3, 5) &= \text{Min}\{C(2, 5), \text{Min}_{1 < x_2 < 4}[C(2, 5 - x_2) + k_2(x_2)]\} \\
&= \text{Min}\{10, C(2, 4) + k_3(1), C(2, 3) + k_3(2), C(2, 2) + k_3(3), C(2, 1) + \\
&k_3(4)\} \\
&= \text{Min}\{10, 9 + 3, 5 + 6, 4 + 9, +\infty\} = 10 \text{ avec } x_3 = 0.
\end{aligned}$$

Donc la solution optimale est :

$x_3 = 0, x_2 = 2, x_1 = 3$  pour un coût total de 10.

## 3.4 Le cas de plusieurs unités de fabrication

### 3.4.1 Position du problème

Nous disposons encore d'un ensemble  $N = \{1, 2, \dots, n\}$  de fournisseurs candidats pour approvisionner un ensemble  $M = \{1, 2, \dots, m\}$  d'unités de fabrication. Les besoins d'une unité de fabrication  $j \in M$  durant chaque période élémentaire sont constants et égaux à  $A_j$ .

Les deux contraintes suivantes s'appliquent aux réapprovisionnements :

- La quantité minimale qu'un fournisseur  $i \in N$  est disposé à fournir à l'unité de fabrication  $j \in M$  est connue et notée  $m_{i,j}$ .

- La quantité maximale qu'un fournisseur  $i \in N$  est capable de fournir durant une période élémentaire est connue et notée  $M_i$ .

Le coût d'une livraison  $x_{i,j}$  effectuée par le fournisseur  $i \in N$  au système de fabrication  $j \in M$  s'écrit  $k_{i,j}(x_{i,j})$  et :

$$k_{i,j}(x_{i,j}) = \left\{ \begin{array}{ll} 0 & \text{si } x_{i,j} = 0 \\ a_{i,j} + g_{i,j}(x_{i,j}) & \text{si } x_{i,j} > 0 \end{array} \right\}$$

avec :

$a_{i,j} \geq 0$  et  $g_{i,j}$  est croissante, concave, continue et continuellement dérivable et  $\lim_{x \rightarrow 0^+} g_{i,j}(x) \geq 0$

Le problème à résoudre s'écrit :

$$\text{Minimiser } \sum_{i=1}^n \sum_{j=1}^m k_{i,j}(x_{i,j}) \quad (3.10)$$

sous les contraintes :

$$\sum_{i=1}^n x_{i,j} = A_j, \forall j \in M \quad (3.11)$$

$$\sum_{j=1}^m x_{i,j} \leq M_i, \quad \forall i \in N \quad (3.12)$$

$$x_{i,j} \in \{0\} \cup [m_{i,j}, M_i], \text{ pour } i = 1, 2, \dots, n \text{ et } j = 1, 2, \dots, m. \quad (3.13)$$

### 3.4.2 Deux résultats de base

Le premier résultat présenté dans ce paragraphe donne une condition nécessaire pour qu'une solution soit optimale.

**Résultat 3**

Soit  $X^1 = \{x_{i,j}^1\}_{i \in N, j \in M}$  une solution admissible du problème. Une condition nécessaire pour que  $X^1$  soit optimale est que, pour tout  $j \in M$ , nous ayons l'une des deux conditions suivantes :

1.

$$x_{i,j} \in \{0\} \cup \{A_j\} \cup \{m_{i,j}\}$$

pour  $i \in E(j)$  où  $E(j) \subset N$

ou :

2.

$$x_{i,j} \notin \{0\} \cup \{A_j\} \cup \{m_{i,j}\}$$

pour  $i \in N/E(j)$  mais, dans ce cas :

$$\sum_{k=1}^m x_{i,k} = M_i$$

pour  $i \in N/E(j)$  sauf, peut-être, pour un indice  $i$ .

Bien entendu,  $E(j)$  peut être vide.

Remarque :

Ce résultat généralise le résultat 1 au cas de plusieurs unités de fabrication.

Il indique que, pour une unité de fabrication  $j \in M$  donnée :

1. il existe un sous-ensemble (éventuellement vide) de fournisseurs dont la livraison est extrême,
2. ou le fournisseur ne livre pas de quantité extrême à  $j$  mais, dans ce cas, le total de ses livraisons aux unités de fabrication est égal à sa capacité maximale.

**Démonstration :**

Soit  $X^1 = \{x_{i,j}^1\}_{i \in N, j \in M}$  une solution admissible qui ne vérifie pas les conditions du résultat 3. Cela signifie qu'il existe deux fournisseurs  $r, s \in N$  et  $j \in M$  tels que :

$$x_{r,j}^1 \notin \{0\} \cup \{A_j\} \cup \{m_{i,j}\} \text{ et } \sum_{k=1}^m x_{r,k}^1 < M_r$$

et :

$$x_{s,j}^1 \notin \{0\} \cup \{A_j\} \cup \{m_{s,j}\} \text{ et } \sum_{k=1}^m x_{s,k}^1 < M_s.$$

Supposons que :

$$\frac{dk_{r,j}}{dx_{r,j}}(x_{r,j}^1) \geq \frac{dk_{s,j}}{dx_{s,j}}(x_{s,j}^1)$$

a. *Construction d'une solution admissible  $X^2$ .*

Nous posons :

$$\delta = \text{Min}\left\{M_s - \sum_{u=1}^m x_{s,u}^1, x_{r,j}^1 - m_{r,j}\right\} \quad (3.14)$$

et :

$$x_{r,j}^2 = x_{r,j}^1 - \delta \quad (3.15)$$

$$x_{s,j}^2 = x_{s,j}^1 + \delta \quad (3.16)$$

$$x_{i,v}^2 = x_{i,v}^1 \quad (3.17)$$

pour  $(i, v) \neq (r, j)$  et  $(i, v) \neq (s, j)$

En additionnant (3.15) et (3.16), nous voyons que :

$$x_{r,j}^2 + x_{s,j}^2 = x_{r,j}^1 + x_{s,j}^1$$

Par conséquent, en tenant compte de (3.17), nous voyons que :

$$\sum_{i=1}^n x_{i,j}^2 = \sum_{i=1}^n x_{i,j}^1 = A$$

car  $X^1$  est admissible.

Donc la solution  $X^2$  vérifie la condition (3.11).

Considérons maintenant la condition (3.12). Elle est vérifiée pour tout  $\{x_{i,j}^2\}_{j \in M}$  à condition que  $i \neq r$  et  $i \neq s$  (voir 3.17).

Montrons maintenant que cette condition est vérifiée pour  $i = r$ . D'après (3.15) :

$$\sum_{k=1}^m x_{r,k}^2 = \sum_{k=1}^m x_{r,k}^1 - \delta$$

Mais, compte tenu de (3.14) :

$$\sum_{k=1}^m x_{r,k}^1 - \delta \leq \sum_{k=1, k \neq j}^m x_{r,k}^1 + m_{r,j}$$

Comme, par hypothèse,

$$x_{r,j}^1 \notin \{0\} \cup \{A_j\} \cup \{m_{r,j}\},$$

alors  $x_{r,j}^1 > m_{r,j}$ , d'où :

$$\sum_{k=1, k \neq j}^m x_{r,k}^1 + m_{r,j} < \sum_{k=1}^m x_{r,k}^1 < M_r$$

Finalement :

$$\sum_{k=1}^m x_{r,k}^2 < M_r$$

Montrons que la condition (3.12) est vérifiée pour  $i = s$ . D'après (3.16) :

$$\sum_{k=1}^m x_{s,k}^2 = \sum_{k=1}^m x_{s,k}^1 + \delta$$

Mais, compte tenu de (3.14) :

$$\sum_{k=1}^m x_{s,k}^1 + \delta \leq \sum_{k=1}^m x_{s,k}^1 + M_s - \sum_{u=1}^m x_{s,u}^1 = M_s$$

Finalement :

$$\sum_{k=1}^m x_{s,k}^2 \leq M_s$$

Nous voyons donc que  $X^2$  vérifie la condition (3.12). Enfin, en considérant les relations (3.14) à (3.17), nous constatons que  $X^2$  vérifie la contrainte (3.13).

Finalement, nous pouvons affirmer que  $X^2$  est admissible.

b.  $X^2$  vérifie les conditions données dans le résultat 3

Considérons la définition (3.14). Deux cas peuvent se présenter :

b1.

$$\delta = M_s - \sum_{u=1}^m x_{s,u}$$

Alors, sachant que :

$$\sum_{k=1}^m x_{s,k}^1 < M_s$$

et compte tenu de (3.16), nous obtenons :

$$\sum_{k=1}^m x_{s,k}^2 = M_s,$$

et  $X^2$  vérifie les conditions du résultat 3.

b2 :

$$\delta = x_{r,j}^1 - m_{r,j}.$$

Dans ce cas, comme nous l'avons vu dans le paragraphe (a) de cette démonstration,  $x_{r,j}^2 = m_{r,j}$  et  $X^2$  vérifie encore les conditions du résultat 3.

c. *Le coût associé à  $X^2$  est inférieur ou égal au coût associé à  $X^1$ .*

Compte tenu de l'hypothèse :

$$\frac{dk_{r,j}}{dx_{r,j}}(x_{r,j}^1) \geq \frac{dk_{s,j}}{dx_{s,j}}(x_{s,j}^1)$$

et du fait que les fonctions de coût sont croissantes et concaves, nous savons que :

$$\frac{dk_{r,j}}{dx}(x_1) \geq \frac{dk_{s,j}}{dx}(x_2)$$

pour tout  $x_1 \leq x_{r,j}^1$  et  $x_2 \geq x_{s,j}^1$ .

Par conséquent :

$$k_{r,j}(x_{r,j}^1) - k_{r,j}(x_{r,j}^2) \geq k_{s,j}(x_{s,j}^2) - k_{s,j}(x_{s,j}^1)$$

soit :

$$k_{r,j}(x_{r,j}^1) + k_{s,j}(x_{s,j}^1) \geq k_{r,j}(x_{r,j}^2) + k_{s,j}(x_{s,j}^1)$$

Comme tous les autres coûts sont inchangés, nous pouvons conclure que le coût associé à  $X^2$  est inférieur ou égal au coût associé à  $X^1$ .

Finalement, nous voyons que  $X^2$  :

- est admissible,
- vérifie les conditions du résultat 3,
- a un coût inférieur ou égal au coût associé à  $X^1$ .

En conclusion, quelle que soit la solution admissible qui ne vérifie pas les conditions du résultat 3, nous pouvons construire une solution au moins aussi bonne qui, soit vérifie les conditions de résultat 3, soit s'en rapproche. Dans ce dernier cas, il suffit de construire  $X^3$  à partir de  $X^2$ , puis  $X^4$  à partir de  $X^3$  et ainsi de suite jusqu'à atteindre une solution qui vérifie les conditions du résultat 3.

Ceci achève la démonstration. ■

Le résultat que nous présentons maintenant fournit une condition nécessaire pour qu'une solution qui vérifie les conditions du résultat 3 soit optimale.

#### Résultat 4

Soit  $X^1 = \{x_{i,j}^1\}_{i \in N, j \in M}$  une solution admissible qui vérifie les conditions du résultat 3. Si deux unités de production  $r, s \in M$  sont approvisionnées par deux fournisseurs  $i, j \in N$  et si :

$$x_{i,r}^1 \notin \{m_{i,r}, A_r\}, x_{i,s}^1 \notin \{m_{i,s}, A_s\}$$

$$x_{j,r}^1 \notin \{m_{j,r}, A_r\}, x_{j,s}^1 \notin \{m_{j,s}, A_s\}$$

et si :

$$A = \frac{dk_{i,r}}{dx_{i,r}}(x_{i,r}^1) + \frac{dk_{j,s}}{dx_{j,s}}(x_{j,s}^1) < \frac{dk_{j,r}}{dx_{j,r}}(x_{j,r}^1) + \frac{dk_{i,s}}{dx_{i,s}}(x_{i,s}^1) = B$$

ou :

$$A = \frac{dk_{i,r}}{dx_{i,r}}(x_{i,r}^1) + \frac{dk_{j,s}}{dx_{j,s}}(x_{j,s}^1) > \frac{dk_{j,r}}{dx_{j,r}}(x_{j,r}^1) + \frac{dk_{i,s}}{dx_{i,s}}(x_{i,s}^1) = B,$$

alors la solution  $X^1$  n'est pas optimale.

### Démonstration

Nous définissons :

$$S_1 = \text{Min}\{x_{j,r}^1 - m_{j,r}, x_{i,s}^1 - m_{i,s}\}$$

et :

$$S_2 = \text{Min}\{x_{i,r}^1 - m_{i,r}, x_{j,s}^1 - m_{j,s}\}$$

Supposons que  $A < B$  et posons :

$$\begin{aligned} x_{i,r}^2 &= x_{i,r}^1 + S_1 \\ x_{j,s}^2 &= x_{j,s}^1 + S_1 \\ x_{j,r}^2 &= x_{j,r}^1 - S_1 \\ x_{i,s}^2 &= x_{i,s}^1 - S_1. \end{aligned} \tag{3.18}$$

Nous notons  $X^2$  la solution obtenue en remplaçant, dans  $X^1$ ,  $x_{i,r}^1$ ,  $x_{j,s}^1$ ,  $x_{j,r}^1$  et  $x_{i,s}^1$  respectivement par  $x_{i,r}^2$ ,  $x_{j,s}^2$ ,  $x_{j,r}^2$  et  $x_{i,s}^2$

**a.  $X^2$  est admissible**

**a1.  $X^2$  vérifie (3.11)**

$$x_{i,r}^2 = x_{i,r}^1 + S_1$$

$$x_{j,r}^2 = x_{j,r}^1 - S_1$$

Donc,

$$x_{i,r}^2 + x_{j,r}^2 = x_{i,r}^1 + x_{j,r}^1$$

et

$$\sum_{k=1}^n x_{k,r}^2 = \sum_{k=1}^n x_{k,r}^1 = A_r$$

De la même manière :

$$x_{j,s}^2 = x_{j,s}^1 + S_1$$

$$x_{i,s}^2 = x_{i,s}^1 - S_1$$

Donc,

$$x_{j,s}^2 + x_{i,s}^2 = x_{j,s}^1 + x_{i,s}^1$$

et

$$\sum_{k=1}^n x_{k,s}^2 = \sum_{k=1}^n x_{k,s}^1 = A_s$$

Enfin :

$$\sum_{k=1}^n x_{k,j}^2 = \sum_{k=1}^n x_{k,j}^1 = A_j.$$

Car  $x_{k,j}^2 = x_{k,j}^1$  pour  $k \in N$  et  $j \in M \setminus \{r, s\}$ .

$X^2$  vérifie donc bien la condition (3.11).

**a2.  $X^2$  vérifie (3.12)**

$$x_{i,r}^2 = x_{i,r}^1 + S_1$$

$$x_{i,s}^2 = x_{i,s}^1 - S_1$$

Donc :  $x_{i,r}^2 + x_{i,s}^2 = x_{i,r}^1 + x_{i,s}^1$  Et :

$$\sum_{k=1}^m x_{i,k}^2 = \sum_{k=1}^m x_{i,k}^1 \leq M_i$$

car  $x_{i,k}^2 = x_{i,k}^1$  pour  $k \in M \setminus \{r, s\}$

De la même manière :

$$x_{j,r}^2 = x_{j,r}^1 - S_1$$

$$x_{j,s}^2 = x_{j,s}^1 + S_1$$

Donc,  $x_{j,r}^2 + x_{j,s}^2 = x_{j,r}^1 + x_{j,s}^1$  et

$$\sum_{k=1}^m x_{j,k}^2 = \sum_{k=1}^m x_{j,k}^1 \leq M_j$$

car  $x_{j,k}^2 = x_{j,k}^1$  pour  $k \in M \setminus \{r, s\}$

$X^2$  vérifie donc bien la condition (3.12).

### a3. $X^2$ vérifie la condition (3.13)

Puisque  $\sum_{k=1}^m x_{j,k}^2 \leq M_j$ , alors  $x_{j,k}^2 \leq M_j$ ,  $\forall k \in M$  et, en particulier pour  $k = s$ .

De la même manière :  $\sum_{k=1}^m x_{i,k}^2 \leq M_i$ , conduit à  $x_{i,k}^2 \leq M_i$ ,  $\forall k \in M$  et, en particulier, pour  $k = r$ .

De plus :

$$\begin{aligned} x_{i,s}^2 &= x_{i,s}^1 - S_1 \geq x_{i,s}^1 - (x_{i,s}^1 - m_{i,s}) = m_{i,s} \\ x_{j,r}^2 &= x_{j,r}^1 - S_1 \geq x_{j,r}^1 - (x_{j,r}^1 - m_{j,r}) = m_{j,r} \end{aligned}$$

Finalement,  $X^2$  est admissible.

### b. $X^1$ n'est pas optimale

Compte tenu du fait que  $A < B$ , du fait que les fonctions sont concaves et croissantes, et des relations (3.18), nous avons :

$$\begin{aligned} & [k_{i,r}(x_{i,r}^2) - k_{i,r}(x_{i,r}^1)] + [k_{j,s}(x_{j,s}^2) - k_{j,s}(x_{j,s}^1)] \\ & < [k_{j,r}(x_{j,r}^1) - k_{j,r}(x_{j,r}^2)] + [k_{i,s}(x_{i,s}^1) - k_{i,s}(x_{i,s}^2)] \end{aligned}$$

Cette inégalité conduit à :

$$\begin{aligned} & k_{i,r}(x_{i,r}^2) + k_{j,s}(x_{j,s}^2) + k_{j,r}(x_{j,r}^2) + k_{i,s}(x_{i,s}^2) \\ & < k_{i,r}(x_{i,r}^1) + k_{j,s}(x_{j,s}^1) + k_{j,r}(x_{j,r}^1) + k_{i,s}(x_{i,s}^1) \end{aligned} \quad (3.19)$$

Comme les coûts correspondant aux autres approvisionnements sont les mêmes pour  $X^1$  et  $X^2$  car ces approvisionnements sont identiques, l'inégalité (3.19) suffit à montrer que la coût correspondant à  $X^2$  est inférieur au coût correspondant à  $X^1$ .

$X^1$  n'est donc pas optimal. ■

### 3.4.3 Un algorithme heuristique.

L'algorithme heuristique que nous présentons maintenant est basé sur les résultats précédents. Il se base également sur l'algorithme 1 (Système de fabrication unique) que nous noterons SYFU dans la suite.

Nous ne classons ni les fournisseurs, ni les systèmes de fabrication. Fournisseurs et systèmes de fabrication sont présentés dans un ordre quelconque. La seule contrainte que nous imposons est que cet ordre soit le même tout au long de l'algorithme.

Cette heuristique, que nous désignerons par SYFM (Système de fabrication multiples) se déroule en deux étapes en boucle. La première étape opère une sélection des fournisseurs. La seconde étape affine ce premier choix.

#### **Algorithme 2 : Systèmes de fabrication multiples (SYFM)**

##### ***Etape1 : Sélection des fournisseurs initiaux.***

Nous recherchons d'abord un fournisseur  $i$  dont la capacité restante est strictement positive et un système de fabrication  $j$  dont les besoins à satisfaire sont strictement positifs. Les recherches sont faites dans l'ordre quelconque donné en entrée.

Nous appliquons alors SYFU au système de fabrication  $j$  en considérant l'ensemble des fournisseurs. Soit  $q_{i,j}$  le réapprovisionnement du fournisseur  $i$  en direction du système de fabrication  $j$ .

Si  $q_{i,j} > 0$ , nous passons à l'étape 2.

Si  $q_{i,j} = 0$ , nous passons au fournisseur  $i$  suivant et recommençons le même processus à partir de la première unité de fabrication. Si les besoins résiduels, notés  $A_j$ , de toutes les unités de fabrication  $j$  sont nuls, la solution est obtenue et nous arrêtons le processus. Si toutes les capacités résiduelles des fournisseurs sont nulles et si au moins un besoin résiduel est strictement positif, alors le problème n'a pas de solution.

**Etape 2 : Affinage**

A cette étape nous considérons une à une les unités de fabrication qui n'ont pas encore été examinées, soit l'ensemble  $F \subset \{1, 2, \dots, m\}$ , et nous calculons les deux coûts suivants :

1. Soit  $k \in F$  l'unité de fabrication examinée avec son besoin résiduel, c'est-à-dire le besoin non encore satisfait précédemment. Nous lui appliquons l'algorithme SYFU en intégrant tous les fournisseurs. Le coût obtenu est le coût le plus bas pour le besoin résiduel de  $k$  : nous notons ce coût  $C_k^i$ . rappelons que ce coût n'est pas toujours optimal car SYFU est une heuristique, mais l'expérience numérique a montré que la solution obtenue est optimale la plupart du temps.
2. Le second coût se calcule comme le précédent, mais après avoir réduit la capacité du fournisseur  $i$  de  $q_{i,j}$ . Nous désignons ce coût par  $C_k^2$ . Le problème n'a pas de solution dans ce cas, nous posons  $C_k^2 = +\infty$ .

Après avoir ainsi exploré toutes les unités de fabrication de  $F$ , nous retenons l'unité  $k^*$  qui vérifie :

$$C_{k^*}^2 - C_{k^*}^1 = \text{Max}_{k \in F} (C_k^2 - C_k^1)$$

L'unité de fabrication  $k^*$  est celle qui conduirait au plus grand supplément de coût pour satisfaire la demande résiduelle : nous la sélectionnons donc pour ne pas avoir à la sélectionner plus tard et, ainsi, être fortement pénalisé. Par conséquent, le fournisseur  $i$  est affecté à l'unité de fabrication  $k^*$ .

La capacité du fournisseur  $i$  et les besoins de l'unité de fabrication  $k^*$  sont réduits en conséquence et nous revenons à l'étape 1.

Le lecteur aura compris que l'étape 1 sert à s'assurer que le fournisseur  $i$  est sélectionnable. L'étape 2 optimise le choix de l'unité de fabrication à laquelle le fournisseur  $i$  va être affecté.

### 3.4.4 Applications numériques

Les applications numériques que nous proposons comportent trois fournisseurs et quatre unités de production. Nous voyons qu'au maximum  $2^3 - 1 = 7$  ensembles de fournisseurs peuvent être sollicités. Bien entendu, certains de ces ensembles ne conduisent à aucune solution.

Un de ces ensembles étant choisi, la solution optimale qui lui correspond peut être calculée en appliquant un algorithme de programmation linéaire, à condition que les coûts soient linéaires sur  $]0, +\infty)$ . C'est afin de pouvoir comparer les résultats donnés par l'heuristique SYFM aux résultats optimaux que nous avons choisi des coûts linéaires sur  $]0, +\infty)$ .

Les données sont les suivantes. Les paramètres  $A_{i,j}$  et  $B_{i,j}$  sont donnés dans tableaux.

Fournisseur 1 :

$$M_1 = 80, m_{1,i} = 10 \text{ pour } i = 1, 2, 3 \text{ et } 4.$$

$$k_{1,i}(x) = \left\{ \begin{array}{ll} 0 & \text{si } x = 0 \\ A_{1,i} + B_{1,i}x & \text{si } x > 0 \end{array} \right\}, \text{ pour } i = 1, 2, 3, 4.$$

Fournisseur 2 :

$$M_2 = 180, m_{2,i} = 10 \text{ pour } i = 1, 2, 3 \text{ et } 4.$$

$$k_{2,i}(x) = \left\{ \begin{array}{ll} 0 & \text{si } x = 0 \\ A_{2,i} + B_{2,i}x & \text{si } x > 0 \end{array} \right\}, \text{ pour } i = 1, 2, 3, 4.$$

Fournisseur 3 :

$$M_3 = 120, m_{3,i} = 10 \text{ pour } i = 1, 2, 3 \text{ et } 4.$$

$$k_{3,i}(x) = \left\{ \begin{array}{ll} 0 & \text{si } x = 0 \\ A_{3,i} + B_{3,i}x & \text{si } x > 0 \end{array} \right\}, \text{ pour } i = 1, 2, 3, 4.$$

Les besoins des unités de fabrication sont donnés dans le tableau 3.2.

Les résultats figurent dans le tableaux 3.3. Ils concernent dix exemples. La première colonne contient les numéros des exemples. La partie gauche du

TAB. 3.2 – Besoins des unités de fabrication

Unités de fabrication	1	2	3	4
Demand	70	50	90	80

tableau contient les valeurs des paramètres. La partie droite du tableau contient les résultats. Dans chaque ligne, le chiffre du haut est la valeur optimale, alors que le chiffre du bas est la valeur fournie par l'heuristique. En tête de colonne, l'indication  $i - j$  signifie : livraison du fournisseur  $i$  à l'unité de fabrication  $j$ .

En observant les coûts (dernière colonne) nous voyons que l'heuristique conduit à l'optimum dans deux cas. Les performances de l'heuristique sont ici moins bonnes que dans le cas d'une unité de fabrication unique.

TAB. 3.3 – Exemple d'un système de production livré par six fournisseurs

Ex. No.	$A_{1,1}$		$A_{1,2}$		$A_{1,3}$		$A_{1,4}$		$A_{2,1}$		$A_{2,2}$		$A_{2,3}$		$A_{2,4}$		$A_{3,1}$		$A_{3,2}$		$A_{3,3}$		$A_{3,4}$		Coût totale	
	$B_{1,1}$	$B_{1,2}$	$B_{1,3}$	$B_{1,4}$	$B_{2,1}$	$B_{2,2}$	$B_{2,3}$	$B_{2,4}$	$B_{3,1}$	$B_{3,2}$	$B_{3,3}$	$B_{3,4}$	$B_{1,1}$	$B_{1,2}$	$B_{1,3}$	$B_{1,4}$	$B_{2,1}$	$B_{2,2}$	$B_{2,3}$	$B_{2,4}$	$B_{3,1}$	$B_{3,2}$	$B_{3,3}$	$B_{3,4}$		
1.	20	10	50	40	30	25	45	25	50	10	10	5	1	1	1	1	1	1	2	2	2	2	2	2	2	294
2.	1	1.1	0.1	0.5	0.8	0.9	0.7	1.2	0.8	1.2	1.4	1.5	0	0	0	80	70	0	0	90	0	0	0	0	0	309
3.	20	10	20	20	30	25	45	25	50	10	10	5	0	0	0	80	70	90	0	0	0	0	0	0	0	272
1	1.1	0.1	0.1	0.5	0.8	0.9	0.7	1.2	0.8	0.2	1.4	1.5	0	0	0	80	0	70	0	0	80	0	50	10	0	279
3.	20	10	50	40	30	40	45	25	30	10	10	5	0	0	0	80	0	0	90	0	70	50	0	0	0	294
1	1.1	0.1	0.1	0.5	0.8	0.9	0.7	1.2	0.8	0.2	1.4	1.5	0	0	0	80	0	70	0	0	80	0	50	10	0	309
4.	20	10	50	40	80	40	45	25	30	10	10	10	0	0	0	80	0	0	90	0	70	50	0	0	0	294
1	1.1	0.1	0.1	0.5	0.8	0.9	0.7	1.2	0.8	0.2	0.8	1.0	0	0	0	80	0	70	0	0	20	0	50	10	60	301
5.	10	10	50	40	30	40	45	25	30	10	10	5	0	0	0	80	0	0	90	0	70	50	0	0	0	294
0.8	0.8	0.1	0.1	0.5	0.8	0.9	0.7	1.2	0.8	0.2	1.4	1.5	0	0	0	80	0	70	0	0	80	0	50	10	0	309
6.	10	10	50	40	30	10	10	25	10	10	10	5	0	0	0	80	0	0	90	0	70	50	0	0	0	239
0.8	0.8	0.1	0.1	0.5	0.8	0.9	0.7	1.2	0.8	0.2	1.4	1.5	0	0	0	80	0	0	90	0	70	50	0	0	0	239
7.	40	40	50	40	30	10	10	25	10	10	10	5	0	0	0	80	0	0	90	0	70	50	0	0	0	239
0.8	0.8	0.5	0.5	0.5	0.8	0.9	0.7	1.2	0.8	0.2	1.4	1.5	0	0	0	80	0	0	90	0	70	50	0	0	0	239
8.	10	10	50	40	30	25	45	25	50	10	30	5	70	10	0	0	0	0	90	0	0	40	0	80	0	293
0.5	1.1	0.1	0.1	0.5	0.8	0.9	0.7	1.2	0.8	0.2	1.0	1.2	70	0	0	0	0	0	90	10	0	50	0	70	0	299
9.	10	10	50	40	30	25	45	25	50	10	10	5	0	0	0	80	70	20	0	0	0	30	90	0	0	253
0.5	1.1	0.1	0.1	0.5	0.8	0.9	0.7	1.2	0.8	0.2	0.2	1.2	70	10	0	0	0	10	0	80	0	30	90	0	0	265
10.	10	10	10	40	30	25	45	25	50	10	10	5	0	0	0	80	70	20	0	0	0	30	90	0	0	253
0.5	1.1	0.1	0.5	0.5	0.8	0.9	0.7	1.2	0.8	0.2	0.2	1.2	70	10	0	0	0	10	0	80	0	30	90	0	0	265

## Chapitre 4

# Planification des capacités lors de la conception d'une chaîne d'approvisionnement

### 4.1 Introduction

Le problème que nous présentons dans ce chapitre se pose au niveau stratégique. L'objectif est de sélectionner des fournisseurs, des producteurs et des distributeurs capables de travailler ensemble pour répondre à une nouvelle demande du marché. La sélection de ces acteurs se fait dans une population connue, définie sur la base de critères qualitatifs comme, par exemple, la solidité financière, la politique de qualité, l'adaptabilité au marché, la capacité d'innovation, les coopérations précédentes avec des entités candidates, pour ne citer que quelques aspects.

Le choix définitif des partenaires de la chaîne d'approvisionnement se fait sur la base de critères quantitatifs comme, par exemple, les coûts d'investissement en ressources de production, de transport, de traitement de l'information, et les coûts de fonctionnement. Des travaux ont déjà été faits dans ce domaine. Nous avons examiné les principaux résultats obtenus dans le premier chapitre. Dans ce chapitre, nous nous attachons à rassembler les capacités nécessaires à la satisfaction d'une nouvelle opportunité offerte par le marché. La demande

à satisfaire durant chaque période élémentaire est connue. Au niveau stratégique, une période élémentaire est le semestre ou l'année. De plus, la demande dont il est question ici est un objectif résultant d'études de marché.

Tous les partenaires potentiels disposent d'une capacité (de production, de transport, de traitement de l'information) limitée qu'il faut compléter pour faire face à la nouvelle demande.

Nous allons montrer que le problème à résoudre est un problème de programmation linéaire mixte, c'est-à-dire un problème dans lequel certaines variables prennent des valeurs réelles et d'autres des valeurs entières. La difficulté se trouve dans la taille que peut atteindre ce problème dans la pratique.

Nous présentons le problème dans le paragraphe suivant.

Des propriétés du problème sont données dans le paragraphe 4.3. Ces propriétés sont utilisées pour établir l'algorithme que nous donnons dans le paragraphe 4.4. Le paragraphe 4.5 est consacré à un exemple numérique. Le paragraphe 4.6 est la conclusion.

## 4.2 Présentation du problème

### 4.2.1 Description du problème

Nous cherchons à constituer une chaîne d'approvisionnement à trois niveaux qui sont :

- le niveau des fournisseurs,
- le niveau des producteurs,
- le niveau des distributeurs.

Tous ces candidats satisfont les critères qualitatifs que nous avons cité en introduction. Chaque distributeur doit être en mesure de satisfaire une demande donnée durant chaque période élémentaire. Les fournisseurs répondent aux demandes des producteurs qui, à leur tour, réapprovisionnent les distributeurs. Chaque fournisseur a son propre coût unitaire et son propre coût de

transport vers chaque producteur. De même chaque producteur a son propre coût de production unitaire et son propre coût de transport vers chaque distributeur.

Nous supposons que, jusqu'à l'horizon choisi pour l'étude, ces coûts sont indépendants du temps. Chaque fournisseur et chaque producteur dispose d'une capacité limitée qu'il peut mettre au service de la chaîne d'approvisionnement envisagée. Ces capacités peuvent être accrues en investissant en ressources physiques et humaines. Nous supposons que l'investissement se fait au début de la conception de la chaîne d'approvisionnement. Il se peut, bien entendu, que certaines capacités se révèlent insuffisantes après quelques années de fonctionnement du fait de l'évolution du marché : alors le problème se posera à nouveau à l'identique.

Nous supposons que les coûts de transport et de production [matières premières et produits finis] sont des fonctions linéaires des quantités. Il s'agit de coûts de fonctionnement.

Les coûts d'investissement sont linéaires sur  $(0, +\infty)$ , avec une discontinuité à l'origine. En d'autres termes, un coût d'investissement se compose d'un coût fixe et d'un coût proportionnel à la capacité additionnelle obtenue.

Le problème qui nous intéresse consiste à définir la combinaison (fournisseurs + producteurs + distributeurs) qui permettra de satisfaire la demande au coût total le plus bas. Notons que la demande qui se présente à un distributeur peut varier d'une période élémentaire à la suivante.

### 4.2.2 Notations et formalisation du problème

Soient  $E_F = \{1, 2, \dots, F\}$  les fournisseurs,  $E_P = \{1, 2, \dots, P\}$  les producteurs et  $E_D = \{1, 2, \dots, D\}$  les distributeurs candidats à la chaîne d'approvisionnement. La demande à satisfaire par chaque distributeur est connue sur chaque période élémentaire incluse dans l'étude. Nous désignons par  $d_k^t, k \in$

$\{1, 2, \dots, D\}$  et  $t \in \{1, 2, \dots, T\}$  la demande que doit satisfaire le distributeur  $k$  au cours de la période  $t$ . Nous supposons que l'étude se déroule sur  $T$  périodes élémentaires. Les autres données du problème sont les suivantes :

**Capacités additionnelles (à déterminer)**

$r_{i,j}, i \in E_F, j \in E_P$  est la capacité additionnelle de transport du fournisseur  $i$  vers le producteur  $j$ ,

$s_{j,k}, j \in E_P$  et  $k \in E_D$  est la capacité additionnelle de transport du producteur  $j$  vers le distributeur  $k$ ,

$g_i, i \in E_F$  est la capacité additionnelle du fournisseur  $i$ ,

$h_j, j \in E_P$  est la capacité additionnelle du producteur  $j$ ,

**Capacité existantes (données)**

$R_{i,j}, i \in E_F, j \in E_P$  est la capacité de transport disponible du fournisseur  $i$  vers le producteur  $j$ ,

$S_{j,k}, j \in E_P$  et  $k \in E_D$  est la capacité de transport disponible du producteur  $j$  vers le distributeur  $k$ ,

$G_i, i \in E_F$  est la capacité disponible chez le fournisseur  $i$ .

$H_j, j \in E_P$  est la capacité de production disponible chez le producteur  $j$ .

**Les variables (à déterminer)**

$x_{i,j}^t, i \in E_F, j \in E_P$ , et  $t \in \{1, 2, \dots, T\}$  est la quantité de matière première transportée du fournisseur  $i$  vers le producteur  $j$  au cours de la période  $t$ .

$y_{j,k}^t, j \in E_P, k \in E_D$ , et  $t \in \{1, 2, \dots, T\}$  est la quantité de matière première transportée du producteur  $j$  vers le distributeur  $k$  au cours de la période  $t$ .

**Les coûts de fonctionnement (donnés)**

$p_i, i \in E_F$  est le coût unitaire de la matière première chez le fournisseur  $i$ ,

$m_j, j \in E_P$  est le coût de production unitaire d'un produit fini chez le producteur  $j$ ,

$u_{i,j}, i \in E_F, j \in E_P$  est le coût unitaire de transport du fournisseur  $i$  vers le producteur  $j$ ,

$v_{j,k}$ ,  $j \in E_P$ ,  $k \in E_D$  est le coût unitaire de transport du producteur  $j$  vers le distributeur  $k$ .

### Les coûts d'investissement (donnés)

Coût d'investissement pour une capacité de transport additionnelle  $r_{i,j}$ ,  $i \in E_F$ ,  $j \in E_P$  du fournisseur  $i$  vers le producteur  $j$  :

$$CT_{i,j}^1(r_{i,j}) = \left\{ \begin{array}{ll} A_{i,j} + a_{i,j} \cdot r_{i,j} & \text{si } r_{i,j} > 0 \\ 0 & \text{sinon} \end{array} \right\}$$

Coût d'investissement pour une capacité de transport additionnelle  $s_{j,k}$ ,  $j \in E_P$ ,  $k \in E_D$  du producteur  $j$  vers le fournisseur  $k$  :

$$CT_{j,k}^2(s_{j,k}) = \left\{ \begin{array}{ll} B_{j,k} + b_{j,k} \cdot s_{j,k} & \text{si } s_{j,k} > 0 \\ 0 & \text{sinon} \end{array} \right\}$$

Coût d'investissement pour permettre au fournisseur  $i \in E_F$  d'augmenter sa capacité de  $g_i$  :

$$CT_i^1(g_i) = \left\{ \begin{array}{ll} E_i + e_i \cdot g_i & \text{si } g_i > 0 \\ 0 & \text{sinon} \end{array} \right\}$$

Coût d'investissement pour permettre au producteur  $j \in E_P$  d'augmenter sa capacité de  $h_j$  :

$$CT_j^2(h_j) = \left\{ \begin{array}{ll} F_j + f_j \cdot h_j & \text{si } h_j > 0 \\ 0 & \text{sinon} \end{array} \right\}$$

### Paramètres financiers (donnés)

Nous ferons encore intervenir deux paramètres financiers :

$\alpha$ , qui est le taux d'actualisation. Ce taux peut être interprété comme l'intérêt servi pour une unité monétaire durant une période,

$\beta$ , qui est le taux de dépréciation des ressources par période.

Avec ces notations, le coût total s'obtient en additionnant les coûts suivants :

1. Coût d'achat de la matière première et le coût de son transport vers les fournisseurs au cours des  $T$  périodes :

$$K_1(p_i, u_{i,j}) = \sum_{t=1}^T \sum_{i \in E_F} \sum_{j \in E_P} C_t(p_i + u_{i,j}) x_{i,j}^t$$

où

$$C_t = (1 + \alpha)^{T-t+1}$$

$C_t$  est le facteur par lequel il faut multiplier un engagement financier effectué au début de la période  $t$  pour connaître ce que cette somme serait devenue à l'horizon du problème si elle avait été placée au taux  $\alpha$  par période.

2. Coûts de production et de transport du producteur vers les distributeurs au cours des  $T$  périodes :

$$K_2(m_j, v_{j,r}) = \sum_{t=1}^T \sum_{j \in E_P} \sum_{k \in E_D} C_t(m_j + v_{j,r}) y_{j,k}^t$$

3. Coût d'investissement pour le transport entre fournisseurs et producteurs :

$$\begin{aligned} I_1(r_{i,j}) &= \sum_{i \in E_F} \sum_{j \in E_P} CT_{i,j}^1(r_{i,j})(C_1 - D_1) \\ &= \sum_{i \in E_F} \sum_{j \in E_P} (C_1 - D_1)(a_{i,j} \cdot r_{i,j} + A_{i,j} \cdot U_{i,j}) \end{aligned}$$

Dans cette formulation :

$C_1 = (1 + \alpha)^T$  :  $C_t$  a été défini plus haut.

$D_1 = (1 - \beta)^T$  :  $D_1$  le facteur par lequel il faut multiplier la valeur des ressources acquises au début de la première période pour connaître leur valeur résiduelle à la fin de la période  $T$ .

Finalement,  $C_1 - D_1$  est le facteur qui, multiplié par l'investissement effectué au début de la première période, nous donne son coût à la fin de la période  $T$ .

Enfin les  $U_{i,j}$  sont des variables binaires qui, du fait des contraintes (4.8) et la forme du coût, seront telles que :

$$U_{i,j} = \left\{ \begin{array}{ll} 1 & \text{si } r_{i,j} > 0 \\ 0 & \text{si } r_{i,j} = 0 \end{array} \right\}$$

4. Coût d'investissement pour le transport entre producteurs et distributeurs :

$$\begin{aligned} I_2(s_j, k) &= \sum_{j \in E_P} \sum_{k \in E_D} CT_{j,k}^2(s_j, k) \\ &= \sum_{j \in E_P} \sum_{k \in E_D} (C_1 - D_1)(b_{j,k} \cdot s_{j,k} + B_{j,k} \cdot V_{j,k}) \end{aligned}$$

où les  $V_{j,k}$  sont des variables binaires qui prennent leur valeur dans  $\{0, 1\}$ . Du fait des contraintes (4.9) que nous introduisons plus loin et de la forme du coût à minimiser, ces variables binaires seront telles que :

$$V_{j,k} = \left\{ \begin{array}{ll} 1 & \text{si } v_{j,k} > 0 \\ 0 & \text{si } v_{j,k} = 0 \end{array} \right\}$$

5. Coût d'investissement pour permettre aux fournisseurs d'augmenter leur capacité :

$$\begin{aligned} J_1(g_i) &= \sum_{i \in E_F} CC_i^1(g_i) \\ &= \sum_{i \in E_F} (C_1 - D_1)(e_i \cdot g_i + E_i \cdot W_i) \end{aligned}$$

Les  $W_i$  sont des variables qui prennent leurs valeurs dans  $\{0, 1\}$ . Du fait des contraintes (4.10) que nous introduisons plus loin et du coût à minimiser, ces variables binaires seront, dans la solution optimale, telles que :

$$W_i = \left\{ \begin{array}{ll} 1 & \text{si } g_i > 0 \\ 0 & \text{si } g_i = 0 \end{array} \right\}$$

6. Coût d'investissement pour permettre aux producteurs d'augmenter leur capacité :

$$J_2(h_j) = \sum_{j \in E_P} CC_j^2(h_j)$$

$$= \sum_{j \in E_P} (C_1 - D_1)(h_j \cdot h_j + F_j \cdot X_j)$$

Les  $X_j$  sont des variables binaires qui prennent leurs valeurs dans  $\{0, 1\}$ . Ces variables binaires obéiront, dans la solution optimale, compte tenu des contraintes (4.11) et de la forme du coût total à optimiser :

$$X_j = \left\{ \begin{array}{ll} 1 & \text{si } h_j > 0 \\ 0 & \text{si } h_j = 0 \end{array} \right\}$$

Finalement, le coût à minimiser est :

$$K_1(p_i, u_{i,j}) + K_2(m_j, v_{j,k}) + I_1(r_{i,j}) + I_2(s_{j,k}) + J_1(g_i) + J_2(h_j) \quad (4.1)$$

pour les valeurs des indices  $i, j$  et  $k$  dans  $E_F, E_P$  et  $E_D$  respectivement.

Les contraintes qui définissent le domaine dans lequel s'opère la minimisation sont les suivantes :

$$x_{i,j}^t \leq R_{i,j} + r_{i,j}, \quad (4.2)$$

$$\text{pour } t \in \{1, 2, \dots, T\}, i \in E_F \text{ et } j \in E_P$$

La contrainte (4.2) indique que le flux transporté du fournisseur  $i$  vers le producteur  $j$  ne doit jamais dépasser la capacité de transport qui s'exprime comme la somme de la capacité disponible et de la capacité acquise en supplément.

$$y_{j,k}^t \leq S_{j,k} + s_{j,k} \quad (4.3)$$

$$t \in \{1, 2, \dots, T\}, j \in E_P \text{ et } k \in E_D$$

La contrainte (4.3) indique que le flux transporté du producteur  $j$  vers le distributeur  $k$  est toujours inférieur ou égal à la somme de la capacité disponible et de la capacité acquise.

$$\sum_{j \in E_P} x_{i,j}^t \leq G_i + g_i, t \in \{1, 2, \dots, T\}, i \in E_F \quad (4.4)$$

La contrainte (4.4) indique que la quantité délivrée par le fournisseur  $i$  durant une période est inférieure ou égale à la capacité disponible plus la capacité acquise.

$$\sum_{k \in E_D} y_{j,k}^t \leq H_j + h_j \quad (4.5)$$

$$t \in \{1, 2, \dots, T\} \text{ et } j \in E_P$$

La contrainte (4.5) indique que la quantité produite par le producteur  $j$  durant une période est inférieure ou égale à la capacité totale de ce producteur (capacité disponible plus capacité acquise).

$$\sum_{i \in E_F} x_{i,j}^t = \sum_{k \in E_D} y_{j,k}^t \quad (4.6)$$

$$t \in \{1, 2, \dots, T\}, j \in E_P$$

Les contraintes (4.6) indiquent la conservation des flux : la quantité totale qui arrive chez un producteur  $j$  durant une période est égale à la quantité totale qui quitte ce producteur durant la même période.

$$\sum_{j \in E_P} y_{j,k}^t = d_k^t \quad (4.7)$$

$$t \in \{1, 2, \dots, T\} \text{ et } k \in E_D$$

Les contraintes (4.7) indiquent que la quantité de produit qui arrive chez un fournisseur durant une période est égale à la demande que doit satisfaire ce fournisseur durant la même période.

Enfin, nous introduisons les contraintes " techniques " suivantes :

$$r_{i,j} \leq GG_{i,j} \cdot U_{i,j}, i \in E_F, j \in E_P \quad (4.8)$$

$$s_{j,k} \leq GG_{j,k} \cdot V_{j,k}, j \in E_P, k \in E_D \quad (4.9)$$

$$g_i \leq GG_i \cdot W_i, i \in E_F \quad (4.10)$$

$$h_j \leq GG_j \cdot X_j, j \in E_P \quad (4.11)$$

où les  $GG_{i,j}$ ,  $GG_{j,k}$ ,  $GG_i$  et  $GG_j$  sont des nombres aussi grand que l'on voudra. L'objectif de ces contraintes est de forcer le membre de gauche à être nul lorsque la variable binaire du membre de droite est nulle.

Enfin :

$$U_{i,j} \in \{0, 1\}, i \in E_F, j \in E_P \quad (4.12)$$

$$V_{j,k} \in \{0, 1\}, j \in E_P, k \in E_D \quad (4.13)$$

$$W_i \in \{0, 1\}, i \in E_F \quad (4.14)$$

$$X_j \in \{0, 1\}, j \in E_P \quad (4.15)$$

Finalement, le problème consiste à minimiser le coût (4.1) sous les contraintes (4.2) à (4.15). Ce problème est clairement un problème de programmation linéaire mixte. Nous désignons ce problème par  $P_1$ . Notons cependant que le nombre de variables et de contraintes est important. Nous en faisons le compte dans le paragraphe suivant.

### 4.2.3 Taille du problème

Nous désignons respectivement par  $F$ ,  $P$  et  $D$  les tailles des ensembles  $E_F$ ,  $E_P$ , et  $E_D$ , c'est-à-dire le nombre de fournisseurs, de producteurs et de distributeurs.

#### a. *Nombre total de variables*

Nombre de variables  $r_{i,j}$  :  $F \cdot P$

Nombre de variables  $s_{j,k}$  :  $P.D$

Nombre de variables  $g_i$  :  $F$

Nombre de variables  $h_j$  :  $P$

Nombre de variables  $x_{i,j}^t$  :  $F.P.$

Nombre de variables  $y_{j,k}^t$  :  $P.D.T$

Soit un total de variables réelles égale à :

$$N_R = P[1 + D(T + 1)] + F[1 + P(T + 1)]$$

A ces variables réelles s'ajoutent des variables binaires :

$F.P$  variables  $U_{i,j}$

$P.D$  variables  $V_{j,k}$

$F$  variables  $W_i$

$P$  variables  $X_j$

Soit un totale de variables binaires égal à :

$$N_B = F(P + 1) + P(D + 1)$$

#### b. *Nombre total de contraintes*

Nombre de contraintes (4.2) :  $F.P.T$

Nombre de contraintes (4.3) :  $P.D.T$

Nombre de contraintes (4.4) :  $F.T$

Nombre de contraintes (4.5) :  $P.T$

Nombre de contraintes (4.6) :  $P.T$

Nombre de contraintes (4.7) :  $D.T$

Nombre de contraintes (4.8) :  $F.P$

Nombre de contraintes (4.9) :  $P.D$

Nombre de contraintes (4.10) :  $F$

Nombre de contraintes (4.11) :  $P$

Nombre de contraintes (4.12) :  $F.P$

Nombre de contraintes (4.13) :  $P.D$

Nombre de contraintes (4.14) :  $F$

Nombre de contraintes (4.15) :  $P$

Soit un total qui peut s'écrire :

$$N_C = F(2 + 2.P + T + P.T) + P(2 + 2.D + 2.T.D.T) + DT$$

Par exemple, un problème de taille moyenne tel que  $F = P = D = T = 10$  aurait :

2440 variables (sans compter les variables d'écart à introduire pour transformer les contraintes d'inégalité en contraintes d'égalité).

2840 contraintes.

Le paragraphe suivant est consacré aux résultats qui seront utilisés pour résoudre le problème.

### 4.3 Propriétés du problème

$P_1$  est le problème que nous venons d'introduire. Nous désignons par  $P_2$  le problème dans lequel les contraintes (4.12) à (4.15) sont relaxées, c'est-à-dire dans lequel les variables  $U_{i,j}$ ,  $V_{j,k}$ ,  $W_i$  et  $X_j$  peuvent prendre leurs valeurs sur  $[0, 1]$ .

#### Résultat 1

Il existe au moins une solution optimale du problème  $P_2$  telle que les contraintes (4.8) à (4.11) sont saturées.

#### Démonstration

Soit  $S_1$  une solution optimale dans laquelle, par exemple,

$$r_{i_1, j_1}^1 < GG_{i_1, j_1} \cdot U_{i_1, j_1}^1 \text{ pour } (i_1, j_1) \in E_F \times E_P$$

L'indice supérieur indique l'appartenance à  $S_1$ . Soit  $S_2$  une solution identique à  $S_1$  mais dans laquelle  $U_{i_1, j_1}^1$  est remplacé par  $U_{i_1, j_1}^2 = r_{i_1, j_1}^1 / GG_{i_1, j_1}$ .

$S_2$  est admissible car toutes les contraintes sont satisfaites,  $y$  compris :

$$r_{i_1, j_1}^2 = r_{i_1, j_1}^1 = GG_{i_1, j_1} \cdot U_{i_1, j_1}^2.$$

En outre, si  $C(\cdot)$  désigne le coût total :

$$C(S_1) - C(S_2) = (C_1 - D_1) \{ a_{i_1, j_1} \cdot r_{i_1, j_1}^1 + A_{i_1, j_1} \cdot U_{i_1, j_1}^1 - a_{i_1, j_1} \cdot r_{i_1, j_1}^2 - A_{i_1, j_1} \cdot U_{i_1, j_1}^2 \}$$

Mais  $r_{i_1, j_1}^1 = r_{i_1, j_1}^2$ . Donc :

$$\begin{aligned} C(S_1) - C(S_2) &= (C_1 - D_1) A_{i_1, j_1} (U_{i_1, j_1}^1 - U_{i_1, j_1}^2) \\ &= (C_1 - D_1) A_{i_1, j_1} (U_{i_1, j_1}^1) - r_{i_1, j_1}^1 / GG_{i_1, j_1} \end{aligned}$$

Mais  $U_{i_1, j_1}^1 > r_{i_1, j_1}^1 / GG_{i_1, j_1}$  et  $(C_1 - D_1) > 0$

Finalement :

$$C(S_1) - C(S_2) > 0$$

La solution  $S_2$  est donc meilleure que  $S_1$  et comporte une contraintes saturée de plus que  $S_1$ .

La même procédure peut être reproduite pour toute contrainte de (4.8) à (4.11) qui serait une inégalité stricte pour obtenir une solution optimale avec contrainte d'égalité.

Ceci achève la démonstration. ■

Ce résultat conduit au corollaire suivant.

### Corollaire 1

Sachant qu'il existe une solution optimale dans laquelle les contraintes (4.8) à (4.11) sont saturées, nous pouvons réduire la taille du problème  $P_2$  en remplaçant :

$U_{i,j}$  par  $r_{i,j}/GG_{i,j}$  pour tout  $i \in E_F$  et  $j \in E_P$

$V_{j,k}$  par  $s_{j,k}/GG_{j,k}$  pour tout  $j \in E_P$  et  $k \in E_D$

$W_i$  par  $g_i/GG_i$  pour tout  $i \in E_F$

$X_j$  par  $h_j/GG_j$  pour tout  $j \in E_P$ .

Nous insistons sur le fait que cette simplification concerne le problème  $P_2$  et non le problème  $P_1$ .

Dans la suite, nous considérons une suite de problèmes dans laquelle le problème de rang  $m$  est construit à partir de la solution du problème  $m - 1$ . Dans cette suite, nous notons  $P_2^m$  le problème de rang  $m$ , le problème  $P_2$  est le problème de rang 0, et nous le notons  $P_2^0$ . Les variables solutions du problème de rang  $m$  se reconnaissent par l'indice supérieur  $m$ .

### Résultat 2

Pour  $m = 1, 2, \dots$ , le problème  $P_2^m$  est obtenu en posant :

$$GG_{i,j}^m = r_{i,j}^{m-1}, \forall i \in E_f \text{ et } j \in E_P \text{ tels que } r_{i,j}^{m-1} > 0,$$

$$GG_{j,k}^m = s_{j,k}^{m-1}, \forall j \in E_P \text{ et } k \in E_D \text{ tels que } s_{j,k}^{m-1} > 0,$$

$$GG_i^m = g_i^{m-1}, \forall i \in E_F \text{ tels que } g_i^{m-1} > 0,$$

$$GG_j^m = h_j^{m-1}, \forall j \in E_P \text{ tels que } h_j^{m-1} > 0 \text{ et :}$$

$$G_{i,j}^m = G_{i,j}^{m-1}, \forall i \in E_F \text{ et } j \in E_P \text{ tels que } r_{i,j}^{m-1} = 0,$$

$$G_{i,j}^m = G_{i,j}^{m-1}, \forall j \in E_P \text{ et } k \in E_D \text{ tels que } s_{j,k}^{m-1} = 0,$$

$$GG_i^m = GG_i^{m-1}, \forall i \in E_F \text{ tels que } g_i^{m-1} = 0,$$

$$GG_j^m = GG_j^{m-1}, \forall j \in E_P \text{ tels que } h_j^{m-1} = 0.$$

Dans ces conditions :

1. Nous pouvons obtenir une solution admissible du problème  $P_1$  à partir de la solution optimale d'un problème  $P_2^m$  en posant :

$$U_{i,j} = 1 \text{ pour tout } (i, j) \text{ tel que } r_{i,j}^m > 0,$$

$$V_{j,k} = 1 \text{ pour tout } (j, k) \text{ tel que } s_{j,k}^m > 0,$$

$$W_i = 1 \text{ pour tout } i \text{ tel que } g_i^m > 0,$$

$$X_j = 1 \text{ pour tout } j \text{ tel que } h_j^m > 0,$$

et en résolvant ce nouveau problème que nous appelons  $P_3^m$ .

2. Désignons par  $S_2^m$  la solution optimale de  $P_2^m$ . Il existe un entier  $m^*$  tel que  $S_2^{m^*}$  est une solution admissible de  $P_1$ .

### Démonstration

a. *Obtention d'une solution admissible à partir de la solution optimale de  $P_2^n$*   
 Soit  $S_2^m$  la solution optimale de  $P_2^m$ . Cette solution vérifie les contraintes (4.2) à (4.7) qui sont aussi les contraintes qui s'appliquent au problème  $P_1$ . De plus, la solution  $S_2^m$  sature les contraintes (4.8) à (4.11) du fait de la relaxation des domaines de définition des variables binaires et de la forme du coût à minimiser. Ces contraintes sont toujours satisfaites si ces variables sont fixées à la valeur 1 lorsque la capacité additionnelle correspondante est strictement positive. De plus, les contraintes (4.12) à (4.15) sont vérifiées de fait, car les variables binaires sont soit nulles dans la solution  $S_2^m$ , soit fixées à 1 lorsque elles sont positives.

Par conséquent, la solution  $S_2$  modifiée comme indiqué pour ce qui concerne les variables binaires, est admissible pour  $P_1$ . En cherchant la solution optimale de  $P_3^m$  qui est le problème dans lequel les variables binaires sont fixées comme indiqué ci-dessus, on obtient donc une solution admissible de  $P_1$  meilleure que la solution  $S_2$  modifiée.

Par conséquent, la solution admissible de  $P_1$  peut s'obtenir :

1. en calculant  $S_2^0, S_2^1, S_2^2, \dots, S_2^n$ , solutions de  $P_2^0, P_2^1, \dots, P_2^m$  respectivement.
2. en fixant à 1 toutes les variables "binaires" strictement positives : Les variables binaires étant fixées, nous sommes en face du problème  $P_3^m$ .
3. en recherchant de la solution optimale de  $P_3^m$  : c'est une solution admissible de  $P_1$ .

La procédure s'applique quelque soit  $m$ .

b. *Existence de  $m^*$  tel que  $S_2^{m^*}$  est admissible pour  $P_1$*

Considérons, par exemple, les variables  $r_{i,j}$  qui sont les capacités de transport additionnelles des fournisseurs  $i$  vers les producteurs  $j$ , et les variables binaires correspondantes  $U_{i,j}$ .

Suivons l'évolution de  $GG_{i,j}^m$  pour  $m = 1, 2, \dots$  chaque fois de  $r_{i,j}^{m-1} > 0$ ,

$G_{i,j}^m$  prend la valeur  $r_{i,j}^{m-1}$ . Par conséquent, et en se souvenant que les variables "binaires" saturent les contraintes, nous avons à l'optimum :

- Soit  $GG_{i,j}^m = r_{i,j}^{m_1}$ , où  $m_1$  est le plus grand entier inférieur ou égal à  $m - 1$  tel que  $r_{i,j}^{m_1} > 0$ ,
- Soit  $GG_{i,j}^m = GG_{i,j}$ , valeur initiale. C'est le cas où les valeurs  $r_{i,j}^k$  des variables  $r_{i,j}$  dans les solution  $S_2^k$ ,  $k = 1, 2, \dots, m$ , sont toutes nulles.

Les contraintes correspondent aux variables binaires, et plus particulièrement la contrainte (4.8) dans notre cas, étant saturées, nous avons :

$$U_{i,j}^k = r_{i,j}^k / GG_{i,j}^k \text{ pour } k = 1, 2, \dots \quad (4.16)$$

Soit, en tenant compte des remarques ci-dessus :

$$U_{i,j}^k = \left\{ \begin{array}{ll} r_{i,j}^k / r_{i,j}^{k_1} & \text{si } A \text{ (voir ci-dessous)} \\ r_{i,j}^k / GG_{i,j}^k & \text{si } B \text{ (voir ci-dessous)} \end{array} \right\}$$

A. Au moins une solution  $S_2^{k_1}$ ,  $k_1 < k$  est telle que  $r_{i,j}^{k_1} > 0$ , et  $k_1$  est le plus grand indice inférieur à  $k$  qui est dans ce cas.

B. Pour tout  $S_2^v$ ,  $v = 1, 2, \dots, k - 1$ , nous avons  $r_{i,j}^v = 0$ .

Nous allons considérer séparément les deux cas ci-dessus :

*Cas 1 :  $m_1$  existe*

Dans ce cas, compte tenu de (4.16) :

$$\prod_{k=1}^m U_{i,j}^k = \prod_{k=1}^m \left( \frac{r_{i,j}^k}{GG_{i,j}^k} \right) = \frac{r_{i,j}^m}{r_{i,j}^{m_1}}$$

Et :

$$r_{i,j}^m = \prod_{k=1}^m U_{i,j}^k \cdot r_{i,j}^{m_1} \quad (4.17)$$

Mais, compte-tenu des contraintes sur les variables binaires, et plus précisément de la contrainte (4.12) dans notre cas,

$$\prod_{k=1}^m U_{i,j}^k \leq 1$$

et donc :

$$r_{i,j}^m \leq r_{i,j}^{m_1}$$

La suite des valeurs prises par  $r_{i,j}$  dans les solutions  $S_2^1, S_2^2, \dots, S_2^m$  est décroissante si l'on considère uniquement les valeurs positives. Cette suite est constante si les valeurs prises par  $U_{i,j}$  dans ces solutions sont toutes égales à 1.

*Cas 2 :  $m_1$  n'existe pas*

Dans ce cas,  $r_{i,j}^k = 0$  pour  $k = 0, 1, 2, \dots, m - 1$ .

L'ensemble des  $(i, j)$  comporte  $F \times P$  éléments. Pour la solution  $S_2^m$ , soit :

$E_1^m$  l'ensemble des couples  $(i, j)$  tels qu'il existe  $m_1 < m$  pour lequel  $r_{i,j}^{m_1} > 0$

$E_0^m$  l'ensemble des couples  $(i, j)$  tels que  $r_{i,j}^k = 0$  pour  $k = 1, 2, \dots, m - 1$ .

Nous savons par ailleurs que pour toute solution  $S_2^k$ , nous avons :

$$\sum_{i \in E_F} r_{i,j}^k = \text{Constante quelque soit } j \in E_P$$

Si donc certains éléments  $r_{i,j}^m, (i, j) \in E_0^m$ , décroissent strictement lorsque  $m$  croît, ils ne peuvent être compensés que par des éléments de  $E_0^m$  qui deviennent strictement positifs.

Finalement, nous arrivons à une solution  $S_2^{m^*}$  dans laquelle :  $r_{i,j}^{m^*} = r_{i,j}^{m^*-1}$  pour  $(i, j) \in E_1^{m^*}$

Cette situation implique  $U_{i,j}^{m^*} = 1$

$r_{i,j}^{m^*} = 0$  pour  $(i, j) \notin E_0^{m^*}$

Cette situation entraîne  $U_{i,j}^{m^*} = 0$ . Ceci complète la démonstration pour les variables  $r_{i,j}$ . La démonstration peut être reprise à l'identique pour les variables  $s_{j,k}, g_i$  et  $h_j$ .

Compte tenu du fait que ces variables concernent des éléments différents du coût, il suffit de conserver les valeurs obtenues à la limite  $m^*$  de chacune des variables pour pouvoir affirmer que la seconde partie du résultat est vérifiée.

Ceci achève la démonstration. ■

**Important :**

Le résultat que nous venons d'établir permet d'aboutir à une solution admissible de  $P_1$  de deux manières :

1. En calculant une solution  $S_2^m$  de  $P_2^m$ , puis en calculent la solution de  $P_3^m$  dérivé de  $P_2^m$  en "forçant" les variables binaires.
2. En calculant la suite de solutions  $S_2^1, S_2^2, \dots, S_2^m, \dots$  jusqu'à aboutir à une solution admissible de  $P_1$ . Nous savons que nous y parviendrons, mais il n'est pas possible de prévoir le nombre de problèmes à résoudre pour y parvenir.

Dans les deux cas, il suffit de résoudre des problèmes de programmation linéaire en nombres réels.

Bien entendu, aucun résultat ne permet d'évaluer la qualité de la solution admissible obtenue.

## 4.4 Calcul d'une solution admissible proche de l'optimum

L'algorithme que nous proposons est dérivé du résultat 2 ci-dessus. Il consiste à calculer les solutions  $S_2^0, S_2^1, S_2^2, \dots, S_2^m, \dots$  des problèmes  $P_2^0, P_2^1, \dots, P_2^m, \dots$  respectivement, puis de calculer les solutions de  $P_3^0, P_3^1, \dots, P_3^m, \dots$  qui sont des solutions admissibles de  $P_1$  et, finalement, de conserver la solution admissible qui conduit au coût le plus bas. La figure 4.1 résume le processus. Le nombre  $Z$  d'itérations est choisi par l'utilisateur car nous ne savons pas évaluer le nombre d'itérations  $m^*$  nécessaires pour que la solution  $S_2^{m^*}$  soit admissible pour  $P_1$ .

Nous conservons la solution  $S_3^k$  qui vérifie :

$$C(S_3^k) = \text{Min}_{m \in \{0,1,2,\dots,Z\}} C(S_3^m)$$

Le paragraphe suivant est consacré à une application numérique qui nous sert à évaluer l'algorithme.

## 4.5 Application numérique

Nous présentons 10 exemples générés au hasard. Comme nous souhaitons calculer la solution optimale pour chacun de ces exemples à l'aide d'une approche par "Séparation et Evaluation" (Branch and Bound) nous nous sommes restreints à des cas comportant 5 fournisseurs, 5 producteurs, 5 distributeurs et 5 périodes élémentaires. En revenant au paragraphe 4.2.3, nous voyons que ce problème comporte :

310 variables réelles.

60 variables binaires.

Notons au passage que la méthode par "Séparation et Evaluation" porte sur les variables binaires.

A chaque noeud, la borne inférieure s'obtient en relaxant les variables binaires non encore fixées, et la borne supérieure s'obtient en appliquant l'heuris-

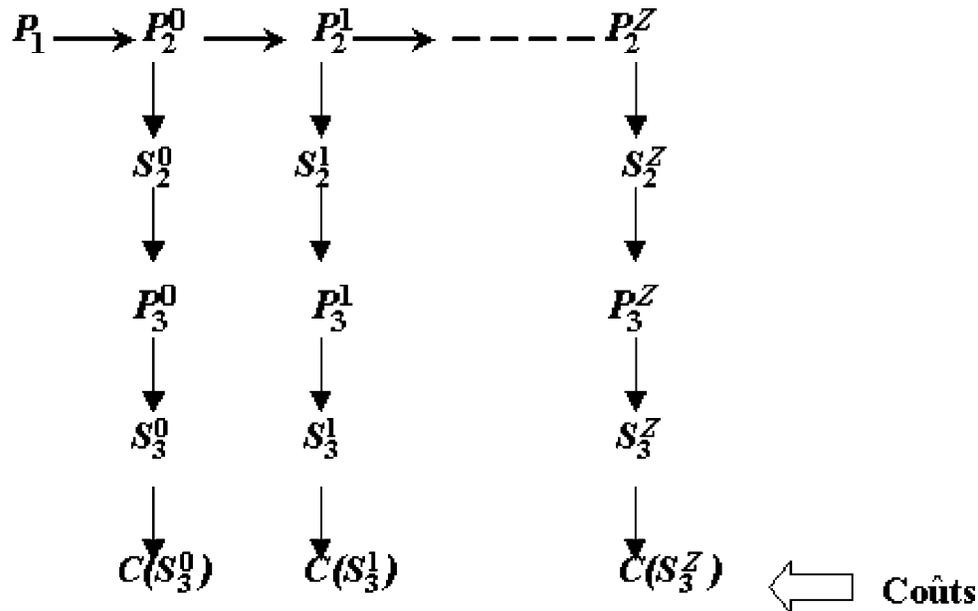


FIG. 4.1 – Schématisation de l'algorithme

TAB. 4.1 – Solutions heuristiques et optimales

#	Solution heuristique (coût)	Solution optimale (coût)	Ecart en %
1	276518	275781	0.267
2	257359	257359	0.0
3	324934	322973	0.607
4	292007	292007	0.0
5	354365	354116	0.070
6	322126	325881	1.910
7	334013	333499	0.154
8	321299	319871	0.466
9	291253	290190	0.366
10	310869	310869	0.0

tique du paragraphe précédent. Nous ne nous attarderons pas sur l'approche, qui est classique. Les exemples sont générés en choisissant de manière aléatoire :

- les capacités disponibles des fournisseurs sur  $[0,500]$ ,
- les capacités transport disponibles chez les fournisseurs sur  $[0,500]$ ,
- les coûts unitaires chez les fournisseurs sur  $[5,10]$ ,
- les capacités de production des producteurs sur  $[0,500]$ ,
- les capacités de transport disponibles chez les producteur sur  $[0,500]$ ,
- les coûts de production unitaires des producteurs sur  $[0,35]$ ,
- les coûts fixes d'investissement sur  $[1000, 5000]$ ,
- les coûts unitaires d'investissement sur  $[1,10]$ ,
- les demandes auxquelles doivent satisfaire les distributeurs sur  $[100,500]$ .

Les résultats sont présentés dans le tableau 4.1.

## 4.6 Conclusion

L'algorithme heuristique que nous avons présenté a l'avantage de n'exiger que la résolution de problèmes de programmation linéaire en nombres réels. Les

---

résultats, testés sur des problèmes de taille raisonnable, sont satisfaisants. Enfin, des problèmes de taille réelle, de l'ordre de 20 fournisseurs, 20 producteurs, 30 distributeurs et 10 périodes élémentaires sont à la portée de cet algorithme.



# Chapitre 5

## Conception d'une chaîne d'approvisionnement multi-niveaux

### 5.1 Introduction

Dans le modèle que nous présentons maintenant, la chaîne d'approvisionnement comporte un nombre quelconque de niveaux. Nous pouvons considérer, par exemple, plusieurs niveaux chez les fournisseurs, plusieurs étapes de fabrication distinctes et plusieurs étapes de distribution, dues à la présence de grossistes. Chaque niveau comporte un ensemble de candidats potentiels. Les demandes ont été évaluées par une étude de marché sur un horizon donné.

Les candidats présents à chaque niveau de la chaîne d'approvisionnement ont les mêmes compétences, mais différent par la capacité dont ils disposent et par leur coût de production.

L'objectif est de sélectionner un partenaire à chaque niveau de la chaîne d'approvisionnement. Ce choix devra conduire à satisfaire les demandes sur l'horizon donné (c'est-à-dire sur l'ensemble des périodes élémentaires sur lesquelles des prévisions de consommation ont été faites) sans retard et, bien entendu, au coût le plus bas.

Notons encore que la capacité offerte par les différents partenaires poten-

tiels peut varier d'une période élémentaire à la suivante, mais que la solution cherchée doit être unique sur tout l'horizon.

Nous introduisons le modèle dans le paragraphe 5.2 et donnons sa formulation mathématique. Une approche dénommée "relaxation des contraintes sur les chemins" est présentée dans le paragraphe 5.3. Le paragraphe 5.4 est consacré à des applications numériques. Le dernier paragraphe est la conclusion.

## 5.2 Présentation du modèle

### 5.2.1 Description

Le modèle que nous considérons est composé de  $n$  étapes notées  $E_1, E_2, \dots, E_n$ . Chaque partenaire potentiel a un coût de stockage et un coût d'activité unitaire. Le terme "activité" recouvre une activité commerciale chez les fournisseurs et les distributeurs, et une activité de production chez les producteurs. De plus, chaque couple de partenaires potentiels situés dans deux étapes successives a un coût de transport et un coût fixe qui représente les sommes à engager, s'ils sont tous deux retenus, pour permettre leur collaboration (gestion de la collaboration, mise en place de réseaux de communication par exemple). Tous ces coûts peuvent varier d'une période élémentaire à la suivante.

Une opération est exécutée à chaque étape. La figure (5.1) représente le modèle.

La figure (5.1) nous montre que le modèle peut être représenté par un graphe  $G(N, A)$ , où  $N$  est l'ensemble des noeuds et  $A$  est l'ensemble des arcs. Un noeud représente la localisation d'un partenaire potentiel et un arc représente la connexion entre deux partenaires potentiels successifs. Notons que deux partenaires potentiels successifs peuvent ne pas être reliés par un arc : cela signifie que, pour une raison ou pour une autre, ils ne peuvent pas collaborer.

Le problème consiste donc à choisir le "meilleur" chemin joignant  $E_0$  à  $E_{n+1}$ .

Rappelons les principales caractéristiques du système :

- Il s'agit de satisfaire une opportunité du marché.
- La demande est connue sur l'horizon, mais peut varier d'une période élémentaire à la suivante.
- La rupture de stock est interdite à la dernière étape. En d'autres termes, le client final doit être satisfait tout au long de l'horizon donné.
- La capacité disponible à chaque noeud est connue sur l'horizon de l'étude, mais peut varier d'une période élémentaire à l'autre.
- La somme du temps opératoire à une étape et du temps de transport de cette étape à la suivante est égale à une période élémentaire.
- Il n'y a pas de limite de capacité sur le transport d'une étape à la suivante.

## 5.2.2 Formulation mathématique

### Les notations

La solution du problème qui nous intéresse est la solution d'un problème de programmation linéaire mixte.

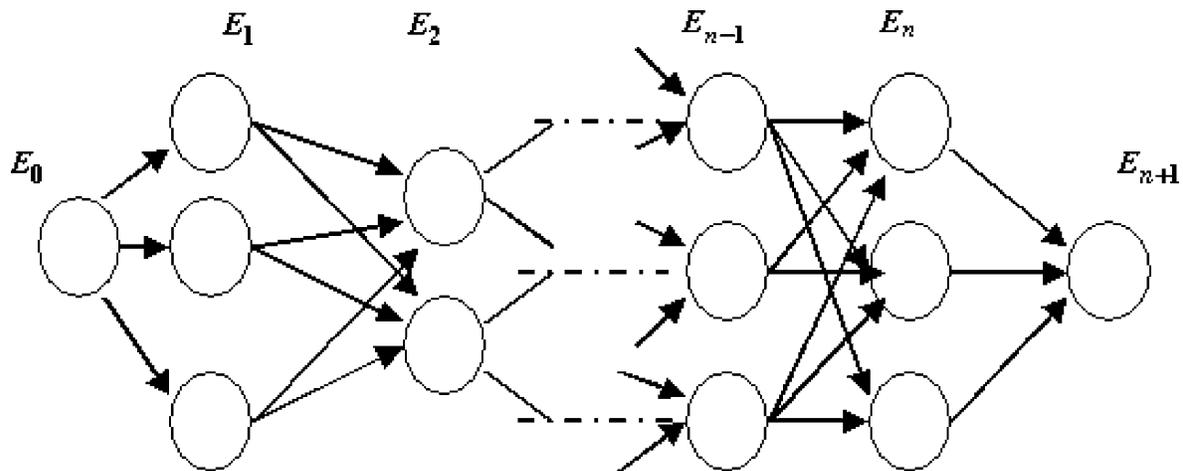


FIG. 5.1 – Le modèle

Les paramètres suivants sont utilisés. Ils prennent comme valeurs les données du problème.

- $N = \{1, 2, \dots, n\}$  est l'ensemble des étapes à parcourir pour réaliser le projet (fabrication ou service).
- $q_i, i \in N$  est le nombre de partenaires potentiels à l'étape  $i$ .
- $F_{u,i,v,i+1}$  est le coût fixe d'établissement d'une connection entre le noeud  $u$  de l'étape  $i$  et le noeud  $v$  de l'étape  $i + 1$ , avec  $i = 1, 2, \dots, n - 1$ ,  $u \in \{1, 2, \dots, q_i\}$  et  $v \in \{1, 2, \dots, q_{i+1}\}$ .
- $C_{u,i,v,i+1,t}$  est le coût de transport unitaire entre le noeud  $u$  de l'étape  $i$  et le noeud  $v$  de l'étape  $i + 1$  pour la période  $t$ . Comme ci-dessus,  $i \in \{1, 2, \dots, n - 1\}$ ,  $u \in \{1, 2, \dots, q_i\}$ ,  $v \in \{1, 2, \dots, q_{i+1}\}$  et  $t \in \{i, i + 1, \dots, T - n + i\}$ .

$T$  est le nombre de périodes sur lesquelles se fait l'optimisation, c'est-à-dire l'horizon du problème.

Nous remarquons que les périodes qui nous intéressent pour la connaissance de la capacité disponible vont de  $i$  à  $T - n + i$  pour l'étape  $i$ . L'explication est la suivante :

- Comme nous sommes au début de la fabrication, la  $i$ -ème étape n'est concernée pour la première fois qu'au cours de la  $i$ -ème période, car une période est nécessaire pour effectuer l'opération à une étape et transporter les produits à l'étape suivante.

- Comme nous sommes intéressés à fournir la demande jusqu'à la période  $T$ , la  $i$ -ème étape n'est concernée que jusqu'à la  $(T - n + i)$ -ème période, période pour laquelle la production de la  $i$ -ème étape aboutira en fin de chaîne au cours de la période  $T$ . Bien entendu, on admet que  $T \geq n$ .

- $H_{u,i,t}^f$  est le coût de stockage à la sortie du noeud  $u$  durant la période  $t$ , avec  $i \in N$ ,  $u \in \{1, 2, \dots, q_i\}$  et  $t \in \{i, \dots, T - n + i\}$ .
- $H_{u,i,t}^r$  est le coût de stockage à l'entrée du noeud  $u$  durant la période  $t \in \{i, \dots, T - n + i\}$ .

Nous voyons donc que chaque noeud, c'est-à-dire chaque partenaire potentiel, a un stock d'entrée et un stock de sortie qui peuvent être utilisés l'un et l'autre.

$U_{u,i,t}$  est le coût unitaire de réalisation de l'opération de l'étape  $i \in N$  par le partenaire (i.e. le noeud)  $u \in \{1, 2, \dots, q_i\}$  durant la période  $t \in \{i, \dots, T-n+i\}$ .

- $d_t$  est la demande qu'il est prévu de satisfaire durant la période  $t \in \{n, \dots, T\}$ .
- $M$  est une valeur numérique supérieure à la somme des demandes qu'il est prévu de satisfaire sur l'horizon  $T$ . Nous verrons que cette valeur n'a pas de signification particulière. Elle n'a pas d'autre but que de nous permettre d'introduire la formulation mathématique.

- $K_{u,i,t}$  est la capacité disponible chez le partenaire  $u \in \{1, 2, \dots, q_i\}$  de l'étape  $i \in N$  durant la période  $t \in \{i, 2, \dots, T-n+i\}$ . Cette capacité s'exprime en nombre d'unités.

Nous introduisons maintenant les variables dont nous voulons chercher la valeur optimale.

- $h_{u,i,t}^f$  est le nombre d'unités en attente à la sortie du noeud  $u \in \{1, 2, \dots, q_i\}$  de l'étape  $i \in N$  durant la période  $t \in \{i, \dots, T-n+i\}$ .

- $h_{u,i,t}^r$  est le nombre d'unités en attente à l'entrée du noeud  $u \in \{1, 2, \dots, q_i\}$  de l'étape  $i \in N$  durant la période  $t \in \{i, \dots, T-n+i\}$ .

- $z_{u,i,t}$  est la quantité produite par le partenaire  $u \in \{1, \dots, q_i\}$  de l'étape  $i \in N$  durant la période  $t \in \{i, \dots, T-n+i\}$ .

- $x_{u,i,v,i+1,t}$  est la quantité transportée du stock de sortie du partenaire  $u \in \{1, 2, \dots, q_i\}$  de l'étape  $i \in \{1, 2, \dots, n-1\}$  vers le stock d'entrée du partenaire  $v \in \{1, \dots, q_{i+1}\}$  de l'étape  $i+1$  durant la période  $t \in \{i, 2, \dots, T-n+i\}$ .

A ces variables s'ajoutent deux séries de variables binaires :

$$W_{u,i} = \left\{ \begin{array}{ll} 1 & \text{si } u \in \{1, \dots, q_i\} \text{ de l'étape } i \in N \text{ est retenu dans la solution} \\ 0 & \text{sinon} \end{array} \right\}$$

$$Y_{u,i,v,i+1} = \left\{ \begin{array}{ll} 1 & \text{si } u \in \{1, \dots, q_i\} \text{ et } v \in \{1, \dots, q_{i+1}\} \\ & \text{pour } i \in \{1, \dots, n-1\} \text{ sont tous deux retenus} \\ 0 & \text{sinon} \end{array} \right\}$$

### Formation du coût à optimiser

Le coût total est formé de trois coûts :

#### a. *Le coût d'établissement des connexions*

Compte tenu des notations précédentes, ce coût s'écrit :

$$\text{Coût1}(Y) = \sum_{i=1}^{n-1} \sum_{u=1}^{q_i} \sum_{v=1}^{q_{i+1}} F_{u,i,v,i+1} \cdot Y_{u,i,v,i+1} \quad (5.1)$$

Ce coût est un coût d'investissement qui intervient en début de mise en place de la chaîne d'approvisionnement ; il ne fait donc pas intervenir le temps.

Dans ce modèle, nous n'avons fait intervenir ni le taux d'actualisation, ni le taux de dépréciation comme nous l'avons fait dans le modèle présenté dans le chapitre précédent : le but est de simplifier les notations. L'introduction de ces taux ne modifierait en rien la suite de la démarche.

#### b. *Le coût de transport à l'horizon T*

Compte tenu des notations introduites ci-dessus, ce coût s'écrit :

$$\text{Coût2}(x) = \sum_{i=1}^{n-1} \sum_{t=1}^{T-n+i} \sum_{u=1}^{q_i} \sum_{v=1}^{q_{i+1}} C_{v,i,v,i+1,t} \cdot x_{u,i,v,i+1,t} \cdot W_{v,i+1} \quad (5.2)$$

Cette expression est la somme des coûts de transport entre deux étapes successives sur l'horizon considéré.

#### c. *Les coûts de stockage et de production*

Nous avons, dans cette somme, trois termes qui représentent, dans l'ordre :

- Les coûts de stockage à l'entrée des noeuds.
- Les coûts de stockage à la sortie des noeuds.
- Les coûts opératoires.

Cette somme s'écrit :

$$\text{Coût3}(h^f, h^r, z) = \sum_{i=1}^n \sum_{t=i}^{T-n+i} \sum_{u=1}^{q_i} (H_{u,i,t}^f \cdot h_{u,i,t}^f + H_{u,i,t}^r \cdot h_{u,i,t}^r + U_{u,i,t} \cdot z_{u,i,t}) \quad (5.3)$$

### Les contraintes

Les contraintes (5.4), ajoutées au fait que les  $W_{u,i}$  sont binaires, forcent à choisir un noeud et un seul à chacune des étapes. Elles s'écrivent :

$$\sum_{u=1}^{q_i} W_{u,i} = 1 \quad \text{pour } i \in N \quad (5.4)$$

Les contraintes (5.5), ajoutées au fait que les  $W_{u,i}$  sont binaires, forcent les variables binaires  $Y_{u,i,v,i+1}$  à être au moins égales à 1 lorsque  $u$  et  $v$  font tous deux parties des candidats retenus.

$$Y_{u,i,v,i+1} \geq W_{u,i} + W_{v,i+1} - 1 \quad (5.5)$$

$$\text{pour } i = 1, 2, \dots, n-1, u = 1, 2, \dots, q_i, v = 1, 2, \dots, q_{i+1}$$

Remarque : Comme les variables binaires  $Y_{u,i,v,i+1}$  figurent dans le coût, et qu'il s'agit de minimiser ce coût, les contraintes (5.5) seront saturées à l'optimum.

Les contraintes (5.6) forcent la production à être inférieure ou égale à la capacité du noeud si le noeud est sélectionné, et à 0 sinon.

$$z_{u,i,t} \leq K_{u,i,t} \cdot W_{u,i} \quad \text{pour } i \in N, u = 1, \dots, q_i, t = i, \dots, T-n+i \quad (5.6)$$

Les contraintes (5.7) interdisent l'expédition à partir du noeud  $u$  de l'étape  $i$  si ce noeud n'est pas retenu, et ceci quelle que soit la période. La valeur de  $M$  est suffisamment importante pour que la contrainte soit inopérante si le noeud  $u$  est retenu.

$$\sum_{t=1}^{T-n+i+1} \sum_{v=1}^{q_i+1} x_{u,i,v,i+1,t} \leq W_{u,i} \cdot M \quad \text{pour } i = 1, 2, \dots, n-1, u = 1, \dots, q_i \quad (5.7)$$

Les contraintes (5.8) interdisent la réception d'une expédition au noeud  $v$  à l'étape  $i + 1$  si le noeud  $v$  n'est pas retenu.  $M$  joue un rôle analogue à celui qu'elle a joué dans les contraintes précédentes. Les contraintes deviennent inopérantes si le noeud  $v$  est choisi, car  $M$  est grand.

$$\sum_{t=1}^{T-n+i+1} \sum_{u=1}^{q_i} x_{u,i,v,i+1,t} \leq W_{v,i+1} \cdot M \quad (5.8)$$

pour  $i = 1, 2, \dots, n - 1$  et  $v = 1, 2, \dots, q_{i+1}$

Les contraintes (5.9) indiquent que le niveau stock à la sortie du noeud  $u$  de l'étape  $i$  à la fin de la période  $t$  est égal à la somme des éléments suivants :

- Le niveau du même stock à la fin de la période  $t - 1$ .
- La quantité fabriquée dans ce noeud durant la période  $t$ .
- La somme des expéditions du noeud  $u$  vers les noeuds de l'étape  $i + 1$  à laquelle on attribue le signe moins.

$$h_{u,i,t}^f = h_{u,i,t-1}^f + z_{u,i,t} - \sum_{v=1}^{q_{i+1}} x_{u,i,v,i+1,t} \quad (5.9)$$

$i = 1, 2, \dots, n - 1, u = 1, 2, \dots, q_i$  et  $t = i, \dots, T - n + i$

Les contraintes (5.10) s'intéressent à l'évolution des stocks de sortie des noeuds de la dernière étape de fabrication. Elles indiquent que le niveau du stock en sortie à la fin de la période  $t$  est égal au niveau du stock en sortie à la fin de la période  $t - 1$  plus la quantité produite dans le noeud au cours de la période  $t$  moins la demande à satisfaire au cours de la période  $t$ .

$$h_{u,n,t}^f = h_{u,n,t-1}^f + z_{u,n,t} - d_t \cdot W_{u,n} \quad u = 1, 2, \dots, q_n \text{ et } t = n, \dots, T \quad (5.10)$$

Les contraintes (5.11) indiquent que le niveau du stock d'entrée d'un noeud à la fin de la période  $t$  est égal au niveau du même stock à la fin de la période précédente, moins la production du noeud au cours de la période  $t$  (quantité

fournie par le stock d'entrée), plus la quantité qui parvient à ce noeud de l'étape précédente. Bien entendu, la première étape n'est pas concernée car elle n'est précédée par aucune autre étape.

$$h_{u,i,t}^r = h_{u,i,t-1}^r - z_{u,i,t} + \sum_{w=1}^{q_i-1} x_{w,i-1,u,i,t} \quad (5.11)$$

$$i = 2, \dots, n \text{ et } t = i, \dots, T - n + i$$

Les contraintes (5.12) et (5.13) spécifient que les stocks sont vides au lancement de la chaîne d'approvisionnement.

$$h_{u,i,0}^r = 0 \text{ pour } i = 1, 2, \dots, n \text{ et } u = 1, \dots, q_i \quad (5.12)$$

$$h_{u,i,0}^f = 0 \text{ } i = 1, 2, \dots, n, \text{ et } u = 1, 2, \dots, q_i \quad (5.13)$$

Les contraintes (5.14) et (5.15) indiquent que les quantités produites ou stockées doivent être positives ou nulles.

$$h_{u,i,t}^r, h_{u,i,t}^f, z_{u,i,t} \geq 0 \quad (5.14)$$

$$\text{pour } i = 1, 2, \dots, n, u = 1, 2, \dots, q_i \text{ et } t = i, \dots, T - n + i$$

$$x_{u,i,v,i+1,t} \geq 0 \quad (5.15)$$

$$i = 1, 2, \dots, n - 1, u = 1, \dots, q_i, v = 1, 2, \dots, q_{i+1} \text{ et } t = i, \dots, T - n + i$$

Les deux dernières séries de contraintes concernent les variables auxiliaires.

$$Y_{u,i,v,i+1} \geq 0 \text{ } i = 1, 2, \dots, n - 1, u = 1, \dots, q_i, v = 1, 2, \dots, q_{i+1} \quad (5.16)$$

Nous n'imposons pas à ces variables d'être binaires car, comme nous l'avons souligné, les contraintes (5.5) sont toujours saturées à l'optimum, ce qui entraîne que ces variables prennent soit la valeur 0, soit la valeur 1 à l'optimum.

$$W_{u,i} \in [0, 1] \text{ pour } i = 1, \dots, n \text{ et } u = 1, \dots, q_i \quad (5.17)$$

Le problème à résoudre consiste donc à minimiser le coût total :

$$\text{Coût1}(Y) + \text{Coût2}(x) + \text{Coût3}(h^f, h^r, z)$$

sous les contraintes (5.4) à (5.17).

## 5.3 Une approche par relaxation des chemins

### 5.3.1 Remarques

La difficulté du problème à résoudre est la conséquence des trois particularités suivantes :

- La présence d'un coût fixe lors de l'établissement d'une connexion entre deux noeuds appartenant à deux étapes successives.
- La capacité finie des noeuds et des systèmes de transport.
- La continuité à assurer entre les différentes étapes de transport.

La taille de ce problème devient rapidement importante.

a. Nombre de variables

Un calcul simple montre que le nombre total de variables est :

$$NV = (T - n + 2) \cdot \sum_{i=1}^{n-1} (q_i \cdot q_{i+1}) + (3T - 3n + 4) \sum_{i=1}^n q_i$$

parmi lesquelles  $\sum_{i=1}^n q_i$  variables binaires. Les variables d'écart ne sont pas incluses.

b. Nombre de contraintes :

La formule suivante donne le nombre de contraintes, tous types confondus :

$$NC = 6(T - n + 2) \sum_{i=2}^{n-1} q_i + q_1 + Q_n(6T - 6n + 11) + (T - n + 3) \sum_{i=1}^{n-1} (q_i \cdot q_{i+1})$$

Considérons un exemple numérique de faible taille :

$$n = 6, q_i = 10 \text{ pour } i=1, \dots, 6, T = 10$$

Nous obtenons :

NV=3960 (sans les variables d'écart)

NC=5300

Nous voyons la nécessité de disposer d'une heuristique efficace.

### 5.3.2 Présentation de la méthode

L'idée qui sous-tend la méthode par relaxation des chemins consiste à compléter chaque noeud des étapes  $i = 1, 2, \dots, n - 1$  par un arc virtuel supposé rattacher ce noeud à n'importe lequel des noeuds de l'étape suivante. Nous définissons ainsi un couple isolé constitué d'un noeud et d'un arc virtuel unique dont ce noeud serait l'origine. Le coût associé à ce couple sera le minimum du coût entre ce noeud et les noeuds de l'étape suivante. Ce coût minimal étant calculé, il est affecté à toutes les paires composées du noeud considéré et d'un noeud de l'étape suivante.

Nous comprenons que si nous cherchons alors le plus court chemin entre le noeud origine (étape 0) et le noeud final (étape  $n+1$  de la figure (5.1)), alors le coût virtuel associé à ce chemin sera une borne inférieure du coût optimal cherché. Il est alors aisé de calculer le coût réel associé au chemin sélectionné. Bien entendu, ce coût réel n'est pas le coût optimal, mais sa comparaison avec le coût virtuel nous donne un écart maximal entre le coût réel et le coût optimal.

De plus, en calculant un certain nombre de meilleurs chemins basés sur les coûts virtuels, en calculant les coûts réels associés ces chemins, puis en conservant le chemin ayant le meilleur coût réel, nous obtenons une solution admissible qui conduit, comme nous le verrons dans les exemples numériques, à un coût tout à fait satisfaisant et, de plus, optimal s'il est inférieur à l'un des coûts virtuels calculés précédemment. Nous démontrerons ce résultat.

### 5.3.3 Calcul du coût associé au couple virtuel

Nous désignerons le problème de programmation linéaire que nous présentons maintenant par  $P_1^{u,i}$ . Il a pour objectif de calculer le coût associé au noeud  $u$  de l'étape  $i$  complété par l'arc virtuel que nous avons introduit dans le paragraphe précédent.

$(P_1^{u,i})$  s'écrit :

$$\begin{aligned} \text{Minimiser} \left\{ \sum_{v=1}^{q_{i+1}} F_{u,i,v,i+1} \cdot W_{v,i+1} + \sum_{t=i}^{T-n+i} \sum_{v=1}^{q_{i+1}} C_{u,i,v,i+1,t} \cdot x_{u,i,v,i+1,t} \cdot W_{v,i+1} \right. \\ \left. + \sum_{t=1}^{T-n+i} (\text{Min}[H_{u,i,t}^f, H_{v,i+1,t}^r \cdot W_{v,i+1}] h_{u,i,t}^f + U_{u,i,t} \cdot z_{u,i,t}) \right\} \end{aligned} \quad (5.18)$$

sous les contraintes :

$$\sum_{v=1}^{q_{i+1}} W_{v,i+1} = 1, \text{ pour } i = 1, \dots, n-1 \quad (5.19)$$

$$z_{u,i,t} \leq K_{u,i,t} \text{ pour } t = i, i+1, \dots, T-n+i \text{ et pour } i = 1, \dots, n-1 \quad (5.20)$$

$$\sum_{t=i}^{k-n+i} x_{u,i,v,i+1,t} \geq W_{v,i+1} \cdot \sum_{t=n}^k d_t \text{ pour } v = 1, \dots, q_{i+1} \text{ et } k = n, \dots, T \quad (5.21)$$

pour  $i = 1, \dots, n-1, u = 1, 2, \dots, q_i$

$$h_{u,i,t}^f = h_{u,i,t-1}^f + z_{u,i,t} - \sum_{v=1}^{q_{i+1}} x_{u,i,v,i+1,t} \quad (5.22)$$

pour  $t = i, i+1, \dots, T-n+i, i = 1, \dots, n-1, u = 1, 2, \dots, q_i$

$$h_{u,i,i-1}^f = 0 \quad (5.23)$$

pour  $i = 1, \dots, n-1$  et  $u = 1, 2, \dots, q_i$

$$h_{u,i,t}^f, z_{u,i,t} \geq 0 \quad \text{pour } t = i, i+1, \dots, T-n+i, \quad t = i, i+1, \dots, T-n+i, \quad \text{et } u = 1, 2, \dots, q_i \quad (5.24)$$

$$x_{u,i,v,i+1,t} \geq 0 \quad (5.25)$$

pour  $v = 1, \dots, q_{i+1}, t = i, i+1, \dots, T-n+i, i = 1, \dots, n-1, u = 1, 2, \dots, q_i$

$$W_{v,i+1} \geq 0 \quad v = 1, \dots, q_{i+1}, i = 1, 2, \dots, n-1 \quad (5.26)$$

La première somme figurant dans le coût correspond au coût fixe pour la connexion du noeud  $u$  de l'étape  $i$  au noeud  $v$  de l'étape  $i+1$  qui sera finalement retenu.

La seconde somme figurant dans le coût correspond au coût de transport entre le noeud  $u$  de l'étape  $i$  et le noeud  $v$  de l'étape  $i+1$  qui sera finalement retenu.

Enfin, la troisième somme figure dans le coût regroupe le coût de stockage et le coût de production. Remarquons que le coût de stockage retenu est le minimum entre le coût de stockage à la sortie du noeud  $u$  de l'étape  $i$  et le coût de stockage à l'entrée des noeuds  $v$  de l'étape  $i+1$ . Le coût de fabrication est celui du noeud  $u$  de l'étape  $i$ .

Dans les contraintes (5.19), les variables  $W_{v,i+1}$  sont considérées comme continues. Cela signifie que nous avons relaxé la contrainte qui impose un seul partenaire à l'étape  $i+1$ . Nous verrons que nous pouvons conserver cette contrainte compte tenu de la taille réduite du problème ( $P_1^{u,i}$ )

Les contraintes (5.20) indiquent que la production du noeud  $u$  de l'étape  $i$  ne peut excéder la capacité de ce noeud au cours des périodes actives.

Les contraintes (5.21) imposent que les quantités cumulées transportées du noeud  $u$  de l'étape  $i$  au noeud  $v$  de l'étape  $i+1$  soient supérieures ou égales

aux besoins sur les périodes correspondantes.

Les égalités (5.22) sont les équations d'état classiques.

L'égalité (5.23) indique que le stock en sortie du noeud  $u$  de l'étape  $i$  est vide à l'instant  $i - 1$ .

Enfin, les égalités (5.24), (5.25) et (5.26) indiquent que les variables sont positives ou nulles.

Remarquons à nouveau que, dans le problème que nous venons de présenter, le noeud  $u$  de l'étape  $i$  est autorisée à approvisionner plusieurs noeuds de l'étape  $i + 1$ .

La solution du problème de programmation linéaire que nous venons de présenter conduit au coût que nous retiendrons pour le couple constitué du noeud  $(u, i)$  et de l'arc virtuel qui admet ce noeud comme origine. Les résultats qui suivent justifient ce choix.

### 5.3.4 Propriétés du problème $(P_1^{u,i})$

Le premier résultat indique que le coût optimal obtenu en résolvant le problème  $(P_1^{u,i})$  est une borne inférieure de tout couple constitué du noeud  $(u, i)$  et du transport de ce noeud à tout autre noeud de l'étape  $i + 1$ .

#### Résultat 1

Considérons un noeud  $u$  à l'étape  $i$  et les arcs dont ce noeud est origine. La solution du problème  $(P_1^{u,i})$  conduit à une borne inférieure de la somme (production du noeud  $u$  de l'étape  $i$  **plus** transport vers le noeud  $v$  ( $v$  est un noeud de l'étape  $i + 1$ ) de l'étape  $i + 1$  **plus** coûts de stockage à la sortie de  $u$  et à l'entrée des noeuds  $v$ ) sur l'horizon  $T$ .

#### Démonstration

Le problème  $(P_1^{u,i})$  est une forme relaxée du problème initial (que nous désignons par  $P$ ) réduit au noeud  $u$  de l'étape  $i$  et au transport entre ce noeud et l'étape suivante.

Nous voyons, en effet, que :

- (a) La production au noeud  $u$  de l'étape  $i$  n'est contrainte ni par le réapprovisionnement qui provient de l'étape précédente (si  $i$  n'est pas la première étape), ni par le réapprovisionnement de l'étape suivante, à condition que la demande cumulée soit satisfaite.
- (b) Le coût de stockage est le minimum entre le coût de stockage à la sortie du noeud  $u$  de l'étape  $i$  et à l'entrée d'un noeud  $v$  de l'étape  $i + 1$ .
- (c) Plusieurs noeuds  $v$  de l'étape  $i + 1$  peuvent être utilisés car les paramètres  $W_{v,i+1}$  ne sont pas forcés d'être binaires.

Ces remarques rendent le résultat 1 évident puisque  $(P_1^{u,i})$  est le problème relaxé du problème  $(P)$  pour la partie englobant le noeud  $u$  de l'étape  $i$  et l'étape  $i + 1$ .

Bien entendu, si nous imposons aux variables  $W_{v,i+1}$  d'être égales soit à 0, soit à 1, c'est-à-dire si nous remplaçons la contraintes (5.26) par :

$$W_{v,i+1} \in \{0, 1\} \quad \text{pour } v = 1, 2, \dots, q_{i+1} \quad (5.27)$$

alors le coût optimal associé au problème ainsi transformé et que nous désignons par  $(P_{1,a}^{u,i})$  est supérieur ou égal au coût optimal obtenu en résolvant le problème  $(P_1^{u,i})$ . Cependant, ce coût continue à jouer son rôle de borne inférieure : c'est ce qu'indique le résultat 2.

### Résultat 2

La solution du problème  $(P_{1,a}^{u,i})$  conduit à une borne inférieure du coût de production au noeud  $u$  de l'étape  $i$  et du transport vers tout noeud  $v$  de l'étape  $i + 1$  sur l'horizon  $T$ . Cette borne inférieure est meilleure que celle donnée par la solution du problème  $(P_1^{u,i})$ .

### Démonstration

Le problème  $(P_{1,a}^{u,i})$  est, comme le problème  $(P_1^{u,i})$ , une forme relaxée du problème  $(P)$  limitée au noeud  $u$  de l'étape  $i$  et à l'étape  $i + 1$ . Les remarques

(a) et (b) faites dans la démonstration du résultat 1 sont encore vraies. Le fait que cette borne est meilleure que la précédente a été souligné plus haut. ■

Il faudra donc résoudre soit le problème  $(P_1^{u,i})$ , soit le problème  $(P_{1,a}^{u,i})$  pour chaque noeud du système.  $(P_1^{u,i})$  est un problème de programmation linéaire (PL) alors que le problème  $(P_{1,a}^{u,i})$  est un problème de programmation linéaire mixte (PLM), mais de taille raisonnable.

Nous pouvons aussi décider de calculer une borne inférieure pour chaque couple  $(u, i)$ ,  $(v, i + 1)$  si le problème  $(P)$  initial est de taille raisonnable. Il faudra alors résoudre ce problème pour chaque couple de noeuds appartenant à deux étapes successives. Nous appelons ce problème  $(P_2^{u,i,v,i+1})$ . Il s'écrit :

$$\begin{aligned} & \text{Minimiser} \left\{ F_{u,i,v,i+1} + \sum_{t=i}^{T-n+i} C_{u,i,v,i+1,t} \cdot x_{u,i,v,i+1,t} \right. \\ & \left. + \sum_{t=i}^{T-n+i} (\text{Min}[H_{u,i,t}^f, H_{v,i+1,t}^r] \cdot h_{u,i,t}^f + U_{u,i,t} \cdot z_{u,i,t}) \right\} \end{aligned} \quad (5.28)$$

sous les contraintes :

$$z_{u,i,t} \leq K_{u,i,t} \quad \text{pour } t = i, i + 1, \dots, T - n + i \quad (5.29)$$

$$\sum_{t=i}^{k-n+i} x_{u,i,v,i+1,t} \geq \sum_{t=n}^k d_t \quad \text{pour } k = n, n + 1, \dots, T \quad (5.30)$$

$$h_{u,i,t}^f = h_{u,i,t-1}^f + z_{u,i,t} - x_{u,i,v,i+1,t} \quad \text{pour } t = i, i + 1, \dots, T - n + i \quad (5.31)$$

$$h_{u,i,i-1}^f = 0 \quad (5.32)$$

$$h_{u,i,t}^f, x_{u,i,v,i+1,t}, z_{u,i,t} \geq 0 \quad (5.33)$$

Compte tenu des explications données pour les autres problèmes nous voyons que :

- Le premier terme du coût est le coût fixe d'établissement d'une connexion entre  $(u, i)$  et  $(v, i + 1)$ .
- Le second terme du coût est le coût de transport entre  $(u, i)$  et  $(v, i + 1)$ .
- Le troisième terme est une minimisation du coût de stockage entre  $(u, i)$  et  $(v, i + 1)$  plus le coût de production au noeud  $(u, i)$ .
- Les contraintes (5.29) indiquent que la production au noeud  $(u, i)$  ne doit pas excéder la capacité de ce noeud sur l'horizon choisi.
- Les contraintes (5.30) indiquent que la production cumulée au noeud  $(u, i)$  doit toujours être supérieure ou égale à la demande cumulée.
- Les contraintes (5.31) sont les équations d'état pour  $(u, i)$ .
- Les contraintes (5.32) indiquent que le stock de sortie du noeud  $(u, i)$  est vide la première fois qu'il est activé.
- Enfin, les contraintes (5.33) sont les contraintes classiques de positivité des variables.

Le problème  $(P_2^{u,i,v,i+1})$  est un problème PL. Le résultat suivant est similaire aux précédents.

### Résultat 3

Le coût optimal obtenu en résolvant le problème  $P_2^{u,i,v,i+1}$  est une borne inférieure du coût de production de noeud  $(u, i)$  et du coût de transport entre  $(u, i)$  et  $(v, i + 1)$  si ces deux noeuds font partie d'un chemin, c'est-à-dire s'ils interviennent dans le processus de production. De plus, en calculant ce coût pour tout couple  $(u, v)$ ,  $v \in \{1, 2, \dots, q_{i+1}\}$ , nous avons une borne inférieure meilleure que la borne inférieure obtenue en résolvant le problème  $(P_{1,a}^{u,i})$ .

### Démonstration

Dans la solution de  $(P_{1,a}^{u,i})$  le coût attribué au transport entre  $(u, i)$  et  $(v, i + 1)$  est le plus petit des coûts de transport entre  $(u, i)$  et un arc de l'étape  $i + 1$ .

Donc la solution de  $(P_{1,a}^{u,i})$  affecte le même coût à tous les couples constitués de  $(u, i)$  et d'un noeud de  $(i + 1)$ . Par contre, la solution de  $(P_2^{u,i,v,i+1})$  donne une borne inférieure spécifique au couple des noeuds traités. ■

### 5.3.5 Les algorithmes

L'algorithme général a déjà été esquissé en début de chapitre. Nous pouvons maintenant préciser notre approche en donnant deux algorithmes voisins.

#### Algorithme 1 : Utilisation des problèmes $(P_1^{u,i})$ ou $(P_{1,a}^{u,i})$

1. Recherche des coûts minima.

Pour chaque étape  $i = 1, 2, \dots, n$  et chaque noeud  $u = 1, \dots, q_i$  :

- (a) Résoudre le problème  $(P_1^{u,i})$  ou  $(P_{1,a}^{u,i})$ .
- (b) Affecter, à tout arc ayant pour origine le noeud  $(u, i)$ , une "longueur" égale à la valeur optimale du critère du problème résolu.

2. Recherche d'une solution proche de l'optimum.

(a) Introduction

Poser :

$k = 1$  (Compteur d'itération)

$LB = 0$  (Borne inférieure)

$UB = \infty$  (Borne supérieure)

$L^* = \emptyset$  (Chemin retenu)

(b) Calculer le  $k$ -ème plus court chemin du graphe en utilisant les "longueurs" calculée en 1(b).

Si un tel chemin n'existe pas, fin de l'algorithme.

Sinon, soit  $l_k$  le chemin obtenu et  $r_k$  sa "longueur", qui est une borne inférieure de la longueur réelle du chemin.

(c) Si  $r_k > LB$ , on pose  $LB = r_k$ .

3. Calculer la "longueur réelle  $r_k^*$  du chemin  $l_k$ , c'est-à-dire le coût réel associé au choix du chemin  $l_k$ . Ce calcul conduit à un problème de programmation linéaire simple dont l'objectif est de trouver la meilleure répartition de la fabrication et du transport pour les partenaires retenus. Bien entendu,  $r_k^*$  est une borne supérieure du coût optimal du problème  $(P)$ .  
Si  $r_k^* < UB$ , on pose  $UB = r_k^*$  et  $L^* = l_k$   
Si  $UB \leq LB$ ,  $L^*$  est le chemin optimal. Fin.
4. Si  $k = K$  ( $K$  est le nombre maximal de chemins que l'utilisateur soull explorer), conserver le chemin  $L^*$  et fin.  
Sinon, faire  $k = k + 1$  et retourner en (b).

Un second algorithme peut être appliqué

#### **Algorithme 2 : Utilisation des problèmes $(P^{u,i,v,i+1})$**

Cet algorithme est le même que le précédent, à l'exception de l'étape 1 qui devient :

Pour tout  $i = 1, 2, \dots, n, u = 1, 2, \dots, q_i$  et  $v = 1, 2, \dots, q_{i+1}$ , résoudre le problème  $(P^{u,i,v,i+1})$  et affecter la valeur optimale du critère à l'arc  $(u, v)$ .

### **5.3.6 Propriété d'optimalité**

Le résultat suivant indique que les algorithmes que nous venons de présenter peuvent conduire à l'optimum.

#### **Résultat 4**

Les algorithmes 1 et 2 conduisent à la solution optimale (si elle existe) si l'utilisateur ne limite pas le nombre de chemins à explorer.

#### **Démonstration**

L'algorithme s'interrompt lorsque la plus petite des bornes supérieures obtenues à cet instant devient inférieure à la borne inférieure du chemin que l'on

vient d'explorer. Comme les chemins sont examinés dans l'ordre croissant de leur borne inférieure, nous pouvons dire que la plus petite des bornes supérieures obtenues à cet instant est inférieure aux bornes inférieures de tous les chemins non encore explorés, donc à la "longueur" de tous les chemins non encore explorés. Comme, par ailleurs, cette plus petite des bornes supérieures obtenues à cet instant est le plus court des chemins obtenus à cet instant, la démonstration est terminée. ■

## 5.4 Exemple numérique et évaluation numérique

### 5.4.1 Un exemple numérique

Nous considérons un problème comportant 4 étapes avec 5 noeuds à chaque étape. Le  $j$ -ème noeud de l'étape  $i$  est numéroté  $5(i - 1) + j$ . Par conséquent :

- les noeuds de la première étape sont 1,2,3,4 et 5.
- les noeuds de la seconde étape sont 6,7,8,9, et 10,
- les noeuds de la troisième étape sont 11,12,13,14, et 15,
- les noeuds de la quatrième étape sont 16,17,18, 19, et 20.

Le figure 5.2 donne le graphe de notre problème avec la numération des arcs. Pour ne pas surcharger la figure, nous n'avons pas tracé tous les noeuds.

L'horizon du problème est  $T = 8$  et les demandes sont :

- au temps 5 : 300
- au temps 6 : 360
- au temps 7 : 340
- au temps 8 : 300

Nous notons en effet que le système est vide à l'origine. La première demande ne peut donc intervenir qu' à la fin de la quatrième période, c'est-à-dire au temps 5.

Tous les noeuds d'une étape sont connectés à tous les noeuds de l'étape suivante, si elle existe.

Les coûts fixes et les coûts de transport sont donnés dans les tableaux 5.1, 5.2 et 5.3.

TAB. 5.1 – Coûts fixes et coûts de transport : étapes 1 à 2

Noeuds	Noeud 6		Noeud 7		Noeud 8		Noeud 9		Noeud 10	
	Fixe	Transp.	Fixe	Transp.	Fixe	Transp.	Fixe	Transp.	Fixe	Transp.
noeud1	1400	10.5	1500	12.5	1600	11.5	1300	14.0	1500	12.0
noeud2	1300	11.0	1500	12.8	1700	12.5	1750	11.0	1900	12.0
noeud3	1100	11.0	2000	14.0	1680	12.0	1300	14.0	2000	14.0
noeud4	1200	12.0	1300	12.0	1450	12.6	1650	14.0	1500	18.0
noeud5	1700	10.5	1600	13.0	1570	10.6	1700	14.0	2000	14.0

Les coûts de production unitaires associés à chaque noeud sont donnés dans le tableau 5.4. Nous avons supposé que les coûts de transport sont les mêmes entre tous les couples de noeuds appartenant à des étapes successives, et que les coûts de transport de l'étape 4 au noeud final sont nuls.

Pour résoudre ce problème, nous utilisons le second algorithme car le problème est de taille raisonnable : il comporte  $3 \times 25 \times +5 = 80$  arcs. Nous ré-

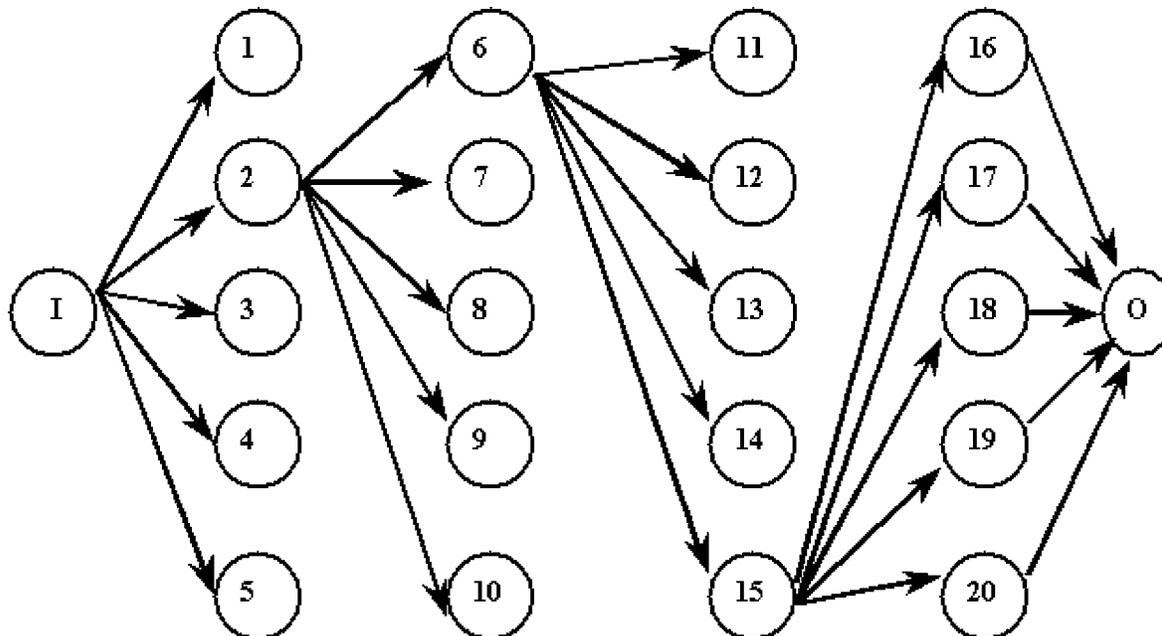


FIG. 5.2 – Le graphe partiel du problème

TAB. 5.2 – Coûts fixes et coûts de transport : étapes 2 à 3

noeuds	Noeud 11		Noeud 12		Noeud 13		Noeud 14		Noeud 15	
	fixe	Transp.								
noeud6	1200	14.2	1700	15.0	1600	12.0	1400	13.5	1780	8.0
noeud7	1800	15.6	1400	12.5	1800	14.0	1750	16.5	1980	11.0
noeud8	1720	19.8	1900	17.6	1900	14.5	1980	12.5	1700	12.5
noeud9	1820	12.5	1600	18.1	2200	12.5	1270	12.0	1900	16.5
noeud10	1880	15.4	1200	12.5	1500	16.0	1180	17.0	1800	16.0

TAB. 5.3 – Coûts fixes et coûts de transport : étapes 3 à 4

noeuds	noeud 16		noeud 17		noeud 18		noeud 19		noeud 20	
	fixe	Transp.								
noeud11	2700	12.5	1800	16.0	1900	16.5	1100	9.8	2500	16.0
noeud12	1800	17.0	1200	13.0	1400	12.0	1900	10.5	1000	11.0
noeud13	1100	10.5	1400	11.0	1900	13.5	1600	18.0	1050	12.0
noeud14	1350	14.5	1600	9.5	1800	19.0	1780	14.0	1850	14.0
noeud15	1600	17.5	1100	8.5	1400	10.5	1300	14.0	1250	10.0

solvons donc le problème ( $P_2^{u,i,v,i+1}$ ) 80 fois pour obtenir une borne inférieure associée à chaque arc.

Nous décidons ensuite de traiter les 11 plus courts chemins du graphe en prenant pour longueur des arcs les bornes inférieures que nous venons de calculer. Les résultats sont donnés dans le tableau 5.5.

Nous observons que la borne inférieure - c'est-à-dire la longueur qui prend en compte les bornes inférieures des arcs du chemin 11 - est plus grande que la borne supérieure, - c'est-à-dire la longueur réelle - du chemin numéro 7. Donc l'optimum est atteint : il consiste à sélectionner les candidats 5,6,15, et 17, et le coût sur l'horizon 8 sera 296910 unités monétaires.

Bien entendu, un problème de taille plus importante nous aurait demandé d'utiliser l'algorithme 1 qui exige moins de calculs pour obtenir des bornes inférieures, mais qui conduit à des bornes inférieures de plus mauvaise qualité ce qui, en moyenne, exige d'explorer plus de chemins si l'on veut atteindre la solution optimale.

TAB. 5.4 – Valeur des paramètres

Paramètres	Capacités				Coût de production unitaire				Coût de stockage (par unité)	
	pour les périodes				pour les périodes				matière première	produits finis
	1	2	3	4	1	2	3	4		
noeud1	333	312	329	321	33	32	35	42	0	10.25
noeud2	340	345	300	360	38	37	39	39	0	8.25
noeud3	330	330	330	330	37	39	37	37	0	8.70
noeud4	400	400	380	380	34	36	35	40	0	9.0
noeud5	370	360	360	360	38	36	36	36	0	8.50
noeud6	340	340	340	340	55	56	58	59	6.15	12.25
noeud7	340	333	330	410	52	54	55	54	7.11	14.75
noeud8	360	340	360	410	54	56	55	54	7.89	12.75
noeud9	340	320	330	390	51	54	52	51	7.55	12.50
noeud10	345	331	330	330	51	53	52	51	7.65	12.0
noeud11	370	360	330	300	49	49	52	51	8.85	14.70
noeud12	370	360	330	300	49	49	50	51	8.80	14.80
noeud13	350	350	350	350	49	49	51	50	8.60	14.50
noeud14	360	350	350	350	53	53	52	53	8.40	14.0
noeud15	360	360	360	360	55	52	54	52	7.5	14.35
noeud16	390	370	340	350	48	51	52	51	6.87	14.55
noeud17	370	360	340	390	49	52	52	51	7.59	15.45
noeud18	350	370	330	400	49	49	52	51	7.65	15.55
noeud19	350	340	350	300	49	49	52	51	7.65	15.0
noeud20	370	380	350	250	49	49	49	49	7.8	15.0

### 5.4.2 Evaluation numérique de l'algorithme 2

Pour évaluer numériquement l'algorithme 2, nous avons généré au hasard 50 exemples comportant chacun 5 étapes avec 2 à 6 noeuds par étape. L'horizon choisi est 8. Les résultats sont résumés dans le tableau 5.6.

Dans plus de 90% des cas, la solution optimale a été trouvée après avoir exploré moins de 30 chemins, mais dans 2 exemples, la solution optimale n'est pas obtenue après avoir exploré 50 chemins.

TAB. 5.5 – Résultats fournis par l'algorithme 2

Chemin n°	Plus court chemin	Borne inférieure	Borne supérieure
1	4-7-12-20	294007	Chemin impossible
2	4-7-12-19	295787	297409
3	5-10-12-20	295840	Chemin impossible
4	5-6-15-20	296040	Chemin impossible
5	4-6-15-20	296190	Chemin impossible
6	4-7-15-20	296257	Chemin impossible
7	5-6-15-17	296430	296910
8	4-6-15-17	296580	297940
9	4-7-15-17	296647	298115
10	5-7-12-20	296907	Chemin impossible
11	4-7-12-18	297147	298761

Note : Le chemin est impossible lorsque l'un de ses arcs n'existe pas, ce qui n'empêche pas de créer le chemin virtuel.

TAB. 5.6 – Résultats de l'évaluation numérique

Nombre de chemins testés entre $n1$ et $n2$	Nombre d'exemples dans lesquels la solution obtenue est optimale en explorant entre $n1$ et $n2$ chemins
1 - 5	7
6 - 10	13
11 - 20	22
21 - 30	3
31 - 40	1
41 - 50	2

## 5.5 Conclusion

Le modèle que nous venons de présenter permet de sélectionner rapidement les participants à une chaîne d'approvisionnement dont l'objectif est de faire face à une opportunité du marché en utilisant les capacités disponibles. Nous avons pu observer que le partenaire le moins cher ou le plus proche n'est pas toujours la meilleure solution lorsqu'il s'agit de coordonner l'action aux

différentes étapes et que le transport et le stockage sont des contraintes qui ne peuvent pas être négligées.

Nous avons développé deux algorithmes heuristiques, sachant que s'ils ne sont pas interrompus prématurément, ils conduisent à la solution optimale en explorant toutes les candidatures. L'originalité de ces heuristiques réside dans la définition d'une borne inférieure du coût de production d'un noeud plus le coût de stockage à la sortie de ce noeud et à l'entrée du suivant, plus le coût de transport entre les deux noeuds.

Comme dans tous les problèmes de ce type, le nombre de chemins à explorer pour atteindre l'optimum ne peut être prévu à l'avance. L'interruption du programme après un nombre donné d'explorations lui vaut le qualificatif d'heuristique.



## Troisième partie

### Quelques problèmes du niveau tactique

Dans cette troisième partie de la thèse, nous traitons des problèmes qui se sont posés au niveau tactique et dont les solutions sont complémentaires du logiciel CAD-SC-SL que nous venons de présenter. Les problèmes traités sont des problèmes d'ordonnancement en temps réel, d'affectation et de partage des pertes et des bénéfices.

# Chapitre 6

## Ordonnancement en temps réel

### 6.1 Introduction

Ce chapitre concerne le niveau tactique. Nous allons voir que l'environnement imposé par les chaînes d'approvisionnement change radicalement la nature des problèmes d'ordonnancement.

Dans le passé, l'objectif était d'ordonnancer les tâches concernant un département. Les tâches à ordonnancer étaient connues en début d'une période (un jour, une semaine ou un mois, suivant le type d'activité), et l'ordonnancement consistait à séquencer ces tâches sur toute la période de façon à optimiser un critère qui, lui aussi, dépendait du type d'activité. Dans le meilleur des cas, l'ordonnancement était rafraîchi à plusieurs reprises au cours de la période pour tenir compte des aléas qui surviennent immanquablement. Une autre technique consistait à procéder par "plan glissant", technique qui consiste à ordonnancer sur un horizon  $T$ , mais à effectuer l'ordonnancement toutes les périodes  $t$  ( $t \ll T$ ) en tenant compte de l'état du système à l'instant où l'ordonnancement est calculé.

La compétition, qui est devenue mondiale, a changé radicalement la manière d'envisager les problèmes d'ordonnancement. Il faut désormais :

- Maintenir les stocks à un niveau aussi faible que possible. La raison est non seulement qu'il faut réduire les coûts de stockage, mais aussi qu'il faut éviter

de se retrouver avec des stocks inutilisables du fait de demandes très versatiles.

- Etre en mesure de répondre aux demandes dans des délais aussi faibles que possible : c'est la réactivité, facteur essentiel de compétitivité. Ici encore, des stocks réduits permettent de diminuer le cycle de production de manière importante.

- Etre en mesure d'informer en temps réel des délais proposés. Cette dernière exigence demande qu'il soit possible d'ordonnancer une demande de la matière première à la livraison, et ceci dès qu'elle se manifeste.

C'est pour répondre à ces objectifs que nous avons développé les travaux exposés dans ce chapitre.

Dans la suite, nous nous référons à l'introduction de l'ordonnancement "temps réel" faite dans le chapitre 1, paragraphe 1.4.2, sous-paragraphe "Le niveau tactique". La première étude menée par l'équipe SAGEP (Simulation, Analyse et GEstion des Systèmes de Production) date de plusieurs années. Elle s'intéressait aux problèmes de type "no-wait" et "limited controllable processing times". Dans ces problèmes, les temps opératoires peuvent être étendus, mais un produit ne peut séjourner entre deux opérations successives. Les premiers travaux consacrés à ce type de problèmes concernaient les traitements de surface [BPV83]. Il s'agissait de problèmes de type job-shop, et une version de l'algorithme développé à cette occasion est rappelée dans le paragraphe suivant. Les évolutions de la démarche sont données dans le chapitre 1. Retenons simplement :

- que, jusqu'à ce jour, aucun algorithme simple n'a été fourni pour les systèmes d'assemblage,

- qu'aucune étude systématique n'a été menée pour mettre en lumière les approches développées pour le contrôle des en-cours, objectif capital dans le nouveau contexte économique, comme nous l'avons souligné plus haut.

## 6.2 Le problème "no-wait" et "limited controllable processing times" original

### 6.2.1 Introduction

Ce problème est, comme nous l'avons souligné plus haut, un problème de type "job-shop". Par ailleurs, nous sommes dans un contexte temps-réel. Cela signifie que les commandes sont ordonnancées "à la volée", et qu'un ordonnancement n'est pas remis en cause par l'arrivée d'une nouvelle commande.

Nous sommes dans le cas où la fabrication démarre avec la matière première, qui subit une suite d'opérations (appelée gamme de fabrication) conduisant au produit fini. Le modèle que nous examinons est donc un modèle qui exclut la convergence de plusieurs activités, et en particulier l'assemblage. Chaque opération de la suite peut être exécutée, éventuellement, par plusieurs ressources différentes mais identiques; nous choisirons celle qui convient le mieux pour terminer la commande le plus tôt possible.

Remarquons qu'une opération peut nécessiter l'utilisation simultanée de plusieurs ressources. Par exemple, un travail d'usinage nécessite la disponibilité d'une machine, d'un outil et d'un employé. De même, le transport d'un produit nécessite un camion et un chauffeur. Nous pourrions encore citer la réception de matière première qui demande que soient disponibles des manutentionnaires, des ressources de levage et des places de stockage. Dans le cas où plusieurs ressources interviennent dans l'exécution d'une opération, c'est la disponibilité de l'ensemble de ces ressources que nous appelons ressource pour l'opération considérée.

### 6.2.2 Notations et classement des fenêtres

Désignons par  $i = 1, 2, \dots, m$  les opérations à exécuter successivement pour réaliser la commande et par  $n_i$  le nombre de ressources (ou d'ensembles de ressources, si plusieurs ressources sont requises conjointement) identiques qui

peuvent être utilisées pour réaliser l'opération  $i$ .

Pour  $i = 1, 2, \dots, m$  et  $j = 1, 2, \dots, n_i$  nous désignons par  $a_{i,j}^k$  le début de  $k$ -ème fenêtre de disponibilité de la ressource  $j$  qui peut effectuer l'opération  $i$ . De la même manière,  $b_{i,j}^k$  est la fin de cette fenêtre. Nous supposons que  $k = 1, 2, \dots, f_{i,j}$ ,  $f_{i,j}$  étant la dernière fenêtre de disponibilité de la ressource  $j$  qui peut exécuter l'opération  $i$ . Evidemment, la fin de la dernière période de disponibilité sur chaque ressource est nécessairement égale à  $+\infty$ .

Nous désignons par  $\theta_i$  et  $\theta_i + \delta_i$  respectivement le temps minimal et le temps maximal requis pour exécuter l'opération  $i$ . Cela signifie qu'il est possible de choisir n'importe quelle valeur située entre ces deux bornes comme temps d'exécution de l'opération  $i$ .

Dans la pratique, cela signifie que  $\theta_i$  est le temps opératoire réel. Si un produit séjourne sur la machine plus que le temps  $\theta_i$ , la machine n'est pas disponible pour un autre produit bien qu'elle soit en arrêt : il s'agit du phénomène de blocage. Le système présente cependant une flexibilité limitée car les temps opératoires peuvent être augmentés dans certaines limites. De plus, nous faisons l'hypothèse qu'il n'y a pas de conflit entre les machines, ce qui signifie qu'une machine n'est pas utilisée pour exécuter deux opérations différentes sur un même produit.

Soulignons que nous ne traitons qu'une commande à la fois. Dès qu'une commande est ordonnancée, les périodes de disponibilité des ressources sont modifiées en conséquence, et la commande suivante est traitée à partir de ces nouvelles données. D'autre part, il faut noter que la suite des opérations à effectuer pour réaliser une commande varie d'une commande à l'autre car nous sommes dans le contexte d'un " job-shop ".

Notons  $x_i$  l'instant où l'opération  $i$  commence. Comme l'attente n'est pas autorisée entre deux opérations (nous sommes dans le cas d'un ordonnancement à temps opératoires contrôlables),  $x_i$  représente également la fin de

l'opération précédente, si elle existe, c'est-à-dire l'opération  $i - 1$  si  $i > 1$ .

$x_{m+1}$  est l'instant de fin d'exécution de la dernière tâche, et donc l'instant de fin d'exécution de la commande.

Nous définissons encore la fenêtre  $F_{i,j}^k = [a_{i,j}^k, b_{i,j}^k]$  qui est ma  $k$ -ème période de disponibilité de la ressource  $j$  qui peut effectuer l'opération  $i$ .

Pour chaque opération  $i$ , après énumération des ressources  $j$  correspondantes et de leurs intervalles de disponibilité, nous interclassons les fenêtres  $F_{i,j}$  de la manière suivante :

- Les fenêtres sont classées dans l'ordre croissant des bornes inférieures  $a_{i,j}^k$ .
- Si deux fenêtres ont la même borne inférieure, nous les classons dans l'ordre croissant des bornes supérieures  $b_{i,j}^k$ .
- Si deux fenêtres ont la même borne inférieure et même borne supérieure, nous les classons dans un ordre quelconque.

Nous obtenons ainsi une suite unique de fenêtres de disponibilité classées par opération. La suite qui correspond à l'opération  $i$  est notée  $S_i = \{s_{i,r}\}_{r=1,2,\dots,q_i} = \{[\alpha_{i,r}, \beta_{i,r}]\}_{r=1,2,\dots,q_i}$ , où  $q_i$  est le nombre total de fenêtres après interclassement. Plus précisément :  $q_i = \sum_{j=1}^{n_i} f_{i,j}$ .

Bien entendu, nous conservons le lien entre chaque fenêtre de disponibilité et la ressource (ou l'ensemble de ressources) qui lui correspond. Notons que  $\beta_{i,q_i}$  est égal à l'infini. En fait,  $n_i$  fenêtres de disponibilité ont une borne supérieure infinie pour l'opération  $i$ .

Soulignons que la suite des fenêtres de disponibilité pour chaque opération doit être mise à jour après l'ordonnancement de chaque produit. En effet l'insertion d'un nouveau produit dans la gamme de fabrication peut réduire ou augmenter le nombre de fenêtres de disponibilité pour l'exécution d'une opération donnée.

### 6.2.3 Le problème

Le problème qui nous intéresse consiste à minimiser le temps de fin d'exécution de la commande en nous servant de la flexibilité offerte par la variabilité des temps opératoires.

Rappelons que ce problème a déjà été publié, comme indiqué en introduction. Ce paragraphe est un rappel.

En nous servant des notations précédentes, le problème s'exprime de la manière suivante :

Trouver  $x_1 < x_2 < \dots, x_m < x_{m+1}$  et une séquence de fenêtres de disponibilité notée  $F_d = \{s_{i,r_i}\}_{i=1,2,\dots,m}$ , avec  $r_i \in \{1, 2, \dots, q_i\}$ , qui minimise  $x_{m+1}$  et satisfait, pour  $i = 1, 2, \dots, m$ , aux contraintes suivantes :

$$\theta_i \leq x_{i+1} - x_i \leq \theta_i + \delta_i$$

$$x_i \geq \alpha_{i,r_i}$$

$$x_{i+1} \leq \beta_{i,r_i}$$

Ces trois contraintes peuvent se réécrire comme les inégalités (6.1) à (6.4).

$$x_{i+1} \geq \theta_i + x_i \tag{6.1}$$

Cette inégalité signifie que l'instant de début de l'opération  $i + 1$ , qui est l'instant de fin de l'opération  $i$ , est supérieur ou égal au temps de fin minimal de l'opération  $i$ .

$$x_i \geq \alpha_{i,r_i} \tag{6.2}$$

L'inégalité (6.2) indique que l'opération  $i$  ne doit pas commencer avant le début de la fenêtre  $S_{i,r_i}$ .

$$x_{i+1} \leq x_i + \theta_i + \delta_i \tag{6.3}$$

L'inégalité (6.3) indique que l'instant de début de l'opération  $i + 1$ , qui est la fin de l'opération  $i$ , est inférieur ou égal à l'instant de fin maximal de l'opération

$i$ .

$$x_{i+1} \leq \beta_{i,r_i} \quad (6.4)$$

Cette inégalité indique que l'opération  $i$  doit se terminer avant la fin de la fenêtre  $S_{i,r_i}$ .

Une séquence  $x_1 < x_2 < \dots, x_m < x_{m+1}$  est dite admissible si et seulement si elle vérifie les inégalités (6.1) à (6.4). Une séquence admissible est optimale si, de plus, elle minimise  $x_{m+1}$  : c'est une telle séquence que nous recherchons.

### 6.2.4 Un algorithme optimal

L'algorithme que nous présentons dans ce paragraphe conduit à une solution optimale. La démonstration figure dans [BPV83]. Nous supposons que les suites  $S_i = \{s_i\}_{i=1,2,\dots,m}$  de fenêtres associées aux opérations sont connues. Nous utilisons les notations introduites précédemment. L'algorithme que nous présentons sera nommé algorithme A dans la suite de cette partie.

#### L'algorithme de base (algorithme A)

1. Poser  $k_i = 1$  pour  $i = 1, 2, \dots, m$ .

Nous initialisons un compteur à 1 pour chaque opération. Notons que nous pourrions initialiser ces compteurs à d'autres valeurs que 1, ce qui obligerait le système à ne pas considérer les fenêtres de rang inférieur à ces valeurs. Nous utiliserons cette possibilité dans l'algorithme d'assemblage.

2. Construction de la séquence  $T = \{t_1, t_2, \dots, t_m, t_{m+1}\}$ .

- (a) Faire  $t_1 = \alpha_{1,k_1}$ .
- (b) Faire  $t_i = \max(\alpha_{i,k_i}, \theta_{i-1} + t_{i-1})$  pour  $i = 1, 2, \dots, m$ .
- (c) Poser  $t_{m+1} = \theta_m + t_m$ .

3. Construction de la séquence  $X = \{x_{m+1}, x_m, \dots, x_1\}$ .

- (a) Poser  $x_{m+1} = t_{m+1}$ .
- (b) Faire  $x_i = \max(t_i, x_{i+1} - \theta_i - \delta_i)$  pour  $i = m, m-1, \dots, 1$ .

4. Test.

- (a) Si  $x_{i+1} \leq \beta_{i,k_i}$ , pour  $i = 1, 2, \dots, m$ , alors la solution optimale est obtenue et l'algorithme se termine.
- (b) Sinon, pour tout  $i = \{1, 2, \dots, m\}$  tel que  $x_{i+1} > \beta_{i,k_i}$ , nous posons  $k_i = k_i + 1$  et nous retournons au point 2 de l'algorithme.

Cet algorithme converge et conduit à la solution optimale cherchée.

### 6.2.5 Des exemples d'application

Nous présentons ici deux exemples d'application. Le premier est un exemple simple dans lequel l'intérêt est que toutes les étapes de l'algorithme sont suivies pas à pas.

Le deuxième exemple reprend le précédent, mais avec des bornes supérieures des temps opératoires inférieures aux précédentes, afin de souligner l'importance de ce paramètre.

#### Exemple 1

L'exemple que nous traitons comporte 4 opérations notées  $R_1, R_2, R_3$  et  $R_4$ , visitées dans cet ordre par la commande qui se présente . Nous supposons que plusieurs ressources identiques sont en mesure d'effectuer chacune de ces opérations, et que le classement des fenêtres de disponibilité a été effectué et a conduit au résultat suivant :

$$R1. [\alpha_{1,1} = 0, \beta_{1,1} = 1], [\alpha_{1,2} = 4, \beta_{1,2} = 7], [\alpha_{1,3} = 10, \beta_{1,3} = 11],$$

$$[\alpha_{1,4} = 13, \beta_{1,4} = 17], [\alpha_{1,5} = 20, \beta_{1,5} = +\infty[$$

$$R2. [\alpha_{2,1} = 0, \beta_{2,1} = 2], [\alpha_{2,2} = 3, \beta_{2,2} = 5], [\alpha_{2,3} = 7, \beta_{2,3} = 14],$$

$$[\alpha_{2,4} = 15, \beta_{2,4} = \infty[$$

$$R3. [\alpha_{3,1} = 0, \beta_{3,1} = 6], [\alpha_{3,2} = 10, \beta_{3,2} = 11], [\alpha_{3,3} = 12, \beta_{3,3} = 18],$$

$$[\alpha_{3,4} = 19, \beta_{3,4} = +\infty[$$

$$R4. [\alpha_{4,1} = 2, \beta_{4,1} = 5], [\alpha_{4,2} = 8, \beta_{4,2} = 12], [\alpha_{4,3} = 16, \beta_{4,3} = \infty]$$

Ces fenêtres de disponibilité sont représentées dans la figure 6.1.

Enfin, les temps opératoires sont, avec les notations introduites précédemment :

$$(\theta_1 = 1, \delta_1 = 3), (\theta_2 = 2, \delta_2 = 2), (\theta_3 = 1, \delta_3 = 3), (\theta_4 = 1, \delta_4 = 2)$$

Nous appliquons maintenant l'algorithme A.

### Première itération

Nous posons  $k_1 = k_2 = k_3 = k_4 = 1$ . Cela signifie que nous cherchons à vérifier d'abord si les opérations peuvent être toutes exécutées dans les premières fenêtres de disponibilité qui les concernent.

Nous obtenons successivement :

$$t_1 = 0$$

$$t_2 = \max(0, 1 + 0) = 1$$

$$t_3 = \max(0, 2 + 1) = 3$$

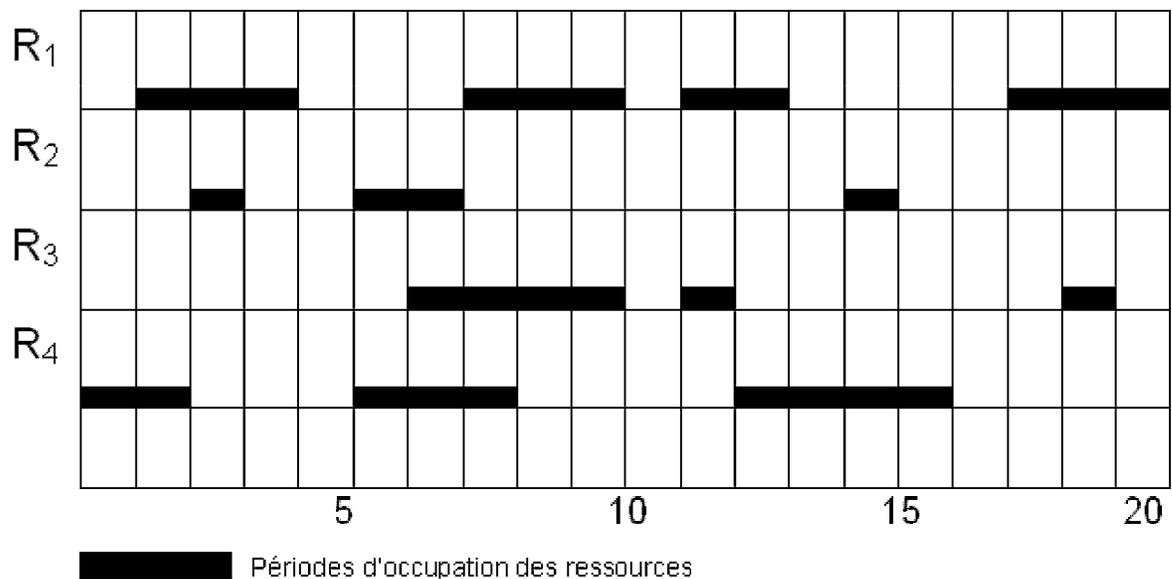


FIG. 6.1 – Occupation des ressources

$$t_4 = \max(2, 1 + 3) = 4$$

$$t_5 = 4 + 1 = 5$$

$$t_5 = 5$$

$$x_4 = \max(4, 5 - 1 - 2) = 4$$

$$x_3 = \max(3, 4 - 1 - 3) = 3$$

$$x_2 = \max(1, 3 - 2 - 2) = 1$$

$$x_1 = \max(0, 1 - 1 - 3) = 0$$

Comme  $x_3 > \beta_{2,1}$ , nous posons  $k_2 = k_2 + 1 = 2$

### Deuxième itération

Nous reprenons l'algorithme au niveau 2 avec :

$$k_1 = k_3 = k_4 = 1, k_2 = 2$$

Sans entrer dans le détail des calculs, nous obtenons successivement :

$$t_1 = 0, t_2 = 3, t_3 = 5, t_4 = 6, t_5 = 7$$

puis :

$$x_5 = 7, x_4 = 6, x_3 = 5, x_2 = 3, x_1 = 0$$

Comme  $x_5 > \beta_{4,1}$  et  $x_2 > \beta_{1,1}$  nous posons  $k_4 = k_4 + 1 = 2$  et  $k_1 = k_1 + 1 = 2$

et nous allons au point 2 de l'algorithme A.

### Troisième itération

$$k_3 = 1, k_1 = k_2 = k_4 = 2$$

Sans entrer dans le détail des calculs, nous obtenons successivement :

$$t_1 = 4, t_2 = 5, t_3 = 7, t_4 = 8, t_5 = 9$$

puis :

$$x_5 = 9, x_4 = 8, x_3 = 7, x_2 = 5, x_1 = 4$$

Comme  $x_4 > \beta_{3,1}$  et  $x_3 > \beta_{2,1}$  nous posons  $k_3 = k_3 + 1 = 2$  et  $k_2 = k_2 + 1 = 3$

et nous allons au point 2 de l'algorithme.

### Quatrième itération

$k_2 = 3, k_1 = k_3 = k_4 = 2$  Sans entrer dans le détail des calculs, nous obtenons successivement :

$$t_1 = 4, t_2 = 7, t_3 = 10, t_4 = 11, t_5 = 12$$

puis :

$$x_5 = 12, x_4 = 11, x_3 = 10, x_2 = 7, x_1 = 4$$

La solution est atteinte. Elle est représentée dans la figure 6.2. Pour permettre de mieux comprendre le graphique, nous avons distingué la partie du temps opératoire correspondant au temps opératoire minimal (rectangles blancs) de la partie opératoire supplémentaire (rectangle grisé). Le temps correspondant à la concaténation de ces deux rectangles reste, bien entendu, inférieur ou égal au temps opératoire maximal de l'opération.

Le graphique indique que le produit se termine à l'instant 12. Il a été démontré que ce temps de fin de fabrication est le minimum. En d'autres termes, l'algorithme que nous proposons est optimal si l'objectif est de minimiser le temps de fin de fabrication (encore appelé makespan dans le vocabulaire de l'ordonnancement, vocabulaire emprunté à la littérature anglo-saxonne).

Nous remarquons enfin que si cet algorithme minimise le makespan, il commence la première opération au plus tôt. Dans l'exemple étudié, la première opération aurait pu commencer à l'instant 6 et ne durer qu'une unité de temps, qui est son temps de fabrication minimal.

### Exemple 2

Dans cette partie on reprendra le même exemple que ci-dessus, après avoir modifié les  $\delta_i$  des opérations utilisant les ressources  $R_1, R_2, R_3$  et  $R_4$ .

Les temps opératoires sont :

$$(\theta_1 = 1, \delta_1 = 1), (\theta_2 = 2, \delta_2 = 1), (\theta_3 = 1, \delta_3 = 1), (\theta_4 = 1, \delta_4 = 1)$$

L'application de l'algorithme  $A$  à ce nouvel exemple donne la solution suivante :

$x_5 = 17, x_4 = 16, x_3 = 14, x_2 = 11, x_1 = 10$ . Elle est représentée dans la figure 6.3

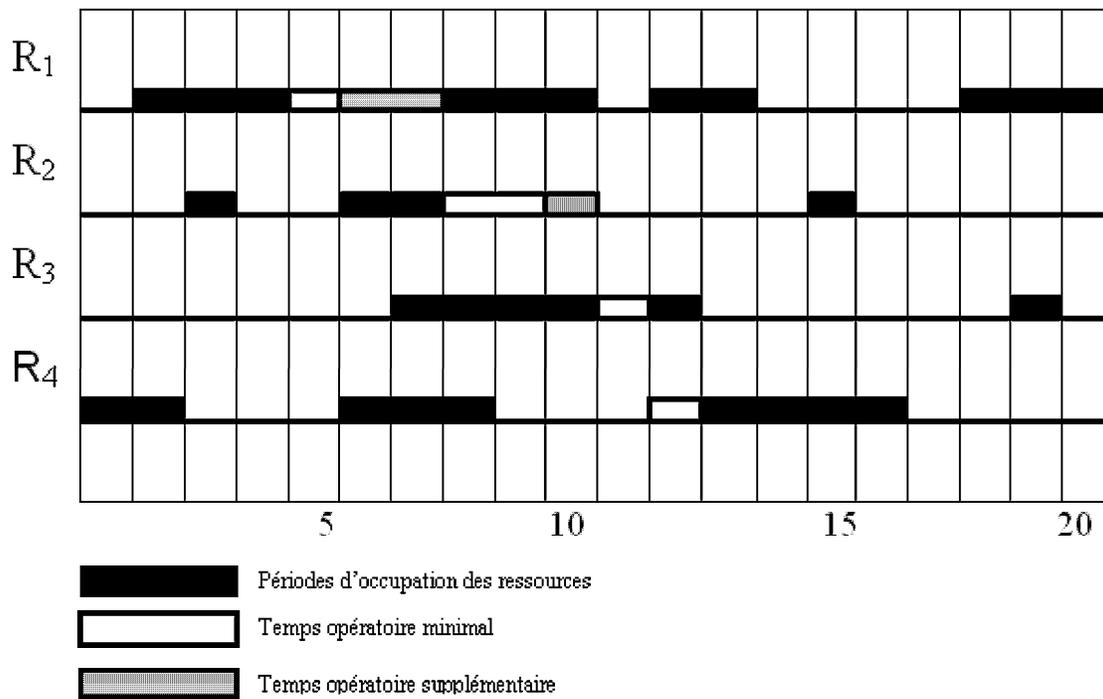


FIG. 6.2 – Ordonnancement du produit

**Remarques :**

- Le cycle de production est majoré par la quantité :

$$\sum_{i=1}^m \theta_i + \sum_{i=1}^m \delta_i$$

- La réduction des  $\delta_i$  a conduit à une réduction du cycle de fabrication ( $17 - 10 = 7$ ) au lieu de ( $13 - 4 = 9$ ), mais aussi à un début de fabrication plus tardif qui s'explique par une difficulté accrue de trouver des fenêtres au plus tôt qui permettent de ne pas stocker entre deux opérations.

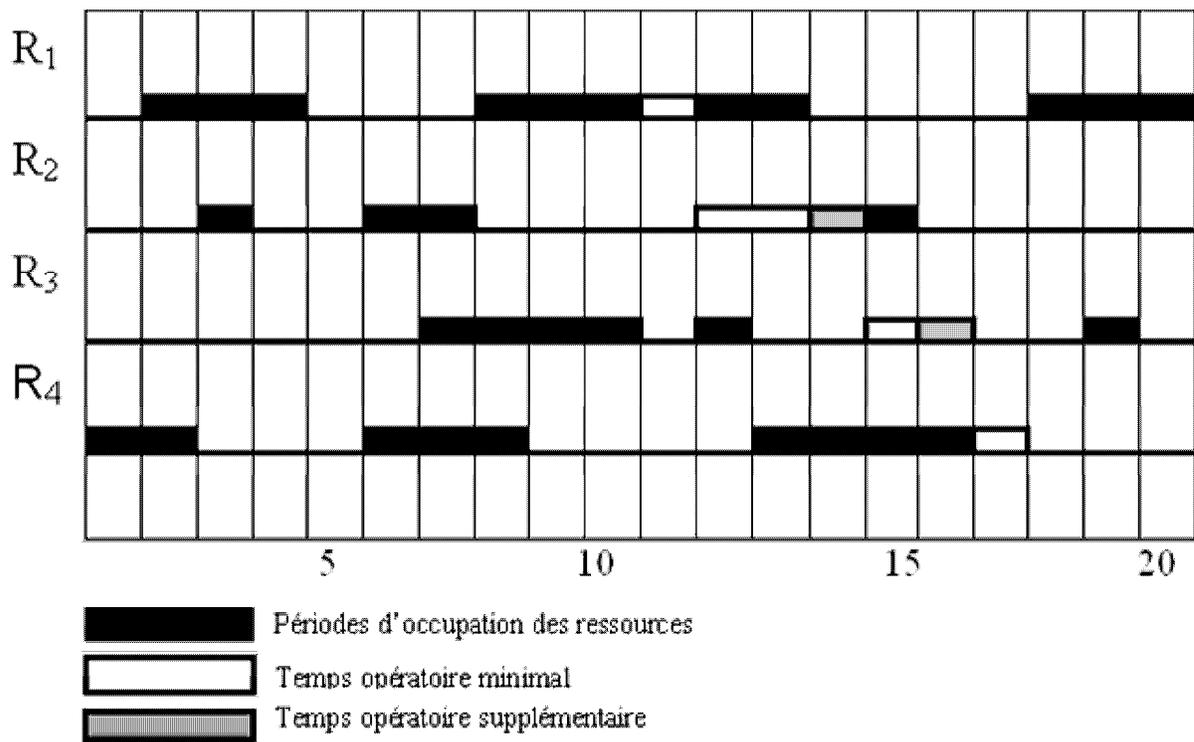


FIG. 6.3 – Ordonnancement du produit

### 6.2.6 Complexité de l'algorithme

Chaque fenêtre de disponibilité de chaque opération est testée au plus une fois (voir étape 4(b) de l'algorithme). Le nombre d'itération est donc borné par :

$$N = \sum_{i=1}^m q_i$$

A chaque itération on effectue  $2(m + 2)$  affectations,  $2(m + 1)$  opérations et  $2m$  calculs de maxima (étape 2 et 3). L'étape 4 se résume à  $m$  tests et l'étape 1 à  $m$  affectations.

Finalement, la complexité de l'ordre de  $8 \times m \times N$ , où  $m$  est le nombre d'opérations et  $N$  le nombre de fenêtres.

### 6.2.7 Gestion des en-cours

#### Le principe

Dans cette partie nous allons utiliser les résultats présentés dans le paragraphe 6.2.4 pour ordonnancer les tâches dès l'arrivée d'une commande tout en contrôlant les en-cours.

Nous cherchons à éviter que les ressources soient immobilisées au-delà du temps opératoire. Le temps supplémentaire  $\delta_i$  affecté à l'opération  $i$  existe toujours, mais il est transféré sur un stock afin de libérer la ressource qui peut alors être utilisée pour effectuer une autre opération.

Dans la pratique, nous remplaçons chaque ressource figurant dans un problème à temps opératoires contrôlables avec un temps opératoire appartenant à  $[\theta_i, \theta_i + \delta_i]$  par :

- Une activité dont le temps opératoire est  $\theta_i$  et qui se déroule sur les ressources chargées des opérations.
- Un stock qui représente une activité dont le temps opératoire appartient à l'intervalle  $[0, \delta_i]$ . Cette ressource a, en outre, la propriété d'avoir toujours une fenêtre de disponibilité égale à  $[0, \infty]$ .

Notre problème est donc un "no-wait & limited controllable processing time" dans lequel les stocks sont considérés comme des ressources dans lesquelles le temps opératoire se situe entre 0 et  $\delta_i$ . Par contre, les temps opératoires sur les machines sont fixes.

Désormais la gamme se définit par :

1. L'ordre d'exécution des opérations. Nous noterons  $O(j)$  la  $j$ -*ime* opération exécutée dans la gamme. Pour simplifier la présentation, nous supposons que toutes les opérations sont exécutées, sans quoi l'ensemble des opérations considérées est le sous-ensemble des opérations exécutées.
2. Nous noterons  $\theta_{O(j)}$  le temps d'exécution de l'opération  $O(j)$ .
3. Enfin,  $\delta_{O(j)}$  est le temps maximal d'attente après exécution de l'opération  $O(j)$ .

Dès lors, notre gamme se compose de  $2m$  opérations qui sont, alternativement, des opérations de transformation et des opérations de stockage. A partir de cette définition, l'algorithme  $A$  s'applique.

Dans cette approche nous sommes intéressés par la limitation des temps de stockage, mais nous ne nous intéressons pas du tout au volume de stockage. En d'autres termes, nous imposons aux produits semi-finis de ne pas séjourner entre deux opérations au-delà d'un temps donné, mais nous ne sommes pas limités par les surfaces de stockage. La conséquence de cette remarque est que, chaque fois que nous traitons un nouveau produit, nous supposons que la ressource de stockage qui suit chaque ressource chargée d'une opération est entièrement disponible.

Pour bien comprendre la modification que nous avons faite pour adapter l'algorithme  $A$  au cas où la flexibilité est reportée sur les stocks, nous l'illustrons par des exemples numériques.

### 6.2.8 Un exemple de système non automatisé

Considérons l'exemple présenté dans la figure 6.4. Il s'agit de 3 opérations. Deux machines sont disponibles pour l'opération 1, une machine pour l'opération 2, et enfin trois machines pour l'opération 3. Dans cet exemple l'ordre ainsi que les temps d'exécution des opérations sont fixés chaque fois que nous introduisons un nouveau produit. Les  $\delta_i$  sont tous égaux à 5 pour les opérations de stockage.

#### Fenêtres classées par opération

R1 :  $[\alpha_{1,1} = 0, \beta_{1,1} = 2], [\alpha_{1,2} = 0, \beta_{1,2} = 4], [\alpha_{1,3} = 7, \beta_{1,3} = 13]$   
 $[\alpha_{1,4} = 10, \beta_{1,4} = 16], [\alpha_{1,5} = 17, \beta_{1,5} = 21], [\alpha_{1,6} = 22, \beta_{1,6} = 25]$   
 $[\alpha_{1,7} = 30, \beta_{1,7} = 34], [\alpha_{1,8} = 33, \beta_{1,8} = +\infty[, [\alpha_{1,9} = 36, \beta_{1,9} = +\infty[$   
R2 :  $[\alpha_{2,1} = 0, \beta_{2,1} = 3], [\alpha_{2,2} = 9, \beta_{2,2} = 13], [\alpha_{2,3} = 20, \beta_{2,3} = 24]$   
 $[\alpha_{2,4} = 33, \beta_{2,4} = \infty[$   
R3 :  $[\alpha_{3,1} = 0, \beta_{3,1} = 3], [\alpha_{3,2} = 0, \beta_{3,2} = 7], [\alpha_{3,3} = 0, \beta_{3,3} = 9]$   
 $[\alpha_{3,4} = 12, \beta_{3,4} = 17], [\alpha_{3,5} = 13, \beta_{3,5} = 19], [\alpha_{3,6} = 19, \beta_{3,6} = 22]$   
 $[\alpha_{3,7} = 27, \beta_{3,7} = \infty[, [\alpha_{3,8} = 28, \beta_{3,8} = 30], [\alpha_{3,9} = 29, \beta_{3,9} = \infty[$

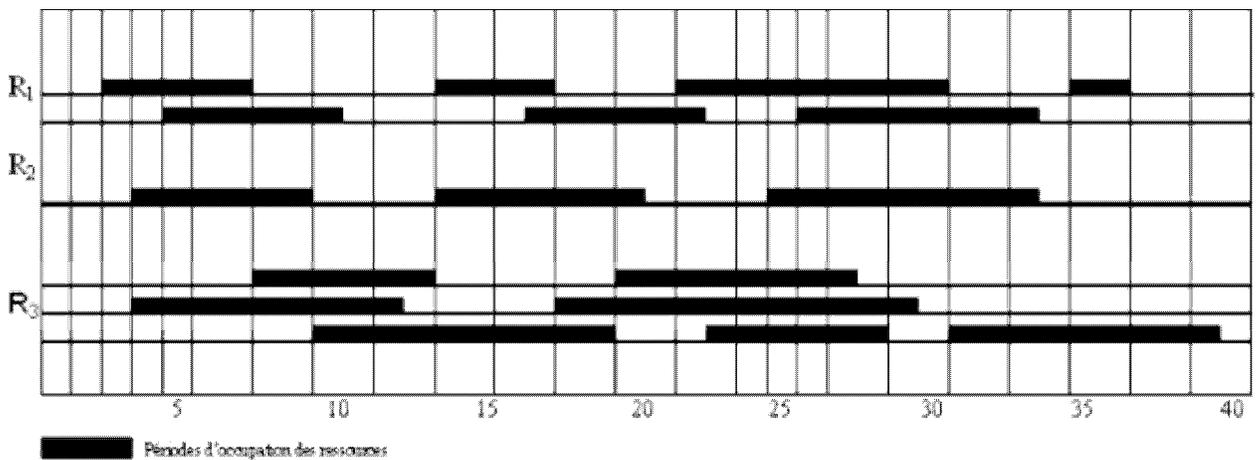


FIG. 6.4 – Occupation des ressources

$$[\alpha_{3,10} = 39, \beta_{3,10} = \infty[$$

**Premier Produit**

On insère le premier produit noté  $P1$  à partir du temps  $t = 0$ . Il passera successivement par les opération 1,2 puis 3 ( $\theta_1 = 2$ ), ( $\theta_2 = 4$ ), ( $\theta_3 = 3$ ).

La solution est la suivante  $x_4 = 16, x_3 = 13, x_2 = 9, x_1 = 2$ . elle est représentée dans la figure 6.5. Nous représentons les périodes de stockage par des carreaux grisés en-dessous de l'axe correspondant à l'opération qui précède le stockage.

**Deuxième Produit**

Mise à jour des fenêtres :

$$R1 : [\alpha_{1,1} = 0, \beta_{1,1} = 2], [\alpha_{1,2} = 0, \beta_{1,2} = 2], [\alpha_{1,3} = 7, \beta_{1,3} = 13]$$

$$[\alpha_{1,4} = 10, \beta_{1,4} = 16], [\alpha_{1,5} = 17, \beta_{1,5} = 21], [\alpha_{1,6} = 22, \beta_{1,6} = 25]$$

$$[\alpha_{1,7} = 30, \beta_{1,7} = 34], [\alpha_{1,8} = 33, \beta_{1,8} = +\infty[, [\alpha_{1,9} = 36, \beta_{1,9} = +\infty[$$

$$R2 : [\alpha_{2,1} = 0, \beta_{2,1} = 3], [\alpha_{2,2} = 20, \beta_{2,2} = 24], [\alpha_{2,3} = 33, \beta_{2,3} = \infty[$$

$$R3 : [\alpha_{3,1} = 0, \beta_{3,1} = 3], [\alpha_{3,2} = 0, \beta_{3,2} = 7], [\alpha_{3,3} = 0, \beta_{3,3} = 9]$$

$$[ \alpha_{3,4} = 12, \beta_{3,4} = 13], [\alpha_{3,5} = 13, \beta_{3,5} = 19], [\alpha_{3,6} = 16, \beta_{3,6} = 17]$$

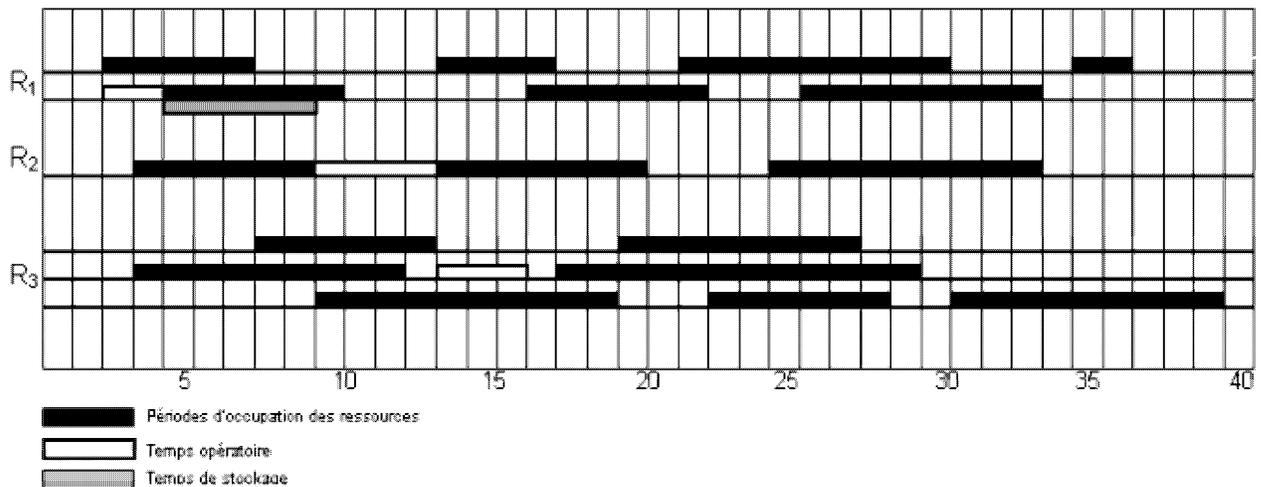


FIG. 6.5 – Ordonnancement du premier produit P1

$$[\alpha_{3,7} = 19, \beta_{3,7} = 22], [\alpha_{3,8} = 27, \beta_{3,8} = \infty], [\alpha_{3,9} = 28, \beta_{3,9} = 30]$$

$$[\alpha_{3,10} = 29, \beta_{3,10} = \infty], [\alpha_{3,11} = 39, \beta_{3,10} = \infty]$$

Nous insérons le nouveau produit  $P2$  à partir de l'instant  $t = 0$  après avoir ordonnancé le produit  $P1$ .  $P2$  passera successivement par les opérations 3, 1 puis 2.  $(\theta_3 = 1, \delta_3 = 0), (\theta_1 = 4, \delta_1 = 0), (\theta_2 = 3, \delta_2 = 0)$ .

La solution est la suivante  $x_4 = 23, x_3 = 20, x_2 = 11, x_1 = 5$ . Elle est représentée dans la figure 6.5.

La figure 6.6 représente les fenêtres de disponibilité après l'insertion du produit  $P2$ .

### Troisième Produit

A ce stade nous allons changer la base du temps en nous plaçant à l'instant  $t = 5$ .

Le changement de la base du temps traduit l'évolution dans le temps. Cependant, une mise à jour des fenêtres de disponibilité des ressources est nécessaire. En nous plaçant au temps  $t_0$ , nous retranchons  $t_0$  à toutes les bornes inférieures et supérieures des intervalles de disponibilité des ressources et nous

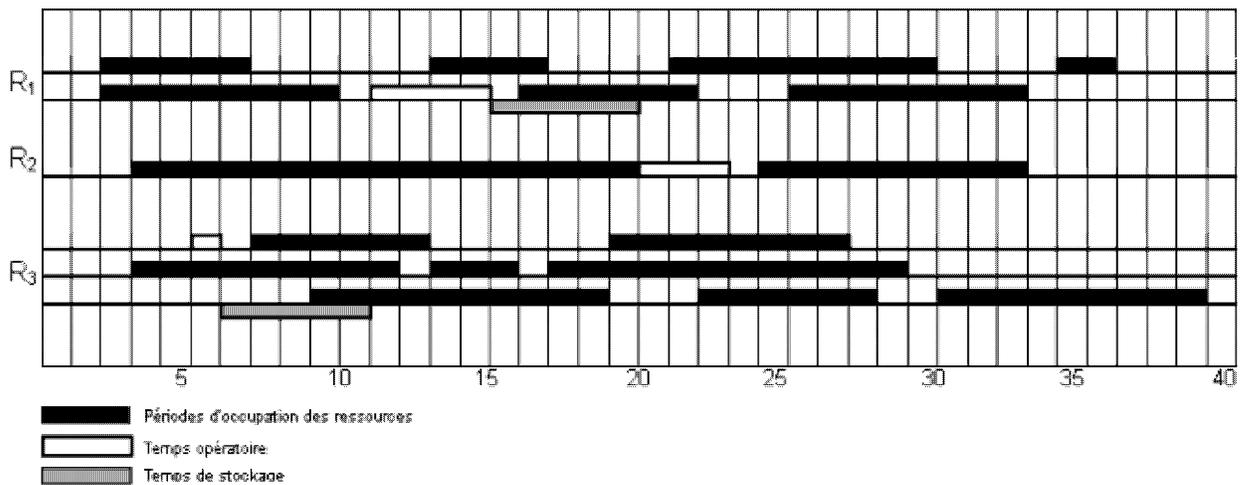


FIG. 6.6 – Ordonnancement du produit P2

affectons 0 aux nombres négatifs.

Algorithme de mise à jour des fenêtres de disponibilité :

Pour  $i = 1, 2, \dots, m, f = 0$

1. Faire  $[\alpha_{i,r}, \beta_{i,r}]_{r=1,2,\dots,q_i} = [\alpha_{i,r}, \beta_{i,r} - t_0]_{r=1,2,\dots,q_i}$ .
2. Si  $\alpha_{i,r} < 0$  et  $\beta_{i,r} > 0$   $\alpha_{i,r} = 0$
3.  $q_i = q_i - f$ .
4. Classer les  $[\alpha_{i,r}, \beta_{i,r}]_{r=1,2,\dots,q_i}$

Retournons à notre exemple, c'est-à-dire à la figure 6.10 après déplacement du temps à +5. Les nouvelles fenêtres de disponibilité après classement sont :

R1 :  $[\alpha_{1,1} = 2, \beta_{1,1} = 8], [\alpha_{1,2} = 5, \beta_{1,2} = 6], [\alpha_{1,3} = 10, \beta_{1,3} = 11]$   
 $[\alpha_{1,4} = 12, \beta_{1,4} = 16], [\alpha_{1,5} = 17, \beta_{1,5} = 20], [\alpha_{1,6} = 25, \beta_{1,6} = 29]$   
 $[\alpha_{1,8} = 28, \beta_{1,8} = \infty], [\alpha_{1,9} = 31, \beta_{1,9} = +\infty[$   
R2 :  $[\alpha_{2,1} = 18, \beta_{2,1} = 19], [\alpha_{2,4} = 18, \beta_{2,4} = \infty[$   
R3 :  $[\alpha_{3,1} = 0, \beta_{3,1} = 4], [\alpha_{3,2} = 1, \beta_{3,2} = 2], [\alpha_{3,3} = 7, \beta_{3,3} = 8]$   
 $[\alpha_{3,4} = 8, \beta_{3,4} = 14], [\alpha_{3,5} = 11, \beta_{3,5} = 12], [\alpha_{3,6} = 14, \beta_{3,6} = 17]$   
 $[\alpha_{3,7} = 22, \beta_{3,7} = \infty], [\alpha_{3,8} = 23, \beta_{3,8} = 25], [\alpha_{3,9} = 24, \beta_{3,9} = \infty[$   
 $[\alpha_{3,10} = 34, \beta_{3,10} = \infty[$

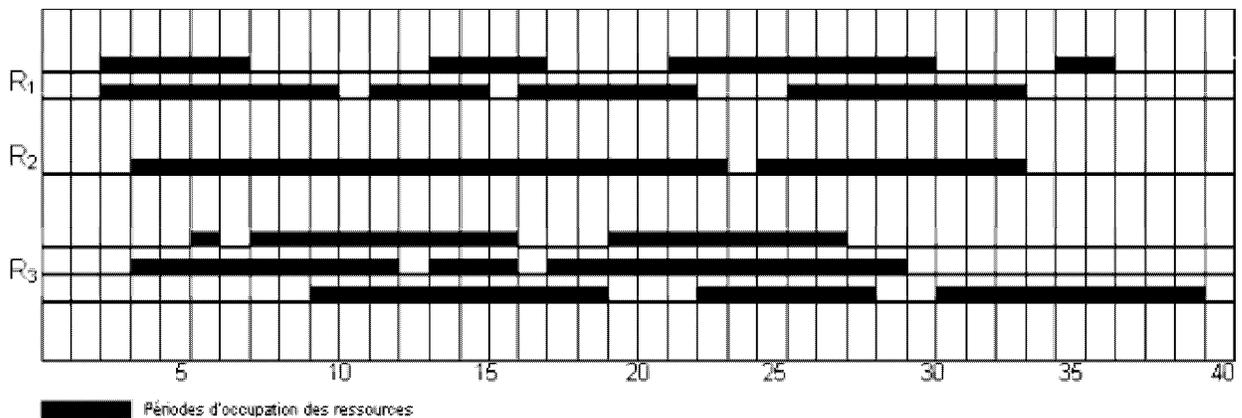


FIG. 6.7 – Occupation des ressources après l'ordonnement de P2

Le troisième produit  $P3$  passera successivement par les opérations 1,2, puis 3.  $(\theta_1 = 4, \delta_1 = 0), (\theta_2 = 2, \delta_2 = 0), (\theta_3 = 2, \delta_3 = 0)$ .

La figure 6.9 représente la solution.

**Quatrième Produit**

Après la mise à jour des fenêtres, nous insérons le produit  $P4$ . On suppose qu'il passe successivement par les opérations 2,3 puis 1  $(\theta_2 = 1, \delta_2 = 0), (\theta_3 = 3, \delta_3 = 0), (\theta_1 = 3, \delta_1 = 0)$ . La solution est la suivante :



FIG. 6.8 – Occupation des ressources après mise à jour

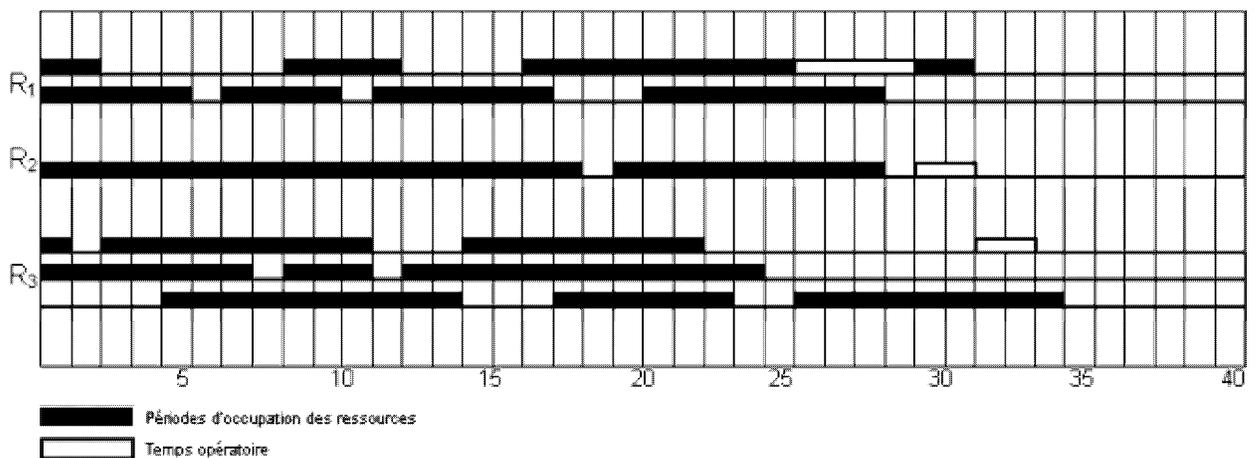


FIG. 6.9 – Ordonnancement du produit P3

$x_4 = 31, x_3 = 28, x_2 = 22, x_1 = 18$ . Elle est représentée dans la figure 6.11.

**Cinquième Produit**

Après la mise à jour des fenêtres, nous insérons le produit  $P5$ . On suppose qu'il passe successivement par les opérations 3,2 puis 1 ( $\theta_3 = 3, \delta_3 = 0$ ), ( $\theta_2 = 3, \delta_2 = 0$ ), ( $\theta_1 = 2, \delta_1 = 0$ ). La solution est la suivante :

$x_4 = 36, x_3 = 34, x_2 = 31, x_1 = 24$ . Elle est représentée dans la figure 6.10.

**Sixième Produit**

Le produit  $P6$  est inséré au temps  $t = 15$  (en se référant à l'origine du temps avant l'insertion du produit 1). Les nouvelles fenêtres de disponibilité sont :

- R1 :  $[\alpha_{1,1} = 0, \beta_{1,1} = 1], [\alpha_{1,2} = 2, \beta_{1,2} = 6], [\alpha_{1,3} = 7, \beta_{1,3} = 10]$   
 $[\alpha_{1,4} = 21, \beta_{1,4} = 24], [\alpha_{1,5} = 21, \beta_{1,5} = \infty], [\alpha_{1,6} = 26, \beta_{1,6} = \infty]$
- R2 :  $[\alpha_{2,1} = 18, \beta_{2,1} = 19], [\alpha_{2,4} = 24, \beta_{2,4} = \infty[$
- R3 :  $[\alpha_{3,1} = 0, \beta_{3,1} = 4], [\alpha_{3,2} = 1, \beta_{3,2} = 2], [\alpha_{3,3} = 4, \beta_{3,3} = 7]$   
 $[\alpha_{3,4} = 13, \beta_{3,4} = 15], [\alpha_{3,5} = 15, \beta_{3,5} = 21], [\alpha_{3,6} = 17, \beta_{3,6} = \infty[$   
 $[\alpha_{3,7} = 23, \beta_{3,7} = \infty], [\alpha_{3,8} = 24, \beta_{3,8} = \infty[.$

Elles sont représentées dans la figure 6.12.

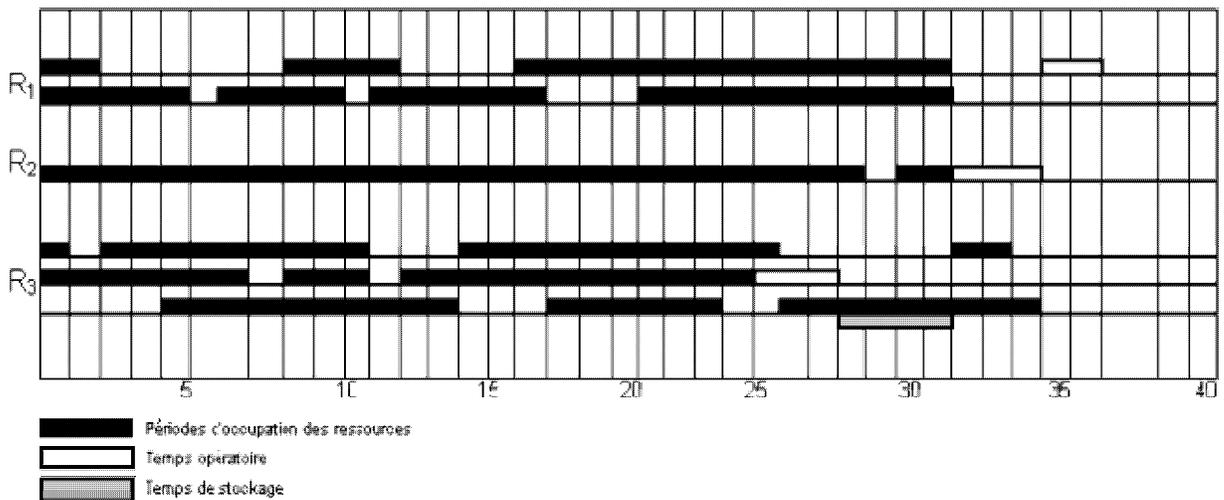


FIG. 6.10 – Ordonnancement du produit P4

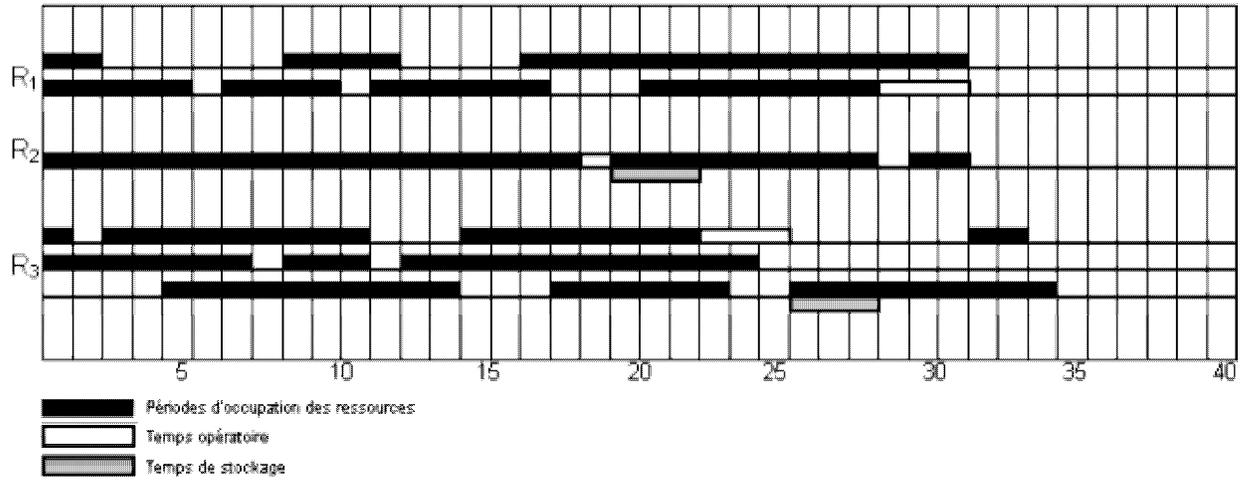


FIG. 6.11 – Ordonnancement du produit P5

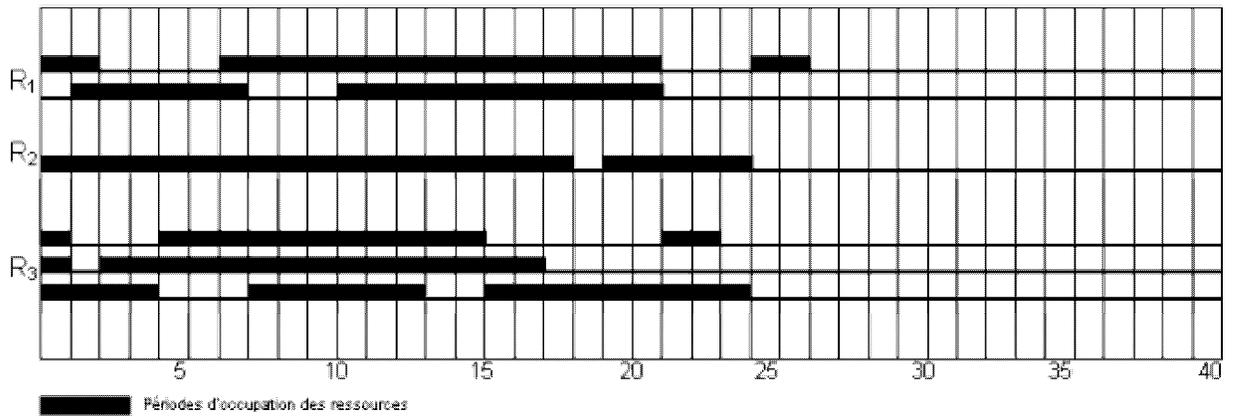


FIG. 6.12 – Occupation des ressources après mise à jour

La produit  $P6$  passera successivement par les opérations 1,3 et 2 ( $\theta_1 = 1, \delta_1 = 0$ ), ( $\theta_2 = 1, \delta_2 = 0$ ), ( $\theta_3 = 1, \delta_3 = 0$ ). La solution est  $x_4 = 19, x_3 = 18, x_2 = 13, x_1 = 7$ . Elle est représentée dans la figure 6.13.

### 6.2.9 Un exemple de système automatisé

Dans le cas d'un système automatisé où les manipulations sont à la charge des robots, le stockage est plus contraint que dans le cas précédent. Nous supposons alors qu'à chaque ressource ou ensemble de ressources chargées d'une opération succède un ensemble de ressources de stockage, chacune de ces ressources de stockage ne pouvant prendre en charge qu'un produit au maximum. Nous contrôlons donc non seulement la durée du stockage entre deux opérations successives, mais aussi le nombre maximal d'unités de produits semi-finis qui peuvent être stockés simultanément après chaque opération.

Bien entendu, cela conduit à associer un axe temporel à chaque place de stockage, et à gérer les fenêtres de disponibilité sur ce dernier axe comme nous le faisons pour les axes correspondant aux opérations. D'où une complexité plus grande que dans le cas de systèmes non automatisés.

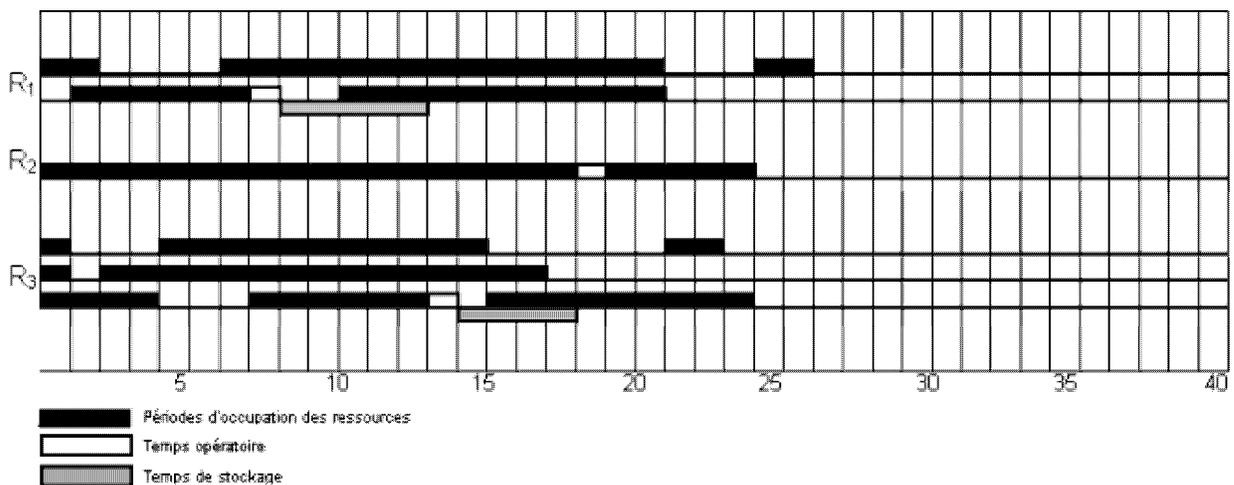


FIG. 6.13 – Ordonnancement du produit P6

### 6.2.10 En Conclusion

La méthode que nous venons de présenter cherche à utiliser d'abord les fenêtres de disponibilité les plus proches dans le temps. Cela assure une bonne utilisation des ressources, car les fenêtres de disponibilité sont utilisées avant que l'écoulement du temps les rejette dans le passé. Comme nous l'avons vu, le taux d'utilisation des ressources croît avec les  $\delta$  affectés aux ressources de stockage, mais au prix d'un en-cours moyen qui augmente parallèlement.

La bonne stratégie est donc d'affecter des valeurs faibles aux  $\delta$  lorsque le système est peu chargé, et d'augmenter ces valeurs lorsque la charge du système augmente. Encore faut-il se souvenir qu'un  $\delta$  faible a tendance à conduire à un makespan important.

## 6.3 Ordonnancement temps réel des processus d'assemblage

Pour faciliter les explications, nous divisons ce paragraphe en deux parties principales. La première partie traite des problèmes d'assemblage simples. La seconde partie généralise la précédente.

### 6.3.1 Ordonnancement d'un processus d'assemblage simple

#### Définition

Un processus d'assemblage simple est un processus dans lequel plusieurs composantes sont obtenues en appliquant des processus linéaires (job-shop), puis assemblées pour obtenir un produit final. Typiquement, un processus d'assemblage simple se termine par une opération d'assemblage, et cette opération est unique pour chaque produit. La figure 6.14 donne la structure d'un processus d'assemblage simple.

Les processus linéaires où séquence d'opérations sont au nombre de  $L$  et désignées par  $l = 1, 2, \dots, L$ . Chaque processus linéaire  $l \in \{1, 2, \dots, L\}$  compte

$m_l$  opérations, ou  $l$  représente l'indice de la séquence considérée.

### Notations

Nous considérons un processus d'assemblage simple, que nous notons  $A$ , composé de  $L$  composantes, chaque composante étant obtenue par application d'une séquence d'opérations. Chaque séquence  $l, 1 \leq l \leq L$ , est composée de  $m_l$  opérations que nous notons  $O_{l,i}$   $i = 1, 2, \dots, m_l$ . Le temps d'exécution d'une opération  $O_{l,i}$  est fixe. Il est noté  $\theta_{l,i}$ . Le temps d'exécution de l'opération d'assemblage est noté  $\theta_A$ .

Un en-cours est considéré comme subissant une opération de stockage d'une durée variant de 0 à  $\delta_{l,i}$  après l'opération  $O_{l,i}$ .  $\delta_{l,i}$  est donné par l'utilisateur et représente la durée maximale de stockage à la sortie de l'opération  $O_{l,i}$ . Les  $\delta_{l,i}$  sont un moyen de contrôler les en-cours; plus les  $\delta_{l,i}$  sont grands, plus le niveau des en-cours est important, et meilleure est l'utilisation des ressources.

Soulignons que plusieurs ressources identiques peuvent être affectées à une

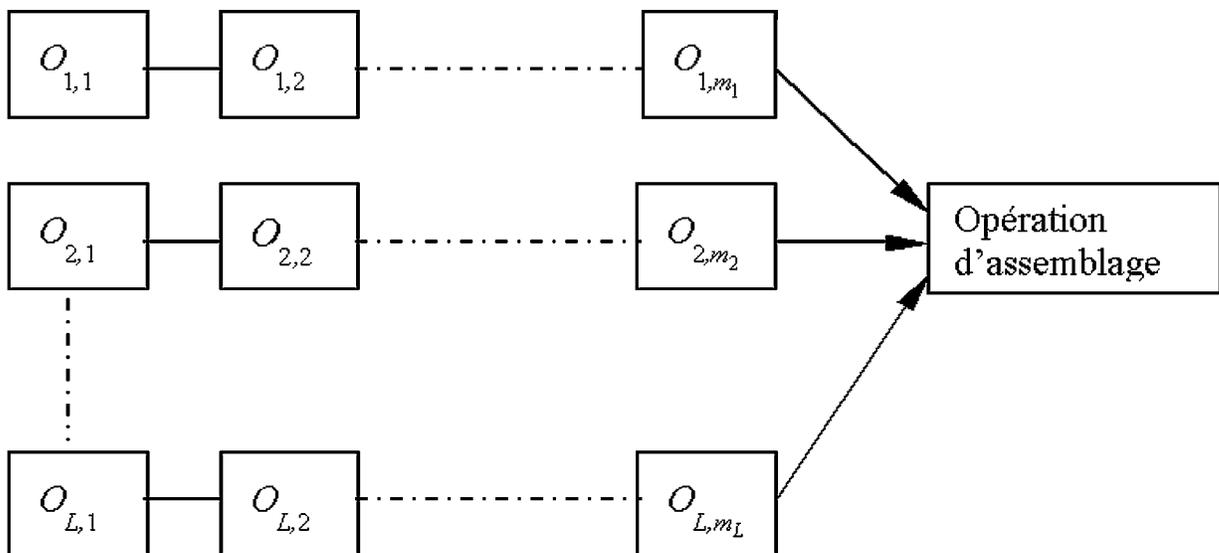


FIG. 6.14 – Processus d'assemblage simple

opération, y compris à l'opération d'assemblage. De plus, une ressource n'est pas utilisée pour exécuter deux opérations différentes sur un même produit. Chaque opération a ses propres fenêtres de disponibilité. Les fenêtres de disponibilité sont obtenues à partir de celles des ressources affectées à chaque opération. Notons  $q_{l,i}$  le nombre total de périodes de disponibilité des ressources pour l'opération  $O_{l,i}$ . Notons que l'opération  $O_{l,i}$  ne peut être exécutée que dans une seule fenêtre. Les fenêtres des ressources disponibles pour exécuter une même opération sont classées dans l'ordre croissant de leurs instants de début et leurs instants de fin comme décrit dans la partie 6.2.2. Nous obtenons ainsi une suite unique de fenêtres de disponibilité classées pour chaque opération. La suite qui correspond à l'opération  $O_{l,i}$  est notée  $S_{l,i} = \{s_{l,i,r_{l,i}}\}_{r_{l,i}=1,2,\dots,q_{l,i}} = \{[\alpha_{l,i,r_{l,i}}, \beta_{l,i,r_{l,i}}]\}_{r_{l,i}=1,2,\dots,q_{l,i}}$ . Rappelons que la borne supérieure de la dernière fenêtre de chaque ressource, y compris celle qui correspond à l'opération d'assemblage, est infinie. Ainsi, nous avons la garantie de trouver une solution réalisable. Nous notons respectivement  $\alpha_{A,r_A}$  et  $\beta_{A,r_A}$   $r_A \in \{1, 2, \dots, q_A\}$  le début et la fin des fenêtres susceptibles de recevoir l'opération d'assemblage  $A$ ,  $q_A$  représente le nombre total d'intervalles de disponibilité des ressources qui peuvent réaliser  $A$ .

### Position du problème

Le problème qui nous intéresse consiste à minimiser le temps de fin d'exécution de l'opération d'assemblage  $A$ .

L'idée que nous avons utilisée pour y parvenir est de compléter chacun des processus linéaire par l'opération d'assemblage, considérée comme faisant partie de la séquence. En conséquence, nous appellerons parfois  $O_{l,m_l+1}$  l'opération d'assemblage lorsqu'elle est ajoutée au processus linéaire  $l \in \{1, 2, \dots, L\}$ . Alors  $\theta_{l,m_l+1} = \theta_A$  désignera le temps opératoire et  $[\alpha_{l,m_l+1,r_{l,m_l+1}}, \beta_{l,m_l+1,r_{l,m_l+1}}]$  désignera la fenêtre associée à  $O_{l,m_l+1}$ . Nous savons résoudre le problème pour

chacune des séquences linéaires lorsque nous avons fixé la première fenêtre à examiner pour chacune des opérations (par l'intermédiaire des  $\theta_{l,i}$  et de l'algorithme qui généralise l'algorithme  $A$  au cas de stockages contrôlés (voir paragraphe 6.2.7)). L'objectif est alors de faire en sorte que l'instant de début de la dernière opération de chacune des séquences (c'est-à-dire l'opération d'assemblage  $A$ ), soit le même.

Nous notons  $x_{l,i}$ ,  $l = 1, 2, \dots, L$  et  $i = 1, 2, \dots, m_l$  l'instant de début d'exécution de l'opération  $O_{l,i}$ . Le problème est donc de trouver  $x_{l,i}$  pour  $l = 1, 2, \dots, L$ ,  $i = 1, 2, \dots, m_l$  et  $r_{l,i} \in \{1, 2, \dots, q_{l,i}\}$ .

Afin de minimiser  $x_A$ , les contraintes suivantes doivent être satisfaites :

$$x_{l,i} \geq \alpha_{l,i,r_{l,i}} \quad (6.5)$$

L'inégalité (6.5) indique que l'opération  $O_{l,i}$  ne doit pas commencer avant le début de la fenêtre  $S_{l,i,r_{l,i}}$ .

$$x_{l,i+1} = x_{l,i} + \theta_{l,i} + \delta_{l,i} \leq \beta_{l,i,r_{l,i}} \quad (6.6)$$

L'inégalité (6.6) indique que l'instant de début de l'opération  $O_{l,i+1}$  qui marque la fin de l'opération  $O_{l,i}$  doit être inférieur à la borne supérieure de  $S_{l,i,r_{l,i}}$ .

$$x_{l,i+1} - x_{l,i} \leq \theta_{l,i} + \delta_{l,i} \quad (6.7)$$

L'inégalité (6.7) indique que l'instant de début de l'opération  $O_{l,i+1}$ , qui est la fin de l'opération  $i$ , est inférieur ou égal à la durée maximale de l'opération  $O_{l,i}$ .

$$x_{l,m_l} + \theta_{l,m_l} + \delta_{l,m_l} = x_A \text{ pour } l = 1, 2, \dots, L \quad (6.8)$$

L'égalité (6.8) indique que le début de l'opération d'assemblage doit coïncider avec l'instant de fin d'exécution de toutes les opération  $O_{l,m_l}$  qui sont les dernières des séquences  $l$ .

$$x_A + \theta_A \leq \beta_{A,r_A} \quad r_A = 1, 2, \dots, q_A \quad (6.9)$$

L'inégalité (6.9) indique que l'opération d'assemblage  $A$  doit se terminer avant la fin de la fenêtre  $S_{A,r_A,i}$ .

$$x_A \geq \alpha_{A,r_A} \quad r_A = 1, 2, \dots, q_A \quad (6.10)$$

Enfin l'inégalité (6.10) indique que l'instant de début de l'opération  $A$  doit être supérieur à la borne inférieure de  $S_{A,r_A}$ .

### Présentation de l'algorithme

Nous présentons l'algorithme qui s'applique à un processus d'assemblage simple. L'algorithme que nous utilisons considère  $L$  fabrications linéaires, la fabrication linéaires  $l = 1, 2, \dots, L$  étant obtenue en ajoutant à la séquence  $l$  l'opération d'assemblage  $A$ .

On traite chacune de ces fabrications linéaires à l'aide de l'algorithme présenté dans le paragraphe 6.2.7. Trois cas peuvent alors se présenter :

1. L'opération d'assemblage n'est pas exécutée dans la même fenêtre pour les  $L$  fabrications linéaires complétées par l'opération d'assemblage.
2. L'opération d'assemblage est exécutée dans la même fenêtre pour les  $L$  fabrications linéaires, mais les temps de début de cette opération d'assemblage ne sont pas les mêmes pour chaque fabrication linéaire.
3. L'opération d'assemblage débute au même instant pour les  $L$  fabrications linéaires.

Dans le premier cas, on relance les calculs en imposant à l'opération d'assemblage des  $L$  fabrications linéaires de ne pas se situer avant la fenêtre de plus haut rang obtenue dans les calculs précédents. Ceci est obtenu en initialisant convenablement les paramètres  $k_i$  de l'algorithme.

Dans le second cas, on relance les calculs après avoir remplacé la borne inférieure de la fenêtre concernée par le maximum des temps de début de l'opération d'assemblage. Dans le troisième cas, la solution optimale est obtenue.

Nous donnons ci-dessous une présentation formelle de l'algorithme à appliquer à un processus d'assemblage simple.

**Algorithme S : Processus d'assemblage simple**

1. Sélection de la première fenêtre dans laquelle peut s'exécuter l'opération d'assemblage.

*Pour sélectionner cette fenêtre, nous calculons le temps minimum nécessaire pour fabriquer chacune des composantes, et nous retenons le plus grand de ces temps, que nous notons  $T$ .*

$$T = \text{Max}_{l=1,2,\dots,L} \sum_{i=1}^{m_l} \theta_{l,i}$$

*La fenêtre cherchée est la première fenêtre  $[\alpha_{A,r_A}, \beta_{A,r_A}]$  telle que*

$$\text{Max}(T, \alpha_{A,r_A}) + \theta_A \leq \beta_{A,r_A}, r_A \in \{1, 2, \dots, q_A\}$$

*Soit  $r_A^*$  cette valeur.*

2. A chaque composante, nous ajoutons l'opération d'assemblage en fin de séquence.

*La composante  $l \in \{1, 2, \dots, L\}$  est donc désormais définie par les opérations  $\{O_{l,i}\}_{i=1,2,\dots,m_l,m_l+1}$ , où l'opération  $O_{l,m_l+1}$  a une durée  $\theta_A$  :  $\theta_{l,m_l+1} = \theta_A$ .*

*Nous sommes donc en face de  $L$  fabrications linéaires que nous pouvons traiter à l'aide de l'algorithme A. Bien entendu, pour chacun des traitements, nous initialiserons  $k_{m_l+1}$ , rang de la première fenêtre à considérer pour l'opération d'assemblage, à  $r_A^*$ .*

$$k_{m_l+1} = r_A^*.$$

Pour chaque opération de chaque composante, nous obtenons :

- (a) la fenêtre de disponibilité où cette opération doit être exécutée,
- (b) l'instant de début et l'instant de fin de cette opération,
- (c) le makespan, qui est l'instant de fin de la dernière opération.

Nous notons  $s^{r_{m_l+1}} = [\alpha_{l,m_l+1,m_l+1}, \beta_{l,m_l+1,r_{m_l+1}}]$ ,  $l \in \{1, 2, \dots, L\}$ , la fenêtre dans laquelle devrait être exécutée l'opération d'assemblage pour la composante  $l$ .  $r_{m_l+1}$  est le rang de la fenêtre de disponibilité en question.

3. Si  $r_{m_{l_1}+1} \neq r_{m_{l_2}+1}$  pour au moins deux composantes  $l_1, l_2 \in \{1, 2, \dots, L\}$ , alors nous calculons :

$$n^* = \text{Max}_{l=1,2,\dots,L}(r_{m_l+1})$$

et nous relançons le calcul en posant

$$k_{m_l+1} = n^* \quad \text{pour } l = 1, 2, \dots, L$$

4. Si tous les  $r_{m_l+1}$  sont égaux pour  $l = 1, 2, \dots, L$  c'est-à-dire si les opérations d'assemblage se déroulent dans la même fenêtre pour toutes les composantes, alors nous augmentons au maximum le temps de début de l'opération d'assemblage pour chaque composante en jouant sur les temps de stockage, sans toutefois quitter la fenêtre commune.

Soient :

- $h_l, l \in \{1, 2, \dots, L\}$  l'instant minimal de début de l'opération d'assemblage dans la composante  $l$ . C'est l'instant donné pour l'algorithme.
- $H_l, l \in \{1, 2, \dots, L\}$  l'instant maximal de début de l'opération d'assemblage dans la composante  $l$ . C'est l'instant que nous venons de calculer.

L'algorithme est donné plus loin.

5. Calculer :

$$S = \cap_{l=1}^L [h_l, H_l]$$

Si  $S = \emptyset$ , nous posons  $k_{m_l+1} = k_{m_l} + 1$  pour  $l = 1, 2, \dots, L$  et nous recommençons les calculs en 2.

Si  $S \neq \emptyset$ , faire débiter l'opération d'assemblage en  $h^* = \text{Min}_{l=1,2,\dots,L} h_l$  et ajuster les débuts des autres opérations en conséquence. L'algorithme d'ajustement est donné dans le paragraphe suivant. Fin des calculs.

### Algorithmes de calcul de $H_l$ et d'ajustement

Dans ce paragraphe, nous montrons comment déterminer les instants  $H_l, l = 1, 2, \dots, L$ , et comment ajuster les temps de début de toutes les opérations.

Considérons un processus linéaire  $l \in \{1, 2, \dots, L\}$  et supposons que nous lui ayons appliqué l'algorithme de minimisation du makespan introduit précédemment. Soient  $[\alpha_{l,i,r_{l,i}}, \beta_{l,i,r_{l,i}}], i = 1, 2, \dots, m_l + 1$ , les fenêtres de disponibilité occupées par les opérations  $i$  du processus linéaire  $l$ . Soit également  $t_{l,i}$  le temps de début de l'opération  $i$  de  $l$ . Le temps de fin de l'opération  $i$  est, bien entendu,  $t_{l,i} + \theta_{l,i}$  compte tenue de l'existence de stockage à durée limitée entre deux opérations successives. Avec les notations utilisées dans les paragraphes précédents,  $h_l = t_{l,m_l+1} + \theta_{l,m_l+1}$ , où  $t_{l,m_l+1}$  et  $\theta_{l,m_l+1}$  sont respectivement l'instant de début et la durée de l'opération d'assemblage qui prolonge le processus linéaire initial  $l$ .

Notons  $\delta_{l,i}$  le temps maximal de stockage à la sortie de l'opération  $i$  du processus linéaire  $l$ , et  $\delta_{l,i}^* \leq \delta_{l,i}$  le temps réel de stockage donné par l'algorithme précédent. Posons  $Z_{l,i} = \delta_{l,i} - \delta_{l,i}^*$ .

Partant de la solution dont nous disposons, et en notant  $x_{l,i}$  l'accroissement que nous souhaitons donner au stockage après exécution de l'opération  $i$  du processus linéaire  $l$ , nous voyons que les inégalités suivantes doivent être vérifiées :

$$t_{l,1} + \sum_{i=1}^j \theta_{l,i} + \sum_{i=1}^{j-1} \delta_{l,i}^* + \sum_{i=1}^{j-1} x_{l,i} \leq \beta_{l,j,r_{l,j}} \quad (6.11)$$

pour  $j = 1, 2, 3, \dots, m_l + 1$ .

$$x_{l,i} \leq Z_{l,i} \text{ pour } i = 1, 2, \dots, m_l \quad (6.12)$$

Finalement, les inégalités suivantes doivent être vérifiées :

$$\sum_{i=1}^{j-1} x_{l,i} \leq \text{Min}\{Z_{l,i}, \beta_{l,j,r_{l,j}} - t_{l,1} - \sum_{i=1}^j \theta_{l,i} - \sum_{i=1}^{j-1} \delta_{l,i}^*\} \quad (6.13)$$

pour  $j = 2, \dots, m_l + 1$

Posons :

$$U_{j-1} = \text{Min}\left\{\sum_{i=1}^{j-1} Z_{l,i}, \beta_{l,j,r_{l,j}} - t_{l,1} - \sum_{i=1}^j \theta_{l,i} - \sum_{i=1}^{j-1} \delta_{l,i}^*\right\} \quad (6.14)$$

pour  $j = 2, \dots, m_l + 1$ .

Les contraintes à retenir sont alors :

$$\sum_{i=1}^{j-1} x_{l,i} \leq U_{j-1} \quad j = 2, \dots, m_l + 1 \quad (6.15)$$

L'algorithme suivant, noté  $H$ , conduit à  $H_l$  tel que défini dans le paragraphe précédent.

### Algorithme H

1. Pour  $j = 2, 3, \dots, m_l + 1$  calculer  $U_{j-1}$   
*Voir égalité (6.14)*
2. Faire  $x_{l,0} = 0$
3. Pour  $j = 1, 2, \dots, m_l$  faire :  $x_{l,j} = \text{Min}_{k=j, \dots, m_l} U_j - x_{l,j-1}$
4. Calculer  $H_l = t_{l,1} + \sum_{i=1}^{m_l+1} \theta_{l,i} + \sum_{i=1}^{m_l} \delta_{l,i} + \sum_{i=1}^{m_l} x_{l,i}$

Nous avons vu que lorsque

$$S = \cap_{l=1}^L [h_l, H_l] \neq \emptyset$$

alors nous fixons l'instant de début de l'opération d'assemblage à la borne inférieure de  $S$ . Supposons que  $h^*$  soit cette borne inférieure. Alors nous ajustons

les temps de début des opérations en appliquant l'algorithme  $AJ$  que nous décrivons maintenant.

**Algorithme AJ**

1. Faire  $x_{l,j}^* = x_{l,j} \frac{h^* - h_l}{H_l - h_l}$  pour  $l = 1, 2, \dots, L$  et  $j = 1, 2, \dots, m_l$ .

*Ces instructions ont pour objectif d'ajuster l'extension des temps de stockage proportionnellement à l'augmentation des temps de fin.*

2. Faire  $t_{l,j}^* = t_{l,1} + \sum_{i=1}^j \theta_{l,i} + \sum_{i=1}^{j-1} (\delta_{l,i}^* + x_{l,i}^*)$   
pour  $l = 1, 2, \dots, L$  et  $j = 2, \dots, m_l + 1$

Les valeurs  $t_{l,j}^*$  obtenues constituent la solution du problème.

**Complexité de l'algorithme  $S$**

Considérons d'abord la seconde étape de cet algorithme. Cette étape, qui consiste à appliquer l'algorithme  $A$  à chaque processus linéaire enrichi de l'opération d'assemblage, est exécutée chaque fois qu'une nouvelle fenêtre est testée pour l'opération d'assemblage.

Le nombre de fenêtres testées pour l'opération d'assemblage est majoré par  $z \cdot q_A$ , où  $q_A$  est le nombre de fenêtres disponibles pour l'opération d'assemblage et  $z$  est le nombre maximal de fois où l'on revient sur une fenêtre d'assemblage après rectification.

Nous savons par ailleurs que la complexité de l'algorithme  $A$  appliqué au processus linéaire  $l \in \{1, 2, \dots, L\}$  enrichi de l'opération d'assemblage est de l'ordre de  $O(\sum_{i=1}^{m_l} q_{l,i} \cdot m_l)$ .

Par conséquent, la complexité à associer à l'étape 2 de l'algorithme est majorée par :

$$O(z \cdot N \cdot M)$$

où  $N$  est le nombre total de fenêtres (les fenêtres de l'opération d'assemblage sont comptées  $l$  fois),  $M$  le nombre total d'opérations sur les processus linéaires et  $z$  peut raisonnablement être pris égal à 2.

La recherche du temps de fin fabrication maximal pour chaque processus linéaire complété est de complexité

$$O(m_l^2) \text{ pour } l = 1, 2, \dots, L$$

d'où, à chaque itération, une complexité majorée par  $O(M^2)$ . Nous aboutissons donc à une complexité de :

$$O(2.M.N.(M + N))$$

### Propriétés de l'algorithme

Dans ce paragraphe, nous fournissons deux propriétés relatives à l'algorithme  $S$ .

#### Résultat 1

L'algorithme  $S$  conduit au makespan minimal.

#### Démonstration

En appliquant l'algorithme  $A$  à une séquence d'opérations  $l \in \{1, 2, \dots, L\}$  complétée par l'opération d'assemblage, nous obtenons le makespan minimal. De plus, la fenêtre dans laquelle est exécutée l'opération d'assemblage ne peut être antérieure à la fenêtre qui lui a été affectée au début de l'algorithme. Enfin, nous affectons à l'opération d'assemblage de chaque séquence  $l \in \{1, 2, \dots, L\}$  :

- à l'entrée de l'algorithme  $S$ , la première fenêtre de disponibilité possible,
- pour les itérations suivantes, la première fenêtre de disponibilité qui peut éventuellement satisfaire toutes les séquences  $l = 1, 2, \dots, L$  et est la plus proche de la fenêtre affectée au début de l'itération précédente.

En conséquence, toutes les fenêtres de disponibilité dans lesquelles pourrait s'exécuter l'opération d'assemblage sont explorées dans l'ordre croissant. La première solution obtenue correspond donc au makespan minimal. ■

Considérons maintenant la solution qui conduit au makespan maximal  $H_l$  d'un processus linéaire  $l \in \{1, 2, \dots, L\}$ . Appelons  $S(H_l)$  la solution correspondante. Alors nous pouvons démontrer le résultat suivant.

Nous reprenons les notations précédentes, et en particulier les notations introduites dans l'algorithme  $AJ$

**Résultat 2**

Soit  $O_{l,j_1}$  la dernière opération de  $S(H_l)$  qui se termine à la fin de la fenêtre de disponibilité dans laquelle elle est exécutée, c'est-à-dire la dernière opération telle que :

$$t_{l,j_1}^* + \theta_{l,j_1} = \beta_{l,j_1,r_{l,j_1}}$$

( $r_{l,j_1}$  est le rang de la fenêtre de disponibilité).

Alors les opérations qui succédant à  $j_1$ , c'est-à-dire les opérations  $j = j_1 + 1, \dots, m_l$ , sont suivies d'une période de stockage maximale.

**Démonstration**

Supposons qu'il existe une solution  $S_1(H_2)$  telle que :

1.  $t_{l,j_1}^{*,1} + \theta_{l,j_1} = \beta_{l,j_1,r_{l,j_1}}$

Les  $t_{l,j_1}^{*,1}$  sont les temps de début des opérations qui correspondent à la solution  $S_1(H_2)$ .

2. Il existe  $j_2 \in \{j_1 + 1, \dots, m_l\}$  tel que

$$\delta_{l,j_2}^{*,1} < \delta_{l,j_2}$$

où  $\delta_{l,j_2}^{*,1}$  est le temps de stockage à la sortie de l'opération  $j_2$ .

Posons :

$$\Delta = \text{Min}\{\delta_{l,j_2} - \delta_{l,j_2}^{*,1}, \text{Min}_{j \in \{j_2+1, \dots, m_l+1\}}(\beta_{l,j,r_{l,j}} - t_{l,j}^{*,1} - \theta_{l,j})\}$$

Alors, en augmentant  $\delta_{l,j_2}^{*,1}$  de  $\Delta$ , nous augmentons le makespan de  $\Delta$ , ce qui montre que  $S(H_l)$  n'était pas la solution qui conduit à  $H_l$ . Ceci achève le démonstration. ■

**Un exemple numérique**

Nous considérons le processus d'assemblage de la figure 6.15. : Soient les temps opératoires suivants :

$$\begin{aligned}
 &(\theta_{1,1} = 3, \delta_{1,1} = 4), (\theta_{1,2} = 2, \delta_{1,2} = 4), (\theta_{1,3} = 2, \delta_{1,3} = 4) \\
 &(\theta_{2,1} = 2, \delta_{2,1} = 4), (\theta_{2,2} = 3, \delta_{2,2} = 4), (\theta_{2,3} = 1, \delta_{2,3} = 4) \\
 &(\theta_{3,1} = 4, \delta_{3,1} = 4), (\theta_{3,2} = 3, \delta_{3,2} = 4), (\theta_{4,1} = 4, \delta_{4,1} = 4) \\
 &(\theta_A = 2, \delta_A = 4)
 \end{aligned}$$

Pour faciliter la présentation de l'exemple nous supposons que chaque opération est exécutée par une seule ressource. Les fenêtres de disponibilité correspondantes à chaque opération sont données dans la figure 6.16.

### Application de l'algorithme

#### *Première itération*

Les résultats trouvés pour les séquences 1,2,3 et 4 sont :  $x_A = 18, x_{1,3} = 16, x_{1,2} = 10, x_{1,1} = 4,$   
 $x_A = 7, x_{2,3} = 5, x_{2,2} = 2, x_{2,1} = 0,$   
 $x_A = 15, x_{3,2} = 8, x_{3,1} = 4,$   
 $x_A = 4, x_{4,1} = 0,$

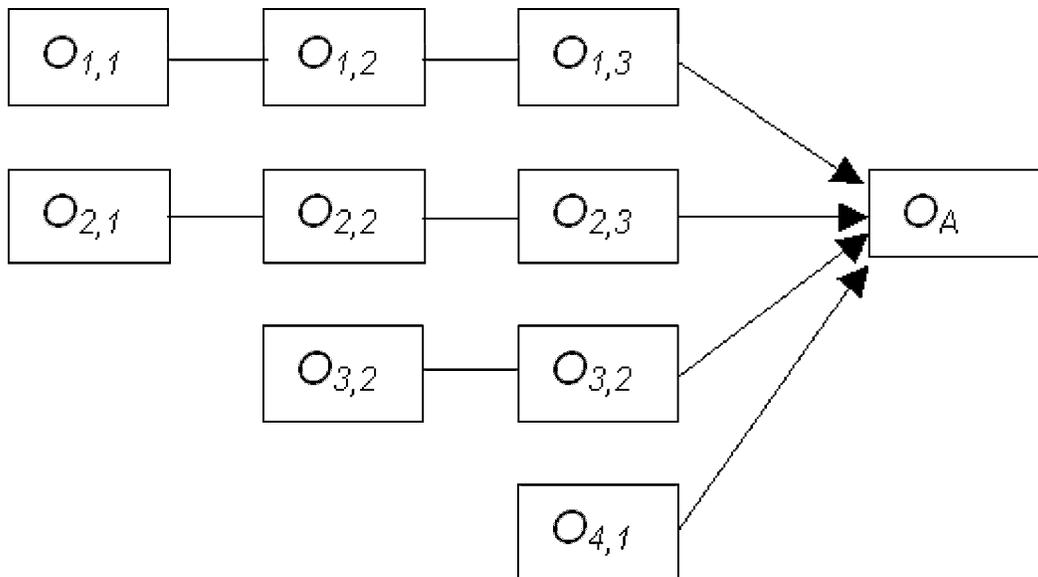


FIG. 6.15 – Processus d'assemblage simple

Nous remarquons que la fenêtre d'exécution n'est pas la même pour toutes les séquences ;  $r_A = 1$  pour les séquences 2 et 4, et  $r_A = 2$  pour les séquences 1 et 3. Nous imposons donc à l'opération d'assemblage des quatre séquences de se situer sur la fenêtre 2 et nous relançons les calculs.

*Deuxième itération :*

$$x_A = 18, x_{1,3} = 16, x_{1,2} = 10, x_{1,1} = 4,$$

$$x_A = 15, x_{2,3} = 11, x_{2,2} = 8, x_{2,1} = 5,$$

$$x_A = 15, x_{3,2} = 8, x_{3,1} = 4,$$

$$x_A = 15, x_{4,1} = 7.$$

L'opération d'assemblage se fait sur la même fenêtre pour toutes les séquences, mais elle ne commence pas au même instant ;  $x_A = 15$  pour les séquences 2,3 et 4 et  $x_A = 18$  pour la séquence 1. Nous sommes dans le cas 2 de l'algorithme proposé. Nous changeons alors la limite inférieure de la fenêtre commune en lui donnant la valeur du plus grand instant de début de l'opération A, soit 18.

*Troisième itération :*

$$x_A = 18, x_{1,3} = 16, x_{1,2} = 10, x_{1,1} = 4, x_A = 22, x_{2,3} = 20, x_{2,2} = 16, x_{2,1} = 12,$$

$$x_A = 18, x_{3,2} = 11, x_{3,1} = 4, x_A = 22, x_{4,1} = 17,$$

Les séquences 2 et 4 ont dépassé la fenêtre 2 et l'opération d'assemblage se fait sur la fenêtre 3. Nous sommes dans le premier cas de l'algorithme. Nous imposons donc à l'opération d'assemblage des quatre séquences de se situer sur la fenêtre 3 et nous relançons les calculs.

*Quatrième itération :*

$$x_A = 22, x_{1,3} = 16, x_{1,2} = 10, x_{1,1} = 4,$$

$$x_A = 22, x_{2,3} = 20, x_{2,2} = 16, x_{2,1} = 12,$$

$$x_A = 22, x_{3,2} = 19, x_{3,1} = 11,$$

$$x_A = 22, x_{4,1} = 17,$$

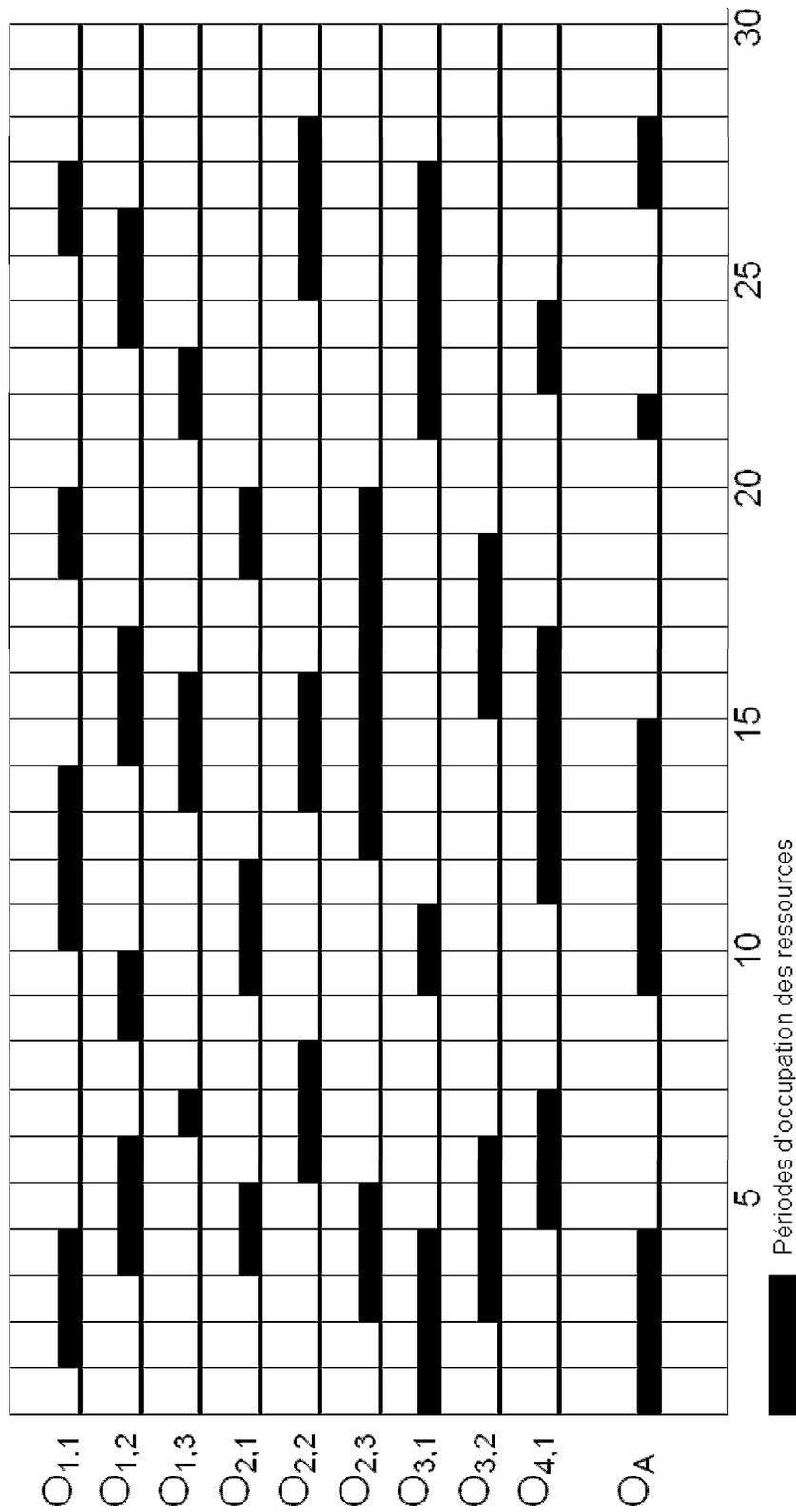


FIG. 6.16 – Occupation des ressources

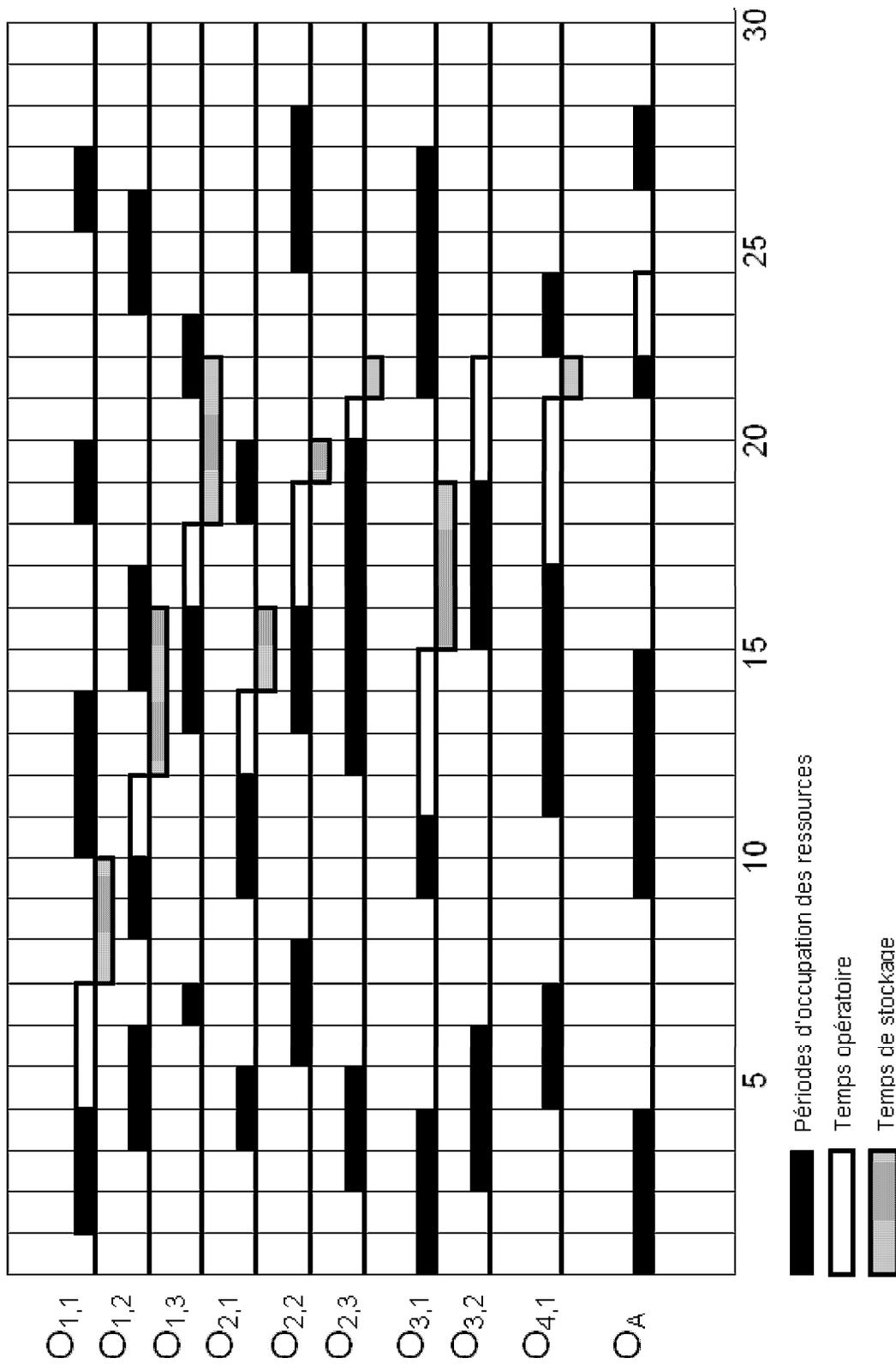


FIG. 6.17 – Ordonnancement du produit

L'opération d'assemblage débute au même instant pour toutes les séquences. Nous arrivons à la fin de l'algorithme, la solution est obtenue et est représentée dans la figure 6.17

### 6.3.2 Ordonnement d'un processus d'assemblage complexe

#### Définition

Un assemblage complexe est un processus qui soit contient une opération d'assemblage qui n'est pas la dernière du processus, soit comporte plusieurs opérations d'assemblage.

La figure 6.18 représente un processus d'assemblage complexe.

Notons que le processus schématisé dans la figure 6.14 pourrait être considéré comme un processus d'assemblage complexe si l'opération A avait été suivie par une autre opération.

#### Position du problème

L'ordonnement que nous cherchons est, comme dans les problèmes développés précédemment, celui qui minimise l'instant de fin de fabrication, encore appelé makespan.

Nous rappelons que nous ne traitons qu'une commande à la fois. Une commande ordonnancée n'est pas remise en question. Rappelons que plusieurs ressources identiques peuvent être affectées à la même opération. Nous utilisons les mêmes notations que celles de la partie 6.3.1.

L'idée que nous avons utilisée pour parvenir à une solution optimale est de décomposer le processus d'assemblage en séquences linéaires. Une séquence linéaire est la suite des opérations situées entre une feuille de l'arborescence et sa racine. Une opération d'assemblage se retrouvera donc dans plusieurs séquences linéaires.

Chaque séquence peut être traitée à l'aide de l'algorithme A modifié pour

tenir compte du stockage. L'objectif est alors d'introduire des contraintes qui obligent à débiter une opération d'assemblage située dans les différentes séquences linéaires au même instant. Nous y reviendrons plus loin.

L'algorithme d'ordonnement que nous proposons dans ce paragraphe exige que le processus d'assemblage complexe soit d'abord décomposé en séquences linéaires de production pour pouvoir appliquer l'algorithme déjà développé pour traiter les cas de processus par transformations successives.

Considérons le processus d'assemblage complexe présenté dans la figure 6.18. La décomposition du processus en séquences linéaires se fait en parcourant l'arborescence du système de production. Ainsi pour notre exemple nous obtenons 5 séquences linéaires. La figure 6.19 représente les séquences obtenues. Les opérations d'assemblage figurent en gras.

Nous remarquons qu'une opération d'assemblage peut appartenir à plusieurs séquences linéaires (l'opération 7 appartient aux séquences  $S_1$ ,  $S_2$  et  $S_3$ ).

Nous appliquons l'algorithme  $A$  modifié pour les séquences linéaires obtenues.

Le résultat direct de l'application de cet algorithme est le makespan des séquences linéaires indépendamment des opérations d'assemblage du processus.

Nous introduisons alors des contraintes qui obligent les opérations d'assemblage à s'effectuer aux mêmes instants pour toutes les séquences linéaires. L'idée est d'ajuster l'instant d'exécution de chaque opération d'assemblage, quand cette dernière ne s'exécute pas au même instant pour toutes les séquences linéaires :

- Soit en relançant les calculs en donnant comme fenêtre de disponibilité initiale la fenêtre la plus éloignée dans le temps lorsque l'opération d'assemblage ne s'exécute pas dans la même fenêtre pour toutes les séquences qui la contiennent.
- Soit en relançant les calculs après avoir corrigé la fenêtre de disponibilité

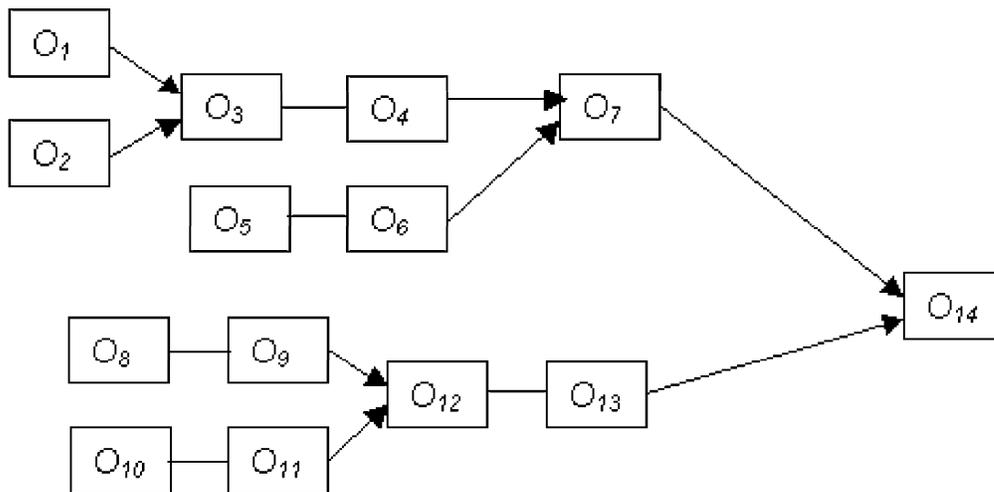


FIG. 6.18 – Processus d'assemblage complexe

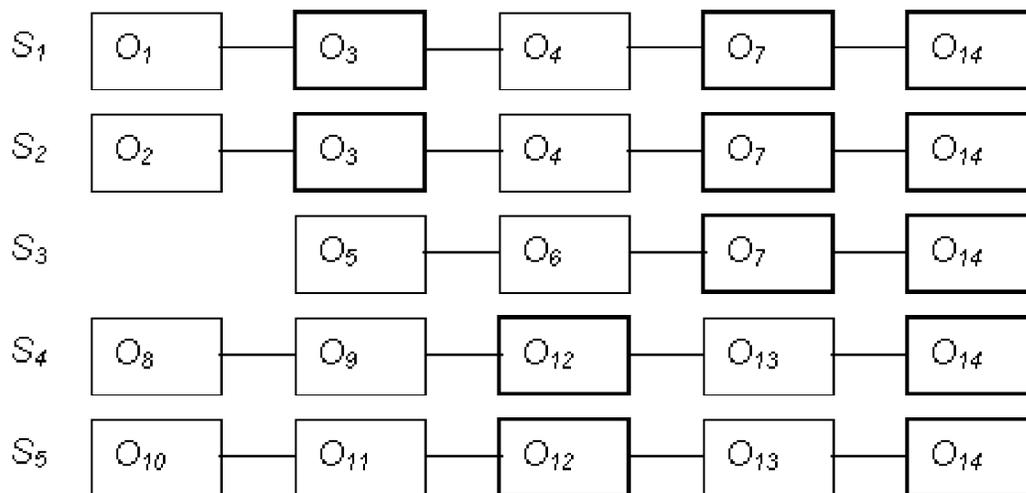


FIG. 6.19 – Décomposition d'un processus d'assemblage complexe

lorsque l'opération d'assemblage s'exécute dans la même fenêtre pour toutes les séquences qui la contiennent ; la correction se fait en prenant le maximum des débuts de cette opération d'assemblage.

Nous utilisons la même philosophie que celle que nous avons développée dans le cas d'un assemblage simple sauf que nous traitons les séquences linéaires une par une en ajustant les instants d'exécution de toutes les opérations d'assemblages à la fois.

### Un exemple

Reprenons l'exemple décrit dans la figure 6.18. Les temps de stockage  $\delta$  sont fixés à 4 pour toutes les opérations. Considérons les temps opératoires suivants :  $(\theta_1 = 2, \delta_1 = 6)$ ,  $(\theta_2 = 4, \delta_2 = 6)$ ,  $(\theta_3 = 3, \delta_3 = 6)$ ,  $(\theta_4 = 2, \delta_4 = 6)$ ,  $(\theta_5 = 2, \delta_5 = 6)$ ,  $(\theta_6 = 1, \delta_6 = 6)$   
 $(\theta_7 = 5, \delta_7 = 6)$ ,  $(\theta_8 = 2, \delta_8 = 6)$ ,  $(\theta_9 = 4, \delta_9 = 6)$ ,  $(\theta_{10} = 1, \delta_{10} = 6)$ ,  $(\theta_{11} = 1, \delta_{11} = 6)$ ,  $(\theta_{12} = 3, \delta_{12} = 6)$   
 $(\theta_{13} = 4, \delta_{13} = 6)$ ,  $(\theta_{14} = 2, \delta_{14} = 6)$

Les fenêtres de disponibilité des ressources des opérations  $i$ ,  $i = 1, 2, \dots, 14$  sont données dans la figure 6.20

### Résolution de l'exemple

*1ère itération :*

$$x_5(14) = 22, x_4(7) = 16, x_3(4) = 10, x_2(3) = 6, x_1(1) = 4,$$

$$x_5(14) = 22, x_4(7) = 16, x_3(4) = 10, x_2(3) = 4, x_1(2) = 0,$$

$$x_4(14) = 15, x_3(7) = 8, x_2(6) = 2, x_1(5) = 0,$$

$$x_5(14) = 35, x_4(13) = 31, x_3(12) = 22, x_2(9) = 12, x_1(8) = 5,$$

$$x_5(14) = 17, x_4(13) = 13, x_3(12) = 4, x_2(11) = 1, x_1(10) = 0,$$

L'opération d'assemblage 14 s'est exécutée dans la fenêtre 2 pour les séquences linéaires 3 et 5, dans la fenêtre 3 pour les séquences linéaires 1 et 2 et

dans la fenêtre 4 pour la séquence linéaire 4.

L'opération d'assemblage 12 s'exécute dans la fenêtre 1 pour la séquence linéaire 5 et dans la fenêtre 2 pour la séquence linéaire 4.

L'opération d'assemblage 7 s'exécute dans la fenêtre 3 pour les séquences linéaires 1 et 2 et dans la fenêtre 2 pour la séquence linéaire 3.

L'opération 3 ne s'exécute dans la même fenêtre 2 pour les deux séquences 1 et 2, mais pas au même instant ( $x(3) = 6$  pour la séquence 1 et  $x(3) = 4$  pour la séquence 2).

Nous relançons les calculs en affectant initialement la fenêtre 4 à l'opération 14, la fenêtre 2 à l'opération 12, la fenêtre 3 à l'opération 7 et la borne inférieure 6 à la fenêtre 2 de l'opération 3.

*2ème itération :*

$$x_5(14) = 28, x_4(7) = 19, x_3(4) = 17, x_2(3) = 13, x_1(1) = 5,$$

$$x_5(14) = 28, x_4(7) = 19, x_3(4) = 17, x_2(3) = 13, x_1(2) = 8,$$

$$x_4(14) = 28, x_3(7) = 17, x_2(6) = 12, x_1(5) = 7,$$

$$x_5(14) = 35, x_4(13) = 31, x_3(12) = 22, x_2(9) = 12, x_1(8) = 5,$$

$$x_5(14) = 35, x_4(13) = 31, x_3(12) = 22, x_2(11) = 17, x_1(10) = 10.$$

L'opération d'assemblage 14 s'exécute dans la même fenêtre 4 pour toutes les séquences linéaires considérées mais pas au même instant.

L'opération d'assemblage 7 s'exécute dans la fenêtre 3 pour les trois séquences linéaires 1,2 et 3 mais pas au même instant.

Les opérations d'assemblage 12 et 3 s'exécutent aux même instant pour toutes les séquences linéaires.

Nous affectons à la fenêtre 4 de l'opération 14 la borne minimale 35 et à la fenêtre 3 de l'opération 7 la borne minimale 19 et nous relançons les calculs.

*3ème itération :*

$$x_5(14) = 35, x_4(7) = 28, x_3(4) = 20, x_2(3) = 13, x_1(1) = 5,$$

$$x_5(14) = 35, x_4(7) = 28, x_3(4) = 20, x_2(3) = 13, x_1(2) = 8,$$

$$x_4(14) = 35, x_3(7) = 40, x_2(6) = 35, x_1(5) = 30,$$

$$x_5(14) = 35, x_4(13) = 31, x_3(12) = 22, x_2(9) = 12, x_1(8) = 5,$$

$$x_5(14) = 35, x_4(13) = 31, x_3(12) = 22, x_2(11) = 17, x_1(10) = 10.$$

Toutes les opérations d'assemblage sont exécutées aux mêmes instants. La solution est obtenue. Elle est représentée dans la figure 6.20. Pour montrer l'importance des temps de stockages permis, nous reprenons le même exemple en réduisant les  $\delta$  à 4. Les deux solutions sont présentées sur le même graphique 6.20.

### Complexité

Elle s'écrit, comme dans le cas d'un assemblage simple :

$$2O(M.N(M + N))$$

Mais les fenêtres des opérations d'assemblage apparaissent dans  $N$  autant de fois qu'elles apparaissent dans le processus linéaire.

### Propriété

#### Résultat 3

L'algorithme appliqué au système d'assemblage complexe conduit au makespan minimal.

#### Démonstration

Analogue à la démonstration du résultat 1.

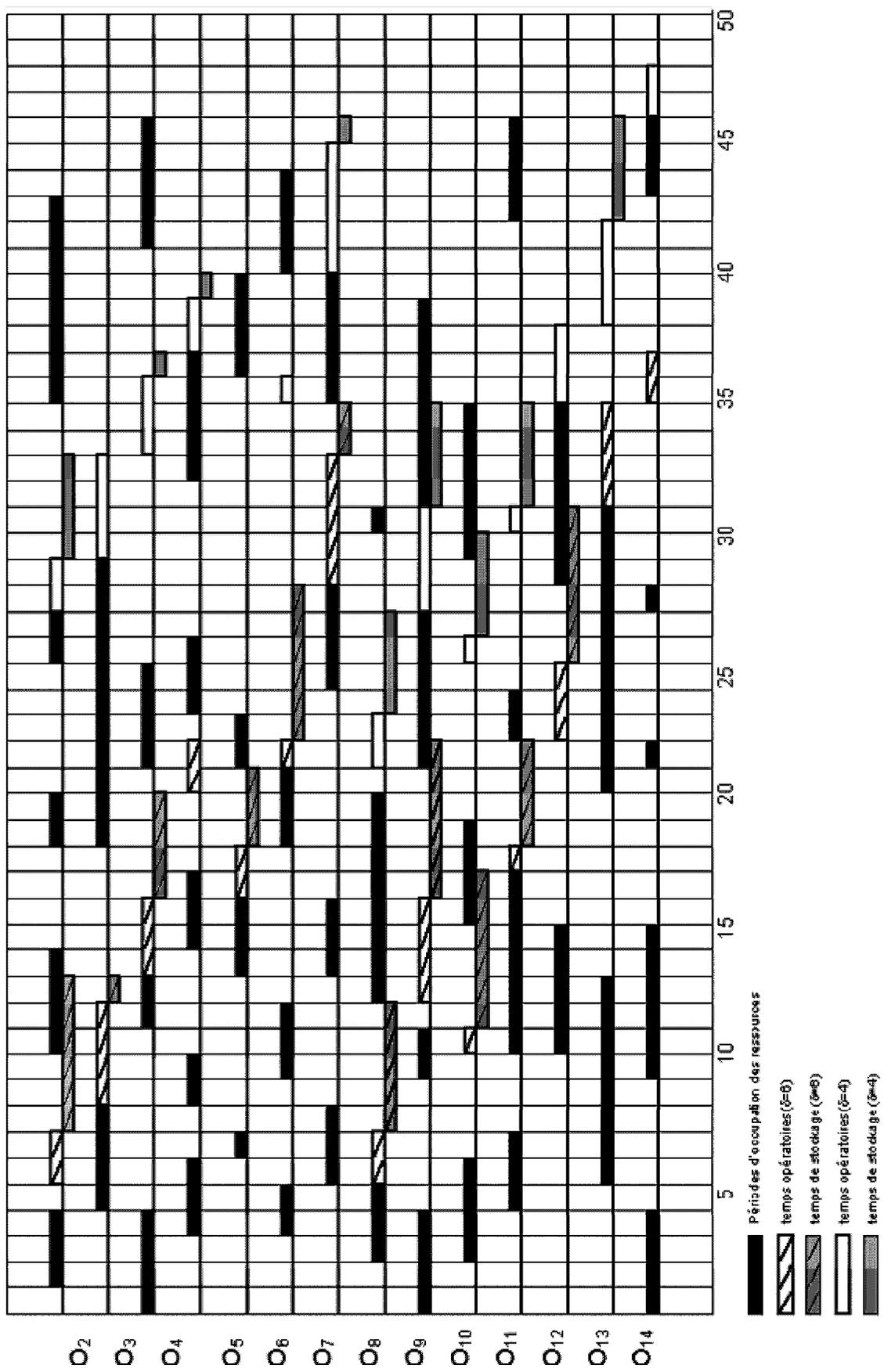


FIG. 6.20 – Ordonnancement d'un produit pour deux différents niveaux d'en-cours permis lors d'un processus d'assemblage

## 6.4 Conclusion

La méthode que nous proposons cherche à ordonnancer toutes les tâches au plus tôt, ce qui garantit une bonne utilisation des ressources, car les fenêtres de disponibilité sont utilisées avant que l'écoulement du temps ne les rejette dans le passé. Comme nous l'avons vu, le taux d'utilisation des ressources croît avec les durées de stockage maximales  $\delta$  affectées aux ressources de stockage, mais au prix d'un en-cours moyen qui augmente parallèlement. La bonne stratégie est donc d'affecter des valeurs faibles aux  $\delta$  lorsque le système est peu chargé, et d'augmenter ces valeurs lorsque la charge du système augmente. Encore faut-il se souvenir qu'un  $\delta$  faible a tendance à conduire à un makespan important.

De plus, l'aspect temps réel de l'algorithme permet un meilleur service des clients. Il est possible de simuler la commande d'un client et de l'informer sur les délais en lui donnant une réponse en temps réel. Ainsi à l'efficacité opérationnelle de la méthode s'ajoute un atout commercial.

Rappelons aussi que cette méthode est particulièrement efficace pour les chaînes d'approvisionnement dans lesquelles l'automatisation garde une part limitée.

Une dernière remarque s'impose : dans tous nos exemples, nous avons considéré qu'une opération fait intervenir une seule ressource, à choisir éventuellement dans un ensemble de ressources identiques. Il se peut cependant que plusieurs ressources soient nécessaires (machines, employé, outil,..etc). Dans ce cas, une ressource disponibilité se définit comme l'ensemble des ressources qui interviennent pour exécuter l'opération.



# Chapitre 7

## Assemblage avec incertitude sur la livraison des composants

### 7.1 Introduction

La sous-traitance est devenue une méthode de gestion de plus en plus populaire du fait de la flexibilité qu'elle offre. Dans le contexte des chaînes d'approvisionnement, la sous-traitance demande une gestion particulièrement fine afin de n'entraîner ni stockage excessif, ni rupture qui désorganise la production.

Ce chapitre s'intéresse à un système d'assemblage dans lequel les composants sont sous-traités. Dans ce système, le délai de livraison de chaque composant est aléatoire, et l'assemblage doit être réalisé à une date donnée. Commandé trop tôt, un composant doit être stocké, ce qui entraîne un coût. Commandé trop tard, un composant arrive après la date à laquelle doit s'effectuer l'assemblage, ce qui entraîne un coût de stockage des composants arrivés à temps et un coût lié au retard de livraison.

Ce problème n'est pas nouveau. Dans [GAL96], Gurrani *et al.* considèrent un problème analogue au problème traité dans ce chapitre, mais leur approche limite l'étude à deux composants. Ils considèrent qu'un composant arrive dans l'une de deux périodes successives. Ce modèle, qui suppose que deux périodes de livraison sont possibles pour chaque composant, autorise une analyse qui conduit à une solution optimale.

Dans [GWY94], Gerchack *et al.* ont proposé un modèle à une période pour deux composants dont les délais de livraison sont des variables aléatoires avec densités de probabilité différentes. Ils aboutissent aux conditions d'optimalité.

Dans [DOL02] Dolgui et Ould-Louly proposent un modèle multi-périodes, c'est-à-dire un modèle dans lequel les composants peuvent arriver à de nombreuses périodes différentes, mais dont les délais de livraison sont des variables aléatoires identiques.

Dans [Mol97], Molinder propose une solution basée sur le recuit simulé pour définir un stock de sécurité et le délai de livraison associé afin de faire face aux aléas des délais de livraison.

Dans [WW76], Whybark et Williams montrent qu'il est plus efficace de régler les points de commande que de rechercher un stock de sécurité.

On retiendra encore le travail de Grasso et Taylor (voir [GT84]) qui se focalise sur la définition d'un stock de sécurité optimal, ainsi que le papier de Proth et al. ([PMW<sup>+</sup>97]) qui ont développé un modèle dans lequel les composants sont communs à plusieurs produits dont la demande est constante à chaque période.

Un état de l'art intéressant sur les problèmes de planification soumis à aléas pourra être trouvé dans [DOL01] et [YWM98]

Le modèle que nous présentons dans ce chapitre est différent des précédents en ce sens que :

- le nombre de composants impliqués dans l'assemblage n'est pas limité,
- la densité de probabilité associée à chaque délai de livraison est continue. En d'autres termes, le temps n'est pas décomposé en périodes au cours desquelles une composante est susceptible d'être livrée.

## 7.2 Formulation du problème

Comme nous l'avons souligné plus haut, nous avons à traiter un problème d'assemblage dans lequel les composants sont confiés à la sous-traitance. Nous

supposons que l'assemblage nécessite  $n$  composants que nous notons  $C_i, i = 1, 2, \dots, n$ . Le délai de livraison, c'est-à-dire la période qui sépare la commande de l'arrivée du composant, est aléatoire, et nous notons  $f_i(t_i - r_i)$  la densité de probabilité de ce délai de livraison pour la composante  $C_i$ . Dans cette fonction,  $r_i$  est l'instant où le composant  $C_i$  a été commandé, et  $t_i$  est l'instant où il a été livré. Chaque composant  $C_i$  a un coût de stockage par unité de temps que nous notons  $s_i$ . Ce coût est applicable entre l'instant  $t_i$  où  $C_i$  est livré, et l'instant où le produit assemblé est livré.

Dans la suite, nous désignons par  $T$  l'instant où le dernier composant arrive et par  $Z$  l'instant où l'assemblage doit débiter pour que la livraison soit effectuée dans les temps.

Bien entendu, si  $T < Z$  le stockage de toute composante  $C_i$  sera effectué durant la période  $Z - t_i$ , et il n'y aura pas de coût associé au retard de livraison. Si, par contre,  $T \geq Z$ , le stockage de toute composante  $C_i$  concernera la période  $T - t_i$  et un coût dû au retard de livraison sera appliqué pour la période  $T - Z$ . Nous désignons par  $h$  le coût de retard d'une période.

Notons que ce modèle n'inclut pas de stock de sécurité. La figure 7.1 schématise un assemblage à 3 composants dans le cas où tous les composants arrivent avant l'instant d'assemblage.

Nous décomposons le coût total en 3 coûts :

- (i) le coût de stockage jusqu'à l'arrivée du dernier composant, c'est-à-dire jusqu'à l'instant  $T$ .
- (ii) le coût de stockage entre l'instant  $T$  et l'instant  $Z$  de début d'assemblage lorsque  $T < Z$ .
- (iii) le coût associé au retard de livraison lorsque  $Z < T$ .

Dans la suite, nous considérons que la durée de l'assemblage est négligeable.

## 7.3 Densité de probabilité associée à T

La densité de probabilité que  $C_i$  soit la composante qui arrive la dernière et qu'elle arrive à l'instant  $T$  est le produit de :

- la probabilité que toutes les composantes autre que  $C_i$  arrivent avant l'instant  $T$ , ce qui s'écrit :

$$\prod_{k=1, k \neq i}^n F_k(T - r_k)$$

où  $F_k$  est la fonction de répartition de la densité de probabilité associée à  $C_k$ .

Cette écriture implique évidemment que les délais de livraison soient indépendants les uns des autres.

- la densité de probabilité associée à  $C_i$  pour  $T - r_i$ .

Finalement, la densité de probabilité cherchée, que nous notons  $P_i(T)$ , s'écrit :

$$P_i(T) = \prod_{k=1, k \neq i}^n F_k(T - r_k) \cdot f_i(T - r_i). \quad (7.1)$$

Bien entendu, la dernière composante peut être n'importe laquelle des  $n$  composantes. En conséquence, la densité de probabilité que la dernière com-

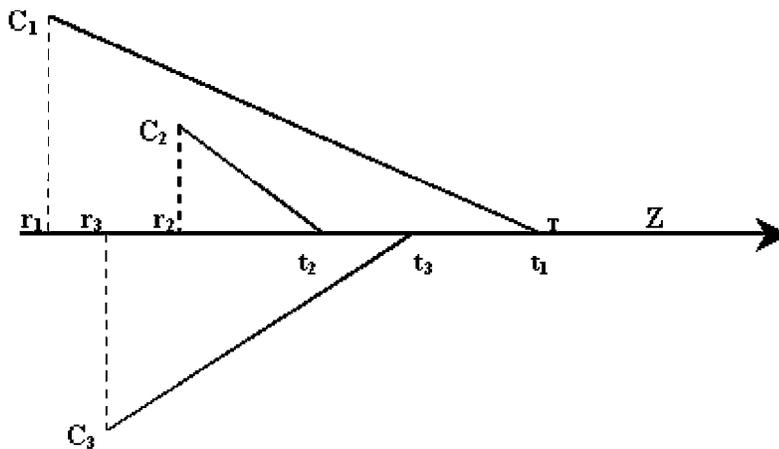


FIG. 7.1 – Une situation d'assemblage

posante arrive à l'instant  $T$  s'écrit :

$$P(T) = \sum_{i=1}^n P_i(T), \quad (7.2)$$

Cette relation donne, avec la relation 7.1 :

$$P(T) = \sum_{i=1}^n \left[ \prod_{k=1, k \neq i}^n F_k(T - r_k) \cdot f_i(T - r_i) \right] \quad (7.3)$$

Note :  $P(T)$  dépend évidemment des instants de commande  $r_1, r_2, \dots, r_n$  mais nous n'avons fait figurer que  $T$  comme variable de  $P$  pour ne pas surcharger les notations.

La relation (7.3) ne s'applique évidemment que si  $T \geq \text{Max}_{j=1,2,\dots,n} r_j$ . Dans le cas contraire,  $P(T) = 0$ .

## 7.4 Les coûts qui composent le coût total

### 7.4.1 Coût de stockage jusqu'à l'instant T

Soit  $C_i$  le dernier composant qui arrive à l'instant  $T$ . Alors le coût de stockage des  $n - 1$  autres composants s'écrit :

$$I(C_i) = \sum_{k=1, k \neq i}^n \left[ s_k \cdot \int_{t_k=r_k}^T (T - t_k) \cdot f_k(t_k - r_k) \cdot dt_k \right] \cdot f_i(T - r_i) dT \quad (7.4)$$

Considérons l'intégrale donnée par la relation suivante :

$$J(C_k) = \int_{t_k=r_k}^T (T - t_k) \cdot f_k(t_k - r_k) dt_k \quad (7.5)$$

En intégrant par parties, nous obtenons :

$$J(C_k) = \int_{t_k=r_k}^T F_k(t_k - r_k) dt_k \quad (7.6)$$

En conséquence la relation (7.4) se réécrit :

$$I(C_i) = \sum_{k=1, k \neq i}^n \left[ s_k \int_{t_k=r_k}^T F_k(t_k - r_k) dt_k \right] \cdot f_i(T - r_i) \quad (7.7)$$

Comme les évènements correspondant à l'arrivée des différents composants sont mutuellement indépendants, et compte tenu du fait que  $T$  peut varier de  $R = \text{Max}\{r_j\}_{j \in \{1,2,\dots,n\}}$  à  $+\infty$ , le coût de stockage moyen pour l'ensemble des composants jusqu'à l'arrivée du dernier composant s'écrit :

$$I_1 = \sum_{i=1}^n \left( \int_R^{+\infty} \left\{ \sum_{k=1, k \neq i}^n [s_k \cdot \int_{t_k=r_k}^T F_k(t_k - r_k) \cdot dt_k] \cdot f_i(T - r_i) \right\} \cdot dT \right) \quad (7.8)$$

Nous observons que  $I_1$  est fonction des instants de commande  $r_1, r_2, \dots, r_n$ , mais non de l'instant  $Z$ .

### 7.4.2 Coût de stockage moyen entre $T$ et $Z$

Ce coût existe lorsque l'instant où le dernier composant arrive, soit  $T$ , est inférieur à  $Z$ , instant du début de l'assemblage. Durant la période  $Z - T$ , le coût de stockage s'applique à tous les composants, et donc le coût de stockage par unité de temps est  $\sum_{k=1}^n s_k$ .

Finalement, le coût de stockage moyen est :

$$I_2 = \left( \sum_{k=1}^n s_k \right) \int_{T=R}^Z (Z - T) \cdot P(T) \cdot dT \quad (7.9)$$

où  $R = \text{Max}\{r_j\}_{j \in \{1,2,\dots,n\}}$  et  $P(T)$  est donné en (7.3).

### 7.4.3 Coût moyen associé à la rupture de stock

Il y a coût de rupture lorsque  $T > Z$ , et ce coût s'élève à  $h$  par unité de temps. Ce coût est

$$B = h \cdot \int_{T=Z}^{+\infty} (T - Z) \cdot P(T) \cdot dT \quad (7.10)$$

Finalement, le coût moyen total est :

$$S = I_1 + I_2 + B$$

## 7.5 Analyse du problème

Le but est de minimiser  $S$ . Or  $S$  est fonction de  $Z$  et des instants  $r_1, r_2, \dots, r_n$  où les commandes sont passées. En considérant l'expression de  $S$ , nous voyons que :

- $I_1$  est constant par rapport à  $Z$ .  $I_1$  n'est fonction que de  $r_1, r_2, \dots, r_n$ , donc des positions relatives des instants où les commandes sont passées.
- $I_2$  et  $B$  ne sont fonctions que de  $Z$  si l'on connaît le dernier instant où des commandes sont passées.

Nous noterons  $W(Z)$  la somme  $I_2 + B$  :

$$W(Z) = \sum_{k=1}^n s_k \int_{T=R}^Z (Z - T) \cdot P(T) \cdot dT + h \int_{T=Z}^{+\infty} (T - Z) \cdot P(T) \cdot dT \quad (7.11)$$

Nous étudions d'abord un cas particulier.

### 7.5.1 Un cas particulier

Nous considérons le cas où la densité associée au délai de livraison de chaque composant est strictement positive sur un intervalle donné et nulle ailleurs :

$$f_i(x_i) = \left\{ \begin{array}{ll} > 0 & \text{si } a_i \leq x_i \leq b_i \\ 0 & \text{ailleurs} \end{array} \right\} \quad (7.12)$$

Dans ce cas, l'expression (7.8) peut se réécrire :

$$I_1 = \sum_{i=1}^n \left( \int_{R_s}^{r_i + b_i} \left\{ \sum_{k=1, k \neq i}^n [s_k \cdot \int_{t_k = r_k + a_k}^T F_k(t_k - r_k) \cdot dt_k] \cdot f_i(T - r_i) \right\} \cdot dT \right) \quad (7.13)$$

où  $R_s = \text{Max}_{j \in \{1, 2, \dots, n\}} \{r_j + a_j\}$  Dans ce cas particulier, nous avons le résultat 1.

#### Résultat 1

Le coût moyen  $I_1$  est minimal pour des instants de passation de commande qui vérifient  $Z - b_i \leq r_i \leq Z - a_i$ ,  $i = 1, 2, \dots, n$ .

#### Démonstration

a. Supposons que, pour le  $p$ -ème composant,  $r_p > Z - a_p$ . Dans ce cas,  $I_1$

(expression 7.13) devient :

$$I_{1,Z < r_p + a_p} = \sum_{i=1, i \neq p}^n \left( \int_{R_s}^{r_i + b_i} \left\{ \sum_{k=1, k \neq i}^n [s_k \cdot \int_{t_k = r_k + a_k}^T F_k(t_k - r_k) \cdot dt_k] \cdot f_i(T - r_i) \right\} \cdot dT \right) \\ + \int_{R_s}^{r_p + b_p} \left\{ \sum_{k=1, k \neq p}^n [s_k \cdot \int_{t_k = r_k + a_k}^T F_k(t_k - r_k) \cdot dt_k] \cdot f_i(T - r_p) \right\} \cdot dT$$

Comme  $r_p > Z - a_p$ , nous avons  $Z < r_p + a_p < r_p + b_p$ . Alors  $I_{1,Z < r_p + a_p}$  peut s'écrire :

$$I_{1,Z < r_p + a_p} = \sum_{i=1, i \neq p}^n \left( \int_{R_s}^{r_i + b_i} \left\{ \sum_{k=1, k \neq i}^n [s_k \cdot \int_{t_k = r_k + a_k}^T F_k(t_k - r_k) \cdot dt_k] \cdot f_i(T - r_i) \right\} \cdot dT \right) \\ + \int_{R_s}^Z \left\{ \sum_{k=1, k \neq i}^n [s_k \cdot \int_{t_k = r_k + a_k}^T F_k(t_k - r_k) \cdot dt_k] \cdot f_i(T - r_p) \right\} \cdot dT \\ + \int_Z^{r_p + b_p} \left\{ \sum_{k=1, k \neq p}^n [s_k \cdot \int_{t_k = r_k + a_k}^T F_k(t_k - r_k) \cdot dt_k] \cdot f_p(T - r_p) \right\} \cdot dT$$

Finalement :

$$I_{1,Z < r_p + a_p} = I_{1,Z = a_p + r_p} + \int_Z^{r_p + b_p} \left\{ \sum_{k=1, k \neq p}^n [s_k \cdot \int_{t_k = r_k + a_k}^T F_k(t_k - r_k) \cdot dt_k] \cdot f_p(T - r_p) \right\} \cdot dT \quad (7.14)$$

Cette égalité montre que  $I_1$  est, lorsque  $r_p + a_p > Z$ , supérieur à  $I_1$  lorsque  $r_p + a_p = Z$ . Cela complète la première partie de la démonstration.

b. Supposons que  $r_p < Z - b_p$ . Dans ce cas, la relation (7.13) devient :

$$I_{1,Z > r_p + b_p} = \sum_{i=1}^n \left[ \int_{\text{Max}(r_j + a_j)}^{r_i + b_i} \left\{ \sum_{k=1, k \neq i, p}^n (s_k \cdot \int_{t_k = r_k + a_k}^T F_k(t_k - r_k) \cdot dt_k) \right. \right. \\ \left. \left. + s_p \cdot \int_{t_p = r_p + a_p}^T F_p(t_p - r_p) \cdot dt_p \right\} \cdot f_i(T - r_i) \cdot dT \right]$$

Comme  $r_p + b_p < Z$ , nous pouvons écrire :

$$\begin{aligned}
I_{1,Z>r_p+b_p} &= \sum_{i=1}^n \left[ \int_{\text{Max}(r_j+a_j)}^{r_i+b_i} \left\{ \sum_{k=1, k \neq i, p}^n (s_k \cdot \int_{t_k=r_k+a_k}^T F_k(t_k-r_k) \cdot dt_k) + s_p \cdot \int_{t_p=r_p+a_p}^{Z-b_p} F_p(t_p-r_p) \cdot dt_p \right. \right. \\
&\quad \left. \left. + s_p \cdot \int_{t_p=Z-b_p}^T F_p(t_p-r_p) \cdot dt_p \right\} \cdot f_i(T-r_i) \cdot dT \right] \\
&= \sum_{i=1}^n \left[ \int_{\text{Max}(r_j+a_j)}^{r_i+b_i} \left\{ \sum_{k=1, k \neq i, p}^n (s_k \cdot \int_{t_k=r_k+a_k}^T F_k(t_k-r_k) \cdot dt_k) + s_p \cdot \int_{t_p=Z-b_p}^T F_p(t_p-r_p) \cdot dt_p \right. \right. \\
&\quad \left. \left. + s_p \cdot \int_{t_p=r_p+a_p}^{Z-b_p} F_p(t_p-r_p) \cdot dt_p \right\} \cdot f_i(T-r_i) \cdot dT \right]
\end{aligned}$$

Finalement :

$$I_{1,Z>r_p+b_p} = I_{1,Z=r_p-b_p} + \sum_{i=1}^n \left[ \int_{\text{Max}(r_j+a_j)}^{r_i+b_i} \left\{ \sum_{k=1, k \neq i, p}^n [s_p \cdot \int_{t_p=r_p+a_p}^{Z-b_p} F_p(t_p-r_p) \cdot dt_p] \cdot f_i(T-r_i) \right\} \cdot dT \right] \quad (7.15)$$

Nous voyons que le coût de stockage moyen est plus petit lorsque  $r_p = Z - b_p$  que lorsque  $r_p < Z - b_p$ . Ceci complète la démonstration. ■

## 7.5.2 Propriété liée au cas général

Le résultat suivant donne une propriété intéressante du coût  $W(Z) = I_2 + B$ , c'est-à-dire de la partie du coût qui dépend de l'instant d'assemblage  $Z$ .

### Résultat 2

Une condition nécessaire et suffisante pour que  $W(Z)$  soit minimal est que la probabilité d'arrivée des composants avant l'instant  $Z$  soit constante et égale à :

$$\frac{h}{\sum_{k=1}^n s_k + h}$$

Cette condition s'écrit :

$$\prod_{k=1}^n F_k(Z - r_k) = \frac{h}{\sum_{k=1}^n s_k + h} \leq 1$$

### Démonstration

Nous montrerons, dans le paragraphe suivant, de  $W(Z)$  est une fonction convexe et continument différentiable de  $Z$  sur  $[R, +\infty)$ . Le dérivée de  $W(Z)$  s'écrit :

$$\frac{dW(Z)}{dZ} = \sum_{k=1}^n s_k \int_{T=R}^Z P(T).dT - h \int_Z^{\infty} P(T).dT$$

où  $R = \text{Max}_{j=1,2,\dots,n} \{r_j\}$ .

Cette dérivée s'annule pour :

$$\sum_{k=1}^n s_k \int_{T=R}^Z P(T).dT = h \int_Z^{\infty} P(T).dT$$

ou,

$$\int_Z^{\infty} P(T).dT = \frac{\sum_{k=1}^n s_k}{\sum_{k=1}^n s_k + h} \quad (7.16)$$

De plus, nous obtenons, à partir de la relation (7.3) :

$$\int_{T=Z}^{\infty} P(T).dT = \int_{T=Z}^{\infty} \sum_{i=1}^n \left[ \prod_{k=1, k \neq i}^n F_k(T - r_k) \cdot f_i(T - r_i) \right] dT$$

La sommation située à droite de cette égalité n'est autre que la dérivée de :

$$\prod_{k=1}^n F_k(T - r_k)$$

et donc

$$\int_{T=Z}^{\infty} P(T).dT = 1 - \prod_{k=1}^n F_k(Z - r_k) \quad (7.17)$$

Par conséquent :

$$\prod_{k=1}^n F_k(Z - r_k) = \frac{h}{\sum_{k=1}^n s_k + h} \leq 1 \quad (7.18)$$

Ceci complète la démonstration. ■

La relation (7.18) montre qu'une infinité de valeurs de  $r_k$ ,  $k = 1, 2, \dots, n$ , satisfont cette équation, donc minimisent  $W(Z)$ . Dans cet ensemble infini de valeurs, nous choisirons l'ensemble  $\{r_k^*\}_{k=1,2,\dots,n}$  qui minimise  $I_1$  : en effet, nous n'oublions pas que nous voulons minimiser  $S = I_1 + W(Z)$ . Cette remarque suggère de proposer un algorithme itératif qui améliore pas à pas la valeur de  $I_1$  tout en restant dans les solutions qui vérifient la relation (7.18). C'est la raison pour laquelle nous proposons un algorithme basé sur le recuit simulé dans lequel la recherche d'une solution voisine, à chaque pas, est contrainte par la nécessité de satisfaire la relation (7.18).

Dans le cas le plus général, c'est-à-dire le cas où les densités de probabilité attachées aux composants sont différentes, nous utilisons une méthode de type Monte Carlo pour évaluer  $I_1$ . Cet algorithme est présenté plus loin.

### 7.5.3 Etude de $W(Z)$

#### Résultat 3 :

$W(Z)$  est une fonction convexe et continûment différentiable de  $Z$  sur  $[R, +\infty)$ .

#### Démonstration

La dérivée de  $W(Z)$  par rapport à  $Z$  s'écrit :

$$\frac{dW(Z)}{dZ} = \sum_{k=1}^n s_k \int_{T=R}^Z P(T).dT - h \int_Z^{\infty} P(T).dT$$

où  $R = \text{Max}_{j=1,2,\dots,n}(r_j)$ . La dérivée seconde de  $W(Z)$  par rapport à  $Z$  est :

$$\frac{d^2W}{dZ^2} = \left( \sum_{k=1}^n s_k + h \right) \cdot P(Z)$$

Comme  $P(Z) > 0$  si  $Z \geq R$ , nous voyons que  $W(Z)$  est convexe.

Il est d'autre part évident que  $W(Z)$  est continûment différentiable. ■

Remarques :

1. Lorsque  $h = 0$ ,  $W(Z)$  est nulle pour  $Z = R$  et croît avec  $Z$ . Elle a donc la forme représentée dans la figure (7.2)
2. Le coût de retard, noté  $B$ , est maximum pour  $Z = R$  et tend vers 0 lorsque  $Z$  tend vers l'infini. Ce coût est représenté dans la figure 7.3

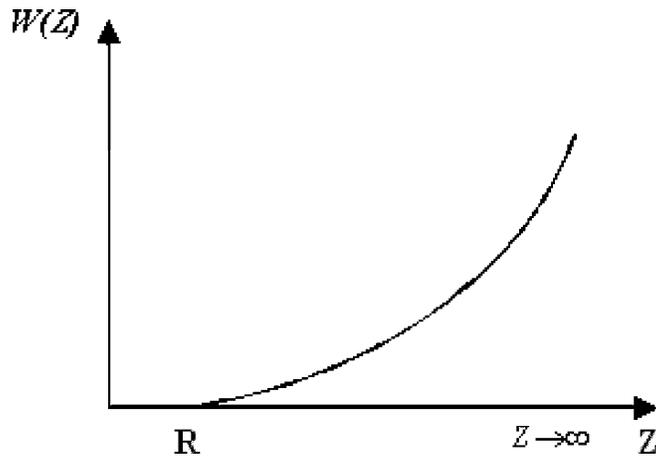
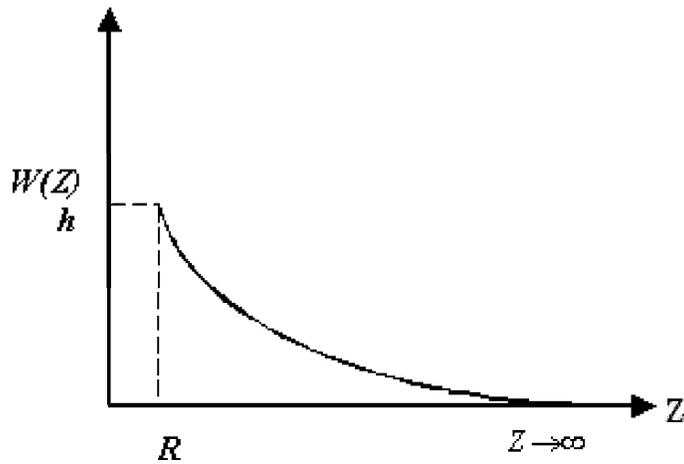
FIG. 7.2 – Fonction  $W(Z)$  pour  $h = 0$ 

FIG. 7.3 – Evolution du coût associé au retard

3. La forme générale de  $W(Z)$  est donnée par figure (7.4)

#### 7.5.4 Un autre cas particulier : les composants ont tous le même comportement

Dans ce paragraphe, nous considérons le cas particulier où tous les composants ont des délais de livraison aléatoires avec une densité uniformément répartie sur le même intervalle. Cette densité est donnée par la relation (7.19)

$$f(t-r) = \left\{ \begin{array}{ll} \frac{1}{b-a} & \text{si } t \in [r+a, r+b] \\ 0 & \text{sinon} \end{array} \right\} \quad (7.19)$$

En considérant les relations (7.8) et (7.18), nous obtenons :

$$I_1 = n.(n-1)s \int_{r+a}^{r+b} \int_{r+a}^T [F(t-r).dt] f(T-r).dT \quad (7.20)$$

ou,

$$I_1 = n.(n-1).s.(b-a)/6 \quad (7.21)$$

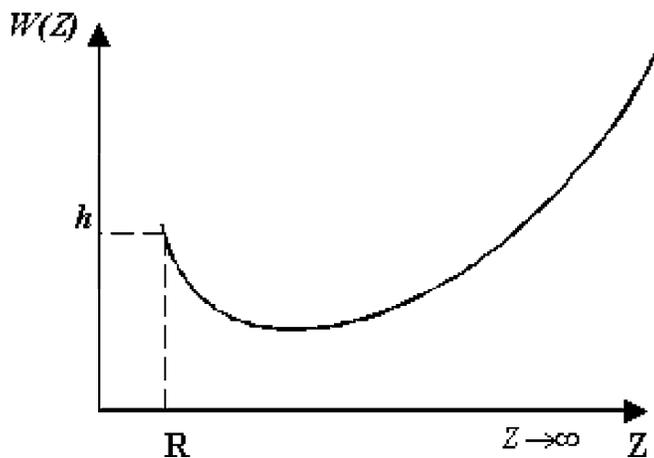


FIG. 7.4 – Forme générale de  $W(Z)$

La relation (7.21) montre que, dans ce cas,  $I_1$  ne dépend pas de l'instant  $r$  où la commande est émise. Par conséquent,  $r$  peut être calculée directement à partir de la relation (7.18).

$$F(Z - r) = \left(\frac{h}{n.s + h}\right)^{\frac{1}{n}} \quad (7.22)$$

Mais :

$$F(Z - r) = \frac{Z - r - a}{b - a}, r \in [Z - b, Z - a]$$

Nous obtenons donc, finalement :

$$r = Z - a - (b - a)\left(\frac{h}{n.s + a}\right)^{\frac{1}{n}}$$

Dans le cas particulier où tous les composants ont un comportement identique avec une densité de probabilité uniforme, l'instant de livraison, qui est le même pour tous les composants, peut être obtenu analytiquement.

### 7.5.5 Présentation de l'algorithme général

Nous calculons d'abord un ensemble de valeurs  $r_1, r_2, \dots, r_n$  qui satisfont la relation (7.18). Cela se fait en affectant la même valeur  $\sqrt[n]{\frac{h}{\sum_{k=1}^n s_k + h}}$  à chaque fonction de répartition  $F_k(Z - r_k)$ , puis en procédant par dichotomie.

Nous évaluons ensuite le coût associé à cet ensemble à l'aide d'une méthode de type Monte Carlo.

Enfin, nous fixons les paramètres du recuit simulé, soit :

$T_0$  : Température initiale.

$T_f$  : Température finale.

$M$  : Nombre maximal d'itérations à la même température.

$\alpha$  : Coefficient multiplicatif de réduction de la température toutes les  $M$  itérations ( $0 < \alpha < 1$ ).

Le but de l'algorithme est de minimiser  $S = I_1 + W(Z)$ . Dans cet algorithme, nous avons décidé de considérer l'égalité (7.18) comme une contrainte

et d'appliquer le recuit simulé à la minimisation de  $I_1$  en recherchant, à chaque itération, une solution voisine qui satisfait (7.18). Nous savons en effet qu'une telle solution minimise  $W(Z)$ .

L'algorithme se formalise comme suit :

### Algorithme général

1. Calculer un ensemble  $r_1, r_2, \dots, r_n$  qui vérifié (7.18). Pour cela :

(a) Calculer  $C = \frac{h}{\sum_{k=1}^n s_k + h}$

(b) Calculer  $c = \sqrt[n]{C}$

(c) Pour  $i = 1, 2, \dots, n$ , calculer par dichotomie  $r_i$  tel que  $F_i(Z - r_i) = c$ .

Soit  $R = \{r_1, r_2, \dots, r_n\}$  la solution trouvée.

2. Initialiser  $T_0, T_f, M, \alpha, ct = 0$

3. Utiliser la méthode de Monte Carlo pour évaluer  $I_1(R)$  à partir de la relation 7.8 et poser :

$$R^* = R$$

$$I_1^* = I_1(R)$$

4. Chercher une solution  $R_1$  du voisinage de  $R^*$  qui satisfait (7.18).

*La recherche de cette solution est donnée dans l'algorithme "Algorithme V" présenté ci-dessous*

5. Evaluer  $I_1(R_1)$  par la méthode de Monte Carlo.

6. Si  $I_1(R_1) < I_1^*$  :

(a) Poser  $R^* = R_1$

(b) Poser  $I_1^* = I_1(R_1)$

(c) Aller en 8.

7. Si  $I_1(R_1) \geq I_1^*$  :

(a) Générer  $x$  au hasard sur  $[0, 1]$ .

(b) Calculer  $y = \exp[-(I_1(R_1) - I_1^*)/T_0]$

(c) Si  $y > x$  poser  $R = R_1$

8.  $ct = ct + 1$

9. Si  $ct > M$ , faire

(a)  $ct = 0$

(b)  $T_0 = T_0 * \alpha$

10. Si  $T_0 > T_f$ , aller en 4.

11. Imprimer  $R^*, I_1^*$ . Fin.

Le recherche d'un voisinage part de la remarque suivante :

a. Si le second membre de (7.18) est égal à 0, cela signifie que  $h = 0$  et donc que la solution optimale est  $r_k = Z$  pour  $k = 1, \dots, n$  : il n'est donc pas nécessaire d'appliquer l'algorithme.

b. Si le second membre de (7.18) est égal à 1, cela signifie que les coûts de stockage sont tous nuls, et donc que la solution optimale est de commander le plus tôt possible.

Par conséquent, l'algorithme ne s'applique que lorsque le second membre de 7.18 est strictement compris entre 0 et 1, ce qui signifie que  $F_k(Z - r_k) > 0$  pour  $k = 1, 2, \dots, n$ , et que une partie au moins de ces fonctions de répartition doivent prendre une valeur inférieure à 1. Nous désignons par  $E(R) = \{k/F_k(Z - r_k) < 1\}$ . D'où l'algorithme de recherche d'une solution voisine.

**Algorithm V**

1. Choisir au hasard  $k_1 \in E(R)$
2. Choisir au hasard  $k_2 \in \{1, 2, \dots, n\}, k_1 \neq k_2$
3. Faire  $r_{k_1}^* = r_{k_1} - \delta$

$\Delta$  est un réel aussi petit que l'on voudra

4. Faire  $r_{k_2}^1 = Z$  et  $r_{k_2}^2 = r_{k_2}$

5. Faire :

$$X = \frac{F_{k_1}(Z - r_{k_1}) \cdot F_{k_2}(Z - r_{k_2})}{F_{k_1}(Z - (r_{k_1} - \Delta))}$$

6. Faire  $r_{k_2}^* = (r_{k_2}^1 + r_{k_2}^2)/2$

7. Si  $F_{k_2}(Z - r_{k_2}^*) < X$  faire  $r_{k_2}^1 = r_{k_2}^*$ , sinon faire  $r_{k_2}^2 = r_{k_2}^*$ .

8. Si  $r_{k_2}^1 - r_{k_2}^2 > \epsilon$ , aller en 6.

9. On choisit le couple  $(r_{k_1}^*, r_{k_2}^*)$  pour remplacer  $(r_{k_1}, r_{k_2})$  dans la solution précédente : c'est la solution voisine.

## 7.6 Exemple numérique

Dans ce paragraphe, nous présentons deux exemples. Le premier exemple concerne un assemblage dans lequel les densités de probabilité des délais de livraison sont différents d'un composant à l'autre. Nous nous intéressons à l'évolution des instants de commande en fonction du coût associé aux retards. Dans le second exemple, nous appliquons l'algorithme général au cas où les composants ont un comportement identique, et nous comparons le résultat à l'optimum que nous pouvons calculer analytiquement dans ce cas.

**Exemple 1 :**

Dans le tableau 1, nous fournissons le coût de stockage unitaire et les paramètres qui définissent les densités de probabilité des délais de livraison. La densité de probabilité est uniforme sur l'intervalle donné. De plus,  $Z = 300$ .

Nous appliquons l'algorithme général pour dix valeurs différentes du coût unitaire de retard (BL). Les résultats sont présentés dans le tableau 2.

Nous observons que lorsque le coût associé aux retards diminue, les instants auxquels les commandes sont passées se rapprochent de  $Z$ . Nous voyons entre autre que la relation  $Z - b_i \leq r_i \leq Z - a_i$  est vérifiée, comme nous l'avons démontré dans le résultat 1.

TAB. 7.1 – Données de l'exemple 1

Composants	Coût de stockage	Densité de probabilité
1	20	[100, 150]
2	15	[50, 170]
3	10	[50, 100]

TAB. 7.2 – Evolution des instants de commande avec BL

BL	50	45	40	35	30	25	20	15	10	5
1	159.08	159.82	161.06	161.99	163.11	164.03	166.15	168.41	171.62	176.69
2	152.96	154.61	156.6	158.90	161.58	164.71	168.99	174.41	182.02	194.31
3	210.23	210.87	211.21	212.14	213.26	215.07	216.35	218.61	221.77	226.89

BL : Coût de rupture.

### Exemple 2

Dans cet exemple, les composants ont tous le même comportement :

$$f(x_i) = \begin{cases} \frac{1}{70} & \text{si } x_i \in [100, 170] \\ 0 & \text{sinon} \end{cases} \text{ pour } i = 1, 2, \dots, n$$

Le coût de stockage est le même pour tous les composants :

$$s_i = 20 \text{ pour } i = 1, 2, 3$$

et

$$h = 50$$

Dans ce cas particulier, la solution optimale est connue, comme nous l'avons signalé précédemment. L'instant de commande  $r$  est le même pour tous les composants et

$$r = Z - a - (b - a) \left( \frac{h}{n.s + a} \right)^{\frac{1}{n}}$$

Ici, nous trouvons

$$r = 146.178$$

Le programme général nous donne :

$r_1 = 146.369$ ,  $r_2 = 143.601$  et  $r_3 = 148.481$  L' écart s'explique d'une part par l'utilisation du recuit simulé, et d'autre part par l'utilisation de la méthode de Monte Carlo.

## 7.7 Conclusion

Nous disposons donc d'une méthode tout à fait générale. Les résultats sont considérablement améliorés lorsque les intégrales sont calculables analytiquement.



# Chapitre 8

## Partage des risques et des bénéfices dans une chaîne d'approvisionnement : un modèle

### 8.1 Introduction

Nous avons développé le concept de chaîne d'approvisionnement dans les deux premiers chapitres de cette thèse. Nous en retenons qu'il s'agit d'une approche globale dans laquelle plusieurs partenaires, qui peuvent être des entreprises différentes, oeuvrent dans le sens d'une compétitivité commune, en prenant soin de partager équitablement bénéfices et pertes.

Le problème que nous traitons dans ce chapitre est une illustration de la collaboration de sociétés différentes au sein d'une chaîne d'approvisionnement. Il s'agit ici d'une coopération entre grossiste et détaillants. Pour simplifier les notations, nous nous limitons à un détaillant, mais un nombre quelconque de détaillants ne modifierait pas le problème.

Le grossiste et le détaillant maintiennent tous deux un stock afin de répondre à la demande. Leur responsabilité est de maintenir un niveau de stock donné. Si la demande excède le stock maintenu par le détaillant, un coût de rupture est appliqué à celui-ci qui se réapprovisionne chez le grossiste. Si la demande excède la somme des stocks maintenus par le grossiste et le détaillant,

un coût de rupture supérieur est appliqué au détaillant, mais le grossiste paye au détaillant une somme proportionnelle à ce qu'il n'a pas été capable de fournir.

Bien entendu, l'objectif est de faire en sorte que grossiste et détaillant aient un bénéfice proportionnel à leur investissement. De plus, la demande diminue quand le prix de vente augmente, et cette conséquence de la compétition est modélisée à l'aide d'une fonction décroissante.

Il faut souligner que les deux acteurs du problème que nous présentons collaborent, et que le modèle que nous présentons est destiné à mettre en place la règle qui gouverne cette collaboration, c'est-à-dire le "sharing process" dans le langage anglo-saxon.

Des approches similaires sont évoquées dans la littérature sous la dénomination de contrat. Dans [CL99], Cachon et Lariviere développent ce qu'ils nomment les "Revenue Sharing (RS) concepts". Dans ce modèle, le distributeur conserve une fraction du bénéfice et retourne le reste au grossiste. Dans [CL00], Giannocarro et Pontrandolfo étendent ces mêmes "RS Contracts" à une chaîne d'approvisionnement à trois niveaux. D'autres types de contrats sont examinés dans la littérature comme par exemple le "Backup agreement" (voir [EG98]), le "Incentive mechanism" (voir [LW99]) et le "Allocation rules-Règles d'affectation" (voir [CL99]). Jonatam *et al.* (voir [JSP02]) ont modélisé un système multi-produits à deux niveaux avec demande déterministe. Le problème s'exprime en termes de problème de programmation linéaire mixte. La littérature que nous venons de citer s'ajoute à la bibliographie donnée dans les deux premiers chapitres.

## 8.2 Formulation du problème

Nous considérons un grossiste qui achète un produit qui lui coûte  $A$  par unité. Il le vend au détaillant au prix de  $w_p$  par unité. Ce détaillant le vend

aux clients au prix de  $w_r$  par unité. Bien entendu,  $A < w_p < w_r$ .

Les besoins des clients s'expriment à l'aide d'une variable aléatoire que nous notons  $u$ . Le marché qui va se reporter sur notre détaillant est  $f(w_r).u$ , où  $f(w_r)$  est une fonction concave et décroissante de  $w_r$ , et  $f(A) \leq 1$ .

Comme nous l'avons souligné précédemment, le grossiste et le détaillant font partie d'une même chaîne d'approvisionnement, et ils collaborent en acceptant de partager les pertes et les bénéfices proportionnellement à leur investissement.

Pour faire face aux besoins  $u$  des clients qui sont aléatoires avec une densité de probabilité  $\phi(u)$ , le grossiste maintient un stock  $I_p$  et le détaillant un stock  $I_r$ . Les coûts de stockage par unité payés par le grossiste et le détaillant sont respectivement  $c_p$  et  $c_r$ . Si le détaillant n'est pas en mesure de satisfaire un client, il lui paye une pénalité de  $r_r$  par unité non livrée jusqu'à épuisement du stock maintenu par le grossiste, et une pénalité supplémentaire de  $\Delta$  au delà de la somme des stocks maintenus par le détaillant et le grossiste. Le coût  $r_r$  se justifie par le retard de livraison, conséquence du transfert du grossiste vers le détaillant. La pénalité supplémentaire  $\Delta$  se justifie par le retard supplémentaire de livraison, conséquence du temps nécessaire au réapprovisionnement du grossiste.

Le coût de retard par produit est, pour le détaillant, défini comme suit :

$$B_{r1} = \begin{cases} 0 & \text{si } f(w_r).u \leq I_r \\ (f(w_r).u - I_r)r_r & \text{si } f(w_r).u > I_r \end{cases} \quad (8.1)$$

Le coût supplémentaire s'écrit :

$$B_{r2} = \begin{cases} 0 & \text{si } f(w_r).u \leq I_p + I_r \\ (f(w_r).u - (I_p + I_r))\Delta & \text{si } f(w_r).u > I_p + I_r \end{cases} \quad (8.2)$$

Les coûts  $B_{r1}$  et  $B_{r2}$  sont payés par le détaillant aux clients.

Si la demande excède  $I_p + I_r$ , le grossiste paye au détaillant :

$$B_p = \begin{cases} 0 & \text{si } f(w_r).u \leq I_r + I_p \\ (f(w_r).u - (I_r + I_p))r_p & \text{si } f(w_r).u > I_r + I_p \end{cases} \quad (8.3)$$

Le coût de retard moyen du grossiste, noté  $\overline{B}_p$ , et du détaillant, noté  $\overline{B}_{r1}$ , ainsi que le coût supplémentaire moyen, noté  $\overline{B}_{r2}$ , sont donc donnés par les formules suivantes :

$$\overline{B}_p = r_p \cdot \left\{ f(w_r) \int_{\frac{I_p+I_r}{f(w_r)}}^{+\infty} u \cdot \phi(u) \cdot du - (I_p + I_r) \int_{\frac{I_p+I_r}{f(w_r)}}^{\infty} \phi(u) du \right\} \quad (8.4)$$

$$\overline{B}_{r1} = r_r \cdot \left\{ f(w_r) \int_{\frac{I_r}{f(w_r)}}^{\infty} u \cdot \phi(u) \cdot du - I_r \int_{\frac{I_r}{f(w_r)}}^{\infty} \phi(u) du \right\} \quad (8.5)$$

$$\overline{B}_{r2} = \Delta \left\{ f(w_r) \int_{\frac{I_r+I_p}{f(w_r)}}^{\infty} u \cdot \phi(u) \cdot du - (I_p + I_r) \int_{\frac{I_r+I_p}{f(w_r)}}^{\infty} \phi(u) du \right\} \quad (8.6)$$

Dans ce modèle, nous supposons que  $r_p \geq r_{r1} + \Delta$ . Cela signifie que la pénalité de retard que le grossiste paye au détaillant est supérieure à la pénalité de retard que le détaillant paye aux clients au cas de dépassement de  $I_p + I_r$ .

Remarque : Le grossiste paye au détaillant, en cas de rupture, une somme supérieure à la somme payée aux clients par le détaillant. Par conséquent, en cas de rupture chez le grossiste, le détaillant reçoit la somme de :

$$(r_p - (r_r + \Delta)) \left\{ f(w_r) \int_{\frac{I_p+I_r}{f(w_r)}}^{+\infty} u \cdot \phi(u) \cdot du - (I_p + I_r) \int_{\frac{I_p+I_r}{f(w_r)}}^{\infty} \phi(u) du \right\}$$

L'objectif du problème est de définir le prix de vente unitaire  $w_p$  du grossiste au détaillant, et le prix de vente unitaire  $w_r$  du détaillant au client de façon à maximiser le profit total tout en s'assurant que le partage des bénéfices, si bénéfices il y a, est équitable.

Le prix de vente de la livraison moyenne du grossiste au détaillant s'écrit :

$$w_p \cdot f(w_r) \cdot \bar{u} = A \cdot f(w_r) \cdot \bar{u} + c_p \cdot I_p + b_p + \overline{B}_p \quad (8.7)$$

Le membre de gauche est le prix de vente d'une quantité égale à la demande moyenne du client. Cette demande moyenne est le besoin moyen  $\bar{u}$  corrigé du facteur représentant la compétition, soit  $f(w_r)$ .

Le membre de droite de l'égalité est la somme :

- du prix d'achat payé par le grossiste pour acquérir la quantité moyenne,
- du coût de stockage de la quantité  $I_p$ ; cette quantité est la quantité que le grossiste s'engage à tenir à la disposition du détaillant. Nous supposons que le grossiste a mis en place le mécanisme qui permet d'être toujours en mesure de proposer cette quantité à la vente. Le coût de ce mécanisme est intégré dans le coût de stockage  $c_p$ .  $I_p$  est un paramètre du système.
- du bénéfice  $b_p$  attendu par le grossiste de la vente au détaillant de la quantité  $\bar{u}.f(w_r)$ . Le bénéfice est un autre paramètre du problème.
- du coût de rupture moyen  $\overline{B_p}$  que le grossiste paye au détaillant.

Le prix de vente appliqué par le détaillant pour la quantité moyenne demandée par les clients s'écrit :

$$w_r.f(w_r).\bar{u} = w_p.f(w_r).\bar{u} + c_r.I_r + b_r + \overline{B_r} - \overline{B_p} \quad (8.8)$$

Le premier terme du second membre représente le coût de l'achat auprès du grossiste. Le second terme est le coût associé au niveau de stock  $I_r$  que le détaillant s'engage à maintenir. Le troisième terme est le bénéfice attendu.  $\overline{B_r}$  est le coût associé à la rupture et qui est payé aux clients.

$$\overline{B_r} = \overline{B_{r1}} + \overline{B_{r2}}$$

Enfin,  $\overline{B_p}$  est soustrait du coût à affecter du détaillant : c'est la somme moyenne que le grossiste verse au détaillant pour cause de rupture.

Le bénéfice global des deux partenaires s'écrit :

$$b_p + b_r = w_r.f(w_r).\bar{u} - [A.f(w_r).\bar{u} + c_p.I_p + \overline{B_p} + c_r.I_r + \overline{B_r} - \overline{B_p}]$$

soit :

$$b_p + b_r = w_r.f(w_r).\bar{u} - [A.f(w_r).\bar{u} + c_p.I_p + c_r.I_r + \overline{B_{r1}} + \overline{B_{r2}}]$$

ou, plus simplement :

$$b_p + b_r = B(I_p, I_r, w_r) = w_r \cdot f(w_r) \cdot \bar{u} - \overline{K(I_p, I_r, w_r)} \quad (8.9)$$

où  $\overline{K(I_p, I_r, w_r)}$  est

$$\begin{aligned} \overline{K(I_p, I_r, w_r)} = & A \cdot f(w_r) \cdot \bar{u} + I_p \cdot c_p + I_r \cdot c_r + \\ & r_r \left\{ f(w_r) \int_{\frac{I_r}{f(w_r)}}^{+\infty} u \cdot \phi(u) du - (I_r) \int_{\frac{I_r}{f(w_r)}}^{+\infty} \phi(u) du \right\} + \\ & (r_r + \Delta) \left\{ f(w_r) \int_{\frac{I_p + I_r}{f(w_r)}}^{+\infty} u \cdot \phi(u) du - (I_p + I_r) \int_{\frac{I_p + I_r}{f(w_r)}}^{+\infty} \phi(u) du \right\} \end{aligned} \quad (8.10)$$

Nous voyons que  $B(I_p, I_r, w_r)$  est une fonction de  $I_p, I_r$  et  $w_r$ . A  $w_r$  fixé, il est possible d'augmenter  $B(I_p, I_r, w_r)$  en modifiant les niveaux des stocks  $I_p$  et  $I_r$  de façon à diminuer  $\overline{K(I_p, I_r, w_r)}$ . A  $(I_p, I_r)$  fixé, il est possible de modifier  $B(I_p, I_r, w_r)$  en changeant la valeur de  $w_r$ .

Les propriétés que nous présentons maintenant vont nous conduire à un algorithme qui va nous permettre de maximiser  $b_p + b_r$  en agissant alternativement sur le couple  $(I_p, I_r)$  et sur  $w_r$ .

### 8.3 Résultats fondamentaux

Le premier résultat s'intéresse à  $\overline{K(I_p, I_r, w_r)}$  comme fonction de  $I_p$  et  $I_r$ .

#### Résultat 1 :

A  $w_r$  fixé,  $\overline{K(I_p, I_r, w_r)}$  est une fonction convexe de  $I_p$  et  $I_r$ .

#### Démonstration :

Considérons la matrice de Hessian :

$$H(I_p, I_r) = \begin{bmatrix} \frac{\partial^2 \overline{K}}{\partial I_p^2} & \frac{\partial^2 \overline{K}}{\partial I_p \partial I_r} \\ \frac{\partial^2 \overline{K}}{\partial I_p \partial I_r} & \frac{\partial^2 \overline{K}}{\partial I_r^2} \end{bmatrix}$$

où :

$$\frac{\partial^2 \bar{K}}{\partial I_p^2} = \frac{r_r + \Delta}{f(w_r)} \cdot \phi\left(\frac{I_p + I_r}{f(w_r)}\right)$$

$$\frac{\partial^2 \bar{K}}{\partial I_p \partial I_r} = \frac{r_r + \Delta}{f(w_r)} \cdot \phi\left(\frac{I_p + I_r}{f(w_r)}\right)$$

et

$$\frac{\partial^2 \bar{K}}{\partial I_r^2} = \frac{r_r + \Delta}{f(w_r)} \cdot \phi\left(\frac{I_p + I_r}{f(w_r)}\right) + \frac{r_r}{f(w_r)} \cdot \phi\left(\frac{I_r}{f(w_r)}\right)$$

Nous voyons que tous les mineurs principaux de  $H(I_p, I_r)$  sont non négatifs.

Cela achève la démonstration. ■

Le second résultat s'intéresse à  $B(I_p, I_r, w_r)$  comme fonction de  $w_r$ .

### Résultat 2

$I_p$  et  $I_r$  étant fixés,  $B(I_p, I_r, w_r)$  est une fonction concave de  $w_r$ .

### Démonstration

Nous calculons successivement :

$$\frac{\partial^2 B}{\partial w_r^2} = 2 \cdot f'(w_r) \bar{u} + f''(w_r) \bar{u} \cdot w_r - \frac{\partial^2 K(I_p, I_r, w_r)}{\partial w_r^2} \quad (8.11)$$

$$\frac{\partial K(I_p, I_r, w_r)}{\partial w_r} = A \bar{u} f'(w_r) + (r_r + \Delta) \int_{\frac{I_p + I_r}{f(w_r)}}^{\infty} u \cdot \phi(u) du + r_r \int_{\frac{I_r}{f(w_r)}}^{\infty} u \cdot \phi(u) du$$

$$\frac{\partial^2 K(I_p, I_r, w_r)}{\partial w_r^2} = A \bar{u} f''(w_r) +$$

$$(r_r + \Delta) \frac{(I_p + I_r)^2}{f^3(w_r)} \phi\left(\frac{I_p + I_r}{f(w_r)}\right) + r_r \frac{(I_r)^2}{f^3(w_r)} \phi\left(\frac{I_r}{f(w_r)}\right) \quad (8.12)$$

Les équations (8.11) et (8.12) nous conduisent à :

$$\frac{\partial^2 B}{\partial w_r^2} = 2 \cdot f'(w_r) \bar{u} - f''(w_r) \bar{u} [A - w_r] -$$

$$\left[ \frac{(I_p + I_r)^2 \cdot (r_r + \Delta)}{f^3(w_r)} \phi\left(\frac{I_p + I_r}{f(w_r)}\right) + \frac{(I_r)^2 \cdot r_r}{f^3(w_r)} \phi\left(\frac{I_r}{f(w_r)}\right) \right] \quad (8.13)$$

En observant le second membre de l'équation (8.13) nous voyons que :

- le premier terme est toujours négatif car  $f'(w_r) < 0$ ,
- le second terme précédé du signe "-", est également négatif lorsqu'une solution admissible existe. En effet,  $f''(w_r) < 0$  car  $f$  est concave,  $A - w_r < 0$  car  $w_r > A$  lorsque la solution est admissible, et le signe "-" rend le terme négatif,
- le troisième terme, avec le signe qui le précède, est également négatif.

Finalement, la dérivée seconde de  $B$  par rapport à  $w_r$  est négative, ce qui achève la démonstration. ■

Le troisième résultat souligne l'unicité de la solution de notre problème.

### Résultat 3

Le problème qui consiste à maximiser la somme des bénéfices a une solution unique.

### Démonstration

Considérons les dérivées premières de  $B(I_p, I_r, w_r)$  par rapport aux trois variables.

$$\begin{aligned} \frac{\partial B}{\partial w_r} &= w_r \bar{u} \cdot f'(w_r) + \bar{u} \cdot f(w_r) - A \bar{u} \cdot f'(w_r) - \\ &(r_r + \Delta) \cdot \int_{\frac{I_p + I_r}{f(w_r)}}^{\infty} u \cdot \phi(u) du - r_r \int_{\frac{I_r}{f(w_r)}}^{\infty} u \cdot \phi(u) du \end{aligned} \quad (8.14)$$

$$\frac{\partial B}{\partial I_p} = -c_p + (r_r + \Delta) \int_{\frac{I_p + I_r}{f(w_r)}}^{\infty} \phi(u) du \quad (8.15)$$

et

$$\frac{\partial B}{\partial I_r} = -c_r + (r_r + \Delta) \int_{\frac{I_p + I_r}{f(w_r)}}^{\infty} \phi(u) du + r_r \int_{\frac{I_r}{f(w_r)}}^{\infty} \phi(u) du \quad (8.16)$$

Supposons que  $s = \{I_p^*, I_r^*, w_r^*\}$  soit une solution optimale. Alors :

$$\frac{\partial B}{\partial w_r}(I_p^*, I_r^*, w_r^*) = \frac{\partial B}{\partial I_p}(I_p^*, I_r^*, w_r^*) = \frac{\partial B}{\partial I_r}(I_p^*, I_r^*, w_r^*) = 0$$

De (8.16) nous déduisons :

$$(r_r + \Delta) \int_{\frac{I_p^* + I_r^*}{f(w_r^*)}}^{\infty} \phi(u) du + r_r \int_{\frac{I_r^*}{f(w_r^*)}}^{\infty} \phi(u) du = c_r$$

En remplaçant cette valeur dans (8.14) et en égalant à 0, il vient :

$$w_r^* \cdot \bar{u} \cdot f'(w_r^*) + \bar{u} \cdot f(w_r^*) - A \cdot \bar{u} \cdot f'(w_r^*) = c_r \quad (8.17)$$

De (8.17), nous déduisons de  $w_r^*$  est unique.

Considérons maintenant l'équation (8.15) elle peut se réécrire :

$$(r_r + \Delta) \int_{\frac{I_p^* + I_r^*}{f(w_r^*)}}^{\infty} \phi(u) du = c_p \quad (8.18)$$

$w_r^*$  étant unique, cela montre que  $I_p^* + I_r^*$  est unique à l'optimum.

Enfin, de (8.18) et (8.16) nous déduisons :

$$r_r \int_{\frac{I_r^*}{f(w_r^*)}}^{\infty} \phi(u) du = c_p + c_r \quad (8.19)$$

Nous voyons que  $I_r^*$  est unique donc que  $I_p^*$  est unique. Cela achève la démonstration. ■

## 8.4 Solution du problème

Nous donnons d'abord l'algorithme qui permet de maximiser le profit de l'ensemble des deux partenaires, c'est-à-dire du grossiste et du détaillant. Nous montrons ensuite comment partager ce profit de manière équitable.

### 8.4.1 Maximisation du profit total

L'algorithme qui conduit à la maximisation du profit total se déduit des résultats donnés dans le paragraphe 8.3.

#### Algorithme de maximisation du profit total

1. Initialisation : faire  $w_r = A$ .

*Bien entendu, cette solution n'est pas admissible*

2. Appliquer la méthode du gradient à la fonction  $\overline{K(I_p, I_r, w_r)}$  en fixant  $w_r$ .  
Nous obtenons ainsi un couple  $(I_p^1, I_r^1)$  qui minimise  $\overline{K(I_p, I_r, w_r)}$  (voir résultat 1).
3. Appliquer la méthode du gradient à la fonction  $B(I_p^*, I_r^*, w_r)$  en fixant  $(I_p^*, I_r^*)$ . Nous obtenons ainsi  $w_r^*$  qui maximise  $B(I_p^*, I_r^*, w_r)$  (voir résultat 2).
4. Recommencer en 2 temps que  $B(I_p, I_r, w_r)$  croit.

Compte tenu du résultat 3, cette solution est optimale. Nous passons maintenant à la seconde partie de la recherche de la solution.

### 8.4.2 Partage équitable du profit

Le profit est partagé proportionnellement aux investissements des partenaires, c'est-à-dire de sorte que :

$$\frac{b_p}{\text{Investissement du grossiste}} = \frac{b_r}{\text{Investissement du détaillant}}$$

$$\frac{b_p}{w_p \cdot f(w_r) \bar{u} - b_p} = \frac{b_r}{w_r \cdot f(w_r) \cdot \bar{u} - b_r}$$

soit

$$\frac{b_p}{A \cdot f(w_r) \bar{u} + c_p \cdot I_p + \overline{B_p}} = \frac{B(I_p, I_r, w_r) - b_p}{A \cdot f(w_r) \cdot \bar{u} + c_p \cdot I_p + b_p + c_r \cdot I_r + B_r}$$

Cette relation conduit successivement à :

$$\frac{b_p}{A \cdot f(w_r) \bar{u} + c_p \cdot I_p + \overline{B_p}} = \frac{B(I_p, I_r, w_r) - b_p}{A \cdot f(w_r) \cdot \bar{u} + c_p \cdot I_p + b_p + c_r \cdot I_r + B_r}$$

$$b_p(A \cdot f(w_r) \cdot \bar{u} + c_p \cdot I_p + b_p + c_r \cdot I_r + \overline{B_r})$$

$$= B(I_p, I_r, w_r) * (A \cdot f(w_r) \bar{u} + c_p \cdot I_p + \overline{B_p}) - b_p(A \cdot f(w_r) \bar{u} + c_p \cdot I_p + \overline{B_p})$$

Nous aboutissons à l'équation second degré :

$$b_p^2 + b_p(2A f(w_r) \bar{u} + 2c_p \cdot I_p + c_r \cdot I_r + \overline{B_r} + \overline{B_p} - B(I_p, I_r, w_r) \cdot (A f(w_r) \bar{u} + c_p \cdot I_p + \overline{B_p}))$$

La solution de cette équation est :

$$b_p = \frac{-Z + \sqrt{Z^2 + 4 * T}}{2} \quad (8.20)$$

où

$$Z = 2.f(w_r).\bar{u} + 2c_p.I_p + 2.\bar{B}_p + c_r.I_r + \bar{B}_r$$

$$T = (A.f(w_r).\bar{u} + c_p.I_p + \bar{B}_p) * B(I_p, I_r, w_r) \text{ et}$$

$$b_r = B(I_p, I_r, w_r) - b_p.$$

Nous illustrons cette approche par des applications numériques

## 8.5 Illustration numérique

Dans les deux exemples que nous présentons maintenant, nous utiliserons la fonction :

$$f(w_r) = 1 - \frac{w_r^2}{k} \quad \text{ou} \quad w_k^2 \leq k$$

pour représenter le coefficient qui appartient à  $(0, 1)$  par lequel est multiplié le besoin total des clients pour représenter la baisse des ventes lorsque le coût augmente. Cette fonction est décroissante et concave.

Dans le premier exemple, nous montrons comment évolue la solution au cours des itérations. Pour ce premier exemple, nous utilisons les paramètres suivants.

Densité de probabilité :

$$\phi(u) = \left\{ \begin{array}{ll} 1/(400) & \text{si } 100 \leq u \leq 500 \\ 0 & \text{sinon} \end{array} \right\}$$

$$c_r = 1.0 \quad c_p = 0.5$$

$$r_r = 5.0 \quad r_p = 6.50$$

$$\Delta = 1.0 \quad A = 1.0$$

$k = 500$  Le tableau 8.1 donne l'évolution de la solution au cours des 7 itérations nécessaires pour aboutir à la solution optimale.

TAB. 8.1 – Suite des itérations

Iteration No.	$w_r$	$B(I_p^*, I_r^*, w_r^*)$	$I_p^*$	$I_r^*$	$w_r^*$
1.	1.0	-480.704	6.35	459.219	12.5329 (Infeasible)
2.	12.5329	1931.29	4.4	315.41	12.8601
3.	12.8601	1936.5	4	307.91	12.9595
4.	12.9595	1937	4.2	305.509	12.9899
5.	12.9899	1937.04	3.8	304.909	13.0026
6.	13.0026	1937.05	4.1	304.509	13.0049
7.	13.0049	1937.05	4.1	304.409	13.0049

La première colonne du tableau donne les rangs des itérations.  $w_r$  (deuxième colonne) est la valeur initiale (première itération) ou la valeur obtenue à la fin de l'itération précédente. La troisième colonne, référencée  $B(I_p^*, I_r^*, w_r)$ , donne le profit total après optimisation par rapport à  $I_p, I_r$ . Les colonnes 4 et 5 donnent les valeurs de  $I_p^*, I_r^*$ . enfin, la dernière colonne donne  $w_r$  après optimisation de  $(B(I_p^*, I_r^*, w_r))$  par rapport à  $w_r$ .

TAB. 8.2 – Evolution de solution par rapport à  $f$ 

$k$	$w_r$	$B(I_p, I_r, w_r)$	$b_p$	$b_r$
100	6.26994	598.582	136.475	462.107
200	8.52749	1042.66	218.772	823.892
600	14.1751	2170.43	390.321	1780.11
1000	18.0642	2951.34	489.894	2461.45
5000	39.3991	7246.07	908.112	6337.96

La seconde série d'expériences donne les bénéfices  $b_p$  et  $b_r$  lorsque  $k$  augmente.  $k$  est le paramètre utilisé dans la définition de  $w_r$ . la fonction  $f(w_r)$  décroît d'autant moins rapidement que  $k$  est bénéficié des partenaires augmente avec  $k$ . Cela s'explique par le fait que les ventes sont, à l'optimum, d'autant moins pénalisées par le coût  $w_r$  que  $k$  est grand.

# Chapitre 9

## Conclusion

Les chaînes d'approvisionnement ont fait passer le gestionnaire d'une vue verticale (gestion par départements) à une vue horizontale (gestion par projets) de leur activité. Cette évolution est fondamentale. Elle entraîne, en outre, l'apparition de problèmes nouveaux, tant au niveau stratégique qu'aux niveaux tactique et opérationnel. De plus, ce changement nécessite la mise en place de nouvelles procédures de conception et d'évaluation.

La première partie de cette thèse s'est donnée comme objectif la mise en place d'un outil d'aide à la conception et à l'évaluation des chaînes d'approvisionnement au niveau stratégique. Nous retiendrons de cette première partie que toute chaîne d'approvisionnement gère cinq macro-activités (acheter, produire, transporter, stocker, et distribuer). Bien entendu, ces macro-activités recouvrent des activités différentes au niveau tactique, suivant le domaine auquel elles s'appliquent, mais l'utilisation de cette décomposition au niveau stratégique permet de simplifier l'analyse que doit mener le concepteur.

La mise en place d'une nouvelle chaîne d'approvisionnement se réduit à une prise de décision primaire. Une telle décision primaire peut concerner le lancement d'un nouveau type de produit, l'automatisation d'une filière existante, ou encore une modification en profondeur de la distribution. Une décision primaire se décline en décisions secondaires, chacune de ces décisions

---

secondaires concernant l'une des macro-activités énumérées ci-dessus. La déclinaison d'une décision primaire en décisions secondaires dépend, bien entendu, du type de système auquel elle s'applique. Nous retrouvons cependant certaines décisions secondaires dans la déclinaison de décisions primaires classiques (lancement d'un nouveau type de produit, par exemple). Ces décisions secondaires peuvent être stockées et servir d'aide-mémoire aux concepteurs futurs : c'est la démarche que nous avons utilisée dans le logiciel CAD-SC-SL.

Ces remarques justifient la manière dont le logiciel CAD-SC-SL a été conçu. Il permet de stocker des décisions primaires avec les décisions secondaires qui leurs correspondent, permettant ainsi au concepteur de tirer profit des expériences antérieures. Les décisions secondaires stockées peuvent être prélevées et rattachées à une nouvelle décision primaire ou à une décision primaire déjà définie.

Les décisions primaires se réduisent au nom qui leur est donné. Les décisions secondaires sont caractérisées par deux coûts : le coût d'investissement et le coût de fonctionnement par période élémentaire. La période élémentaire peut-être le mois, le trimestre ou l'année, suivant le type de chaîne d'approvisionnement considérée. Pour faciliter la définition de ces coûts, le logiciel aide l'utilisateur en procédant en trois temps.

L'utilisateur est d'abord appelé à décider si le coût est de type quantitatif ou non. Un coût est qualifié de "non quantitatif" s'il ne dépend d'aucun des paramètres qui caractérisent la chaîne d'approvisionnement. Le terme de "coût fixe" eut été préférable si ce terme n'avait pas été utilisé par ailleurs. Un coût est qualifié de "quantitatif" dans le cas contraire. Plusieurs coûts quantitatifs attachés à une même chaîne d'approvisionnement peuvent ne pas dépendre du même paramètre, d'où un mécanisme qui permet de ramener tous ces coûts à un paramètre unique. Plusieurs aspects importants doivent être soulignés à cet instant :

(i) Les coûts considérés sont des coûts incrémentaux. En d'autres termes, seuls les coûts qui s'ajoutent aux coûts antérieurs à l'introduction de la nouvelle chaîne d'approvisionnement sont pris en considération.

(ii) Les coûts sont attachés aux décisions secondaires : cela implique que si plusieurs partenaires participent à la même décision secondaire, d'autres mécanismes existent pour répartir ces coûts - et donc également les bénéfices - entre eux ; ces mécanismes sont connus sous le nom de "sharing processes".

Enfin, le logiciel CAD-SC-SL que nous avons développé permet d'optimiser la valeur du paramètre unique lorsque l'utilisateur est capable de fournir une borne inférieure et une borne supérieure pour ce paramètre.

Il manque à ce logiciel un habillage qui le rende encore plus simple d'utilisation, mais sa version actuelle est déjà d'utilisation aisée. Une surcouche est prévue afin d'y intégrer le "sharing process" lorsque plusieurs sociétés sont impliquées dans la même chaîne d'approvisionnement.

La seconde partie de la thèse concerne les problèmes nouveaux qui apparaissent au niveau stratégique. Nous retiendrons, dans cette seconde partie, les problèmes qui concernent la sélection des partenaires d'une chaîne d'approvisionnement. Nous en avons examiné deux. Le premier problème consiste à sélectionner, dans un système de fabrication à trois niveaux, les partenaires qui conduiront à la chaîne d'approvisionnement la moins onéreuse. Les coûts considérés sont les coûts d'investissement et de fonctionnement sur un horizon donné. Dans ce problème, le nombre de partenaires sélectionnés à chaque niveau n'est pas limité. Le second problème considéré est un problème multi-niveaux auquel s'appliquent deux contraintes : (i) il n'est permis de choisir qu'un seul partenaire par niveau, et (ii) il n'est pas question d'investir, ce qui signifie que seules les capacités disponibles peuvent être utilisées. Les coûts de transport inter-niveaux ont une importance majeure dans ce problème. Le lecteur comprendra que de nombreux autres modèles restent à étudier lorsqu'il

s'agit du choix des partenaires, et une modification même légère des contraintes qui s'appliquent à ces choix risque de modifier fondamentalement la nature mathématique du problème. L'optimisation d'autres modèles de sélection des partenaires de chaînes d'approvisionnement sera une des voies que je suivrai dans ma recherche future.

La troisième partie de la thèse vise à donner les moyens d'une réactivité importante sur tout le spectre des activités de production. L'ordonnancement en temps réel des systèmes d'assemblage est la partie centrale de cette dernière partie, avec son corollaire qui est la gestion des stocks intermédiaires. Rappelons que cette gestion est capable de limiter le temps de séjour entre deux opérations successives et, si on le souhaite, de limiter le nombre d'unités stockées à chaque instant entre deux opérations successives.

De nombreux problèmes restent à examiner à ce niveau. En particulier, le critère qui est actuellement la minimisation du temps de fin de fabrication peut devenir, dans certains cas, la minimisation du coût de l'énergie consommée : ce problème reste ouvert.

Enfin, le problème de marketing qui s'intéresse au prix de vente souhaité pour augmenter la compétitivité face à la concurrence n'a été qu'effleuré dans le dernier chapitre. Il mérite d'être analysé en profondeur, mais cette analyse dépend d'une étude détaillée des réactions des clients dans un environnement de compétition, et cette étude reste à faire.

## Quatrième partie

### Bibliographie

# Bibliographie

- [AG86] Afentakis, P. and Gavish, B. (1986), Optimal lot sizing algorithms for complex product structures, *Operations Research*, 34, pp 237-244.
- [AK84] Afentakis, P. and Karmarkar, U. (1984), Computationally efficient solutions to the lot sizing problem in multistage assembly systems, *Management Science*, 30, pp 222-239.
- [AM88] Ansari, A. and Modarress, B. (1988), Jit purchasing as a quality and productivity centre, *International Journal of Production research*, 26(1) pp. 19-26.
- [Ber85] Bertand, J. (1985), Multiproduct optimal batch sizes with in-process inventories and multi work centers, *IIE Transactions*, 17, pp 157-163.
- [BK90] Benton, W. and Krajewski, L. (1990), Vendor performance and alternative manufacturing environments, *Decision Sciences*, 21, pp 403-415.
- [Bla87] Blazewicz, J. (1987), Selected topics in scheduling theory, *Annals of Discrete Mathematics*, 31, pp 1-60.
- [BPV83] Bolignano, D., Proth, J.-M., and Voyatzis, K. (1983), Ordonnement dans un système de traitement de surface, *Rapport de Recherche INRIA*, No. 234.

- [BSSW94] Bretthaire, K., Shetty, B., Syam, S., and White, S. (1994), A model for resource constrained production and inventory management., *Decisions Sciences*, 25(4) pp. 561-580.
- [Cal71] Callahan, J. (1971), *The nothing hot delay problems in the production of steels*, Ph.D. thesis, Department of Industrial Engineering, University of Toronto, Canada.
- [CAP03] Chauhan, S., Alexandre, D., and Proth, J.-M. (2003), Ordering policy in an assembly system under lead-time uncertainty, *INRIA Research Report*, RR-4802.
- [CCLL95] Chretienne, P., Coffman, J., E.G., Lenstra, J., and Liu, Z. (1995), *Scheduling theory and its application*, Wiley, NewYork, U.S.A.
- [CcP98] Chen, H., chu, C., and Proth, J.-M. (1998), Job-shop scheduling using lagrangian relaxation, *IEEE Transactions on Robotics and Automation*, 4, n°5.
- [CDP02] Chauhan, S., Dridi, N., and Proth, J.-M. (2002), Cost optimization in water a supply system, *INRIA Research Report*.
- [CDP03] Chauhan, S., Duron, C., and Proth, J.-M. (2003), *Les chaînes d'approvisionnement*, Hermès, ISBN 274620617X.
- [CEKS02] Chauhan, S., Ereemeev, A. V., Kolokolov, A. A., and Servakh, V. V. (2002), On solving concave cost supply management problem with single manufacturing unit, *SCM Conference, Poland*, page 147-154.
- [CGP02] Chauhan, S., Gordon, V., and Proth, J.-M. (2002), No-wait scheduling in supply chain environment, *INRIA Research Report*, RR-4467.
- [Cha99] Chauvet, F. (1999), *Ordonnancement en temps réel dans les problèmes à en-cours limités*, Ph.D. thesis, Université de Metz.

- [CHH96] Choi, T., Hartley, J., and Hahn, C. (1996), Defining supply chain management : Synthesis and implications, *Working paper, Department of Management, Bowling Green state University, Bowling Green, OH.*
- [Chr92] Christopher, M. (1992), *Logistics and supply chain management : strategies for reducing costs and improving services*, Financial time : Pitman publishing, ISBN : 0273034154.
- [Chr97] Christensen, C. (1997), *The innovator's dilemma : When new technologies cause great firms to fail*, Harvard business school press, Boston, MA, USA.
- [Chr98] Christopher, M. (1998), *Logistics and supply chain management : strategies for reducing costs and improving services*, Second edition, Financial time : Pitman publishing, ISBN : 0273034154.
- [CL88] Cohen, M. and Lee, H. (1988), Strategic analysis of integrated production-distribution systems : models and methods, *Operations Research*, 36(2), pp 216-228.
- [CL99] Cachon, G. and Lariviere, M. (1999), "Capacity choice and allocation : strategic behavior and supply chain performance.", *Management Science*, 45.
- [CL00] Cachon, G. and Lariviere, M. (2000), "Supply chain coordination with revenue sharing contracts : Strength and limitations", *Working paper, The Wharton School of Business, University of Pennsylvania, Philadelphia.*
- [CLP00] Chauvet, F., Levner, E., and Proth, J.-M. (January 2000), Online part scheduling in a surface treatment system, *Journal of Operational Research, IJOR.*, vol. 120/2.

- [CNP02] Chauhan, S., Nagi, R., and Proth, J.-M. (2002), Strategic capacity planning in supply chain design for a new market opportunity, *Inria research report*, RR-4658.
- [Cop97] Copacino, W. (1997), *Supply chain management : The basics and beyond*, The St. Lucie press, Boca Raton, Florida, APICS Series on resource management, Falls church, Verginia.
- [CP01a] Chauhan, S. and Proth, J.-M. (2001), A supply chain model at the strategic level, *INRIA Research Report*, RR-4274.
- [CP01b] Chauvet, F. and Proth, J.-M. (2001), On-line scheduling in assembly processes, *Information Systems and Operation Research*, vol.39 n°1.
- [CP03a] Chauhan, S. and Proth, J.-M. (2003), Concave cost supply problem, *European Journal of Operational Research*, 148, pp. 374-383.
- [CP03b] Chauhan, S. and Proth, J.-M. (2003), Supply chain partnership based on revenue sharing, *INRIA Research Report*, RR-4846.
- [CPW01] Chauvet, F., Proth, J.-M., and Wardi, Y. (2001), Scheduling no-wait production with time windows and flexible processing times, *IEEE Transactions on Robotics and Automations*, 17, n°1, pp. 60-69.
- [CRJMM02] Chauhan, S., Rakesh, N., Jean-Marie, P., and Maria, S. A. (2002), Strategic supply chain design for a new market opportunity from qualified partner set, *INRIA Research Report*, RR-4508.
- [CWW73] Crowston, W., Wagner, M., and Williams, J. (1973), Economic lot size determination in multistage assembly systems, *Management Science*, 19, pp 517-526.

- [DOL01] Dolgui, A. and Ould-Louly, M.-A. (2001), "Supply chain planning under uncertainties.", In *P.Groumpos and A.Tzes(eds), Manufacturing, Modelling, Management and Control : A Proceedings Volume of the IFAC Symposium (Amsterdam : Pergamon Elsevier)*.
- [DOL02] Dolgui, A. and Ould-Louly, M.-A. (2002), "Generalized newsboy model to compute the optimal planned lead times in assembly systems.", *International Journal of Production Research*, 40.
- [DS90] Dueck, G. and Scheuer, T. (1990), Threshold accepting : A general purpose optimization algorithm appearing superior to simulated annealing, *Journal of computational Physics*, 90, pp 161-175.
- [Dur02] Duron, C. (2002), *Ordonnancement en temps réel*, Ph.D. thesis, Université de Metz.
- [EG98] Emmons, H. and Gilbert, S. (1998), "The role of returns policies in pricing and inventory decisions for catalogue goods", *Management Science*, 44.
- [ER81] Eggli, U. and Rippin, D. (1981), Short-term scheduling for multi-product batch chemicals plants, *American Institute of Chemical Engineers Meeting, Houston, Texas, USA*.
- [ESV99] Ereng, S., Simpson, N., and Vakharia, A. (1999), Integrated production/distribution planning in supply chains : An invited review, *European Journal of Operational Research*, 115, pp 219-236.
- [Fin98] Fine, C. (1998), *Clock speed. winning industry control in the age of temporary advantage*, Parseus books, reading, Massachusetts, Library of congress catlog number 98-86945.
- [Fos96] Foster, T. (1996), Supply chain black holes, *Distribution*, 63.

- [GAL96] Gurnani, H., Akella, R., and Lehoczky, J. (1996), "Optimal order policies in assembly systems with random demand and random supplier delivery.", *IIE Transactions*, 28 : 865–878.
- [Gat98] Gattorna, J. e. (1998), *Strategic supply chain alignment. Best practice in supply chain management*, Gover publishing limited, Gover House, Creft road, Aldershot, Hamshire GU11 3hr, England. ISBN 0-566-07825-2.
- [GG74] Geoffrion, A. and Graves, G. (1974), Multicommodity distribution system design by benders' decomposition, *Management Science*, 20(5) pp. 822-844.
- [GG80] Gavish, B. and Graves, S. (1980), A one-product production/inventory problem under continuous review policy, *Operations Research*, 28, pp 1288-1235.
- [GLLR79] Graham, R., Lawler, E., Lenstra, J., and Rinnoykan, A. (1979), Optimization and approximation in deterministic sequencing and scheduling : a survey, *Annals of Discrete Mathematics*, 5, pp 287-326.
- [Gol89] Goldberg, D. (1989), *Genetic Algorithm in Search Optimization and Machine Learning*, Addison-Wesley, New York.
- [GP02] Govil, M. and Proth, J. M. (2002), *Supply chain design and management. strategic and tectical perspectives*, Academic press, ISBN 0-12-294151-9, Library of Congress Catlog Card Number 2001094751.
- [GQSL96] Gellego, G., Queyranne, M., and Simchi-Levi, D. (1996), Single resource multi-item inventory systems, *Operations Research*, 44(4) pp 580-595.

- [GS86] Golden, B. and Skiscim, C. (1986), Using simulated annealing to solve routing and location problems, *Naval Research Logistics quarterly*, 33, pp. 261-279.
- [GT84] Grasso, E. and Taylor, B. (1984), "A simulation based experimental investigation of supply/timing uncertainty in mrp systems.", *International Journal of Production Research*, 22 : 485-497.
- [GWY94] Gerchak, Y., Wang, Y., and Yano, C. (1994), "Lot sizing in assembly systems with random components yields.", *IIE Transactions*, 26 : 19-24.
- [HA92] Heragu, S. and Alfa, A. (1992), Experimental analysis of simulated annealing based algorithm for the layout problem, *European Journal of Operational Research*, 57, pp. 190-202.
- [Han59] Hanssmann, F. (1959), Optimal inventory location and control in production and distribution networks, *Operations Research*, 7, pp 483-498.
- [HS96] Hall, N. and Sriskandarajah, C. (1996), A survey of machines scheduling problems with blocking and no-wait process, *Operations Research*, 44, pp 510-525.
- [JSP02] Jonatan, G., Shah, N., and Papageorgiou, L. G. (2002), "Fair transfer proce and inventory holding policies in two-enterprise supply chains", *European Journal of Operational Research*, 143 : 582-599.
- [KGV83] Kirkpatric, S., Gelatt, C., and Vecchi, M. (1983), Optimization by simulated annealing, *Science*, 220, pp 671-680.
- [KK94] Kogut, B. and Kulatilaka, N. (1994), Operationg flexibility, global manufacturing, and the option value of a multinational network, *Management Science*, 40(1), pp. 123-139.

- [KL94] Kamath, R. and Liker, J. (1994), A second look at japanese product development, *Harvard Business Review*, 154-170.
- [KR96] Krajewski, L. and Ritzman, L. (1996), *Operations Management : Strategy and Analysis*, Addison-Wesley, Reading, MA.
- [LB92] Lee, H. and Billington, C. (1992), Managing supply chain inventory : Pitfalls and opportunities, *Sloan Management Review*, 65-73.
- [LB93a] Lee, H. and Billington, C. (1993), Material management in decentralized supply chains, *Operations Research*, 41(5), pp 835-852.
- [LB93b] Lee, H. and Billington, C. (1993), Material management in decentralized supply chains., *Operations Research*, 41(5) pp. 835-852.
- [LBC93] Lee, H., Billington, C., and Carter, B. (1993), Hewlett-Packard gains controls of inventory and services through design for localization, *Interfaces*, 23, pp. 1-11.
- [LMR94] Larsson, T., Migdalas, A., and Ronnqvist, M. (1994), A langrangian heuristic for the capacited concave minimum cost network flow problem, *European Journal of Operational Research*, 78, pp 116-129.
- [LW99] Lee, H. and Whang, S. (1999), "Decentralized multi-echelon supply chains : incentives and information", *Management Science*, 45.
- [MDE93] Martin, C., Dent, D., and Echart, J. (1993), Integrated production, distribution, and inventory planning at libbey-owens-ford, *Interfaces*, 23(3), 68-78.

- [MK93] Maloney, B. and Klein, C. (1993), Constrained multi-item inventory systems : An implicit approach, *Operations Research*, 20(6) pp. 639-649.
- [MN98] Mazzola, J. and Neebe, A. (1998), Establishing consistent and realistic reorder intervals in production-distribution systems, *Operations Research*, 33, pp. 1316-1344.
- [Mol97] Molinder, A. (1997), "Joint optimisation of lot-sizes, safety stocks and safety lead times in an mrp system.", *International Journal of Production Research*, 35 : 983-994.
- [NDZ02] Nagurney, A., Dong, J., and Zhang, D. (2002), A supply chain network equilibrium model, *Transportation research, Part E* 38(2), pp 281-303.
- [O'N87] O'Neal, C. (1987), The buyer-seller linkage in a just-in-time environment, *Journal of Purchasing and Materials Management*, 7-13.
- [OSM98] Orman, A., Shahani, A., and Moore, A. (1998), Modelling for the control of a complex radar system, *Computers and Operations Research*, 25, n°3, pp 239.
- [PMW<sup>+</sup>97] Proth, J.-M., Mauroy, G., Wardi, Y., Chu, C., and Xie, X. (1997), "Supply management for cost minimization in assembly systems with random component yield times", *Journal of Intelligent Manufacturing*, 8 : 385-403.
- [Rek82] Reklaitis, G. (1982), Review of scheduling of process operations, *AIChE Symposium Series*, 78, pp.119-133.
- [RR72] Reddi, S. and Ramamoorthy, C. (1972), On the flow shop sequencing problem with no-wait in process, *Operations Research Quarterly*, 23, pp. 323-331.

- [RR92] Ramudhin, A. and Ratliff, H. (1992), Generation daily production schedule in process industries, *Working paper, Faculté des Sciences de l'Administration, Université de Laval, Québec, Canada.*
- [Sal73] Salvador, M. (1973), *A solution of a special class of flowshop scheduling problems, in the proceedings of the Symposium on the Theory of Scheduling and its Applications*, Springer-Verleg, Berlin, Allemagne.
- [SSL95] Schary, P. and Skjott-Larsent, T. (1995), *Managing global supply chain*, Handelshojskolens forlag. Distribution : Munksgaard International Publisher Ltd, Copenhagen, ISBN 87-16-13278-5.
- [TTM89] Trigeiro, W., Thomas, L., and McClain, J. (1989), Capacitated lot sizing with setup times, *Management Science*, 35(3) pp. 353-366.
- [Und96] Underhill, T. (1996), *Strategic alliances : managing the supply chain*, Pennwell Books, Tulsa, Okla.
- [VG97] Vidal, C. and Goetschalckx, M. (1997), Strategic production-distribution models : A critical review with emphasis on global supply chain models, *European Journal of Operational Research*, 98, pp. 1-18.
- [WCB91] Weber, C., Current, J., and Benton, W. (1991), Vendor selection criteria and methods, *European Journal of Operational Research*, 50, pp 2-18.
- [Wil81] William, J. (1981), Heuristic techniques for simulataneous scheduling of production and distribution in multi-echelon structures : Theory and empirical comparisons, *Management Sciences*, 27, pp 336-352.

- [Wil84] Williams, M., J (1984), Special products and uncertainty in production/inventory systems, *European Journal of Operational Research*, 15, pp 46-54.
- [WW76] Whybark, D. and Williams, J. (1976), "Material requirements under uncertainty.", *Decision Science*, 7 : 595–606.
- [YL99] Yan, S. and Luo, S.-C. (1999), Probabilistic local search algorithms for concave cost transportation network problems, *European Journal of Operational Research*, 117, pp 511-521.
- [YWM98] Yeung, J., Wong, W., and Ma, L. (1998), "Parameters affecting the effectiveness of mrp systems : a review.", *International Journal of Production Research*, 36 : 313–331.





# Chaînes d'approvisionnement : approches stratégique et tactique.

*Thèse de Doctorat soutenue par Satyaveer Singh Chauhan*

**Résumé** - Cette thèse se divise en trois parties. La première est consacrée à la conception d'une chaîne d'approvisionnement au niveau stratégique. Une approche systématique est proposée; elle est construite à partir de cinq macro-activités. Un logiciel a été développé à partir de cette approche. Il est présenté en détails.

Dans la seconde partie de la thèse, des problèmes de recherche opérationnelle liés aux chaînes d'approvisionnement vues au niveau stratégique sont exposés. Il s'agit en particulier de problèmes relatifs à la sélection des partenaires d'une chaîne d'approvisionnement.

Enfin, la dernière partie de la thèse est dédiée à des problèmes sensibles au niveau tactique dans l'environnement des chaînes d'approvisionnement. Il s'agit d'ordonnancement en temps réel, d'un problème d'assemblage lorsque les composants sont sous-traités et de la conception d'un processus de partage des bénéfices et des pertes entre partenaires.

**Mots-clés** - Chaîne d'approvisionnement, Conception, Ordonnancement et temps réel, Affectation Partage des parties et des bénéfices, Assemblage, Evaluation, Modélisation.

# Supply chains : strategic and tactical approaches.

*Ph. D. Thesis by Satyaveer Singh Chauhan*

**Abstract** - This thesis is divided in three parts : The first part is devoted to the design of a supply chain at the strategic level. A systematic approach is proposed based on five macro-activities. Software has been developed to support this approach. It is presented in detail. In the second part of the thesis, solutions of O.R. problems related to supply chains at the strategic level are proposed. In particular, models related to the selection of partners are proposed. The last part of the thesis is dedicated to problems that are the most sensitive in a supply chain environment that is real time scheduling, assembly problem when components are outsourced and definition of a sharing process in a specific example.

**Keywords** - Supply chains, Design, Real-Time Scheduling, Assignment, Sharing Process, Assembly, Evaluation, Modelling.

---

INRIA-Lorraine  
Université de Metz - Faculté des Sciences  
UFR Mathématiques, informatique, mécanique  
Ile du Saulcy - 57000 Metz

Cette thèse a été préparée à l'INRIA (Institut National de Recherche en Informatique et en Automatique) de Octobre 2000 à septembre 2003 et co-financée par la région Lorraine.