



HAL
open science

Modèles génériques et méthodes de résolution pour la planification tactique mono-site et multi-site

David Lemoine

► **To cite this version:**

David Lemoine. Modèles génériques et méthodes de résolution pour la planification tactique mono-site et multi-site. Recherche opérationnelle [math.OC]. Université Blaise Pascal - Clermont-Ferrand II, 2008. Français. NNT : 2008CLF21892 . tel-00860292

HAL Id: tel-00860292

<https://theses.hal.science/tel-00860292>

Submitted on 10 Sep 2013

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.

N° d'Ordre : D.U. 1892
EDSPIC : 425

UNIVERSITE BLAISE PASCAL - CLERMONT-FERRAND II
ECOLE DOCTORALE
SCIENCES POUR L'INGENIEUR DE CLERMONT-FERRAND

THESE

présentée par
David LEMOINE

Ingénieur Diplômé du Conservatoire National des Arts et Métiers
Diplômé du Master de Recherche

pour obtenir le grade de

Docteur d'université

Spécialité : INFORMATIQUE

**Modèles Génériques et Méthodes de Résolution
pour la Planification Tactique
Mono-site et Multi-site**

Soutenue publiquement le 04 décembre 2008 devant le jury :

Madame	Valérie Botta-Genoulaz	Rapporteur
Monsieur	Philippe Chrétienne	Rapporteur
Monsieur	Stéphane Dauzère-Pérès	Rapporteur
Monsieur	Imed Kacem	Examineur
Monsieur	Michel Gourgand	Directeur de thèse
Madame	Sylvie Norre	Directeur de thèse

*A moi qui tout à l'heure incertain, agité,
Et sur les flots du doute à tout vent ballotté,
Cherchais le bien, le vrai, dans les rêves des sages,
Et la paix dans des cœurs retentissants d'orages.
A peine sur mon front quelques jours ont glissé,
Et que, séparé d'eux par un abîme immense,
Un nouvel homme en moi renaît et recommence.*

A. de Lamartine

C'est très bien ! Mais,

S. Norre

Remerciements

A l'issue de mon « année à plein temps » clôturant le cursus d'ingénieur C.N.A.M., Michel Gourgand m'a proposé d'entamer une nouvelle aventure (car c'est bien de cela dont il s'agit). Et cette aventure, une idée presque folle à laquelle Michel Gourgand croyait sans doute beaucoup plus que moi-même, trouve sa conclusion dans ce manuscrit. J'aimerais donc, en tout premier lieu, remercier Michel Gourgand pour avoir suscité en moi l'envie de faire cette thèse.

Je tiens également à remercier très chaleureusement Valerie Botta-Genoulaz, Philippe Chrétienne et Stéphane Dauzère-Pérès pour m'avoir fait l'honneur de rapporter ce travail ainsi qu'Imed Kacem pour avoir accepté de faire partie de mon jury de thèse.

Les travaux présentés dans ce manuscrit ont été réalisés au Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes (LIMOS) dirigé par Alain Quilliot, à l'Université Blaise Pascal (Clermont Ferrand II), dans l'équipe « Modélisation et Aide à la Décision », animée par Michel Gourgand. Ils ont été financés par le conseil général de l'Allier à qui j'adresse mes plus sincères remerciements pour la bourse dont ils m'ont fait bénéficier durant ces trois années.

Je n'aurais pu mener à bien (et au bout) ce travail si je n'avais pas été conseillé et guidé par mes deux directeurs de thèse, Michel Gourgand et Sylvie Norre. Leur soutien, leurs conseils avisés ainsi que leur grande rigueur et honnêteté scientifiques alliés à leur très grande disponibilité ont permis à cette thèse d'exister. Leurs encouragements m'ont été d'une aide très précieuse, et je tiens encore une fois à les remercier de tout cœur pour ces trois années passées ensemble.

J'aimerais également avoir une pensée pour Michael Comelli dont la collaboration m'a beaucoup enrichi. Merci pour le travail que nous avons accompli ensemble et pour ton sens de l'humour et de la dérision que même « Nidami » ne saurait te renier. Cependant cette collaboration n'aurait pas été ce qu'elle fut sans la présence de « Gipi », qui m'a même recueilli un temps, lorsque j'étais à plusieurs domiciles pas très fixes. Merci Patrice pour ta disponibilité et ton amitié.

Je tiens également à remercier toute l'équipe « Modélisation et Aide à la Décision » pour l'accueil et le soutien qu'elle m'a apportés durant ces trois ans et en particulier Nathalie, my crystallized fruits designer as well as my personal tour operator. Je crois que je te dois, à toi et à tes 72831 pâtes de fruit, les quelques kilos que j'ai pu prendre durant ma thèse et qui m'ont valu quelques surnoms et affiches placardées sur la porte de mon bureau... Une pensée également pour Laurent au flegme quasi british et pour David, mon garagiste personnel.

Je remercie également le personnel de l'IUT de Montluçon pour l'accueil qu'ils m'ont réservé et les facilités qu'ils m'ont faites pour mener à bien mes recherches. Merci au département GLT pour m'avoir accueilli durant ces trois années ! Merci pour votre soutien Anne-céline, Anne-Claire, Christine, Catherine, Cécile, Marc, Nadine, Virginie et ... j'en oublie ! Merci à mes étudiants que j'ai sans doute fait (parfois, pas trop quand même) souffrir, et qui m'ont tellement apporté durant toutes ces années.

Je tiens également à manifester l'affection que j'ai aussi pour les autres personnes qui ont fait partie

de ma vie durant cette thèse et même avant.

En plus de ma famille, je pense à tous mes amis proches et très proches (*Béatrice, Bruno, Jean-Jacques, Olivier, Pascale, ...*) que je ne citerai pas nommément tant la liste est longue mais qui sont bien présents au plus profond de mon cœur. Merci Olivier pour ton soutien et pour être toi.

Mes remerciements seraient incomplets si je n'y associais pas mes trois « compères » : Damien (Macrobio), Frank (Ada Bouchra's spiritual son) et Jérôme (The Business Man) avec qui j'ai suivi la majorité des cours du C.N.A.M. et qui m'ont permis, par leur motivation et leur amitié, d'en être là aujourd'hui. Merci Frank d'avoir été présent.

Une pensée pour deux personnes qu'il m'est inutile de nommer mais dont je sens plus que je ne ressens la présence, tous les jours d'avantage, et qui ont fait de ce que j'étais, ce que je suis désormais.

Certes, c'est très bien tout ça, mais... *C'est quand qu'on mange de la brioche ?* (Gourgand *et al.* 2007)

Table des matières

Introduction	15
1 La problématique de la planification tactique	19
Introduction	20
I L'approche MRP II	20
1 Les différents plans dans MRP II	21
2 MRP II et les niveaux décisionnels de la planification industrielle	26
3 Les limites de MRP II	27
II La chaîne logistique et le management de la chaîne logistique	28
1 Notion de chaîne logistique	28
2 Le management de la chaîne logistique	30
III Classification des modèles mathématiques pour la planification tactique	36
1 Les modèles mono-site	38
2 Les modèles multi-site	55
Conclusion	59
2 Proposition de modèles génériques pour la planification tactique	61
Introduction	63
I La méthodologie ASCI	63
1 Analyse et spécification	65
2 Conception et implantation	66
3 Analyse des résultats de l'étude et prise de décisions	67
II Modèle de connaissance générique pour la planification multi-site	67
1 Le sous-système physique	67
2 Le sous-système logique	73
3 Le sous-système décisionnel	74
4 Les dépendances entre les différents sous-systèmes	77
5 Le modèle UML représentant les trois sous-systèmes et les liens les unissant	80
III Modèle mathématique générique pour la planification multi-site	82
1 Notations mathématiques des ressources « production »	83
2 Notations mathématiques pour les ressources « stock »	85
3 Modélisation des ressources « transport »	87
4 Modélisation des flux de produits	88
IV Deux exemples d'instanciation du modèle générique	95
1 Le MLCLSP	96
2 Un modèle de planification multi-site avec prise en compte des transports	100
Conclusion	105

3	Etat de l'art et proposition de méthodes pour le Capacitated Lot Sizing Problem	109
	Introduction	111
I	Le modèle mathématique	111
	1 Le modèle de base	111
	2 La reformulation en problème de plus court chemin	112
	3 Le modèle approximatif de Manne	114
	4 Les extensions du CLSP	115
II	Etat de l'art sur les méthodes de résolution	117
	1 Calcul de bornes inférieures	117
	2 Méthodes pour le modèle du CLSP	119
	3 Méthodes pour le modèle du CLSP avec Setup Time	123
	4 Conclusion sur l'état de l'art	128
III	Proposition de méthodes pour le CLSP	128
	1 Calcul d'une borne inférieure	129
	2 Chaînage de deux métaheuristiques	132
	3 La Métaheuristique « bi-objectif »	147
	4 Mise en œuvre et résultats	148
IV	Adaptation au CLSP avec Setup Time	154
	1 Prise en compte des temps deancements	154
	2 Mise en œuvre et résultats	157
	Conclusion	158
4	Méthodes approchées pour la planification d'une chaîne logistique de type Flow-Shop Hybride	159
	Introduction	160
I	Modèle mathématique pour une chaîne logistique de type flow-shop hybride	161
	1 Notations utilisées	162
	2 Les contraintes du modèle mathématique	164
	3 La fonction objectif	168
	4 Le modèle mathématique définitif	168
II	Etat de l'art	170
	1 Méthodes pour le MLLP	170
	2 Méthodes pour le MLCLSP	174
	3 Conclusion	181
III	Méthodes d'optimisation pour la planification d'une chaîne logistique de type Flow-Shop Hybride	181
	1 Résolution du problème de planification des « usines étages »	182
	2 Résolution du problème d'affectation	191
	3 La méthode d'optimisation pour le problème de planification de la chaîne logistique de type flow shop hybride	192
	4 Mise en œuvre et résultats	192
IV	Vers une nouvelle approche	195
	1 Résolution du problème « usines étages »	195
	2 Résolution du problème d'affectation	197
	3 Une nouvelle méthode d'optimisation	198
	Conclusion	198
5	Pistes de recherche pour une synchronisation des plans tactique et opérationnel	201
	Introduction	202
I	Etat de l'art	202
	1 Intégration de contraintes opérationnelles	202
	2 Estimation des capacités du système de production	203

II	Approche par couplage de modèles mathématiques	206
III	Approche par couplage d'une méthode approchée et d'un modèle de simulation	209
IV	Illustration de l'approche utilisant la simulation sur la planification d'un atelier de type Flow-Shop hybride	211
	Conclusion	219
6	Gestion des stocks à demande différenciée : application à un contexte industriel	221
	Introduction	222
I	Le contexte de l'étude	223
1	Les principales politiques de gestion des stocks	223
2	La gestion des stocks à demande différenciée	224
3	Les travaux du LIMOS	224
II	Le projet	226
1	Etude de faisabilité et de condition du test	226
2	Conception de l'application informatique permettant le calcul des seuils	235
	Conclusion	240
	Conclusion	240
	Annexes	253
	Annexe 1 - Comparaison des modèles de lot-sizing mono-niveau	256
1	Comparaison des trois modèles «small bucket»	256
2	Comparaison entre le CLSP et le GLSP	259
	Annexe 2 - Les procédures de réparations	263

Liste des tableaux

1.1	Les hypothèses prises en compte par les modèles de la littérature	59
2.1	grille de lecture du chapitre	83
2.2	Le modèle générique proposé et les modèles de la littérature	106
2.3	Prise en compte des différents coûts dans les modèles multi-site de la littérature	106
2.4	Prise en compte des différentes contraintes dans les modèles multi-site de la littérature	107
3.1	Etat de l'art pour le CLSP	122
3.2	Etat de l'art pour le CLSP avec setup time	127
3.3	Résultats du calcul de borne inférieure	149
3.4	Résultats de la méthode d'optimisation sur les instances répertoriées	150
3.5	Tableau comparatif des résultats pour les deux métaheuristiques	151
3.6	Résultats obtenus par la métaheuristique pour le CLSP sur les instances de (Eppen et Martin 1987)	152
3.7	Comparaison des résultats obtenus avec ceux de la littérature	153
3.8	Résultats pour le CLSP avec setup time	157
4.1	Etat de l'art pour le MLLP	173
4.2	Etat de l'art pour le MLCLSP	180
4.3	Résultats obtenus sur les instances de petite taille	193
4.4	Décomposition des coûts	194
4.5	Résultats obtenus sur les instances de taille industrielle	195
5.1	La demande client à satisfaire	212
5.2	Temps de gamme pour les produits finis considérés	212
5.3	Coûts engendrés par la fabrication des produits	213
5.4	Planning obtenu à la première itération	214
5.5	Makespans obtenus à la première itération	214
5.6	Planning obtenu à la seconde itération	215
5.7	Makespans obtenus à la seconde itération	215
5.8	Planning obtenu à la troisième itération	216
5.9	Makespans obtenus à la troisième itération	216
5.10	Planning obtenu à la quatrième itération	217
5.11	Makespans obtenus à la troisième itération	217
5.12	Planning obtenu à la dernière itération	218
5.13	Makespans obtenus à la dernière itération	218
7.1	jeu d'essai : demandes et capacités « small bucket »	256
7.2	jeu d'essai : Stock initial, Capacité requise, Coût de stockage, de lancement	256
7.3	Résultats obtenus au moyen de Cplex	258
7.4	jeu d'essai : demande et capacité « big bucket »	260
7.5	jeu d'essai : Stock initial, capacité requise, Coût de stockage, de lancement « big bucket »	260
7.6	Résultats obtenus au moyen de Cplex	260

7.7	Tableau récapitulatif pour les modèles dans la logique MRP	262
-----	--	-----

Table des figures

1	Les trois niveaux décisionnels en planification industrielle	16
1.1	La structure hiérarchique de planification MRP II	21
1.2	Cadre décisionnel du PIC	23
1.3	MRP II et la planification tactique	27
1.4	La mise en œuvre temporelle	28
1.5	Les activités du processus logistique d'une chaîne logistique	30
1.6	La maison du management de la chaîne logistique	30
1.7	Problématique de la planification des chaînes logistiques	31
1.8	Intégration du DRP dans MRP II	32
1.9	L'intégration horizontale	32
1.10	Des progiciels complémentaires.	33
1.11	Les fonctionnalités des APS	35
1.12	La structure de planification dans un APS	36
1.13	classification des modèles mono-site	39
1.14	Représentation graphique des différents coûts pour le modèle EPQ	41
1.15	Les différentes structures	49
1.16	Les modèles de planification tactique	57
2.1	la chaîne logistique considérée	63
2.2	Processus de modélisation ASCI	64
2.3	Interaction entre les trois sous-systèmes	66
2.4	Sous-système physique	68
2.5	Modélisation UML des usines et des machines qui leur sont associées	68
2.6	Modélisation UML des ressources de stockage	69
2.7	Modélisation UML des moyens de transport	70
2.8	positionnement des stocks	70
2.9	Modélisation UML des dépendances Usine-Stocks	71
2.10	Les transports entre stocks	71
2.11	Modélisation UML des dépendances Stock-Transport	72
2.12	Modélisation UML des dépendances StockProduit-Fournisseur	72
2.13	Modélisation UML des dépendances StockProduit-Centre_Distribution	73
2.14	Sous-système logique	73
2.15	Les contraintes du système de production	74
2.16	Les contraintes de capacité	75
2.17	Les contraintes d'utilisation	75
2.18	Les contraintes de stock	76
2.19	Les contraintes demande et livraison	76
2.20	Les coûts pris en compte au niveau de la chaîne logistique	76
2.21	La règle d'équilibre des stocks	77
2.22	Les caractéristiques générales de la chaîne logistique	77
2.23	Modélisation UML de la dépendance Usine-Produit	78

2.24	Modélisation UML de la dépendance entre Moyen_Transport et Produit	78
2.25	Modélisation UML du lien entre Centre_Distribution et Produit	79
2.26	Modélisation UML de la dépendance entre Machine et Produit	79
2.27	Modélisation UML du lien de dépendance entre Produit et StockProduit	79
2.28	Modélisation UML de la relation entre Fournisseur et Produit	80
2.29	Modélisation UML des liens entre le sous-système physique et le sous-système décisionnel	80
2.30	Modélisation UML des liens entre les trois sous-systèmes	81
	la chaîne logistique considérée (Figure 2.1)	82
2.31	Les ressources de production	83
2.32	Flux produit lié au stock « ante »	89
2.33	Flux produit lié au stock « post »	91
2.34	Le modèle générique et les principaux modèles de la littérature	96
2.35	Le modèle générique instancié avec une usine, un client et sans fournisseur	98
2.36	Le modèle générique instancié pour le MLCLSP	99
2.37	La chaîne logistique	102
2.38	Association usine fictive, client	103
2.39	Les possibilités d’instanciation du modèle générique de connaissance	108
3.1	Graphe pour un problème de « lot-sizing » à 6 périodes pour 1 produit	113
3.2	Les méthodes de résolution pour le CLSP	121
3.3	Les méthodes de résolution pour le CLSP avec Setup-Time	126
3.4	Algorithme pour la descente par gradient	130
3.5	Algorithme pour la méthode par sous-gradient	132
3.6	Algorithme de notre méthode d’optimisation	133
3.7	Algorithme de base pour les voisinages	136
3.8	Effet du déplacement de quantité sur les stocks (cas 1)	137
3.9	Effet du déplacement de quantité sur les stocks (cas 2)	137
3.10	Algorithme pour le voisinage 1	139
3.11	Algorithme pour le voisinage 2	141
3.12	Algorithme pour le voisinage 3	142
3.13	Algorithme pour le voisinage 4	143
3.14	Deux solutions de même coût	145
3.15	Effet de la nouvelle fonction objectif sur les solutions précédentes	145
3.16	Algorithme pour le voisinage 6	146
3.17	Algorithme de principe de notre méthode d’optimisation avec fonction bi-objectif . . .	148
4.1	Cas d’étude	160
4.2	Chaîne logistique de type Flow Shop Hybride	161
4.3	Instanciation des stocks	163
4.4	Les stocks « ante » de l’étage k	165
4.5	Les stocks « post » de l’étage k	166
4.6	Les stocks « post » de l’étage M	166
4.7	Les méthodes de résolution pour le MLLP	172
4.8	Les méthodes de résolution pour le MLCLSP	175
4.9	Les méthodes de résolution pour le MLCLSPST	179
4.10	Principe d’agrégation	181
4.11	prise en compte de la demande perdue sous l’effet des contraintes de périodicité	183
4.12	Le voisinage « augmentation »	184
4.13	Le voisinage « regroupement à gauche »	185
4.14	Exemple d’utilisation des procédures de réparation sur le produit i	186
4.15	Exemple d’utilisation des deux autres procédures de réparation sur le produit i	187

4.16	Exemple d'utilisation des procédures de réparation sur le produit i dans le cas d'un voisinage regroupement à gauche	188
4.17	Schéma d'optimisation pour la métaheuristique « usines etages »	190
4.18	heuristique d'affectation	191
4.19	Schéma d'optimisation pour la métaheuristique «Flow Shop Hybride»	192
4.20	Nouvelle approche d'optimisation	198
5.1	Approche par intégration de contraintes opérationnelles	203
5.2	Approche proposée	204
5.3	Approche par couplage modélisation - simulation	205
5.4	Le principe de fonctionnement de notre méthodologie	206
5.5	Le principe de fonctionnement de notre méthodologie	210
5.6	L'atelier simulé	212
5.7	Evolution des capacités suivant les itérations	219
6.1	l'approvisionnement, la production et la distribution dans une chaîne logistique	222
6.2	Cas d'étude	225
6.3	Résultats obtenus	226
6.4	Représentation des stocks dans les outils de gestion des stocks et de prise de commande	227
6.5	Cas d'une politique de rationnement par seuil avec prise en compte du cours de route	229
6.6	Cas d'une politique de rationnement par seuil sans prise en compte du cours de route	230
6.7	Situation initiale	231
6.8	Modélisation du seuil de rationnement dans l'outil de gestion des stocks	232
6.9	Utilisation des lots dans l'outil de prise de commande	232
6.10	Comparaison du scénario 3 avec une politique de rationnement par seuil	233
6.11	Utilisation des stocks logiques dans l'outil de gestion des stocks	234
6.12	Configuration du test	235
6.13	Le modèle de données	236
6.14	Schéma de fonctionnement de l'application	237
6.15	Enchaînement des algorithmes	238
6.16	L'algorithme de gestion des stocks	239
6.17	Processus de modélisation ASCI	243
7.1	Diagramme de Gantt obtenu pour le DLSP	257
7.2	Diagramme de Gantt obtenu pour le CSLP	257
7.3	Diagramme de Gantt obtenu pour le PLSP	258
7.4	Evolution des stocks cumulés pour les différents modèles	259
7.5	CLSP	261
7.6	GLSP	261
7.7	Procédure 1	264
7.8	Procédure 2	265
7.9	Procédure 3	266
7.10	Procédure 4	267
7.11	Procédure 5	268
7.12	Procédure 6	269
7.13	Procédure 7	270
7.14	Procédure 8	271
7.15	Procédure 9	272
7.16	Procédure 10	273

Introduction

La gestion de production s'intéresse à l'ensemble des décisions à prendre pour qu'une entreprise arrive à fabriquer des produits conformes à la demande, en temps voulu et à des coûts permettant d'assurer sa pérennité. On voit donc apparaître quatre concepts fondamentaux :

- la maîtrise des délais,
- la maîtrise des coûts,
- la maîtrise de la qualité,
- la satisfaction du client.

Gérer la production, c'est trouver un compromis entre ces impératifs parfois contradictoires.

D'après (Chassang 1983) : « Bien gérer la production, c'est d'abord bien connaître et comprendre les besoins et les objectifs dictés par la stratégie de l'entreprise. La gestion de production est un outil au service de la stratégie de l'entreprise. »

Afin de bien situer les divers problèmes rencontrés en gestion de production, (Giard 2003) définit trois niveaux décisionnels, suivant l'horizon de temps sur lequel les décisions s'appliquent, le niveau de compétence hiérarchique et celui d'agrégation de la décision :

- *Le niveau stratégique* : Les décisions stratégiques concernent la conduite générale de l'entreprise sur le long-terme (vision, en général, à plus de 18 mois), elles déterminent ainsi la politique majeure de l'entreprise. Cela implique une définition volontariste et cohérente du portefeuille d'activités qu'elle entend avoir et des ressources stables qu'elle compte mettre en œuvre pour parvenir à ses fins. Par ressources stables, on entend aussi bien les machines, les hommes, que les informations relatives à la production (procédures de gestion formalisées, bases de données techniques...). En ce qui concerne les machines, les décisions stratégiques concernent donc leurs achats, remplacements, radiations, cessions, localisations ... ; pour les hommes leurs embauches, licenciements, plans de formation ... ; pour les informations, le choix de nouveaux traitements *etc.* Ces décisions seront ensuite mises en œuvre dans les niveaux inférieurs.
- *Le niveau tactique* : Les décisions tactiques concernent le moyen-terme (horizon de 6 à 18 mois). Parmi celles-ci, on retrouve l'élaboration du *plan de production* : c'est-à-dire, d'une part les personnes à prévoir et à former compte tenu de l'effectif et de la législation du travail, d'autre part la *planification de la production* qui est une programmation prévisionnelle, agrégée par famille de produits, pour un ensemble de périodes (agrégation du temps) dont l'amplitude varie du jour à la semaine (selon les entreprises). On retrouve également l'élaboration du *plan de transport* : cela correspond à un ensemble de tournées type de distribution ou d'approvisionnement qui sera utilisé dans les entreprises où ces problèmes sont relativement stables et ont une certaine incidence économique. Il est important de noter que l'ensemble de ces décisions s'inscrit dans le cadre général défini par les décisions stratégiques.
- *Le niveau opérationnel* : Les décisions opérationnelles assurent la flexibilité quotidienne nécessaire pour faire face aux fluctuations prévues de la demande et des disponibilités de ressources (mode prévisionnel) et réagir aux aléas (mode correctif), dans le respect des décisions tactiques.

Cela implique que la mise en œuvre au niveau opérationnel nécessite une désagrégation complète du plan défini au niveau hiérarchique supérieur. Parmi les décisions opérationnelles concernant la gestion de production, on trouve : la *gestion des stocks*, qui assure la mise à disposition des matières premières et des composants ; l'*ordonnancement* qui consiste en une programmation détaillée des ressources mobilisées (opérateurs, équipements et outillages) dans l'exécution des opérations nécessaires à la production élémentaire de biens ou de prestations de service (pour un client final ou pour des besoins internes, s'il s'agit d'opérations de maintenance ou de manutention), sur un horizon ne dépassant pas quelques dizaines d'heures, dans le cadre d'un découpage temporel généralement de l'ordre de la minute.

Transversalement à ces trois niveaux, il existe un problème majeur qui est celui de la *planification industrielle*. Son but est de livrer aux clients des produits en quantité demandée et au moment voulu, au coût de revient le plus bas. Autrement dit, l'enjeu est de trouver un équilibre entre le taux de satisfaction client et les différents coûts engendrés par la fabrication effective de la demande (production, stockage etc.). Partie intégrante de la gestion de production, la planification industrielle se décline donc en trois niveaux hiérarchiques comme l'illustre la figure 1 :



Figure 1 – Les trois niveaux décisionnels en planification industrielle (Genin 2003)

- la *planification stratégique* (long-terme : de 2 à 5 ans) a pour objectif de satisfaire le marché dans son ensemble dans le cadre de la stratégie générale de l'entreprise. Plus précisément, elle cherche à déterminer les objectifs de croissance, de développement et de rentabilité de celle-ci. Sa responsabilité première est la recherche d'une stratégie globale permettant d'atteindre les objectifs à long terme au travers de modifications substantielles des capacités de production (nouvelles unités de production, nouvelles localisations de celles-ci, adoption de nouvelles technologies...).
- la *planification tactique* (moyen-terme : de 1 à 18 mois) définit les conditions de satisfaction de la demande des clients : elle vise à déterminer les quantités de produits à fabriquer à l'intérieur d'un horizon à moyen ou à court terme (planification des priorités) et à préciser les quantités de ressources à utiliser (planification des capacités) afin de répondre le mieux possible aux objectifs opérationnels visés en matière de qualité, de volume, de lieu, de temps et de coût. La capacité des installations ne pourra varier significativement au cours de la période de planification et les seuls moyens qui s'offriront au gestionnaire pour modifier le volume de la production résideront dans la modulation de l'utilisation des ressources existantes de production (heures supplémentaires...), la variation des quantités de ressources, à l'intérieur de limites souvent étroites, pourvu qu'elle puisse s'appliquer rapidement (embauche ou mise à pied temporaire d'ouvriers, location temporaire de pièces d'équipement), ou le recours à des capacités externes.

- *la planification opérationnelle* (court-terme : quotidien voire hebdomadaire) assure le suivi et le contrôle du flux physique afin d'assurer la mise à disposition des produits à chaque maillon selon les conditions définies au niveau tactique. Cela consiste donc à ordonnancer la production sur un horizon allant de la journée à la semaine, c'est-à-dire à déterminer les dates de début des tâches sur les différentes ressources afin d'assurer la fabrication des quantités d'articles dans les temps définies au niveau tactique. C'est pour cela qu'on appelle traditionnellement cette approche la « planification par date ». Elle est généralement traitée par la « communauté d'ordonnancement ».

Les entreprises sont confrontées à une concurrence de plus en plus forte et à un marché souvent fluctuant. Parmi les niveaux de décisions définis par (Giard 2003), c'est au niveau tactique que l'entreprise peut prendre des mesures lui permettant de devenir réactive et concurrentielle. En effet, la planification stratégique consiste, entre autres, en la définition de l'infrastructure du système de production (localisation d'usines, d'entrepôts...) et fournit donc un cadre physique à la planification tactique. La planification stratégique est donc peu flexible aux adaptations. D'un autre côté, en raison de son horizon court-terme, la planification opérationnelle ne dispose pas de suffisamment de temps pour réagir aux incertitudes. C'est dans le cadre de la planification tactique que s'inscrivent les problématiques rencontrées dans cette thèse.

Au fur et à mesure de l'avancement de nos travaux de thèse, nous nous sommes rendus compte que contrairement à la communauté d'ordonnancement, celle s'intéressant aux problématiques de planification tactique ne disposait pas d'un cadre de travail aussi formalisé (modèles et instances de référence, notation des problèmes rencontrés etc.). Au travers d'un état de l'art sur les modèles mathématiques de planification tactique rencontrés dans la littérature, nous avons essayé de définir quelles instances seraient susceptibles d'être utilisées par la communauté comme instances de référence. Nous nous sommes également tournés vers l'élaboration d'un modèle de connaissance générique afin de proposer une formalisation mathématique dont l'instanciation permet de retrouver les différents modèles de référence de la littérature et de permettre par la suite l'élaboration d'un modèle de données unique permettant d'unifier l'écriture des différentes instances rencontrées. Les travaux de thèse présentés ont également poursuivi deux autres buts : d'une part apporter des solutions satisfaisantes en terme de planification de la production, c'est-à-dire proposer des méthodes d'optimisation rapides et efficaces pour l'élaboration de plannings de production dont les coûts logistiques sont les plus faibles possibles ; d'autre part donner des méthodologies pour tenter de résoudre les divers problèmes liés aux problèmes de cohérence entre les niveaux de planification tactique et opérationnelle. Enfin, nous avons travaillé sur un problème industriel dont l'objectif est de maîtriser l'impact des aléas de la demande des clients sur leur taux de satisfaction au travers d'une politique de gestion de stock particulièrement innovante : dans le cadre d'une collaboration industrielle, il nous a été confié l'étude de faisabilité d'un test « grandeur nature » de cette politique ainsi que l'élaboration d'une application informatique permettant de mettre en œuvre un tel test.

Le manuscrit est organisé de la manière suivante :

Le premier chapitre définit le contexte des travaux que nous présentons. Nous montrons les différentes façons dont ont été abordées les problématiques relatives à la planification tactique et mettons en avant les difficultés liées à la mise en œuvre de ces approches. Nous dressons également un état de l'art des différents modèles mathématiques de référence conçus pour répondre aux problèmes d'élaboration de plans de production selon que la planification s'adresse à une entreprise mono-site, multi-site ou même une chaîne logistique.

L'objectif du second chapitre est de proposer un modèle de connaissance générique pour la planification tactique. Ce dernier est donné sous la forme d'une représentation UML. Il permet d'obtenir un modèle mathématique générique pour l'élaboration de plans de production. Ce modèle mathématique

est ensuite instancié dans différentes configurations (mono-site, multi-site voire chaîne logistique) et nous montrons que les modèles de référence rencontrés dans la littérature sont obtenus par une simple instanciation de ce modèle.

Le troisième chapitre s'intéresse à l'élaboration de plans directeurs de production dans un cadre mono-site. Pour répondre à cette problématique, un modèle mathématique (le Capacitated Lot Sizing Problem) a été développé (Drexel et Kimms 1997). Après un état de l'art sur les méthodes de résolution liées à ce modèle, nous présentons plusieurs méthodes d'optimisation dont l'une utilise une fonction bi-objectif qui lui permet une plus grande flexibilité dans son parcours de l'espace des solutions admissibles. Pour juger de la qualité des solutions obtenues, un calcul de borne inférieure basé sur une relaxation lagrangienne a été développé. Nous testons ces méthodes sur des instances de la littérature puis nous proposons de l'appliquer sur une variante de ce problème (intégrant une difficulté supplémentaire liée à la recherche d'une solution faisable).

Le quatrième chapitre s'appuie sur un cas d'étude industriel dont le sujet est l'optimisation de la production dans une chaîne logistique dont la topologie peut être rapprochée de celle d'un atelier de type « Flow-Shop Hybride ». Après avoir proposé un état de l'art sur les méthodes de résolution liée au modèle de référence pour l'élaboration des plans de production multi-niveau (c'est à dire s'intéressant à la planification de la fabrication des produits finis mais également de leurs composants), nous instancions le modèle mathématique générique sur la structure de la chaîne considérée et nous adaptons le modèle obtenu aux spécificités industrielles de l'entreprise. Ensuite, nous proposons une méthode d'optimisation pour tenter d'apporter une réponse satisfaisant à cette problématique. Nous testons notre méthode sur des instances de taille industrielle.

Le cinquième chapitre tente d'apporter une réponse satisfaisante au problème de cohérence entre la planification tactique et la planification opérationnelle. Autrement dit, la question à laquelle on essaie de répondre est la suivante : « Comment s'assurer qu'un plan de production tactique peut effectivement être réalisé dans le temps imparti au niveau des ateliers de production ? ». Après avoir décrit les différentes méthodologies rencontrées dans la littérature pour répondre à cette problématique, nous proposons deux nouvelles approches basées l'une sur un couplage de modèles mathématiques, l'autre sur un couplage entre une méthode d'optimisation et des modèles de simulation.

Le sixième et dernier chapitre porte sur une problématique industrielle issue d'une collaboration entre le LIMOS et une entreprise clermontoise. Cette problématique porte sur la gestion des stocks à demande différenciée qui vise à minimiser les stocks tout en maintenant le niveau de satisfaction désiré pour chaque client. Suite à une étude précédemment effectuée au sein du laboratoire, l'entreprise nous a demandé de développer une application permettant de mettre en œuvre une telle gestion des stocks et d'étudier la faisabilité d'un test « grandeur nature » de cette politique. Nous présenterons l'état d'avancement des travaux réalisés.

Chapitre 1

La problématique de la planification tactique

Sommaire

Introduction	20
I L'approche MRP II	20
1 Les différents plans dans MRP II	21
1.1 Le plan stratégique	22
1.2 Le Plan Industriel et Commercial (PIC)	22
1.3 Le Plan Directeur de Production (PDP)	24
1.4 Le calcul des besoins nets (MRP)	25
1.5 Le pilotage de l'atelier	26
2 MRP II et les niveaux décisionnels de la planification industrielle	26
3 Les limites de MRP II	27
3.1 Synchronisation des capacités et des matières	27
3.2 Synchronisation verticale	28
II La chaîne logistique et le management de la chaîne logistique	28
1 Notion de chaîne logistique	28
2 Le management de la chaîne logistique	30
2.1 La planification tactique : partie intégrante du management de la chaîne logistique	31
2.2 Le problème d'intégration horizontale	32
2.3 Solutions logicielles pour la planification de la chaîne logistique	33
III Classification des modèles mathématiques pour la planification tactique	36
1 Les modèles mono-site	38
1.1 Les modèles mono-niveau	40
1.2 Les modèles multi-niveau	48
2 Les modèles multi-site	55
Conclusion	59

Introduction

L'objectif de ce premier chapitre est de situer notre problématique de planification tactique au sein des méthodologies couramment utilisées en gestion de production et de répertorier les modèles mathématiques de référence y répondant.

Les problèmes liés à la planification industrielle sont vastes. (Wight 1984) a proposé une méthode visant à apporter une réponse satisfaisante à l'ensemble de ceux-ci : MRP II (Manufacturing Resource Planning). Cette approche définit une série de plans hiérarchisés où chacun donne lieu à un ensemble de décisions d'échelle et d'horizon spécifiques, fonction de sa position hiérarchique. Chaque plan, révisé avec une périodicité qui lui est spécifique, sert de cadre de décisions pour l'établissement du plan de niveau inférieur. Dans une première partie, nous allons définir plus précisément l'approche MRPII et nous montrerons comment et avec quels plans sont abordés les problèmes liés à la planification tactique. Nous mettrons en exergue les faiblesses liées à la mise en œuvre de cette méthodologie.

La méthode MRPII étant initialement orientée pour la planification d'un seul site industriel, nous définirons, dans une seconde partie, le « Supply Chain Management » que (Thomas et Lamouri 2000) considèrent comme une évolution de MRPII permettant de prendre en compte, entre autres choses, la planification de toute une chaîne logistique. Après avoir mis en avant les difficultés de synchronisation des différents plans de production induites par l'aspect multi-site d'une chaîne logistique, nous donnerons un descriptif des solutions logicielles élaborées pour sa planification.

Pour finir, nous donnerons un état de l'art des modèles mathématiques de référence pour l'élaboration de plan de production en nous attachant à identifier les modèles répondant aux problématiques de planification tactique.

I L'approche MRP II

Le concept MRP (Material Requirements Planning) est apparu aux Etats-Unis en 1965 sous l'impulsion de Joseph Orlicky (Orlicky 1975). Son principe est né du besoin de synchronisation des quantités disponibles de matières, de composants et de sous-ensembles dans un contexte de délais fixes et de produits stables. En effet, Orlicky a mis en évidence deux types de demande à satisfaire :

- la demande externe. Celle-ci est composée des commandes clients en produits finis, pièces de rechange etc. Cette demande est le plus souvent estimée (prévisions de vente) mais peut également inclure les commandes fermes.
- la demande interne qui est déduite de la demande externe et qui est le résultat d'un calcul basé sur la décomposition des produits finis en composants jusqu'à la matière première (par le biais de leur nomenclature). Ce calcul est le cœur de la méthode MRP.

Le MRP transforme donc les données commerciales relatives aux ventes en données de production.

Initialement, le MRP ne tenait pas compte des contraintes extérieures et planifiait les besoins uniquement en fonction de la demande ou de la prévision de demande, c'est-à-dire :

- en considérant une capacité de production illimitée toujours capable de fournir,
- sans pouvoir réagir aux aléas (pannes, retards de livraison...)

En 1971, Orlicky définit la méthode MRP I qui permet de prendre en compte les capacités finies.

En 1984, Wight lance les bases de la méthode MRP II (Manufacturing Resource Planning) en élargissant le concept au « management resources planning » et à l'intégration des données financières et logistiques (Wight 1984) : cette évolution s'est justifiée par le besoin de prendre en compte les capacités

des ressources de production, d'approvisionnement, de sous-traitance, de stockage, de distribution et également financières (Genin 2003).

1 Les différents plans dans MRP II

Cette approche définit une série de cinq plans :

- le plan stratégique,
- le plan industriel et commercial,
- le programme directeur de production,
- la planification des besoins en composants (Material Requirements Planning, MRP en anglais),
- le pilotage d'atelier.

La figure 1.1 illustre l'approche MRP II.

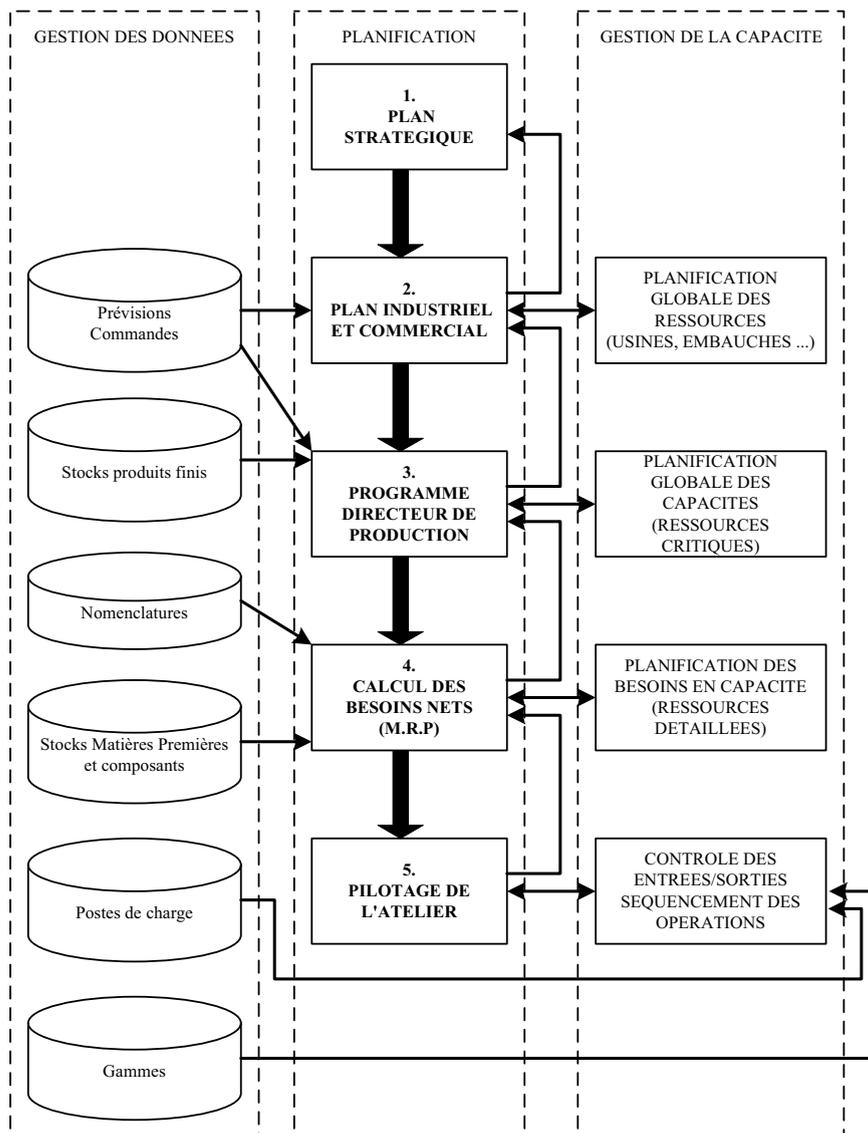


Figure 1.1 – La structure hiérarchique de planification MRP II (Vollmann et al. 1997)

Chaque niveau hiérarchique possède donc ses propres caractéristiques et la manière d'agrégier les données dépend de nombreux facteurs dont :

- la longueur de l'horizon de planification,
- le niveau de précision de l'information requise et de la demande.

1.1 Le plan stratégique

La planification stratégique consiste à valider la stratégie que déploiera l'entreprise en termes de croissance, de développement et d'investissement sur un horizon de 2 à 5 ans. Cette validation est effectuée à l'aide d'une estimation d'adéquation charge/capacité à long terme permettant de valider les moyens mis en oeuvre (nouvelles unités de production, fermetures de sites, adoption de nouvelles technologies...) pour la réalisation de ces objectifs stratégiques.

1.2 Le Plan Industriel et Commercial (PIC)

Le Plan Industriel et Commercial (PIC) définit les différents niveaux d'activités de l'entreprise, c'est-à-dire qu'il établit un compromis entre les objectifs commerciaux et les capacités de production.

Le délai nécessaire à l'application des décisions induites par le PIC étant généralement assez long, son élaboration s'appuie donc sur des prévisions de vente et son horizon est de 12 à 18 mois. De plus, lorsqu'il s'agit de décisions majeures, à long terme, l'entreprise ne cherche pas à déterminer à l'avance les ventes de chacun de ses produits, c'est pour cela que le PIC s'appuiera plutôt sur des prévisions par famille d'où un niveau d'agrégation élevé. Il peut même arriver, dans le cas où les produits et les ressources de l'entreprise sont assez homogènes, qu'on établisse des prévisions en heures ou en euros.

D'après (Baglin et al. 2005), la définition d'une famille de produits peut se faire comme suit :

- soit en la constituant en fonction de la ressource à planifier (dans le cadre d'une ressource critique par exemple) : la famille reprendra tous les produits dont la fabrication nécessite l'utilisation de cette ressource,
- soit en fonction d'un composant critique : la famille reprendra tous les produits dont la nomenclature contient ce composant.

Le PIC joue trois rôles majeurs :

1. *un rôle d'anticipation* : dans de nombreuses situations, les quantités de produits souhaitées au cours du temps par le client varient fortement. Le PIC devra anticiper les possibilités d'évolution des ventes (donc des charges de travail) de façon à disposer d'un délai suffisant pour pouvoir donner au système de production les moyens lui permettant de s'adapter à son marché. Il doit donc étudier les rapports charge/capacité pour les ressources critiques, évaluer les quantités de matières nécessaires en vue d'une négociation avec les fournisseurs, identifier les plans d'action adéquats pour réaliser un bon équilibre charge/capacité pour les ressources en surcharge (ou en sous-charge).
2. *un rôle de répartition* : il réalise une répartition de flux et de charges entre les ressources critiques si elles sont multiples.
3. *un rôle de négociation* : il reflète un compromis entre l'objectif du département commercial (maximiser les ventes au travers, par exemple, d'une personnalisation des objets produits) et celui de la production (minimiser les coûts de revient industriels).

A ce niveau, les leviers d'action déterminent les trois stratégies de base (éventuellement combinées) lors de la mise en oeuvre du PIC pour faire face à la fluctuation de la demande. Celles-ci sont :

- *la stratégie de niveau* : la création de stock d'anticipation pour absorber les forts pics de demande,
- *la stratégie de poursuite* : fait-on appel à du personnel intérimaire ou à temps partiel et aux heures supplémentaires ?

- *le management de la demande* : on modifie la demande par une politique de changement de prix (hausse des prix, promotions...).

La figure 1.2 illustre les fonctions du PIC.

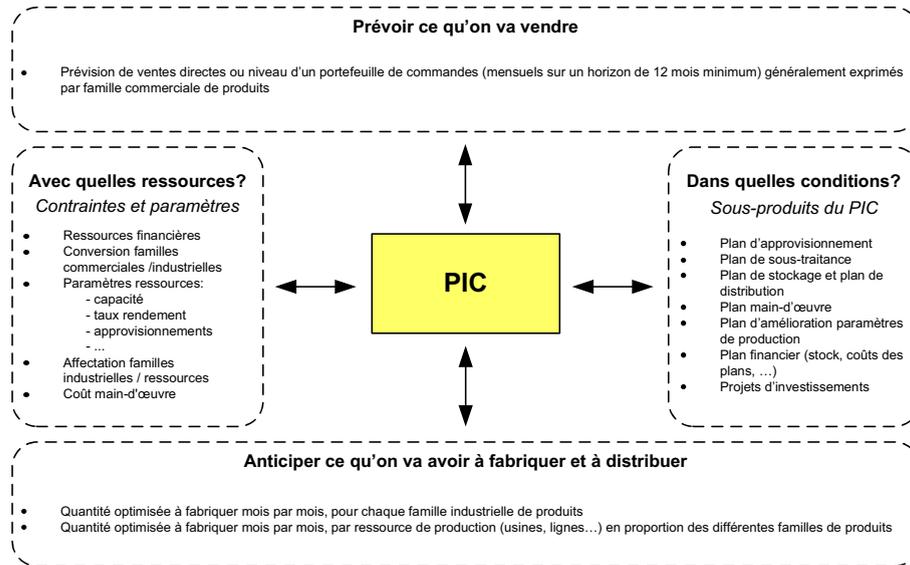


Figure 1.2 – Cadre décisionnel du PIC
(Womack et al. 1992)

L'élaboration traditionnelle du PIC repose sur des techniques graphiques ou sur des modèles de programmation linéaire ((Giard 2003), (Heizer et Render 1995), (Lamouri et Thomas 1999) et (Baglin et al. 2005)).

Les techniques graphiques et par tableaux sont répandues car faciles à comprendre et à utiliser. Ces plans fonctionnent avec peu de variables à la fois pour permettre au planificateur de comparer la demande prévue avec la capacité existante. Ces méthodes graphiques procèdent par calculs successifs d'évaluation des coûts; elles permettent ainsi d'identifier différents plans intégrés valables mais dont les coûts ne sont pas nécessairement les plus faibles. Le gestionnaire doit donc faire appel à son bon sens pour déterminer un plan approprié.

Ces méthodes suivent généralement 5 étapes :

1. détermination de la demande de chaque mois ;
2. détermination de la capacité en horaire normal, en heures supplémentaires et en sous-traitance par mois ;
3. recensement des coûts de main-d'œuvre, de possession de stock, *etc.* ;
4. évaluation des stratégies influençant la main-d'œuvre ou le niveau de stock ;
5. établissement d'alternatives et comparaison de leurs coûts totaux.

Ces outils de management aident à évaluer des stratégies mais ne les génèrent pas. Or les décideurs demandent une approche systématique qui considère tous les coûts et fournit une réponse efficace au problème de l'élaboration du PIC.

Les modèles mathématiques s'appuyant sur la programmation linéaire proposent une telle approche (Crandall 1998). Contrairement à la méthode graphique, cette dernière permet de générer des stratégies optimales pour un ensemble donné de paramètres. Mais il faut garder à l'esprit que celles-ci sont

généérées pour des paramètres « prévisionnels ». Par conséquent, il est tout à fait envisageable qu'un aléa de la production fasse fortement diverger l'optimum et modifie ainsi la stratégie optimale. Cela pourrait avoir de fâcheuses conséquences (notamment financières en cas de mise en place d'une stratégie de remplacement par exemple...). Il s'ensuit que le PIC doit être relativement stable, en raison du poids des décisions prises. Il faut donc, durant son élaboration, rechercher la stabilité de la solution (robustesse) plutôt que son optimisation. Ainsi (Genin et al. 2001) concluent : « Le PIC devient ainsi l'instance de réflexions qui oriente les décisions pour rendre robuste la stratégie proposée par le modèle selon les conditions données. L'outil d'aide à la décision doit donner une réponse moyenne qui sera la meilleure possible pour plusieurs ensembles de conditions. Le PIC ne sera alors pas optimisé au sens mathématique mais variera peu lors d'une modification des paramètres. »

Le PIC fournit donc un plan de capacité et de vente, points d'entrée au niveau suivant de la planification tactique qui a pour but de définir un plan directeur de production.

1.3 Le Plan Directeur de Production (PDP)

Le Plan Directeur de Production (PDP) est un plan établissant pour chaque période (semaine, plus rarement à la journée) les quantités des principaux produits qui seront fabriquées sur un horizon de 3 à 6 mois : il donne des volumes de fabrication de *produits directeurs* pour chaque période sur l'horizon de planification. Il établit donc des objectifs de production exprimés en termes de références contrairement au PIC qui lui, raisonne en familles de produit. Bien évidemment, ceux-ci respectent les décisions globales prises dans le cadre du PIC.

Le PDP étant utilisé comme outil d'analyse charge/capacité et pour le calcul des besoins nets, le choix des articles pour lesquels il est développé est primordial :

- si on a peu de produits finis, ils constitueront les produits directeurs,
- sinon, les produits directeurs seront formés de composants critiques qui permettront de piloter les flux en amont.

Puisque l'élaboration du PDP se base sur des commandes fermes et des prévisions, son horizon de planification est, en général, découpé en deux parties : une zone *ferme* et une zone *prévisionnelle*.

- *Dans la zone ferme*, les ordres de fabrication ne devront plus être remis en cause (si ce n'est en des circonstances exceptionnelles). Cette partie a pour mission d'éviter une remise en cause permanente des ordres de fabrication qui empêcherait toute planification des activités.
- *Dans la zone prévisionnelle*, les ordres sont incertains et susceptibles d'être modifiés.

Pour chaque article directeur, le PDP est constitué de quatre échéanciers :

- les besoins à chaque période : ils correspondent aux commandes fermes et prévisionnelles mais également à la mise en œuvre concrète de la politique de stockage/déstockage d'anticipation décidée au niveau du PIC,
- les quantités de produits finis à réceptionner pour couvrir les besoins : on tient compte des coûts induits, notamment des économies potentielles en cas de regroupement de plusieurs ordres (si les coûts fixes - de réglage par exemple - sont très importants), et de la capacité du système,
- les quantités de produits à lancer en fabrication en tenant compte de différents délais pour respecter les réceptions planifiées,
- les stocks prévisionnels par période qui se déduisent des besoins et des réceptions planifiées.

On voit donc que le PDP joue un rôle fondamental au niveau de la planification tactique. D'ailleurs pour (Courtois et al. 2003), il s'agit même « d'un *contrat* qui définit de façon précise l'échéancier des quantités à produire pour chaque produit fini ». Il en définit même les principales fonctions :

- il dirige le calcul des besoins-matières : en donnant les ordres de fabrication pour les produits finis, il induit le calcul des besoins-matières au travers des nomenclatures,

- il concrétise le PIC puisqu'il traduit chaque famille en produits finis,
- il permet de suivre les ventes réelles en comparant les commandes reçues avec les prévisions,
- il met à disposition du service commercial le *disponible à vendre* qui est un outil donnant le nombre de produits finis disponibles à la vente sans remettre en cause le PDP prévu et donc sans déstabiliser la production,
- il permet enfin de mesurer l'évolution du stock.

Le Plan Directeur de Production joue un rôle fondamental dans le fonctionnement d'un système intégré de planification et de contrôle de la production et des stocks, en se positionnant comme le lieu où s'établit à chaque période l'équilibre entre ressources de l'entreprise et demande à satisfaire. C'est en ce sens que (Tchernev 2003) précise : « Le PDP se trouve donc au coeur des tensions engendrées par les perturbations susceptibles de survenir dans l'entreprise, tels les retards de production ou dans son environnement, telles les modifications de la demande. L'objectif premier du PDP est le service : livrer les commandes à temps dans un délai concurrentiel. Le deuxième objectif est l'utilisation optimale des ressources, soit la possession minimale d'un stock d'une part, et des conditions de fonctionnement favorisant l'efficacité du système opérationnel et des achats d'autre part. »

Le PDP ne traitant que de la planification en produit fini, le MRP permet de planifier, sur le même horizon, le déstockage ou la fabrication des composants en remontant les besoins au moyen de nomenclatures (il est possible de noter la limite de cette technique pour les industries de process et les industries à gammes multiples).

1.4 Le calcul des besoins nets (MRP)

Pour (Nollet et al. 1994), la planification des besoins-matières se situe à l'interface de la planification détaillée des opérations et de l'ordonnancement : « Son rôle est de déterminer les quantités et les dates de fabrication, d'assemblage des articles à demande dépendante, c'est-à-dire des matières premières et des composants, afin de produire à temps des quantités prévues dans le plan directeur de production. Elle assure la coordination entre ce que l'entreprise s'est engagée à livrer sur le marché et la mise en branle des opérations de production devant être réalisées pour pouvoir terminer les commandes à temps. »

Autrement dit, le calcul des besoins est un outil de programmation à moyen terme qui génère des propositions de fabrication et d'achat pour l'ensemble des articles à tous les niveaux de la nomenclature à partir du PDP.

Outre la planification, le MRP sert également de système de contrôle : en effet, on peut assurer le suivi des étapes, des détails et remédier à tout écart défavorable jugé important pouvant remettre en cause et modifier le PDP. Il permet aussi de « diriger » les ressources ; en effet, une fois les quantités et les dates de fabrication et de commande connues, il est alors possible de passer à l'étape de l'ordonnancement.

En définitive, le MRP répond aux trois questions suivantes :

- De quels composants ou matières premières doit-on disposer pour réaliser le PDP ?
- Combien d'unités de chaque article doit-on acheter et fabriquer, compte-tenu des niveaux de stock ?
- Quand passer la commande d'achat et quand démarrer la fabrication ?

On répond à ces questions en « éclatant » chaque produit du PDP et en établissant les calendriers de fabrication et d'achat, en prenant soin de décaler les délais respectifs prévus pour chaque opération de livraison, de fabrication et d'assemblage de ces articles. Les quantités déterminées par le MRP sont fortement corrélées avec les décisions inscrites dans le PDP. Le degré d'exactitude dans la détermination du PDP influence donc la justesse des quantités à produire selon le MRP.

La simplicité du principe MRP ne doit pas cacher les difficultés de sa mise en oeuvre en vraie grandeur dans un environnement industriel car il génère une masse considérable de calculs qui engendrent des milliers d'ordres de fabrication et d'achat qu'il faut contrôler et dont il faut suivre la réalisation.

1.5 Le pilotage de l'atelier

Le pilotage de l'atelier assure la flexibilité quotidienne nécessaire pour faire face aux fluctuations prévues de la demande et des disponibilités de ressources. (Thomas et Lamouri 2000) précisent : « Le pilotage de l'atelier est responsable de l'exécution du PDP et de la planification du besoin en composants (MRP), ainsi que de la bonne utilisation de la main-d'œuvre et des machines; il doit aussi minimiser les encours et assurer le taux de service client. »

On va retrouver à ce niveau la gestion des stocks, qui assure la mise à disposition des matières premières et des composants; l'ordonnancement qui consiste en une programmation détaillée des ressources mobilisées (opérateurs, équipements et outillages) et l'organisation effective des transports.

Nous avons vu que les problèmes liés à la planification industrielle se répartissaient selon trois niveaux : stratégique, tactique et opérationnel. Regardons comment la logique MRP II s'intègre dans cette classification.

2 MRP II et les niveaux décisionnels de la planification industrielle

(Genin 2003) définit un plan tactique comme étant « l'ensemble des plans définissant les volumes de distribution de production et d'approvisionnement et la modulation de capacité des ressources pour satisfaire les besoins des clients finaux ».

En se basant sur cette définition, les plans constitutifs du niveau tactique sont donc le PIC et le PDP. Cependant, (Drexel et Kimms 1997) montrent que l'établissement du PDP et du MRP sont des processus dont l'interaction est très forte et, par suite, que ces deux plans sont intimement liés. En effet, le principe utilisé pour établir ces plans se découpe en trois phases :

Phase I : En commençant par les produits finis, on calcule les quantités à fabriquer pour chaque produit, et ceci, niveau par niveau pour tous les produits (finis ou composants). Durant cette étape, les contraintes de capacité sont ignorées,

Phase II : Les résultats obtenus durant la Phase I violent généralement les contraintes de capacité pendant certaines périodes, la Phase II consiste à décaler certaines fabrications dans le but de fournir un plan qui les respecte. En faisant ceci, on ignore les relations de précedence entre les produits,

Phase III : La dernière phase consiste à établir le séquençement de la fabrication des produits et à délivrer des ordres de fabrication au système de production.

C'est pour cela que beaucoup d'auteurs ((Stadler 2005), (Meyr et al. 2005)) considèrent que le MRP se trouve à la frontière entre le niveau tactique et le niveau opérationnel.

La figure 1.3 montre comment s'articule la méthode MRP II dans le cadre de la planification industrielle.

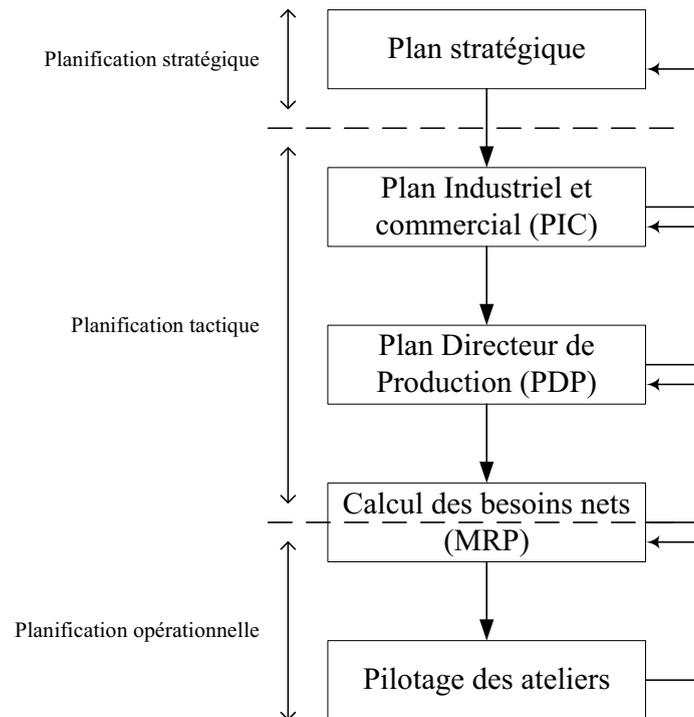


Figure 1.3 – MRP II et la planification tactique

3 Les limites de MRP II

La littérature fait état d'un bon nombre de critiques concernant la logique MRP II. Mais comme le souligne (Genin 2003), ces dernières sont particulièrement axées sur des problèmes de synchronisation entre les différents plans, engendrés par le concept même de planification hiérarchisée. En effet, lorsque nous avons introduit le concept MRP II nous avons utilisé la phrase suivante : « Chaque plan, révisé avec une périodicité qui lui est spécifique, sert de cadre de décisions pour l'établissement du plan de niveau inférieur ». Bien que d'apparence anodine, cette dernière fait clairement apparaître les difficultés de synchronisation liées à l'approche MRP II.

3.1 Synchronisation des capacités et des matières

Comme illustrée par la figure 1.1, c'est la fonction de gestion de la capacité qui assure la synchronisation du flux physique et des ressources. Pour illustrer notre propos, prenons l'exemple du PIC et du PDP : le PIC détermine les niveaux de production pour chaque famille de produits et fournit donc un plan de production agrégé. Le PDP a pour fonction la planification de la production exprimée au niveau des produits finis, dans le cadre de décisions fixé par le PIC : la somme des quantités de production des produits finis doit être égale à la quantité indiquée par le PIC de la famille de ces derniers. Cela a deux implications fortes :

- le PIC doit assurer l'existence d'au moins un PDP,
- le PDP doit être compatible avec le PIC.

(Hennet 1997) parle respectivement de *robustesse* du plan agrégé (PIC) et de *cohérence* du plan détaillé (PDP). L'obtention d'une telle robustesse (resp. cohérence) est assurée par la gestion des capacités. Cependant, la façon dont sont prises en compte ces dernières dépend du niveau du plan dans lequel elles s'inscrivent. En effet, plus le plan se situe à un niveau hiérarchique élevé, plus la capacité considérée est agrégée : au niveau du PIC, elle le sera au niveau usine, pour le PDP, elle le sera au niveau des ressources critiques de chaque usine. En conséquence, outre le fait que ces capacités soient difficilement estimables, le PIC peut prévoir certaines quantités à fabriquer qui ne seront pas réalisables au niveau du PDP car l'estimation des capacités au niveau du PIC sera trop grossière pour assurer sa robustesse pour le PDP.

Précédemment, nous avons décrit les trois phases permettant d'établir le PDP et le MRP en commençant par planifier à capacité infinie. Cela engendre un grand nombre d'itérations pour obtenir des plans réalisables : c'est également symptomatique de cette difficulté de synchronisation des capacités matières et de plus, ce procédé « heuristique » n'est en rien optimisant en termes de coûts.

3.2 Synchronisation verticale

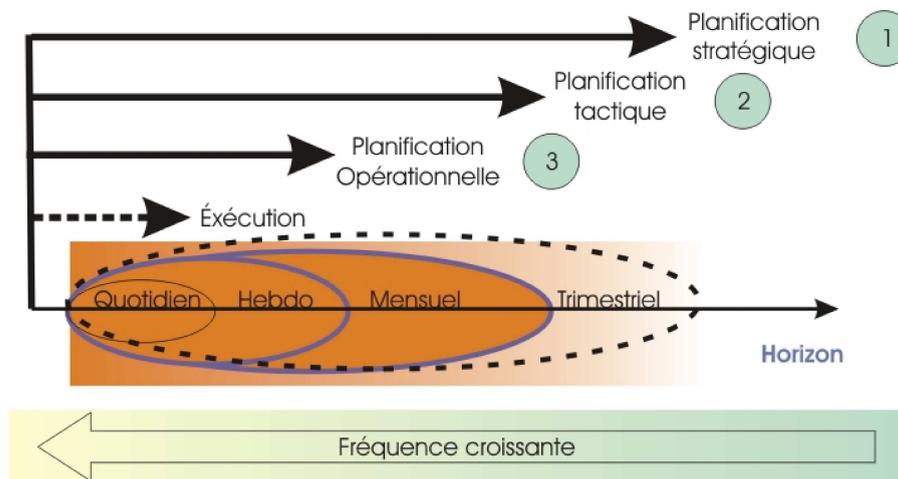


Figure 1.4 – La mise en œuvre temporelle

Comme le montre la figure 1.4, pour chaque niveau hiérarchique, chaque plan est revu périodiquement et indépendamment des plans de niveau inférieur. De plus, les fréquences de révision de ceux-ci ne sont pas identiques : plus l'horizon du plan est petit, plus sa fréquence de révision est élevée. De sorte que si un aléa survient, il ne sera pas pris en compte simultanément dans l'ensemble des plans concernés ce qui induira une désynchronisation de ceux-ci.

II La chaîne logistique et le management de la chaîne logistique

Dans la section précédente, nous avons présenté la méthode MRPII, initialement orientée mono-site. Cependant, les mécanismes de fusion/acquisition entre entreprises ont amené à définir, dans un premier temps, la notion de groupe industriel et donc à considérer des entreprises multi-site faisant ainsi émerger une nouvelle problématique liée à la planification tactique : la synchronisation horizontale qui vise non seulement à générer des plans de production pour chaque site de production, mais aussi à en assurer la synchronisation (des produits fabriqués par un site pouvant être nécessaires à la fabrication d'un produit sur un autre site de production). Puis est apparu le concept de chaîne logistique : au travers d'un jeu d'alliance, d'externalisation et de sous-traitance, les groupes industriels ont pu étendre leur champ d'influence sans pour autant consacrer leurs ressources financières à des acquisitions ne rentrant pas dans leur « cœur de métier ».

1 Notion de chaîne logistique

(Porter 1986) a introduit le concept de « chaîne de valeur » dans le but de décomposer l'activité de l'entreprise en séquence d'opérations élémentaires afin de permettre le développement d'un avantage compétitif sur le marché. (Gervais 1995) précise : « La valeur est la somme que les clients sont prêts à payer pour obtenir le produit qui leur est offert. Cette valeur résulte de différentes activités réalisées

par les fournisseurs, la firme et les circuits de distribution, que ceux-ci soient intégrés ou non à la firme. (...) Au sein de cet agrégat, M. Porter chercha à identifier les lieux essentiels de création de la valeur, de manière à cerner les modalités permettant d'obtenir un avantage concurrentiel ». Forte de son succès, la chaîne de la valeur a été suivie par d'autres : chaîne de la demande, chaîne relationnelle et bien sûr chaîne logistique (supply chain).

Dans la littérature, on peut retrouver nombre de définitions de la chaîne logistique. (Féniès et al. 2004) la définissent comme étant : « un ensemble ouvert traversé par des flux, composé d'entités et d'acteurs variés autonomes qui utilisent des ressources restreintes (capital, temps, matériel, matières premières...) et qui coordonnent leur action par un processus logistique intégré afin d'améliorer prioritairement leur performance collective (satisfaction du client final, optimisation globale du fonctionnement de la Supply Chain) mais aussi à terme leur performance individuelle (maximisation du profit d'une entité) ». Cette définition montre qu'une chaîne logistique se définit par la réunion de trois ensembles :

- un réseau composé d'entités physiques (usines, ateliers, entrepôts, distributeurs, grossistes, détaillants,...) et d'organisations autonomes (firmes, filiales, ...) ;
- un ensemble ouvert traversé par des flux (financier, matériel, informationnel, ...) ;
- un ensemble d'activités regroupées dans un processus logistique intégré dont l'agencement constitue une chaîne de valeur intra et inter-organisationnelle.

C'est sur ce premier ensemble que se focalise généralement les définitions de la chaîne logistique. En effet, (Lee et Billington 1993) définissent la chaîne logistique comme : « un réseau de sites de production et de sites de distribution qui se procurent des matières premières, les transforment et les distribuent au consommateur ». Pour (Ganeshan et Harrison 1995), la chaîne logistique est « un réseau d'entités de production et de sites de distribution qui réalise les fonctions d'approvisionnement de matières, de transformation de ces matières en produits intermédiaires et/ou finis, et de distribution de ces derniers jusqu'aux clients ».

Dans le second ensemble, on retrouve :

- le flux physique (achat de matières, transformation des matières premières en produits, livraison des produits). L'optimisation de ce flux vise à satisfaire les clients du système.
- le flux monétaire ou financier. Ce flux circule en sens inverse du flux physique mais de manière erratique. L'optimisation du flux financier est faite de manière locale, dans chaque maillon de la chaîne logistique, mais rarement de manière globale. Pour (Shapiro 2001) : « L'optimisation de ce flux financier (cash-flow) permet a priori d'obtenir la satisfaction des actionnaires des entités de la chaîne logistique et d'améliorer son fonctionnement global ». Le flux financier est la contrepartie monétaire du flux physique.
- le flux informationnel qui permet la coordination des flux financiers comme physiques entre chaque nœud, et une coordination globale. Quel que soit le niveau hiérarchique du système logistique, les flux d'information contiennent les données nécessaires au pilotage et à la gestion de l'ensemble des activités de cette unité. Pour chaque activité, les informations suivantes doivent être disponibles :
 - l'information propre aux moyens qui la réalisent (l'état de ces moyens, la connaissance et les données nécessaires à leur fonctionnement, les règles locales pour leur pilotage) ;
 - l'information concernant l'état des activités qui dépendent de celle-ci ;
 - l'information sur son état concernant les activités dépendantes ;
 - l'information sur son état concernant le système de gestion de l'unité flexible de production/stockage ou le retour de l'information (feed-back) ;
 - les règles globales pour le pilotage et la gestion de l'ensemble des activités qui lui sont relatives

(Tchernev 1997) définit le processus logistique comme « un ensemble d'activités ordonnées ayant comme objectif la maîtrise et la gestion des flux matières à travers le système logistique, l'affectation et la gestion des ressources du système logistique afin d'assurer un niveau de service à moindre coût ».

D'après (Stadler 2005), les processus logistiques d'une Supply Chain sont composés de quatre grandes familles d'activités génériques (Figure 1.5) formant le dernier ensemble définissant la chaîne logistique :

- approvisionner en matières,
- fabriquer des produits,
- distribuer les produits,
- vendre les produits.

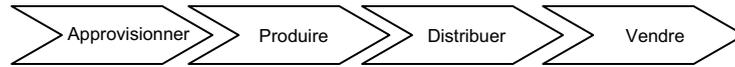


Figure 1.5 – Les activités du processus logistique d'une chaîne logistique (Stadler 2005)

La littérature traitant de la chaîne logistique est vaste et traite de problèmes aussi variés que sa conception, son optimisation et sa gestion. Ces problématiques sont usuellement regroupées sous le vocable de « management de la chaîne logistique ». Plus précisément, (Samii 2006) en donne la définition suivante : le supply chain management est « la coordination systémique et stratégique des fonctions traditionnelles de l'entreprise dans un réseau inter-entreprises avec, pour objectif, d'améliorer les performances à long terme à la fois de l'entreprise concernée et de la supply chain dans son ensemble ». C'est en ce sens que (Thomas et Lamouri 2000) définissent le management de la chaîne logistique comme une évolution de MRPII.

2 Le management de la chaîne logistique

Le concept de management de la chaîne logistique (supply chain management -SCM-) a été introduit par (Oliver et Webber 1992) et a fait l'objet de nombreux ouvrages. Dans leurs travaux de synthèse, (Stadler et Kilger 2004) le définissent comme « une démarche permettant l'intégration d'unités organisationnelles le long d'une chaîne logistique et la coordination des flux physiques, informationnels, et financiers dans le but de satisfaire le consommateur final et d'améliorer la compétitivité de la supply chain dans son ensemble » et en illustrent les multiples facettes grâce à la « Maison du management de la chaîne logistique » (Figure 1.6).

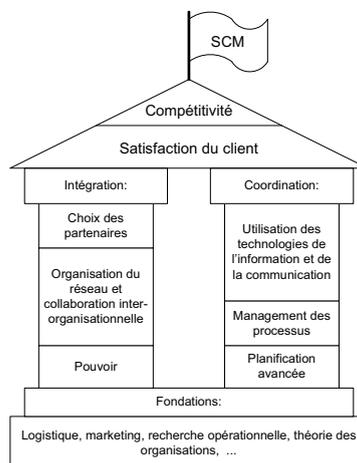


Figure 1.6 – La maison du management de la chaîne logistique (Stadler 2002)

Au travers de cette illustration, ces auteurs montrent qu'une démarche de management de la chaîne

logistique vise à accroître la compétitivité de l'ensemble de la chaîne logistique pour une meilleure satisfaction du client. Pour atteindre cet objectif (symbolisé par le toit de la maison), on s'appuie sur deux composantes majeures du management de la chaîne logistique : l'intégration du réseau des acteurs de la chaîne logistique et la coordination des flux physiques, financiers et informationnels (les deux piliers de la maison). Ces deux piliers reposent sur des fondations que sont la logistique, le marketing etc.

Pour (Stadler et Kilger 2004), l'intégration du réseau des acteurs de la chaîne logistique consiste :

- à choisir les partenaires de la supply chain en fonction de leur cœur de métier,
- à organiser le réseau en définissant les collaborations interorganisationnelles,
- à définir les relations de pouvoir au sein de la chaîne logistique : en effet, certaines décisions doivent être prises pour l'ensemble de la chaîne logistique (par exemple, l'intégration d'un nouveau partenaire dans la chaîne). Il faut donc définir le(s) instance(s) capable(s) de prendre et d'imposer de telles décisions.

Dans la coordination des flux, on retrouve :

- l'utilisation des technologies de l'information et de la communication telles que l'Echange de Données Informatisé (EDI),
- le management des processus qui vise à coordonner l'ensemble des activités participant à la réalisation des commandes clients de la manière la plus efficiente possible,
- la planification avancée qui intègre les problématiques de planification long, moyen et court terme : c'est à ce niveau qu'on retrouvera les problématiques liées à la planification tactique.

2.1 La planification tactique : partie intégrante du management de la chaîne logistique

Nous avons vu que les problèmes liés à la planification de la chaîne logistique se retrouvaient au niveau de la planification avancée dans la maison du management de la chaîne logistique. La figure 1.7 détaille les diverses problématiques qu'on y retrouve.

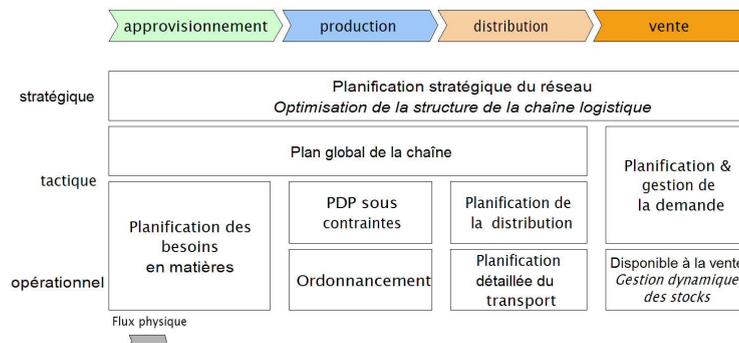


Figure 1.7 – Problématique de la planification des chaînes logistiques (Meyr et al. 2002)

Aux trois niveaux de décision de la planification industrielle s'ajoute une dimension formée des processus industriels définis par (Stadler 2005) enrichissant ainsi ces niveaux hiérarchiques de la prise en compte des problématiques liées à la chaîne logistique. Au niveau tactique, nous retrouvons le « plan global de la chaîne » (Master Planning) qui étend la problématique du PIC à l'ensemble de la chaîne, englobant ainsi des problématiques d'approvisionnement, de production et de distribution. Nous retrouvons également le PDP et le calcul des besoins nets. Au niveau de la distribution, nous trouvons un nouveau plan concernant la distribution. Ce dernier a donné lieu à une nouvelle technique : le Distribution Resource Planning (DRP). S'appuyant sur le système logistique externe aval (système physique de distribution), il a pour objectif de maintenir au niveau strictement nécessaire l'ensemble des stocks dans tout le système de distribution (Thomas et Lamouri 2000). Le fonctionnement du DRP est similaire à un MRP : partant de la demande client et suivant une logique analogue à celle de

la méthode MRP, la demande prévisionnelle au niveau de l'usine est calculée à partir des demandes prévisionnelles au niveau aval (en suivant la nomenclature de distribution), en tenant compte des stocks existants et des règles de réapprovisionnement : délai, stock de sécurité, quantité standard de réapprovisionnement. Ainsi, la planification des besoins de produits finis crée des besoins matières ou composants qui nécessitent des ordres de réapprovisionnement, qui eux-même génèrent des demandes de produits finis aux entités amonts, etc. (Thomas et Lamouri 2000) proposent le schéma suivant pour expliciter l'intégration du DRP à la structure de planification MRPII (Figure 1.8).

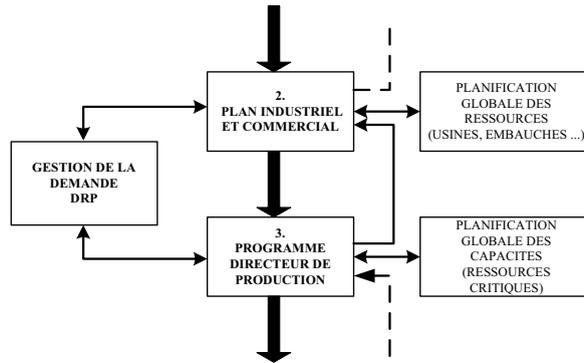


Figure 1.8 – Intégration du DRP dans MRPII (Thomas et Lamouri 2000)

2.2 Le problème d'intégration horizontale

La prise en compte des éléments constitutifs de la chaîne logistique dans le processus de planification tactique étend le problème de synchronisation horizontale des sites de production à tous les acteurs de la chaîne : on parle alors d'intégration horizontale (Genin 2003). La figure 1.9 illustre le principe de l'intégration horizontale.

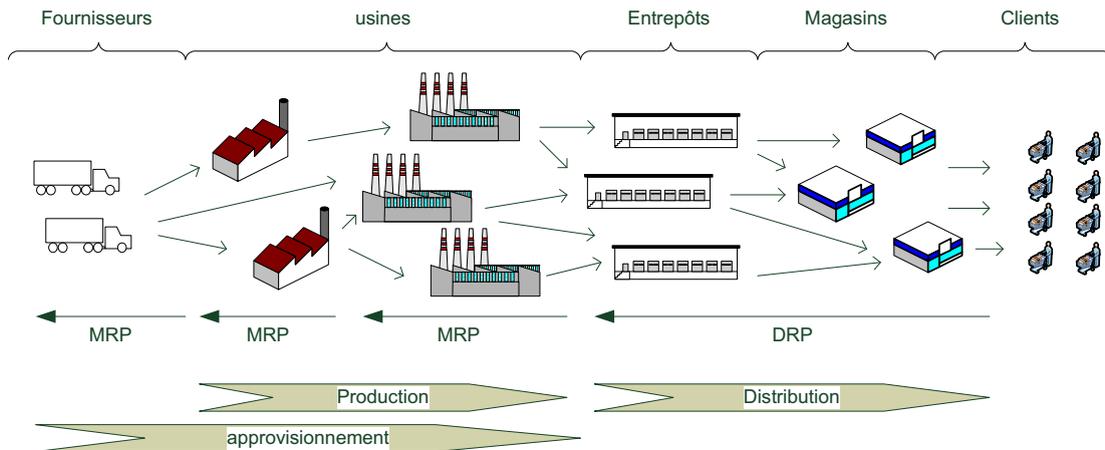


Figure 1.9 – L'intégration horizontale

Les demandes exprimées par les clients sont remontées séquentiellement de l'aval de la chaîne vers l'amont. Le résultat du DRP est transmis au MRP des premiers sites de production ce qui leur génère des besoins en composants qu'ils transmettent à leur tour au MRP des sites amonts et ainsi de suite jusqu'aux fournisseurs. (Genin 2003) remarque que les processus de planification menés à l'aide de ces systèmes peuvent aboutir à des plans réalisables pour chaque entité logistique mais qu'ils ne sont pas optimisés dans leur ensemble. De plus, il se peut qu'on arrive à des plans irréalisables en raison d'un

manque de cohérence entre eux (par exemple, un PDP peut avoir été élaboré pour une usine mais être irréalisable en raison de contraintes d’approvisionnement) ce qui entraîne une désynchronisation horizontale (Crama et al. 2001).

Ce principe d’intégration horizontale induit deux approches possibles de planification dans la chaîne logistique :

- une approche décentralisée. Les acteurs de la chaîne logistique ne partagent qu’une certaine quantité d’informations et génèrent leur planning de production isolément, induisant par suite une optimisation locale (pour chaque maillon de la chaîne) du processus de planification,
- une approche centralisée. Supposant une très forte (mais hypothétique) collaboration des différents acteurs de la chaîne logistique, elle permet de concevoir des plans de production apportant une solution globalement meilleure que celle obtenue par une approche décentralisée (Thierry et al. 1994). Mais il est évident qu’une telle approche ne devient totalement pertinente que dans le cas d’une chaîne logistique interne (les acteurs appartiennent au même groupe industriel par exemple), où le partage de l’information s’il n’est pas total est pour le moins très fort.

Bien que beaucoup d’auteurs considèrent comme très difficile l’adoption d’une approche centralisée, ces résultats en terme d’optimisation poussent ainsi de nombreux auteurs à étudier la pertinence du partage de l’information au cœur des chaînes logistiques afin d’en améliorer la performance globale.

Notre problématique étant l’élaboration de plans de production (master planning, PDP et MRP) pour la chaîne logistique, nous nous placerons dans le cas d’une approche centralisée.

2.3 Solutions logicielles pour la planification de la chaîne logistique

Comme la gestion industrielle regroupe toutes les fonctions de l’entreprise, elle comprend, outre la gestion de la production, la gestion des ressources humaines, des achats, de la qualité, financière... Il existe donc plusieurs types de progiciels répondant aux multiples besoins d’une entreprise. Nous trouvons :

- les ERP (Enterprise Resource Planning) : progiciels de gestion intégrée,
- les APS (Advanced Planning and Scheduling) : progiciels de planification et d’optimisation,
- les LES (Logistics Execution System) : progiciels d’optimisation de la logistique de distribution,
- les MES (Manufacturing Execution System) : progiciels de contrôle et d’optimisation de l’exécution de la production,
- les outils dédiés (à une ou plusieurs fonctions).

La figure 1.10 (exp 2006) présente le positionnement simplifié des principaux progiciels pour la gestion industrielle :

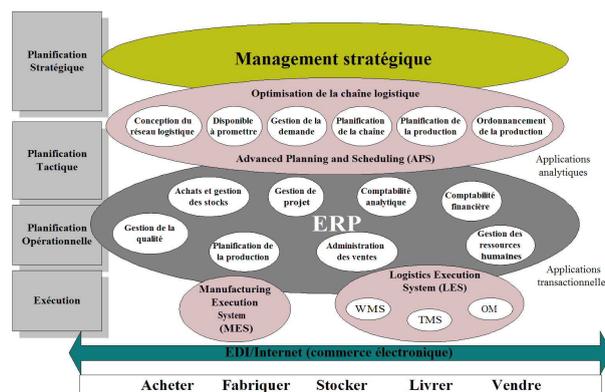


Figure 1.10 – Des progiciels complémentaires.

Ces offres sont complémentaires et permettent de couvrir l'ensemble des besoins de l'entreprise.

Les LES sont des progiciels qui rationalisent la totalité du cycle de traitement des commandes (de sa connaissance à la livraison). Ils regroupent 3 fonctions :

- la gestion de l'entrepôt (WMS pour Warehouse Management System),
- la gestion des transports (TMS pour Transport Management System),
- la gestion avancée des commandes (OM pour Order Management).

Leur principal objectif est d'apporter une réponse la plus rapide possible à une demande complexe des clients. Ils réagissent à des délais courts, des fluctuations de la demande au jour le jour ou en temps réel, au fort taux de rotation des produits, etc. Leur périmètre fonctionnel est restreint, limité à la logistique de distribution au niveau exécutif.

MES est un acronyme inventé par l'AMR-Research, et désigne des systèmes qui délivrent des informations pertinentes en temps réel sur l'exécution des ordres de fabrication, dans le but de les contrôler depuis leur lancement jusqu'à l'obtention des produits finis et d'optimiser les activités de production. Ceci s'effectue en cherchant à effectuer un meilleur suivi et à gérer un planning de production et d'ordonnement réaliste (par identification des goulots et réduction des stocks et des encours). Ils ont un cadre fonctionnel restreint, et sont fortement dépendants du mode de fabrication.

Les outils dédiés représentent des solutions se trouvant sur le marché en position de niches. Nous pourrions citer par exemple les moteurs d'optimisation ou encore les modules de prévisions des ventes.

Tous ces outils reposent sur l'EDI qui permet aux entreprises d'échanger des données avec des prestataires ou collaborateurs externes par des moyens de communication respectant les normes UN/EDIFACT (Echange de Données InFormatisé pour l'Administration, le Commerce et les Transports) et EBXML (Electronic Business using eXtensible Markup Language).

Nous présentons de manière plus détaillée les progiciels de type ERP et APS, offrant des solutions pour répondre aux problématiques de planification tactique.

2.3.1 Enterprise Resource Planning (ERP)

Un ERP est un Progiciel de Gestion Intégré (PGI en français) qui permet de gérer et d'intégrer l'ensemble des processus d'une entreprise (gestion des ressources humaines, gestion comptable et financière, aide à la décision, approvisionnement,...). Il est basé sur deux grands principes :

- *l'intégration* consiste en l'élaboration d'applications informatiques (paie, comptabilité...) modulaires (indépendantes entre elles) tout en partageant une base de données unique et commune. Les données sont standardisées et partagées, ce qui élimine l'ambiguïté des données multiples,
- *le temps réel* : un moteur de Workflow permet, lorsqu'une donnée est entrée dans le système d'information, de la propager dans tous les modules du système qui en ont besoin, selon une programmation prédéfinie.

Selon l'APICS (agence américaine de standardisation des termes industriels, aujourd'hui désignée par The Educational Society for Resource Management), l'ERP est un « progiciel qui assure la mise en temps réel de l'ensemble des fonctions de l'entreprise, tant du point de vue des ordres d'achat, de vente que de gestion des stocks et produits en assurant la charge des différentes ressources et un lien avec la gestion financière ».

En ce qui concerne la gestion de la production, le module correspondant de l'ERP permet de gérer des nomenclatures de produits et des gammes de produits d'un point de vue du suivi de la production.

Il permet en effet de planifier la production, de calculer des besoins, de prévoir les ventes au niveau entreprise (PIC), de prévoir la production (PDP), de calculer des besoins et des ressources (hommes et machines : MRP II), de planifier des capacités, de contrôler la fabrication, de suivre la production, de calculer les coûts de revient. Cela signifie que ce module souffre de carences dues à la méthode MRPII : il ne permet pas, notamment, d'optimiser la planification de la production. De nouveaux outils nommés APS (Advanced Planning System ou Advanced Planning and Scheduling) ont été élaborés pour pallier ces faiblesses.

2.3.2 Advanced Planning System (APS)

D'après (Fleischmann 2004), un « Advanced Planning System » (APS) est un logiciel qui couvre l'ensemble des tâches de planification de la supply chain d'une entreprise manufacturière. Par suite, les APS sont des progiciels qui sont censés optimiser la planification et synchroniser les flux de la chaîne logistique en tenant compte simultanément d'un grand nombre de contraintes (ressources, capacités, délais, coûts) et ce, sur plusieurs niveaux de planification (stratégique, tactique et opérationnel) comme l'illustre la figure 1.11.

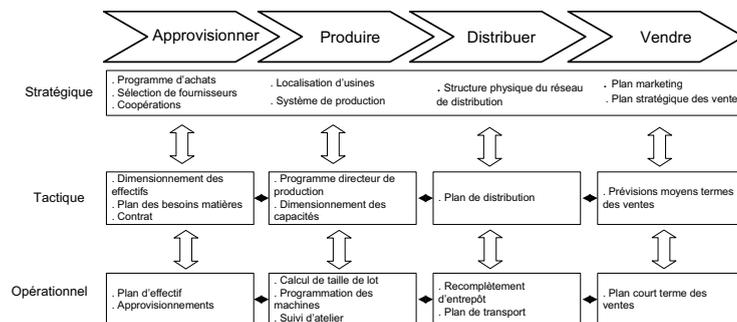


Figure 1.11 – Les fonctionnalités des APS
(Meyr et al. 2005)

D'après (Fleischmann 2004), la réelle avancée des APS consiste en l'adjonction aux ERPs traditionnels d'un véritable module de planification basé sur des modèles mathématiques qui permet l'accession pour une entreprise à de réelles fonctions de planification. Ces outils s'appuient donc sur de puissants solveurs et permettent de répondre en partie aux problèmes de la planification industrielle. Au travers de ceux-ci, ils sont donc capables d'élaborer des solutions globales pour la chaîne logistique (« synchronisation horizontale ») et d'optimiser celle-ci. De plus, leur faculté à générer rapidement une solution « multi-niveau » permet de replanifier l'ensemble de la chaîne à chaque événement considéré comme majeur (contrairement à l'approche périodique de MRP II), comme le montre la figure 1.12. Ce dernier point ne résout pas les problèmes concernant la consolidation et la replanification (consommation des commandes fermes) : en effet, une replanification à chaque événement mineur ou majeur entraîne une instabilité pouvant être néfaste à la performance globale du système (Genin 2003).

De plus, comme le soulignent (Lamouri et al. 2005), les APS sont conçus pour élaborer des plannings optimaux mais, comme ils ne prennent pas en compte l'incertitude des données (prévisions de la demande...), ils n'élaborent pas de solutions robustes c'est-à-dire capables d'assurer une certaine stabilité de la planification face aux fluctuations possibles des données.



Figure 1.12 – La structure de planification dans un APS
(Lamouri et al. 2005)

Pour finir les calculs des besoins nets (MRP) sont assignés à la fonction d’approvisionnement au niveau opérationnel (Tempelmeier 2001). Or ceci n’est évidemment pas le cas dans un contexte industriel puisqu’ils sont rattachés à la fonction de production. Cela a une conséquence majeure : l’approche des APS pour la planification n’est pas meilleure que celle des ERPs dans le sens où elle s’appuie sur la méthode MRP, négligeant totalement le calcul des tailles de lots.

La littérature traitant des problèmes de planification tactique est vaste. Cependant, une grande partie des travaux qui leur sont consacrés, s’est axée vers la formalisation mathématique de ces problématiques et l’élaboration de méthodes dédiées à ces modèles : ainsi a émergé toute une famille de modèles appelée « Lot Sizing Problem » et dédiée à l’élaboration des différents plans de production (PIC, Master Planning, PDP et MRP). Dans la section suivante, nous proposons un état de l’art de ces modèles et nous verrons que suivant les hypothèses prises en compte, certains d’entre eux ne peuvent pas être considérés comme répondant à une problématique de planification tactique.

III Classification des modèles mathématiques pour la planification tactique

Dans cette partie, nous proposons un état de l’art des principaux modèles de « lot-sizing » et dont un extrait a fait l’objet d’une publication (Comelli et al. 2008).

Comme nous l’avons décrit précédemment, les problématiques liées à la planification tactique sont nombreuses et fonction de la typologie de la structure à laquelle elles se rapportent (entreprise mono-site, multi-site; chaîne logistique...). C’est pour cela que nous avons commencé par recenser les critères discriminants qui nous ont permis de classifier les modèles rencontrés :

- la typologie de la structure considérée : nous allons retrouver les problématiques mono-site et multi-site. Comme nous le verrons par la suite, la planification tactique de la chaîne logistique est usuellement traitée comme un problème de planification multi-site (avec les spécificités d’approvisionnement et de distribution en sus),
- la modélisation de la demande (constante ou variable) : une demande est dite constante si elle est répartie uniformément sur tout l’horizon. Dans le cas contraire, on parle de demande variable. on discrétise alors l’horizon en périodes et on agrège la demande sur celles-ci. Cette distinction fondamentale a permis de définir deux approches de résolution très différentes :
 - *les modèles à demande constante* ont pour caractéristique de fournir comme solution un cycle

- de production, reproductible au cours de l'horizon, minimisant les coûts de production,
- *les modèles à demande variable* déterminent des quantités à produire sur des périodes préalablement définies sur l'horizon de planification permettant de minimiser les coûts de production et de répondre au besoin client.
- la façon de prendre en compte la nomenclature des produits : les problèmes de planification tactique présentés dans leur globalité consistent en la planification des produits finis et de leurs composants. Ceux-ci peuvent être traités de deux manières différentes :
 - *niveau par niveau*. On effectue une planification à partir de la logique MRP qui consiste à remonter le besoin en composants à partir des produits finis (dits demande externe). Cette logique définit ainsi une planification niveau par niveau. On peut donc utiliser des modèles mono-niveau, la quantité déterminée au niveau supérieur engendrant la demande (dite interne) au niveau en cours. La limite de cette approche se situe au niveau de la cohérence entre les différentes planifications niveau par niveau.
 - *à partir des modèles dits multi-niveau (Multi Level, ML)* qui consistent à lier les niveaux entre eux en exprimant la demande interne et externe ce qui assure la réalisabilité de la solution. Les modèles multi-niveau se différencient des modèles mono-niveau par les notions de structures arborescentes (nomenclature) et de délai d'obtention des divers composants (lead time),
- la modélisation de l'horizon de planification,
- la prise en compte des capacités des ressources utilisées.

Les difficultés liées à la modélisation des systèmes physiques ont conduit à l'élaboration de deux types de modèles. En effet, il est nécessaire pour certains cas industriels de définir une solution de planification ayant pris en compte des contraintes opérationnelles notamment de séquençement (la phase III du principe de l'élaboration du PDP et du MRP exposée précédemment (Drexl et Kimms 1997)). C'est pour cela que la littérature distingue aussi deux grandes classes de modèles : les modèles dits de « big bucket » et de « small bucket ». Les modèles « big bucket » permettent de modéliser la production de plusieurs produits au cours d'une même période tandis que les modèles « small bucket » ne modélisent que la production d'un seul produit par période et permettent, par suite, de proposer un séquençement de la solution proposée. Les modèles « small bucket » ne sont donc pertinents que dans le cas où les périodes sont très courtes (micro-période, par exemple la journée). En effet, ces modèles supposent que le système de production ne fabrique qu'un seul type de produits par période. En revanche, ces modèles permettent de modéliser plus finement les coûts de lancement (reconfiguration du système de production) en séquençant la production. A l'opposé, les modèles « big bucket » sont pertinents pour des périodes plus longues (macro-période, la semaine par exemple). Ainsi, suivant la façon dont est modélisée l'horizon de planification (micro-périodes ou macro-périodes), nous retrouvons deux types de modèles très différents. Cependant, il est important de noter que les modèles « small bucket », en s'intéressant au séquençement de la production, se trouvent vraiment à la frontière avec les problématiques d'ordonnancement : c'est pour cela qu'ils sont souvent considérés comme ne faisant pas partie des modèles de planification tactique.

Toutes les différenciations citées précédemment sont liées à la pertinence de la modélisation. Cependant, pour aider à la résolution de ceux-ci, d'autres modèles ont été développés notamment ne prenant pas en compte la capacité du système de production : ainsi, dans la littérature il est aussi possible de distinguer deux autres critères de différenciation qui sont la prise en compte ou non de la capacité et les modèles mono ou multi-produit. Bien qu'étant irréalistes, les modèles sans capacité et/ou mono-produit permettent d'élaborer des stratégies de résolution pour des problèmes plus complexes en fournissant, par exemple, des bornes inférieures à ceux-ci. Dans l'état de l'art que nous proposons, nous ne ferons pas mention explicitement des modèles mono-produit car ils sont obtenus comme cas particulier des modèles multi-produit. Bien évidemment, dans ce cas, les concepts de « small » ou « big bucket » ne présentent plus aucun intérêt. Le lecteur intéressé pourra se référer à l'état de l'art présenté par (Brahimi et al. 2006).

Les modèles mathématiques divergent aussi par la prise en compte de diverses autres hypothèses telles que le backlogging (retard dans la demande client encore appelé backorder) ou les temps de set-up (temps de lancement de la production) qui, représentant des contraintes génériques pour les modèles de planification, ne peuvent être considérées comme discriminantes.

Reprenant et complétant le schéma proposé par (Rizk et Martel 2001), la figure 1.13 récapitule la classification des différentes formalisations présentées dans le cadre mono-site.

1 Les modèles mono-site

Un problème de planification mono-site peut se formaliser ainsi :

Objectif :

Déterminer sur un horizon moyen terme (6 à 18 mois) des quantités de produits ou de famille de produits de telle façon à optimiser une fonction objectif. Cette dernière, selon les cas d'étude, exprime un compromis entre les coûts de production, de lancement, de stockage. Lorsque la demande client doit être entièrement satisfaite et sans retard, la planification consiste dans un premier temps à trouver une solution répondant à cette exigence. Dans le cas contraire, divers coûts peuvent être imputés (retard, annulation de commande etc.).

Données :

Les données d'entrée du modèle sont :

- la durée de l'horizon de planification,
- la demande client par produit ou par famille de produits pour chaque période.

Paramètres :

Les paramètres pris en compte sont le plus souvent :

- le coût unitaire de stockage,
- le coût unitaire de production,
- le coût unitaire de lancement d'une campagne de production,
- la capacité de production du système de production,
- la capacité de production nécessaire pour la fabrication d'un produit,
- la capacité nécessaire pour effectuer un lancement de production.

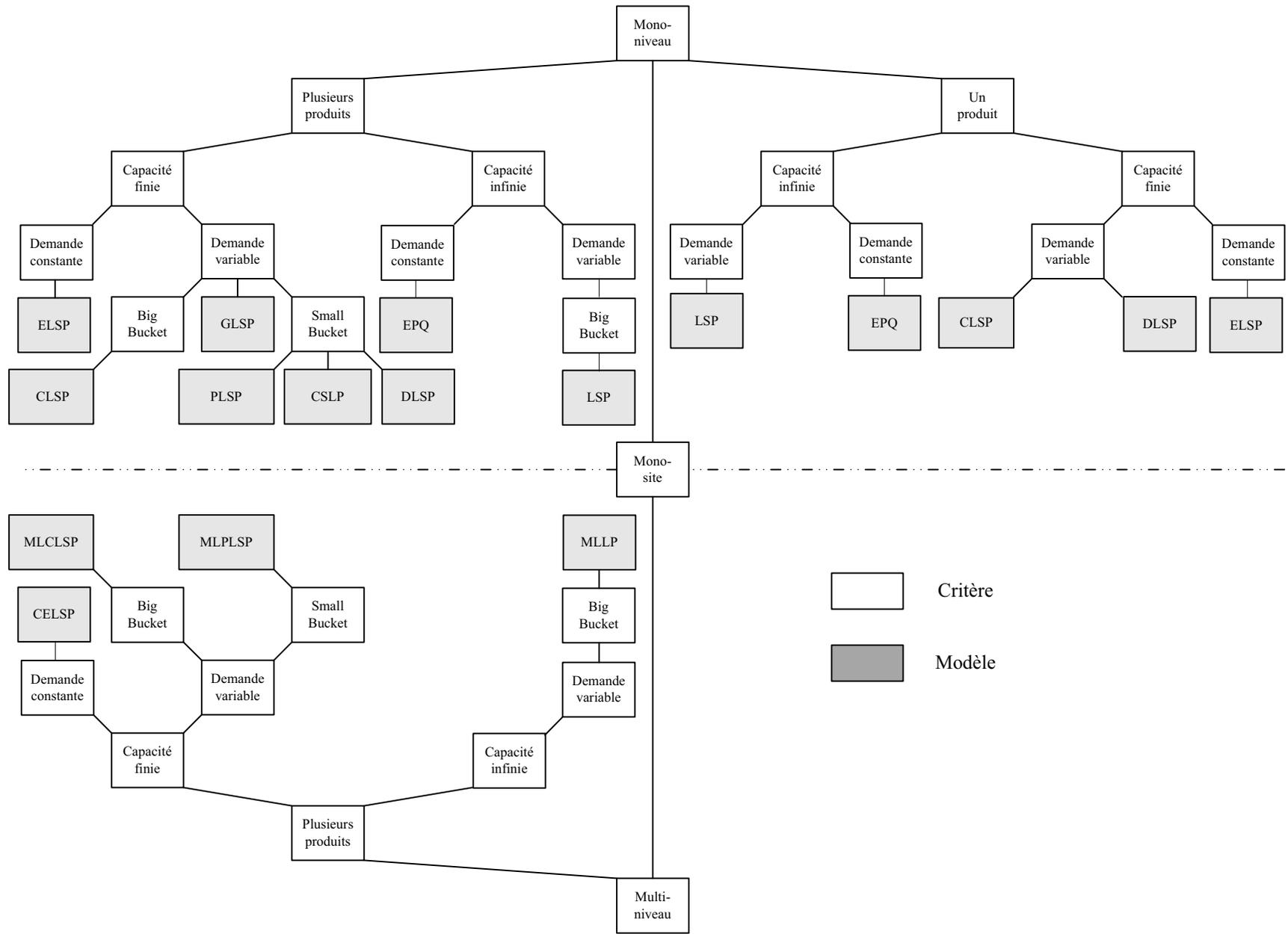
Variables :

Elles représentent :

- les quantités produites,
- les quantités en stocks,
- les débuts de campagne (afin d'imputer les coûts de lancement de campagne).

Nous commençons par présenter les principaux modèles mono-niveau en les déclinant suivant les critères discriminants énoncés ci-dessus.

Figure 1.13 – classification des modèles mono-site



1.1 Les modèles mono-niveau

Dans notre classification, nous distinguons les modèles à capacité infinie et à capacité finie.

1.1.1 Capacité infinie

On suppose donc dans ce paragraphe que la capacité de production est assez grande pour qu'on puisse la considérer comme infinie. Selon notre classification, nous commençons par présenter le modèle à demande constante puis celui à demande variable.

1.1.1.1 Demande constante

Supposons que nous ayons à planifier la fabrication de N types de produits. La capacité de production étant supposée infinie et les produits n'ayant aucune relation entre eux, on remarque alors que planifier la fabrication des N types de produits revient à élaborer un planning par type de produits fabriqués. C'est pour cela que nous ne nous intéresserons ici qu'au cas où nous n'avons qu'un seul type de produits à fabriquer : ce problème a donné lieu au modèle dit du « lot économique (Economic Product Quantity) ».

Cette approche consiste en un problème de gestion d'un seul produit pendant une durée indéterminée sous les hypothèses suivantes :

- la demande est constante et intervient de manière uniforme au cours du temps, μ pièces étant retirées du stock à chaque unité de temps,
- à chaque production, un coût fixe $K \geq 0$ doit être payé, ainsi qu'un coût variable cQ proportionnel au nombre Q d'unités produites ($c \geq 0$),
- la production est immédiate,
- les retards (*backlogging*) ne sont pas admis,
- pour chaque unité de temps et chaque produit dans le stock, un coût unitaire de stockage $h \geq 0$ doit être payé.

L'objectif est de déterminer une politique de fabrication minimisant la somme des coûts de fabrication et de stockage par unité de temps. Dans le cas d'une politique de gestion optimale, il est clair qu'on ne doit pas passer d'ordre de production tant que le stock n'est pas vide car la production étant immédiate, une telle décision n'aurait comme conséquence que d'augmenter inutilement les coûts de stockage. De plus les retards étant interdits, une fabrication doit prendre place au plus tard lorsque le niveau du stock atteint zéro. Les coûts ne variant pas au cours du temps et le processus étant supposé se répéter à l'infini, le nombre de produits à fabriquer pour chaque ordre de production doit être le même. D'où :

Théorème 1.1 *Dans le modèle du lot économique, la politique optimale est cyclique et consiste à produire par lots de Q unités tous les T jours.*

Se basant sur le fait que $Q = \mu T$ (production lorsque le stock est vide), la taille Q du lot ainsi que le nombre de jours T formant le cycle sont donnés par les formules de Wilson :

$$Q = \sqrt{\frac{2\mu K}{h}} \text{ et } T = \sqrt{\frac{2K}{\mu h}}$$

La figure 1.14 illustre l'évolution des différentes coûts pris en compte dans le modèle EPQ.

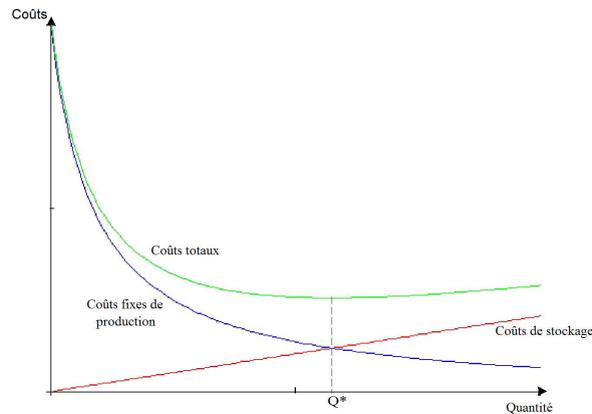


Figure 1.14 – Représentation graphique des différents coûts pour le modèle EPQ

Il existe beaucoup d'extensions à ce modèle :

- les délais de production sont constants et non nuls. Un tel délai n'aura naturellement aucune influence sur les coûts de gestion du stock et les formules de Wilson définiront toujours une politique optimale. Mais pour satisfaire la demande sans retard (pas de pénurie) les ordres de fabrication doivent prendre place suffisamment tôt afin que la fin de fabrication des produits coïncide avec le moment où le stock sera nul. Le niveau du stock où un ordre devra être émis sera appelé *point de fabrication* : (Hêche et al. 2003) nous dit que si le délai de production L est inférieur ou égal à la durée T du cycle optimal, ce point de commande est égal à μL . Si L est supérieur à T alors le point de fabrication est égal au reste de la division euclidienne de μL par Q .
- il existe également une variante du modèle qui prend en compte les retards (*backlogging*) en introduisant des coûts de pénuries dans le modèle de base et un nombre S d'articles non livrés déclenchant un nouvel ordre de fabrication.

1.1.1.2 Demande variable

Dans ce sous-paragraphe nous supposons que la demande est variable. Le but est de déterminer des quantités à produire sur des périodes préalablement définies sur l'horizon de planification permettant de minimiser les coûts de gestion et de répondre au besoin client. Ce modèle appelé le « Lot Sizing Problem » prend parfois le nom de modèle de Wagner Within, du nom des auteurs qui ont trouvé le premier algorithme polynômial capable de le résoudre (Wagner et Whitin 1958). Puisqu'il permet de planifier plusieurs produits par période, c'est un modèle « big bucket ».

Notations :

Les données du modèle sont :

- T : Longueur (en périodes) de l'horizon de la planification,
- N : Nombre de produits,
- d_{it} : Demande en produit i à la fin de la période t , $i \in \llbracket 1, N \rrbracket, t \in \llbracket 1, T \rrbracket$
- h_i : Coût de stockage du produit i durant une période, $i \in \llbracket 1, N \rrbracket$
- I_{i0} : Stock initial en produit i , $i \in \llbracket 1, N \rrbracket$
- p_i : Capacité nécessaire pour produire une unité de produit i , $i \in \llbracket 1, N \rrbracket$
- s_i : Coût de lancement du produit i . $i \in \llbracket 1, N \rrbracket$

Les variables de décisions du modèle sont :

$$\begin{aligned}
Q_{it} &: \text{Quantité de produit } i \text{ fabriquée à la période } t, & i \in \llbracket 1, N \rrbracket, t \in \llbracket 1, T \rrbracket \\
I_{it} &: \text{Niveau de stock en produit } i \text{ à la période } t, & i \in \llbracket 1, N \rrbracket, t \in \llbracket 1, T \rrbracket \\
X_{it} &: \begin{cases} 1 & \text{s'il y a lancement pour le produit } i \text{ à la période } t, \\ 0 & \text{sinon.} \end{cases} & i \in \llbracket 1, N \rrbracket, t \in \llbracket 1, T \rrbracket
\end{aligned}$$

En utilisant ces notations, on peut écrire ce problème sous forme d'un programme linéaire mixte :

$$\text{Minimiser } \sum_{i=1}^N \left(\sum_{t=1}^T (s_i X_{it} + h_i I_{it}) \right) \quad (1.1)$$

Sous les contraintes :

$$I_{it} = I_{i(t-1)} + Q_{it} - d_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.2)$$

$$Q_{it} \leq \left(\sum_{t=1}^T d_{it} \right) X_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.3)$$

$$X_{it} \in \{0, 1\} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.4)$$

$$I_{it}, Q_{it} \geq 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.5)$$

L'objectif (1.1) est de minimiser la somme des coûts de lancement et de stockage de tous les produits pour toutes les périodes.

La contrainte (1.2) représente l'équilibre des stocks.

La contrainte (1.3) impose qu'un lancement pour le produit i soit effectué à la période t si on a une fabrication de celui-ci durant cette période.

La contrainte (1.4) définit les variables de lancement comme étant binaires et (1.5) est une condition de non-négativité.

(Wagner et Whitin 1958) ont proposé un algorithme en $O(T^2)$ à base de programmation dynamique pour résoudre ce problème. Ensuite la complexité de celui-ci a été améliorée en $O(T \ln(T))$ par divers auteurs tels que (Wagelmans et al. 1992), (Aggarwal et Park 1993) et (Federgruen et Tzur 1991).

1.1.2 Capacité finie

Contrairement au paragraphe précédent, la capacité de production est supposée finie. Selon notre classification, nous commençons par présenter le modèle à demande constante puis ceux à demande variable.

1.1.2.1 Demande constante

Le problème du « **Economic Lot Scheduling Problem** » (ELSP) est une extension du problème du lot économique dans lequel sont prises en compte les capacités du système de production. Son but est de planifier la fabrication d'un ensemble de produits sur une seule machine (c'est un modèle mono-ressource) dans le but de minimiser le coût moyen de stockage et de lancement sous les hypothèses d'une demande et d'un taux de production constants et finis par produit. Le fait que les taux de production soient finis montre que la ressource admet une capacité maximale de fabrication par produit et par unité de temps. (Hsu 1983) a montré que ce problème est \mathcal{NP} -difficile. Mais il a été montré par (Carlier et Chrétienne 1988) que sur un horizon infini, ce problème admet une solution périodique. Le but est donc de rechercher un cycle de production pour chaque produit qui minimise la somme des coûts moyens de stockage et de lancement de chaque produit sur ces cycles.

Comme pour le modèle EPQ, chaque cycle de production pourra se diviser en deux parties distinctes :

- une partie où il y aura production et consommation (pour répondre à la demande durant cette partie),

- une seconde où il y aura consommation des produits mis en stock pendant la première phase.

Les notations du modèle sont :

N	: Nombre de types de produits à fabriquer.	
T_i	: Cycle de production pour le produit i	$i \in \llbracket 1, N \rrbracket$
d_i	: Demande constante en produit i par unité de temps	$i \in \llbracket 1, N \rrbracket$
p_i	: Taux de production constant de produit i par unité de temps	$i \in \llbracket 1, N \rrbracket$
c_i	: Coût de lancement du produit i	$i \in \llbracket 1, N \rrbracket$
h_i	: Coût de stockage du produit i par unité de temps	$i \in \llbracket 1, N \rrbracket$

Le problème du ELSP peut s'écrire sous la forme :

$$\min_{(\{T_i\}_{i \in \llbracket 1, N \rrbracket}) \in \Gamma} \sum_{i=1}^N \left(\frac{c_i}{T_i} + \frac{1}{2} h_i d_i T_i \left(1 - \frac{d_i}{p_i} \right) \right)$$

où Γ représente l'ensemble des n -uplets des cycles T_i pour i allant de 1 à n qui forment une solution réalisable.

Dans son état de l'art sur le ELSP, (Elmaghraby 1978) distingue deux types d'approches de résolution : l'approche analytique qui vise à trouver une solution optimale au problème muni de restrictions convenables et l'approche heuristique qui tend à déterminer des solutions acceptables pour le problème original.

L'approche analytique :

Les restrictions prises dans ce type d'approche portent sur l'ensemble Γ . Plus précisément, on suppose qu'il existe un cycle T dit « de base » tel que tous les autres cycles sont des multiples entiers (k_i) de ce cycle de base (T) (Bomberger 1966). Un cas particulier important est celui où tous les cycles sont égaux au cycle de base : c'est ce qu'on appelle dans la littérature « l'approche par période commune ».

L'approche heuristique :

La plupart des heuristiques proposées dans la littérature s'appuie sur le postulat dont (Boctor 1987) a prouvé la nécessité, que tout temps de cycle est un multiple d'un cycle de base. (Salomon 1991) a décrit les heuristiques basées sur le temps de cycle de base en montrant qu'elles s'appuient sur trois procédures essentielles :

- une qui calcule les paramètres k_i et T ,
- une qui détecte lorsqu'une solution proposée n'est pas réalisable,
- une qui modifie les paramètres en cas de non réalisabilité.

La littérature traitant du ELSP est riche. Comme le prouve l'abondance des publications récentes sur ce modèle, il reste d'actualité (nous avons pu répertorier plus de 12 articles traitant de cette problématique parus entre Janvier 2006 et Décembre 2008). Cependant, l'hypothèse concernant la demande constante est bien trop restrictive pour qu'on puisse considérer son applicabilité dans un contexte industriel classique. Pour cette raison, nous ne présentons pas d'état de l'art sur les méthodes de résolution de ce dernier. Pour celui-ci, le lecteur pourra se reporter à (Elmaghraby 1978) qui fait référence sur ce sujet, ou, plus récemment, à celui présenté par (Raza et Akgunduz 2008) qui comparent dans leur article différentes approches de résolution (heuristiques, métaheuristiques etc.)

1.1.2.2 Demande variable

Nous avons également vu que pour un problème de pertinence de la modélisation des systèmes physiques, la littérature distinguait deux grandes classes de modèles : les « big bucket » et les « small bucket ». Les uns autorisent la production de plusieurs types de produits au sein d'une même période tandis que les autres, en ne permettant qu'un seul type de produit par période, fournissent un séquençement de la solution proposée.

Les modèles « small Bucket »

Comme dit dans l'introduction, les modèles « small bucket » supposent qu'on ne produise pas plus d'un type de produits par période et ne sont pertinents que dans le cas où les périodes sont très courtes. Mais, contrairement au modèle « big bucket », ils fournissent un séquençement pour la production. Comme nous l'avons dit précédemment, ces modèles ne sont pas à proprement parlé dédiés à la planification tactique, c'est pour cela que nous nous bornons à les décrire sans rentrer dans les méthodes utilisées pour les résoudre.

Le Discrete Lot sizing and Scheduling Problem (DLSP)

Outre le fait que ce modèle soit un « small bucket », le DLSP fait l'hypothèse du « tout ou rien », c'est-à-dire que dès qu'on décide de produire durant une micro-période, alors on le fait à capacité maximale. En reprenant le modèle de (Wagner et Whitin 1958), on introduit les notations complémentaires suivantes :

$$\begin{aligned} p_i & : \text{Capacité nécessaire pour produire une unité de produit } i, & i \in \llbracket 1, N \rrbracket \\ C_t & : \text{Capacité disponible à la période } t, & t \in \llbracket 1, T \rrbracket \end{aligned}$$

ainsi que de nouvelles variables de décision :

$$Y_{it} : \begin{cases} 1 & \text{si la machine est configurée pour le produit } i \text{ à la période } t, \\ 0 & \text{sinon.} \end{cases} \quad i \in \llbracket 1, N \rrbracket, t \in \llbracket 1, T \rrbracket$$

Le modèle du DLSP reprend la fonction objectif (1.1) du modèle de Wagner-Within ainsi que toutes ses contraintes à l'exception de la contrainte (1.3) qui est remplacée par :

$$p_i Q_{it} = C_t Y_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.6)$$

$$\sum_{i=1}^N Y_{it} \leq 1 \quad \forall t \in \llbracket 1, T \rrbracket \quad (1.7)$$

$$X_{it} \geq Y_{it} - Y_{i(t-1)} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.8)$$

$$Y_{it} \in \{0, 1\} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.9)$$

C'est la contrainte (1.6) qui exprime le « tout ou rien ».

La contrainte (1.7) est la contrainte qui fait de ce modèle un modèle « small bucket » : en effet, on ne peut fabriquer plus d'un type de produits par période.

La contrainte (1.8) définit les périodes où on aura des coûts de lancement. Si la production est déjà lancée à la période précédente, alors il n'est pas nécessaire de la préparer. On voit donc qu'en

optimisant ce problème, le solveur va tenter de *séquence* la production de façon à avoir le moins possible de lancement.

La contrainte (1.9) définit les variables de configuration comme étant binaires.

D'après (Bruggemann et Jahnke 2000), la complexité de ce modèle a été très difficile à établir. Cependant, ces derniers ont prouvé qu'il est NP-difficile.

Le Continuous Setup Lot sizing Problem (CSLP)

Dans le modèle du DLSP, la contrainte du « tout ou rien » (1.6) est extrêmement stricte. Elle peut même avoir comme conséquence d'engendrer du stock pour un produit à la fin de l'horizon de planification. D'après (Drexel et Kimms 1997), celle-ci est plus motivée par les propriétés qu'elle engendre mathématiquement que par sa réalité industrielle. L'idée du modèle du CSLP est donc de relaxer cette contrainte. Le modèle du CSLP reprend donc l'intégralité des contraintes du DLSP. Cependant, la contrainte (1.6) devient

$$p_i Q_{it} \leq C_t Y_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.10)$$

C'est la contrainte (1.10) qui impose que la machine soit configurée pour le produit i à la période t si on a une fabrication de celui-ci durant cette période.

Le fait d'avoir relaxé la contrainte du « tout ou rien » (1.6) engendre une très grande différence entre le DLSP et le CSLP. En effet, dans la contrainte (1.10), on peut très bien avoir $Y_{it} = 1$ sans pour autant qu'il y ait production durant cette période. Plus précisément, supposons qu'à la période t , on fabrique un produit i : alors $Y_{it} = 1$. Supposons qu'à la période $t + 1$, on décide de ne rien produire : contrairement au DLSP qui impose une production si la machine est configurée à la période $t + 1$, on peut laisser la machine configurée pour cette période ($Y_{i(t+1)} = 1$). Si on décide à la période $t + 2$ de reproduire des articles i , alors, comme $Y_{i(t+1)} = 1$, nous n'aurons pas de lancement pour cette période, d'où la dénomination de ce modèle.

(Florian et al. 1980) ont prouvé que ce problème était NP-difficile.

Le Proportional Lot sizing and Scheduling Problem (PLSP)

Tout comme le DLSP, le CSLP ne permet de produire qu'un seul type de produit par période. Or, comme il n'est pas soumis à la loi du « tout ou rien », le CSLP permet de ne pas produire à pleine capacité : la capacité restante est donc inutilisée. Le modèle du PLSP remédie à cela : l'idée sous-jacente est d'utiliser la capacité restante pour fabriquer un autre type de produits, sous certaines conditions. Bien qu'on puisse produire plus d'un type de produits par période, on classe celui-ci parmi les modèles « small bucket » (car, à l'instar des modèles « small bucket » il ne permet qu'un seul lancement de production par période).

Si deux produits sont fabriqués durant la même période, il est important (pour les coûts de lancement) de savoir l'ordre dans lequel ils sont fabriqués. Pour cela on va redéfinir la signification des variables de décision Y_{it} de la façon suivante : pour ce modèle, Y_{it} vaudra 1 si la machine est configurée pour fabriquer le produit i à la fin de la période t . Ainsi, la conséquence fondamentale de cette redéfinition est qu'on pourra reconfigurer une fois la machine au cours de la période $t + 1$ et la production de l'article i pourra se faire à la période $t + 1$ si la machine est configurée correctement soit à la fin de la période précédente (donc au début de la période $t + 1$), soit à la fin de la période $t + 1$: cela nous autorise donc bien à fabriquer deux types d'articles par période.

Le modèle du PLSP est identique à celui du CSLP à l'exception de la contrainte (1.10) qui est remplacée par :

$$p_i Q_{it} \leq C_t (Y_{i(t-1)} + Y_{it}) \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.11)$$

$$\sum_{i=1}^N p_i Q_{it} \leq C_t \quad \forall t \in \llbracket 1, T \rrbracket \quad (1.12)$$

(Haase 1993) a montré que ce problème est NP-difficile.

Le modèle « big Bucket »

Dans la littérature, on retrouve ce modèle sous le nom de « Capacitated Lot Sizing Problem » (CLSP). Il reprend l'ensemble des contraintes du modèle de Wagner-Within en y adjoignant des contraintes de capacités. Plus précisément, on introduit les notations supplémentaires suivantes :

- C_t : Capacité disponible à la période t , $t \in \llbracket 1, T \rrbracket$
 p_i : Capacité nécessaire pour produire une unité de produit i , $i \in \llbracket 1, N \rrbracket$

Et la contrainte (1.3) est remplacée par :

$$p_i Q_{it} \leq C_t X_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.13)$$

$$\sum_{i=1}^N p_i Q_{it} \leq C_t \quad \forall t \in \llbracket 1, T \rrbracket \quad (1.14)$$

La contrainte (1.13) impose qu'un lancement pour le produit i soit effectué à la période t si on a une fabrication de celui-ci durant cette période.

La contrainte (1.14) est une contrainte de capacité.

D'après (Drexel et Kimms 1997), ce modèle est dédié à la problématique de l'élaboration de PDP et du PIC. Ils le considèrent comme étant le modèle de référence pour ces problématiques. (Bitran et Yanasse 1982) ont prouvé que sa résolution est NP-difficile. Un état de l'art des méthodes de résolution pour ce problème sera présenté dans le chapitre III de cette thèse.

Un modèle hybride : le General Lot sizing and Scheduling Problem (GLSP)

Le modèle du « General Lot Sizing and Scheduling Problem » a été conçu pour faire la synthèse entre les modèles « small » et « big bucket ». En effet, il a été développé en exprimant les coûts de stockage et la réponse à la demande sur des macro-périodes tout en permettant un séquençement de la production sur des micro-périodes. La création des micro-périodes est dynamique contrairement aux modèles des « small bucket » où ces dernières sont fixées. Plus précisément, les micro-périodes (dont le numéro sera appelé « position » dans ce modèle) correspondent au temps nécessaire pour fabriquer les lots de produits déterminés : leur durée est donc variable dans le GLSP. L'idée est donc de fixer un nombre maximal de type de produits (ce qui détermine le nombre de micro-périodes) à fabriquer durant une macro-période et de répartir la capacité disponible durant cette macro-période sur les micro-périodes où il y aura production (ce qui déterminera la taille, éventuellement nulle, de celles-ci). Cependant, planifier le séquençement de la fabrication des produits sur un horizon tactique n'a pas vraiment de sens industriellement. Ceci explique peut-être que ce modèle n'est pas utilisé en pratique pour les problématiques tactiques et qu'il n'est que très peu étudié dans la littérature : à notre connaissance, il n'existe qu'un seul article qui lui est consacré (Fleischmann et Meyr 1997).

Notations :

Les données du modèle sont :

T	: Longueur (en périodes) de l'horizon de la planification.	
N	: Nombre de produits.	
N_t	: Nombre maximal de lots durant la période t .	
F_t	: Première position de la période t : $F_t = 1 + \sum_{\tau=1}^{t-1} N_\tau$.	
L_t	: Dernière position de la période t : $L_t = F_t + N_t - 1$.	
Nb	: Nombre total de position : $\sum_{t=1}^T N_t$.	
C_t	: Capacité disponible à la période t	$t \in \llbracket 1, T \rrbracket$
d_{it}	: Demande en produit i à la fin de la période t	$i \in \llbracket 1, N \rrbracket, t \in \llbracket 1, T \rrbracket$
h_i	: Coût de stockage du produit i durant une période	$i \in \llbracket 1, N \rrbracket$
I_{i0}	: Stock initial en produit i	$i \in \llbracket 1, N \rrbracket$
p_i	: Capacité nécessaire pour produire une unité de produit i	$i \in \llbracket 1, N \rrbracket$
s_i	: Coût de lancement du produit i	$i \in \llbracket 1, N \rrbracket$
Y_{i0}	: $\begin{cases} 1 & \text{si la machine est initialement configurée pour le produit } i, \\ 0 & \text{sinon.} \end{cases}$	$i \in \llbracket 1, N \rrbracket$

Les variables de décisions du modèle sont :

Q_{in}	: Quantité de produit i fabriquée à la position n	$i \in \llbracket 1, N \rrbracket, n \in \llbracket 1, Nb \rrbracket$
I_{it}	: Niveau de stock en produit i à la période t	$i \in \llbracket 1, N \rrbracket, t \in \llbracket 1, T \rrbracket$
X_{in}	: $\begin{cases} 1 & \text{s'il y a lancement pour le produit } i \text{ à la position } n, \\ 0 & \text{sinon.} \end{cases}$	$i \in \llbracket 1, N \rrbracket, n \in \llbracket 1, Nb \rrbracket$
Y_{in}	: $\begin{cases} 1 & \text{si la machine est configurée pour } i \text{ à la position } n, \\ 0 & \text{sinon.} \end{cases}$	$i \in \llbracket 1, N \rrbracket, n \in \llbracket 1, Nb \rrbracket$

En utilisant ces notations, on peut écrire le problème du GLSP sous forme d'un programme linéaire mixte :

$$\text{Minimiser } \sum_{i=1}^N \left(\sum_{n=1}^{Nb} s_i X_{in} \right) + \sum_{i=1}^N \left(\sum_{t=1}^T h_i I_{it} \right) \quad (1.15)$$

Sous les contraintes :

$$I_{it} = I_{i(t-1)} + \sum_{n=F_t}^{L_t} Q_{in} - d_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.16)$$

$$p_i Q_{in} \leq C_t Y_{in} \quad \forall (i, n, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, Nb \rrbracket \times \llbracket 1, T \rrbracket \quad (1.17)$$

$$\sum_{i=1}^N \sum_{n=F_t}^{L_t} p_i Q_{in} \leq C_t \quad \forall t \in \llbracket 1, T \rrbracket \quad (1.18)$$

$$\sum_{i=1}^N Y_{in} \leq 1 \quad \forall n \in \llbracket 1, Nb \rrbracket \quad (1.19)$$

$$X_{in} \geq Y_{in} - Y_{i(n-1)} \quad \forall (i, n) \in \llbracket 1, N \rrbracket \times \llbracket 1, Nb \rrbracket \quad (1.20)$$

$$Y_{in} \in \{0, 1\} \quad \forall (i, n) \in \llbracket 1, N \rrbracket \times \llbracket 1, Nb \rrbracket \quad (1.21)$$

$$I_{it} \geq 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.22)$$

$$Q_{in}, X_{in} \geq 0 \quad \forall (i, n) \in \llbracket 1, N \rrbracket \times \llbracket 1, Nb \rrbracket \quad (1.23)$$

L'objectif (1.15) est de minimiser la somme des coûts de lancement et de stockage de tous les produits pour toutes les périodes.

La contrainte (1.16) représente l'équilibre des stocks.

La contrainte (1.17) impose que la machine soit correctement configurée pour produire le produit i à la position n .

La contrainte (1.18) est la contrainte de capacité : la somme des capacités requises pour produire les articles dans les positions de la période t ne peut pas excéder la capacité globale de la période t .

La contrainte (1.19) impose qu'on ne puisse pas produire plus d'un type d'articles à la position n .

La contrainte (1.20) définit s'il y a un lancement à la position n .

La contrainte (1.21) définit les variables de configuration comme étant binaires et (1.22), (1.23) sont des conditions de non-négativité.

1.2 Les modèles multi-niveau

Le problème de planification de la production à plusieurs niveaux considère les produits fabriqués à partir de plusieurs autres composants. Plus précisément, les produits finis requièrent des composants pour lesquels il faut aussi déterminer les plans de production correspondants. Ces composants peuvent aussi avoir des sous-composants, etc. Cela signifie donc qu'on prend en compte les différents niveaux de nomenclature d'où le vocable « multi-niveau ». Dans ces modèles, on retrouve donc les deux types de demandes (externe et interne) et la demande interne est déduite de la demande externe grâce aux liens de nomenclature.

Généralement, la nomenclature est représenté par un graphe acyclique où les sommets représentent les produits (ou composants) et où les arcs représentent les liens entre les différents sommets. Pour chaque produit fini i , on peut lister l'ensemble des produits j intervenant directement dans sa fabrication et on définit ainsi le coefficient technique a_{ij} de la façon suivante : a_{ij} est la quantité de produit i directement utilisée pour fabriquer une unité de produit j . Le graphe valué ainsi obtenu est appelé « graphe gozinto ». Le caractère acyclique du graphe permet de le partitionner en niveaux. Par convention et en cohérence avec la définition d'une nomenclature produit, les produits de niveau 0 sont les produits finis. Ils correspondent uniquement à une demande externe. Si un produit i est de niveau n , l'ensemble des produits directement impliqués dans sa fabrication seront de niveau $n + 1$. On code ce graphe grâce à une matrice carrée appelée « Matrice gozinto » dont l'ordre est égal au nombre de produits dont on doit planifier la fabrication et où chaque élément a_{ij} représente le nombre de produits j entrant directement dans la fabrication du produit i . Le graphe étant acyclique, la matrice peut être représentée sous une forme triangulaire inférieure dont la diagonale est nulle.

Dans la littérature, on retrouve quatre types de structure (graphes gozinto) représentant les différentes nomenclatures possibles :

- la structure en série : chaque produit a au plus un prédécesseur et au plus un successeur,
- la structure d'assemblage : chaque produit a au plus un successeur,
- la structure arborescente : chaque produit a au plus un prédécesseur,
- la structure générale où il n'y pas de limitation sur l'ensemble des prédécesseurs et successeurs.

La figure 1.15 illustre ces différents graphes.

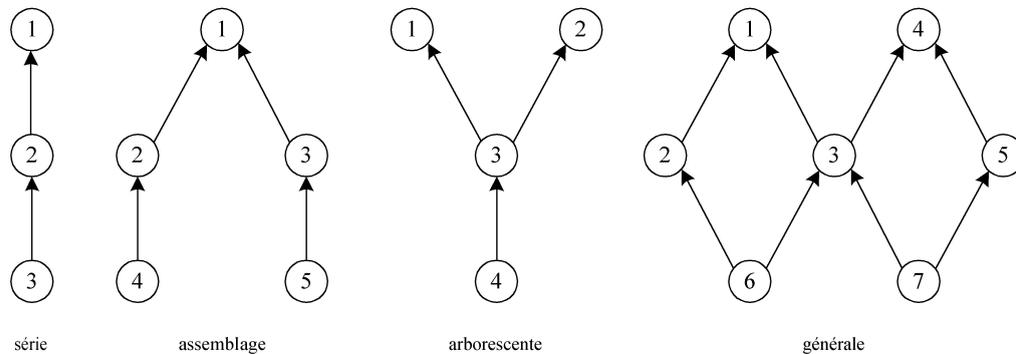


Figure 1.15 – Les différentes structures

Cependant, les composants fabriqués à une certaine période ne sont pas forcément disponibles directement pour la production du produit fini (encours, temps de transport, opérations de contrôle non modélisées,...). Considérer la planification de composants et de produits finis sous-entend donc la nécessité de modéliser le temps d'obtention de ces composants (lead time). En effet, si les contraintes de capacité modélisent sur une période la possibilité d'effectuer les composants nécessaires, le « lead time » modélise le temps de parcours et d'encours entre la production et la disponibilité réelle.

Nous présentons maintenant les principaux modèles multi-niveau rencontrés dans la littérature en suivant les critères définis précédemment.

1.2.1 Capacité infinie

Ici, nous supposons que les capacités de production sont assez grandes pour qu'on puisse les considérer comme infinies. Dans la littérature, nous n'avons trouvé qu'un seul modèle de référence : c'est un modèle « big bucket » dans lequel la demande est variable. Le cas de la demande constante peut se ramener aisément au cas mono-niveau avec demande constante et capacité infinie (la demande externe constante engendrant la demande interne constante) et ne présente pas donc pas d'intérêt en soi.

Dans la littérature, on désigne le modèle « big bucket » à demande variable et capacité infinie par « Multi-Level Lot-sizing Problem » (MLLP). Nous reprenons le modèle donné par (Clark et Armentano 1995)

Notations :

Les données du MLLP sont :

- T : Longueur (en périodes) de l'horizon de la planification.
 N : Nombre de produits.
 K : Nombre de ressources utilisées pour la production.
 S_i : Ensemble des composants succédant directement à i , $i \in \llbracket 1, N \rrbracket$
 a_{ij} : Nombre de composants i utilisés pour fabriquer j , $(i, j) \in \llbracket 1, N \rrbracket \times S_i$
 d_{it} : Demande en produit i à la fin de la période t , $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
 c_{it} : Coût de production pour un produit i à la période t , $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
 s_{it} : Coût de lancement du produit i durant la période t , $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
 h_{it} : Coût de stockage du produit i durant la période t , $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
 L_i : Lead time pour le produit i , $i \in \llbracket 1, N \rrbracket$
 $T_i + 1$: période à partir de laquelle, sur un horizon glissant, on planifie la fabrication de i ,
 $i \in \llbracket 1, N \rrbracket$
 M_{it} : une borne supérieure pour Q_{it} , $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$

Les variables de décisions du modèle sont :

- Q_{it} : Quantité de produit i fabriquée à la période t $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
 I_{it} : Niveau de stock en produit i à la période t $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
 Y_{it} : $\begin{cases} 1 & \text{si on produit } i \text{ à la période } t, \\ 0 & \text{sinon.} \end{cases}$ $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$

$$\text{Minimiser } \sum_{i=1}^N \sum_{t=T(i)+1}^{T(i)+T} (s_{it}Y_{it} + c_{it}Q_{it} + h_{i(t+L(i))}I_{i(t+L(i))}) \quad (1.24)$$

Sous les contraintes :

$$I_{i(L(i)+t)} = I_{i(L(i)+t-1)} + Q_{it} - d_{i(L(i)+t)} - \sum_{j \in S(i)} a_{ij}Q_{j(L(i)+t)} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket T(i) + 1, T(i) + T \rrbracket \quad (1.25)$$

$$Q_{it} \leq M_{it}Y_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.26)$$

$$Y_{it} \in \{0, 1\} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.27)$$

$$Q_{it}, I_{i(L(i)+t)} \geq 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket T(i) + 1, T(i) + T \rrbracket \quad (1.28)$$

L'objectif (1.24) est de minimiser la somme des coûts de lancement, de production et de stockage de tous les produits pour toutes les périodes. On remarque que la production de chaque composant i se fait sur un horizon glissant.

La contrainte (1.25) représente l'équilibre des stocks.

La contrainte (1.26) impose qu'un lancement pour le produit i soit effectué à la période t si on a une fabrication de celui-ci durant cette période.

La contrainte (1.27) définit les variables de lancement comme étant binaires et (1.28) est une condition de non-négativité.

Contrairement au modèle mono-niveau à demande variable et à capacité infinie (Wagner et Whitin 1958), ce modèle est NP-difficile (Arkin et al. 1989).

1.2.2 Capacité finie

Dans le cas à capacité finie, nous retrouvons les deux modélisations possibles de la demande. Conformément à notre classification, nous commençons par le modèle à demande constante.

1.2.3 Demande constante

A l'instar de l'approche mono-niveau, nous allons détailler le cas où la demande en produits finis est constante : ce problème porte le nom de « Cyclic Economic Lotsizing and Scheduling Problem » (CELSP).

(Bourland et Yano 1997) ont prouvé que ce problème était NP-difficile. Mais il a été montré par (Carlier et Chrétienne 1988) que sur un horizon infini, ce problème admet une solution K -périodique. Le but est donc de rechercher un cycle de production pour chaque produit (fini ou composant) i qui minimise la somme des coûts moyens de stockage et de lancement de chaque produit (fini ou composant) i sur ces cycles. En raison de la complexité de la recherche d'une telle solution pour des valeurs quelconques de l'entier K , de nombreux auteurs ont limité la recherche aux solutions 1-périodiques, ce qui est le cas, en particulier, dans l'approche par période commune (Hennet 2001) qui permet la détermination conjointe des tailles de lots et de l'ordonnancement de leur fabrication.

Les systèmes de production considérés sont décrits grâce aux paramètres suivants :

- N : Nombre de produits finis.
- M : Nombre total de produits.
- A : Matrice carrée triangulaire inférieure « gozinto » $M \times M$
représentant les liens entre les produits et leurs composants.
- d_i : Taux de demande (externe) pour le produit i , si $i > N$ alors $d_i = 0$ $i \in \llbracket 1, M \rrbracket$
- η_i : Coût unitaire de stockage du produit i $i \in \llbracket 1, M \rrbracket$
- p_i : Taux de production du produit i $i \in \llbracket 1, M \rrbracket$
- f_i : Coût de lancement de production pour le produit i $i \in \llbracket 1, M \rrbracket$

On suppose également qu'à chaque produit $i \in \llbracket 1, M \rrbracket$ est associée une ressource de production $r = R(i)$, la même ressource r pouvant être affectée à plusieurs produits mais ne pouvant en traiter qu'un seul à la fois.

Nous donnons la modélisation de ce problème proposée par (Hennet 2001) qui le formule comme un problème de Job-Shop de la façon suivante :

Il commence par définir la notion de job générique en développant le graphe « gozinto » en N sous-graphes disjoints ayant chacun une structure anti-arborescente (nomenclature convergente) et correspondant chacun à la nomenclature d'un et d'un seul produit fini. Ainsi, il associe à chaque sous-graphe (qu'il nomme job générique) un unique produit fini : ce job générique devra donc être « produit » pour un cycle de longueur T en quantité Td_i .

Pour chaque job générique i , il définit des « opérations génériques » qui correspondent à l'ensemble des chemins du sous-graphe associé au job i dont la valuation est calculée en fonction de la nomenclature du produit et de la longueur du cycle de production : par exemple, s'il existe un chemin de longueur 2, (j, l, i) du produit j au produit fini i , l'opération associée à ce chemin est la production de la quantité $a_{j,l}a_{l,i}Td_i$. Dans le graphe de précédence du job i , l'arc entre l'opération (j, l, i) et l'opération (l, i) définit une contrainte de précédence entre ces deux opérations. Soit p le nombre d'opérations génériques. Par définition, l'opération générique $s \in \llbracket 1, p \rrbracket$ appartient au job i si i est le sommet terminal du sous-graphe auquel appartient s .

Pour chaque produit fini i , (Hennet 2001) partitionne les opérations génériques appartenant au job i en deux ensembles :

- les opérations génériques terminales : $s \in \llbracket 1, p \rrbracket$ est une opération générique terminale si elle

correspond à la production du produit fini i en quantité $d_i T$. Sa durée opératoire sera donc $\tau_s T$ avec $\tau_s = \frac{d_i}{p_i}$,

- les opérations génériques non-terminales : dans ce cas, l'opération unique qui lui succède est notée $\nu(s)$. Si l'opération s est définie par le chemin (j, l, \dots, i) , alors $\nu(s)$ est définie par (l, \dots, i) . L'opération s consiste donc à fabriquer le produit $j = P(s)$ en quantité $q_s T$, avec $q_s = a_{jl} \dots a_{\nu_i} d_i$. La durée de l'opération générique $s \in \llbracket N + 1, p \rrbracket$ est $\tau_s T$ avec : $\tau_s = \frac{q_s}{p_j}$.

La formalisation du problème donné par (Hennet 2001) est par période commune. En choisissant arbitrairement un instant initial noté 0, l'auteur définit les variables du modèle :

T : la longueur de la période commune,

$T(s)$: la date de début de l'opération générique s , $T(s) \in [0, T]$ $s \in \llbracket 1, p \rrbracket$.

Les dates suivantes d'exécution des opérations semblables à s sont $T(s) + kT$ avec $k \in \mathbb{N}$ ce qui garantit le caractère périodique de l'ordonnancement.

Les décisions d'allocation des ressources aux activités de production sont supposées fixées a priori. La ressource associée à l'opération s (notée $r = \Gamma(s)$) produisant j (notée $j = P(s)$) est la ressource allouée à la production de produit j :

$$r = \Gamma(s) \iff (r = R(j) \text{ avec } j = P(s))$$

Les ressources étant supposées simples et unitaires, elles ne peuvent donc pas traiter deux produits en simultanée, donc les contraintes de ressources sont de type disjonctif. Elles sont formulées sur la période générique $[1, T]$ et portent sur toutes les paires d'opérations utilisant la même ressource :

$$T(s) + \tau_s T \leq T(t) \text{ ou } T(t) + \tau_t T \leq T(s) \quad \forall (s, t) \in \llbracket 1, p \rrbracket^2 \mid \Gamma(s) = \Gamma(t) \quad (1.29)$$

Les contraintes de capacité sur les ressources s'appliquent entre produits différents aussi bien qu'entre sous-lots du même produit.

En raison de l'objectif de minimisation du coût moyen total, les produits intermédiaires (composants) utilisés au cours de la période générique ont été produits soit à la période courante, soit à la période précédente : soit s l'opération périodique visant à fabriquer un type de composant pour un produit fini i , on notera $h(s)$ la différence entre l'indice de la période d'utilisation et celui de la période de production. $h(s)$ vaudra donc 0 si le composant a été utilisé durant la période où il a été produit, 1 si cela a eu lieu à la période suivante. Les contraintes de précédence entre opérations doivent donc être formulées sur deux périodes consécutives, par exemple sur l'intervalle $[-T, T]$. On obtient donc les contraintes suivantes :

$$h(s) = 0 \quad \forall s \in \llbracket 1, N \rrbracket \quad (1.30)$$

$$h(s) = 0 \text{ ou } 1 \quad \forall s \in \llbracket N + 1, p \rrbracket \quad (1.31)$$

$$h(\nu(s)) \leq h(s) \quad \forall s \in \llbracket N + 1, p \rrbracket \quad (1.32)$$

$$T(s) + \tau_s T - h(s)T \leq T(t) - h(t)T \quad \forall s \in \llbracket N + 1, p \rrbracket \mid t = \nu(s) \quad (1.33)$$

$$T(s) + \tau_s T \leq T \quad \forall s \in \llbracket 1, N \rrbracket \quad (1.34)$$

En raison de la définition des opérations génériques, le même produit j peut être fabriqué en plusieurs sous-lots au sein de la période générique. Pour garantir que la fabrication du produit j n'occasionne qu'un seul coût de lancement par période, il est nécessaire d'introduire des contraintes de non-interruption entre sous-lots du même produit :

$$\max_{s|P(s)=j} (T(s) + \tau_s T - h(s)T) - \min_{s|P(s)=j} (T(s) - h(s)T) = T \sum_{s|P(s)=j} \tau_s \quad (1.35)$$

L'objectif recherché est de trouver les valeurs de la période commune et des dates de lancement qui minimisent le coût moyen de production :

$$J = c_1 + c_2 + c_3 \quad (1.36)$$

où :

- c_1 représente le coût de lancement de fabrication par unité de temps : $c_1 = \frac{\sum_{i=1}^M f_i}{T}$,
- c_2 représente le coût moyen de stockage des produits finis par unité de temps : $c_2 = T k_2$ avec $k_2 = \sum_{i=1}^N \frac{\eta_i}{2} \left(1 - \frac{d_i}{p_i}\right) d_i$,
- c_3 représente le coût de stockage des produits primaires et intermédiaires par unité de temps : $c_3 = \sum_{s=N+1}^p \eta_j q_s \left[T(\nu(s)) - T(s) - h(\nu(s))T + h(s)T + q_s \left(\frac{T}{2p_j} - \frac{T}{2p_l} \right) \right]$ avec $j = P(s)$ et $l = P(\nu(s))$.

1.2.4 Demande variable

Dans cette partie, nous présentons les modèles à demande variable. A l'instar des modèles mono-niveau, nous retrouvons les modèles « small bucket » et « big bucket ». Bien évidemment, les remarques que nous avons formulées dans le cas mono-niveau restent valables dans le cas multi-niveau.

1.2.4.1 Le modèle « Small Bucket » : le MLPLSP

Dans la littérature, nous ne trouvons qu'un seul modèle « small bucket » multi-niveau (Drexler et Kimms 1997) : le modèle Multi Level Proportional Lot sizing and Scheduling Problem (MLPLSP) qui est une extension du PLSP prenant en compte les liens de nomenclature. Le nombre de produits fabriqués sur une micro-période est donc limité à deux produits.

Notations :

Les données du modèle sont :

T	: Longueur (en périodes) de l'horizon de la planification.	
N	: Nombre de produits.	
M	: Nombre de machines.	
\mathcal{J}_m	: Ensemble de produits fabriqués sur la machine m	$m \in \llbracket 1, M \rrbracket$
S_i	: Ensemble des composants succédant directement à i	$i \in \llbracket 1, N \rrbracket$
L_i	: Lead time exprimé en nombre de périodes pour le produit i	$i \in \llbracket 1, N \rrbracket$
m_i	: Machine qui fabrique le produit i	$i \in \llbracket 1, N \rrbracket$
h_i	: Coût de stockage du produit i durant une période	$i \in \llbracket 1, N \rrbracket$

s_i	: Coût de lancement du produit i durant une période	$i \in \llbracket 1, N \rrbracket$
p_i	: Capacité nécessaire pour produire i	$i \in \llbracket 1, N \rrbracket$
a_{ij}	: Nombre de composants i utilisés pour fabriquer j	$(i, j) \in \llbracket 1, N \rrbracket \times S_i$
C_{mt}	: Capacité de la machine m disponible à la période t	$(m, t) \in \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket$
d_{it}	: Demande externe en produit i à la fin de la période t	$(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
I_{i0}	: Stock initial en produit i	$i \in \llbracket 1, N \rrbracket$
Y_{i0}	: $\begin{cases} 1 & \text{si la machine est initialement configurée pour le produit } i, \\ 0 & \text{sinon.} \end{cases}$	$i \in \llbracket 1, N \rrbracket$

Les variables de décisions du modèle sont :

Q_{it}	: Quantité de produit i fabriquée à la période t	$i \in \llbracket 1, N \rrbracket, t \in \llbracket 1, T \rrbracket$
I_{it}	: Niveau de stock en produit i à la période t	$i \in \llbracket 1, N \rrbracket, t \in \llbracket 1, T \rrbracket$
X_{it}	: $\begin{cases} 1 & \text{s'il y a lancement pour le produit } i \text{ à la période } t, \\ 0 & \text{sinon.} \end{cases}$	$i \in \llbracket 1, N \rrbracket, t \in \llbracket 1, T \rrbracket$
Y_{it}	: $\begin{cases} 1 & \text{si la machine est configurée pour le produit } i \text{ à la fin de } t, \\ 0 & \text{sinon.} \end{cases}$	$i \in \llbracket 1, N \rrbracket, t \in \llbracket 1, T \rrbracket$

En utilisant ces notations, on peut écrire le problème du MLPLSP sous forme d'un programme linéaire mixte (Drexel et Kimms 1997) :

$$\text{Minimiser } \sum_{j=1}^N \sum_{t=1}^T (s_j X_{jt} + h_j I_{jt}) \quad (1.37)$$

Sous les contraintes :

$$I_{it} = I_{i,t-1} + Q_{i,t} - d_{it} - \sum_{j \in S_i} a_{ij} Q_{jt} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.38)$$

$$I_{it} \geq \sum_{j \in S_i} \sum_{\tau=t+1}^{\min\{t+L_i, T\}} a_{ij} Q_{j\tau} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 0, T-1 \rrbracket \quad (1.39)$$

$$p_i Q_{it} \leq C_{m_j, t} (Y_{i(t-1)} + Y_{it}) \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.40)$$

$$\sum_{i \in \mathcal{J}_m} p_i Q_{it} \leq C_{mt} \quad \forall (m, t) \in \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket \quad (1.41)$$

$$\sum_{i \in \mathcal{J}_m} Y_{it} \leq 1 \quad \forall (m, t) \in \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket \quad (1.42)$$

$$X_{it} \geq Y_{it} - Y_{i(t-1)} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.43)$$

$$Y_{it} \in \{0, 1\} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.44)$$

$$Q_{it}, I_{it}, X_{it} \geq 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (1.45)$$

L'objectif (1.37) est de minimiser la somme des coûts de lancement et de stockage de tous les produits pour toutes les périodes.

La contrainte (1.38) représente l'équilibre des stocks.

La contrainte (1.39) est essentielle puisque c'est elle qui introduit la notion de lead time dans le modèle. Elle impose que le stock en produit i couvre les besoins en demande interne de celui-ci durant toutes les périodes consécutives constituant le lead time.

C'est la contrainte (1.40) qui impose que la machine sur laquelle est fabriqué le produit i soit configurée pour ce produit durant la période t si on a une fabrication de celui-ci durant cette période.

La contrainte (1.41) est la contrainte de capacité.

La contrainte (1.42) impose qu'il n'y ait pas plus d'une reconfiguration de la machine m pendant la période t .

La contrainte (1.43) définit lorsqu'il y a des lancements (si il y a eu reconfiguration de la machine).

La contrainte (1.44) définit les variables de configuration comme étant binaires et (1.45) est une condition de non-négativité.

Le MLPLSP est un problème NP-difficile (le PLSP, qui est NP-difficile, est un MLPLSP à un niveau). Il y a peu de littérature consacrée au MLPLSP. Cependant Kimms (Kimms 1999) propose une méthode de résolution basée sur un algorithme génétique. A l'instar des modèles « small bucket » mono-niveau et pour les mêmes raisons, le MLPSP n'est pas considéré comme un modèle dédié à la planification tactique.

1.2.4.2 Le modèle « Big Bucket »

Dans la littérature, ce modèle porte le nom de Multi-Level Capacitated Lot-Sizing Problem (ML-CLSP). Pour (Drexl et Kimms 1997), ce modèle est dédié à la problématique de l'élaboration de PDP et du MRP. Ils le considèrent comme étant le modèle de référence pour ces problématiques.

Nous reprenons le modèle donné par (Clark et Armentano 1995) (paragraphe 1.2.1) pour le MLLP auquel nous ajoutons les notations suivantes :

Les données du modèle sont :

b_{kt} : Capacité de la ressource k disponible à la période t , $(k, t) \in \llbracket 1, K \rrbracket \times \llbracket 1, T \rrbracket$

f_{ikt} : capacité fixe de ressource k utilisée pour configurer celle-ci pour produire i à la période t ,
 $(i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, K \rrbracket \times \llbracket 1, T \rrbracket$

v_{ikt} : Quantité unitaire en ressource k utilisée pour produire i à la période t ,
 $(i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, K \rrbracket \times \llbracket 1, T \rrbracket$

Le modèle du MLCLSP reprend les contraintes du MLLP en rajoutant la contrainte :

$$\sum_{i=1}^N (f_{ikt}Y_{it} + v_{ikt}Q_{it}) \leq b_{kt} \quad \forall (k, t) \in \llbracket 1, K \rrbracket \times \llbracket 1, T \rrbracket \quad (1.46)$$

La contrainte (1.46) est une contrainte de capacité.

Il est à noter que les contraintes de capacité sont exprimées en fonction de plusieurs ressources (au contraire du CLSP) traduisant le fait que les composants et les produits finis peuvent être fabriqués sur différentes ressources. Par contre, sa complexité est également NP-difficile.

2 Les modèles multi-site

Initialement, les modèles multi-site ont consisté à planifier la production sur les différentes usines composant le système de production en prenant en compte les transports entre les différents sites sur un horizon moyen terme. Avec l'avènement de la chaîne logistique, cette problématique a intégré en sus la conception des plans d'approvisionnement et de distribution pour chaque entité de la chaîne (figure 1.7). Ainsi un modèle multi-site pour la chaîne logistique doit répondre aux problématiques suivantes :

- planifier l'activité d'approvisionnement consiste d'une part, à sélectionner les fournisseurs chez lesquels on s'approvisionne et, d'autre part, à déterminer les quantités de produits qui leur seront achetées,

- la planification de la production va s'attacher à établir les quantités de produits à fabriquer par chaque usine composant la chaîne logistique et à déterminer les quantités de produits devant être acheminées d'une entité de production à une autre,
- planifier la distribution consiste à déterminer les quantités de produits finis à livrer à chaque centre de distribution.

Ainsi les modèles multi-site doivent planifier les quantités à fabriquer par usine (en tenant compte, éventuellement, des nomenclatures produits) et les quantités à transporter entre chaque entité de la chaîne logistique (fournisseur, usine, entrepôts). Ils peuvent se formaliser ainsi :

Objectif :

Déterminer sur un horizon moyen terme (6 à 18 mois) des quantités de produits ou de famille de produits à fabriquer et à transporter de telle façon à optimiser une fonction objectif. Cette dernière selon les cas d'étude exprime un compromis entre les coûts de production, de lancement, de stockage et de transport.

Données :

Les données d'entrée du modèle sont :

- la durée de l'horizon de planification,
- la demande client par produit ou par famille de produits.

Paramètres :

Les paramètres pris en compte sont le plus souvent :

- le coût unitaire de stockage,
- le coût unitaire de production sur l'usine k ,
- le coût unitaire de lancement d'une campagne de production du produit i sur l'usine k ,
- le coût unitaire de transport du produit i de l'entité k à l'entité l ,
- la capacité de production de l'usine k ,
- la capacité de production nécessaire pour la fabrication d'un produit i ,
- la capacité nécessaire pour effectuer un lancement de production pour le produit i ,
- la capacité de transport disponible entre l'entité k et l .

Variables :

Elles représentent :

- les quantités produites sur l'usine k
- les quantités en stocks dans l'usine k
- les débuts de campagne (afin d'imputer les coûts de lancement de campagne)
- les quantités transportées de l'usine k à l'usine l .

En comparant les paramètres et variables définis dans un modèle mono-site et un modèle multi-site, on s'aperçoit que ces derniers peuvent être vus comme une généralisation des modèles mono-site. C'est ce qu'illustre la figure 1.16

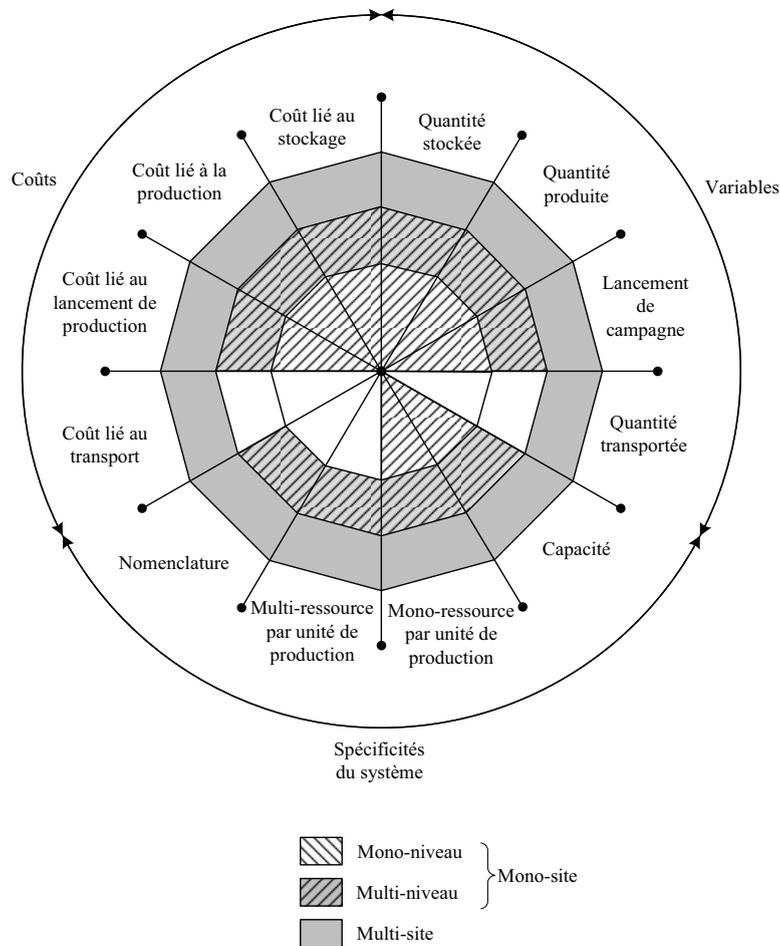


Figure 1.16 – Les modèles de planification tactique

Prendre en compte toutes ces activités dans un même modèle n'est pas chose aisée : dans la littérature, les modèles rencontrés se focalisent sur une voire deux de ces activités. Nous avons répertorié également les modèles de planification multi-site (au nombre de 5) pouvant être considérés comme générique et nous les avons situés par rapport aux activités précédemment décrites. Ces derniers sont des modèles multi-niveau considérant plusieurs sites de production mais se distinguent sur les contraintes d'équilibre des stocks et les contraintes de distribution :

- Le modèle de (Thierry 2003) [1] s'appuie sur une contrainte générique d'équilibre des stocks prenant en compte les quantités produites, les quantités transportées entre deux sites de production et les quantités transportées au client (les lead time de transport sont modélisés). Cependant, les stocks comme les transports sont supposés avoir une capacité infinie. De plus, les stocks de sécurité ne sont pas considérés et la demande de chaque client doit être obligatoirement honorée (il n'y a pas de backorder, ni d'annulation de commande). Chaque usine est mono-ressource et sa capacité de production est finie, l'emploi de capacité supplémentaire et la perte de capacité due à la reconfiguration de la ressource de production ne sont pas envisagées. De plus, les lead time de production ne sont pas pris en compte.
- Le modèle de (Mc Donald et Karimi 1997) [2] est assez proche du précédent. Outre le fait que l'annulation de commande et que les stocks de sécurité sont pris en compte et que les usines considérées sont multi-ressource, la vraie différence réside dans la détermination des quantités transportées entre les sites de production : n'est transportée entre deux sites de production à

la période t que la quantité exacte de produit dont l'usine destination aura besoin durant cette même période. De prime abord, cette hypothèse peut paraître très restrictive. Mais, ce parti pris est justifié par le fait que les coûts de transport sont constants sur l'horizon et ne dépendent que de la quantité transportée, conjugués au fait que les délais d'acheminement ne sont pas pris en compte et que les capacités de transport sont infinies.

- Le modèle de (Timpe et Kallrath 2000) [3] est un modèle issu d'un cas d'étude portant sur une industrie chimique. Ce modèle prend en compte l'activité de production et de distribution. Il prend également en compte l'approvisionnement mais un seul fournisseur est considéré. Les stocks sont à capacité finie et les stocks de sécurité sont modélisés. En ce qui concerne les transports, ils sont à capacité finie et les temps de transport sont modélisés. Cependant, la façon dont sont modélisées les capacités de transport est dédiée au cas d'étude sur lequel s'appuie le modèle : en effet, les transports s'effectuent par containers et donc la capacité est exprimée en termes de nombre de containers disponibles et nous avons donc une capacité disponible par container.
- Le modèle de (Pibernic 2007) [4] est présenté comme étant un modèle générique pour la planification tactique multi-site. Cependant, la chaîne logistique est modélisée par des étages de production, utilisant une configuration de type « Flow shop hybride ». Les centres de distribution se trouvent sur le dernier étage. Les transports considérés ne relient que les usines se trouvant sur des étages consécutifs et sont à capacité infinie. Les stocks de sécurité, les backorders, l'annulation de commande ne sont pas pris en compte. A l'instar du modèle de (Thierry 2003), chaque usine est mono-ressource et sa capacité de production est finie. Il n'y a pas de prise en compte de capacité supplémentaire ni de perte de capacité due à la reconfiguration de la ressource de production. De la structure même de la chaîne logistique considérée, on voit bien que ce modèle ne pourra pas s'adapter à tout type de chaîne logistique.
- Le modèle de (Lin et Chen 2007) [5] provient d'un cas d'étude issu d'une entreprise d'électronique. Ce modèle est très proche de celui de (Mc Donald et Karimi 1997), à ceci près que la structure de la chaîne logistique est aussi de type « Flow shop hybride » et qu'il considère des fournisseurs auprès desquels sont achetées les matières premières. De plus, la nomenclature de fabrication modélisée est de type série et chaque usine est modélisée par une unique ressource. Aucun centre de distribution n'est pris en compte.

En terme de résolution, (Thierry 2003) utilise la programmation par contrainte sur des instances de taille industrielle contenant 50 unités de production, 5 périodes de temps et 50 produits finis (avec une nomenclature sur 10 niveaux). Cependant, aucune comparaison avec d'autres méthodes n'est donnée. Pour (Mc Donald et Karimi 1997), deux instances sont résolues grâce à l'utilisation de solveurs (OSL et Cplex) : la première contient 2 usines, 4 périodes et 14 produits et la seconde 1 usine, 12 périodes et 34 produits. (Timpe et Kallrath 2000) utilisent pour la résolution de son modèle le solveur Xpress. La taille des instances est de 4 usines, 12 périodes et 200 produits finis. Dans cette étude, l'optimisation se borne à trouver une solution réalisable. (Lin et Chen 2007) utilise quant à lui le solveur Lingo. La taille de ces instances est de 6 usines, 2 produits et 3 niveaux de nomenclature. Cependant, aucune information concernant les temps de résolution et la qualité des résultats n'est donnée. (Pibernic 2007) se focalise sur les problèmes de collaboration. Aucune approche de résolution n'est mentionnée.

Cet état de l'art permet de conclure qu'aucun modèle de la littérature ne peut être considéré comme un modèle de référence pour la problématique multi site au contraire des modèles CLSP et MLCLSP pour les problématiques PDP et MRP. En ce sens, nous rejoignons (Drexl et Kimms 1997) qui n'en mentionnent aucun. Concernant la résolution de ces problèmes, l'absence d'un modèle étudié par plusieurs auteurs ne nous permet pas d'en déduire des conclusions pertinentes. Néanmoins, il est possible de relever que la majorité des travaux se limitent à utiliser des solveurs et ne proposent pas de méthodes de résolution dédiées et permettant de traiter des instances de taille industriel.

Le tableau 1.1 récapitule les principales hypothèses prises en compte dans ces modèles.

Hypothèses	[1]	[2]	[3]	[4]	[5]
Approvisionnement			X		X
Production	X	X	X	X	X
Distribution	X	X	X	X	
Backorder					X
Annulation de commande		X			
Stock de sécurité		X	X		
Stock à Capacité finie			X		
Mono-Ressource	X		X	X	X
Multi-Ressource		X			
Ressource à capacité finie	X	X	X	X	X
Lead Time de production					X
Temps de reconfiguration					X
Nomenclature	Générale	Générale	Générale	Générale	Série
Capacité de transport	infinie	infinie	finie	infinie	infinie
Lead Time de transport	X		X		
Lead Time d'approvisionnement					X

Tableau 1.1 – Les hypothèses prises en compte par les modèles de la littérature

Conclusion

Dans ce chapitre, nous avons présenté les problématiques liées à la planification tactique et nous avons montré comment elles étaient prises en compte au sein de MRPII et du management de la chaîne logistique. Nous avons mis en avant les principales difficultés liées à la mise en œuvre des différents plans tactiques préconisés par ces méthodologies puis nous avons présenté les modèles mathématiques de « lot sizing » conçus pour élaborer divers plans de production.

Nous avons constaté que parmi ces derniers, il n'y avait pas de modèle de référence pour l'élaboration du master planning. Le chapitre II tentera de pallier ce manque par la proposition d'un modèle générique dont les principaux modèles de planification (mono et multi-site) pourront être déduits par instanciation. Nous avons également remarqué que parmi les modèles de « lot sizing », deux seulement avaient été développés pour élaborer des plans tactiques (PIC-PDP-MRP) : le CLSP et le MLCLSP. Ceux-ci feront l'objet respectivement des chapitres III et IV de ce mémoire.

Nous avons vu que l'élaboration de plans tactiques comportait des difficultés majeures autres que leur optimisation en termes de coûts : les problèmes liés à la synchronisation des capacités et des matières et aux aléas liés à la demande client. Le chapitre V proposera deux approches permettant d'assurer la faisabilité d'un plan tactique au niveau opérationnel et le sixième et dernier chapitre montrera comment au travers d'une politique de gestion de stock innovante testée sur un cas industriel puis mise en œuvre sur un cas réel, assurer un taux de satisfaction minimal moyen pour des clients qu'on aura désiré « privilégier ».

Chapitre 2

Proposition de modèles génériques pour la planification tactique

Sommaire

Introduction	63
I La méthodologie ASCI	63
1 Analyse et spécification	65
1.1 Analyse	65
1.2 Spécification	65
2 Conception et implantation	66
2.1 Conception et implantation du modèle d'action	66
2.2 Conception et implantation du modèle de résultat	67
3 Analyse des résultats de l'étude et prise de décisions	67
II Modèle de connaissance générique pour la planification multi-site	67
1 Le sous-système physique	67
1.1 Les usines	68
1.2 Les stocks	69
1.3 Les moyens de transport	69
1.4 Les fournisseurs et les centres de distribution	70
1.5 Les dépendances entre les différentes ressources composant le sous-système physique	70
2 Le sous-système logique	73
3 Le sous-système décisionnel	74
3.1 Les contraintes du système de production	74
3.2 Les coûts pris en compte	76
3.3 La règle de gestion : équilibre des stocks	77
3.4 Les caractéristiques générales de la chaîne logistique	77
4 Les dépendances entre les différents sous-systèmes	77
4.1 Les dépendances entre le sous-système physique et le sous-système logique	77
4.2 Les dépendances entre le sous-système physique et le sous-système décisionnel	80
5 Le modèle UML représentant les trois sous-systèmes et les liens les unissant	80
III Modèle mathématique générique pour la planification multi-site	82
1 Notations mathématiques des ressources « production »	83
2 Notations mathématiques pour les ressources « stock »	85
3 Modélisation des ressources « transport »	87
4 Modélisation des flux de produits	88
4.1 Les contraintes liées aux ressources « production »	88
4.2 Les contraintes liées aux ressources « stock »	89

4.3	Les contraintes liées aux ressources « transport »	93
4.4	Fonction objectif	94
IV	Deux exemples d’instanciation du modèle générique	95
1	Le MLCLSP	96
2	Un modèle de planification multi-site avec prise en compte des transports	100
	Conclusion	105

Introduction

Dans le chapitre I, nous avons montré qu'il n'existait pas de modèle de référence pour la planification multi-site. Le but de ce chapitre est de pallier cette carence.

A cet effet, nous avons considéré la chaîne logistique interne suivante (figure 2.1) :

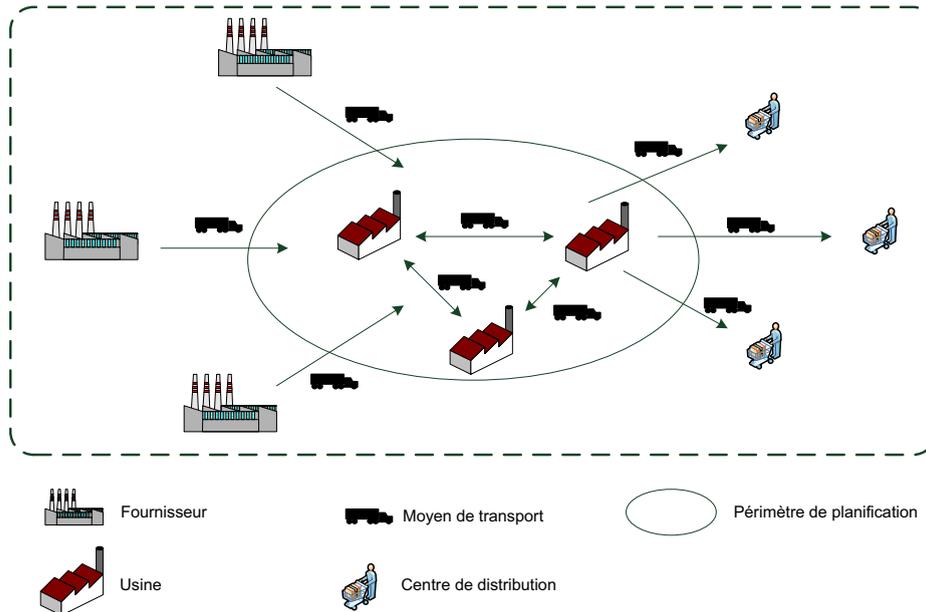


Figure 2.1 – la chaîne logistique considérée

Dans celle-ci apparaît trois types d'acteurs : les fournisseurs, les usines et les centres de distribution pouvant être reliés deux à deux par des moyens de transport. Nous nous intéressons donc à la planification de la production au sein de la chaîne logistique interne ce qui signifie que nous ne considérons pas les problèmes de transport entre les fournisseurs et les usines ainsi qu'entre les usines et les centres de distribution. Cependant, nous montrerons au travers d'un exemple qu'il sera possible de les prendre en compte. Par ailleurs, nous nous attachons à intégrer toutes les hypothèses de modélisation référencées dans le tableau 1.1 (stock de sécurité, stock à capacité finie, mono-ressource, multi-ressource, ressource à capacité finie, lien de nomenclature, capacité de transport, lead time de production, lead time de transport et d'approvisionnement) ainsi que les extensions classiques de la littérature (backorder, temps de reconfiguration, annulation de commande, stock initial).

Après avoir présenté rapidement la méthodologie de modélisation utilisée, nous décrivons le modèle de connaissance générique pour la planification tactique mono et multi-site que nous avons élaboré. Ensuite, nous donnons le modèle mathématique générique que nous obtenons par traduction du modèle de connaissance. Pour finir, nousinstancions ce dernier sur deux exemples correspondants à des modèles de la littérature.

I La méthodologie ASCI

Dans un premier temps, nous présentons succinctement la méthodologie de modélisation ASCI (Analyse, Spécification, Conception et Implémentation) définie par (Gourgand et Kellert 1991) et sur laquelle nous nous sommes appuyés pour élaborer notre modélisation générique pour la planification

tactique.

Celle-ci met en jeu deux processus parallèles. Le premier processus se focalise sur le domaine de l'étude, le second sur un système dédié du domaine considéré.

Les domaines d'étude peuvent être aussi divers que les systèmes industriels, les chaînes logistiques ou les systèmes hospitaliers, etc. Le système est donc par définition un système précis du domaine considéré. L'objectif de ce premier processus est de capitaliser la connaissance acquise en termes de modèle de connaissance et d'outils par l'étude successive de systèmes appartenant au domaine. Cette capitalisation permet alors d'aider à l'étude des prochains systèmes en offrant un modèle de connaissance et des outils.

Les deux processus sont mis en œuvre parallèlement au cours de l'étude d'un système. En fonction de la maturité du domaine étudié, le processus focalisé sur le domaine sera plus ou moins enrichi par le processus focalisé sur un système et « alimentera » ce dernier plus ou moins. Les deux processus sont composés des quatre phases : Analyse, Spécification, Conception et Implantation. La principale différence entre les deux processus réside dans les phases de conception et d'implantation. Pour le processus focalisé sur le domaine, ces deux phases doivent être considérées comme l'élaboration de composants génériques permettant la constitution d'une bibliothèque d'outils dédiés à l'étude du domaine.

Après avoir décrit globalement la méthodologie de modélisation ASCI, nous nous proposons de nous focaliser sur la description du processus dédié à l'étude d'un système.

La figure 2.2 illustre le processus de modélisation dédié au système dans sa dernière version (Norre 2005).

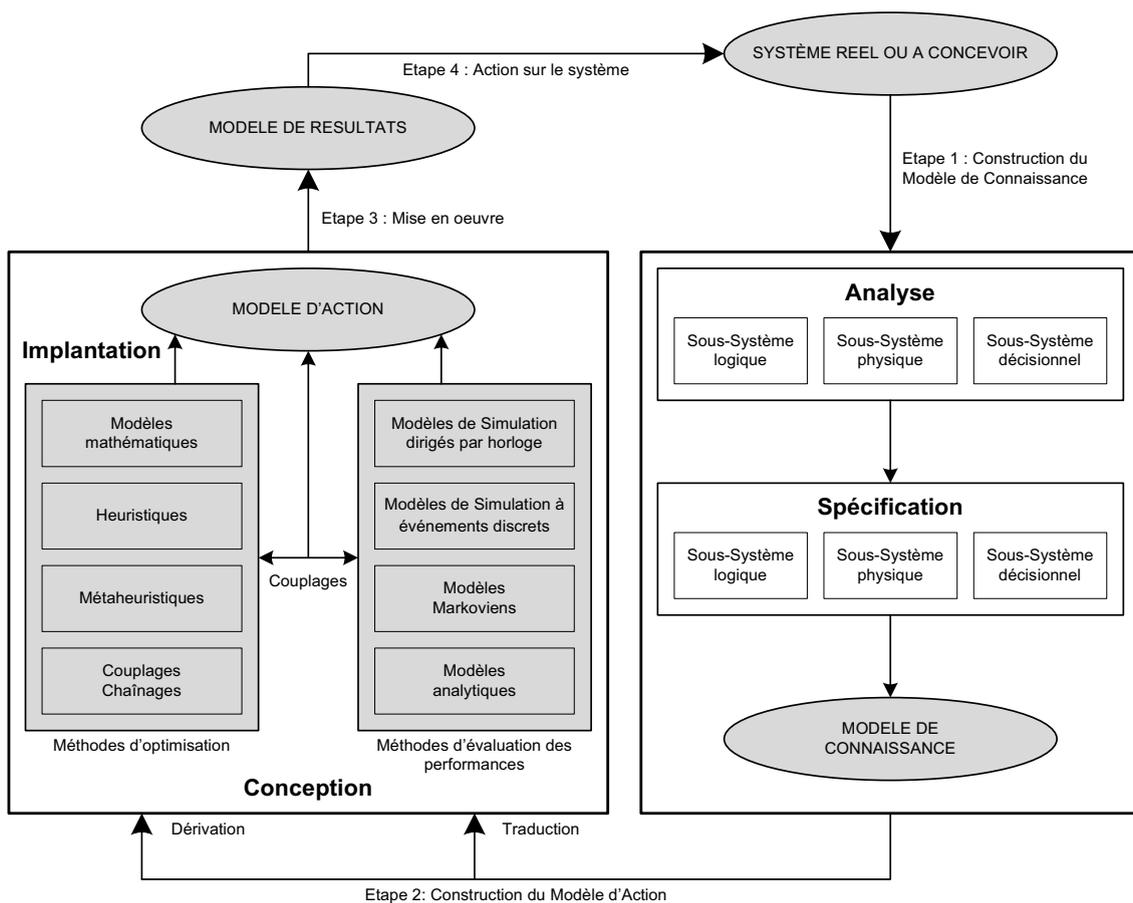


Figure 2.2 – Processus de modélisation ASCI

Le processus de modélisation est formé par l'exploitation du modèle de connaissance (modèle de fonctionnement du système) et du modèle d'action (traduction du modèle de connaissance dans un formalisme mathématique, dans un langage de programmation, dans un langage de simulation ...). Ce processus est itératif et est composé de quatre phases qui sont :

- l'élaboration du modèle de connaissance,
- l'exploitation du modèle d'action à partir du modèle de connaissance,
- l'exploitation du modèle d'action pour évaluer les performances,
- l'interprétation des résultats, correspondant à la déduction des actions à effectuer sur le système.

1 Analyse et spécification

1.1 Analyse

La première étape du processus de modélisation réside dans l'activité suivante « Comprendre le système et les acteurs ». Lors de cette étape, les acteurs du système et les experts en modélisation concentrent leurs efforts sur la compréhension du système à étudier, l'étude des informations sur le système à modéliser, la définition des objectifs de l'étude et les contraintes à respecter. Les besoins des décideurs comme des acteurs, les objectifs et les contraintes sont énoncés une première fois à ce niveau du processus. Une vérification de la formulation du problème peut être menée à l'aide de questionnaires pour déterminer si la formulation du problème correspond bien au problème étudié. Les résultats de cette étape du processus de modélisation sont la définition du système et la formulation du problème.

1.2 Spécification

Lors de cette étape, les entités et leurs interrelations jugées pertinentes, sont recueillies, structurées. Les données et les règles de fonctionnement sont également présentées. Le but de cette étape est d'aboutir à un modèle de connaissance qui est une formalisation dans un langage naturel ou graphique de la structure et du fonctionnement du système. Ce modèle doit contenir, entre autres, les informations pertinentes pour résoudre les problèmes posés. Il doit mettre en évidence de manière claire l'ensemble des problèmes induits par les systèmes étudiés. L'obtention du modèle de connaissance d'un système complexe est un processus qui doit être conduit avec beaucoup de rigueur. Il faut, d'une part, conduire une analyse structurelle et fonctionnelle du système et, d'autre part, spécifier les activités et donc les flux qui traversent ce système, et les règles de fonctionnement qui gèrent ces flux. Pour conduire cette analyse, une décomposition en trois sous-systèmes complémentaires et communicants est préconisée :

- Le sous-système physique comprend les entités physiques qui composent le système étudié. Il permet de décrire la topologie des unités de production, de stockage, les caractéristiques techniques des moyens de production ainsi que les liens physiques et logiques entre ces moyens.
- Le sous-système logique est composé des gammes et nomenclatures des pièces que peut fabriquer le système étudié ainsi que des règles opératoires. La gamme opératoire est constituée des opérations concernant les pièces en termes de moyen de production et de temps de traitement.
- Le sous-système décisionnel permet de spécifier l'ensemble des règles de fonctionnement du système. Il est formé d'un ensemble de règles de gestion qui agissent sur les sous-systèmes logique et physique.

Les interactions entre les trois sous-systèmes sont présentées sur la figure 2.3.

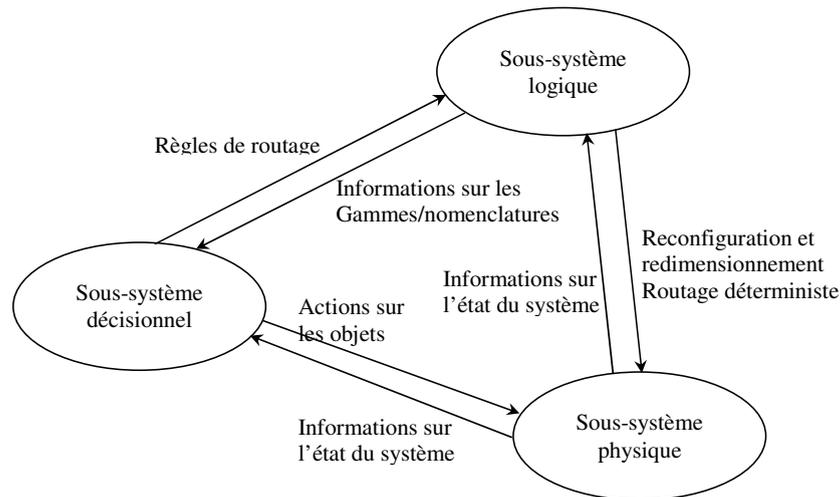


Figure 2.3 – Interaction entre les trois sous-systèmes

En fonction des informations sur l'état du sous-système physique (état des ressources) et du sous-système logique (positions des entités de flux, contraintes), le sous-système décisionnel a pour but de :

- modifier par ses décisions l'état du sous-système physique,
- déterminer l'ordonnancement des entités de flux (choix de l'entité de flux à traiter) et l'affectation des ressources (choix de la ressource allouée),
- gérer l'attribution et la restitution des ressources.

Les interactions entre les sous-systèmes physique et logique sont les liens entre les entités de flux et les ressources.

L'étape de spécification est très délicate car elle fixe le niveau de détail et le type d'abstraction, et donc les entités et les types d'événements jugés pertinents. La validation/vérification du modèle de connaissance et des données nécessaires permet de s'assurer que le modèle est homomorphe au système, et qu'il est pertinent relativement aux objectifs de l'étude. Il faut enfin s'assurer que le modèle de connaissance est cohérent et complet par rapport à la méthodologie retenue pour le construire, et par rapport aux savoirs que les acteurs ont de leur système. Cette validation/vérification est effectuée par les acteurs du système et les experts en modélisation aidés par des outils informatiques qui implantent des mécanismes de contrôle systématique. Le résultat final de cette étape est l'obtention d'un modèle de connaissance approuvé et validé qui devient la référence pour élaborer un modèle informatique dédié au système.

2 Conception et implantation

2.1 Conception et implantation du modèle d'action

Le modèle de connaissance approuvé et validé est l'entrée principale de l'activité d'élaboration du modèle d'action. Ce dernier est une traduction du modèle de connaissance dans un formalisme mathématique (par exemple une méthode analytique qui exploite la théorie des réseaux de file d'attente, un modèle informatique, ...) ou dans un langage de programmation (par exemple un modèle de simulation). Il est exploitable sur ordinateur et fournit des critères de performance du système dont l'analyse peut permettre d'agir sur le système, et donc sur le modèle de connaissance, en modifiant sa structure et son fonctionnement. Plusieurs modèles d'action peuvent être déduits d'un même modèle de connaissance, selon le formalisme choisi et/ou le degré de finesse souhaité. La démarche de modélisation ASCI, conçue pour les problèmes d'évaluation des performances, a été par la suite étendue aux problèmes d'optimisation. Aussi, un modèle d'action désigne tout aussi bien un modèle d'évaluation

des performances qu'une méthode d'optimisation ou un couplage entre une méthode d'optimisation et un modèle d'évaluation des performances. On retrouvera donc :

- les modèles de simulation,
- les modèles markoviens,
- les modèles analytiques,
- les modèles mathématiques (solveur),
- les heuristiques,
- les métaheuristiques,
- les couplages et chaînages de méthodes,
- les couplages et chaînages de méthodes et modèles.

2.2 Conception et implantation du modèle de résultat

L'entrée principale pour élaborer le modèle de résultats est le modèle d'action vérifié. L'exploitation du modèle d'action fournit des résultats à partir desquels on construit un modèle de résultats (quantitatif et qualitatif). On parle ici de modèle de résultats car, pour un même modèle d'action, on choisit les types de résultats qu'on veut obtenir ainsi que la manière de les présenter. Par exemple, l'exploitation d'un modèle de simulation peut fournir une trace de fonctionnement du système qui est ensuite animée, des critères de performances d'entités du système, ou bien encore une analyse statistique d'indicateurs de performance de fonctions du système que l'on structurera par exemple dans un tableau de bord prospectif. Cette étape comprend la conception des expériences (la planification de la validation et la planification des expériences), la validation du modèle d'action, l'exécution des expériences et, à la fin, la construction du modèle des résultats. L'élaboration des plans d'expérience fournit la planification de la validation et la planification des expériences. La planification de la validation a pour objectif de choisir les outils et les méthodes de validation en fonction du formalisme choisi et la méthode de résolution du modèle d'action, de déterminer les paramètres (variables) cibles à comparer par mesures sur le système réel, l'analyse de données nécessaires pour la validation...

3 Analyse des résultats de l'étude et prise de décisions

Une fois le modèle de résultats élaboré, le processus de prise de décisions est activé. Lors de cette étape du processus de modélisation, les experts en modélisation, les acteurs et les managers du système examinent le(s) modèle(s) de résultats obtenus. En fonction de l'analyse des résultats, des décisions sont prises pour agir sur le système et/ou pour tester des hypothèses nécessitant des modifications du modèle de connaissance, ceci en vue d'améliorer les performances qualitatives ou quantitatives du système. En d'autres termes, soit on intervient sur le système et le processus de modélisation est terminé, soit on réitère le processus de modélisation. Les managers du système modélisé prennent leur décision à l'aide d'un tableau de bord prospectif, comparable à celui qu'ils utilisent pour piloter leurs activités réelles.

Dans la suite, nous utiliserons cette méthodologie afin d'élaborer notre modèle de connaissance générique pour la planification tactique.

II Modèle de connaissance générique pour la planification multi-site

Dans cette partie, nous élaborons un modèle de connaissance générique des chaînes logistiques dans un contexte de planification tactique. En reprenant le principe de décomposition en trois sous-systèmes (physique, logique et décisionnel) de la méthodologie ASCI, nous les présentons tour à tour ainsi que leurs dépendances en utilisant le langage de modélisation UML

1 Le sous-système physique

Dans le contexte de la planification tactique multi-site, on va retrouver dans le sous-système physique les éléments constitutifs de la chaîne logistique, c'est-à-dire :

- les usines,
- les stocks,
- les moyens de transport,
- les fournisseurs,
- les centres de distribution.

La figure 2.4 donne le modèle UML du sous-système physique dans sa globalité.

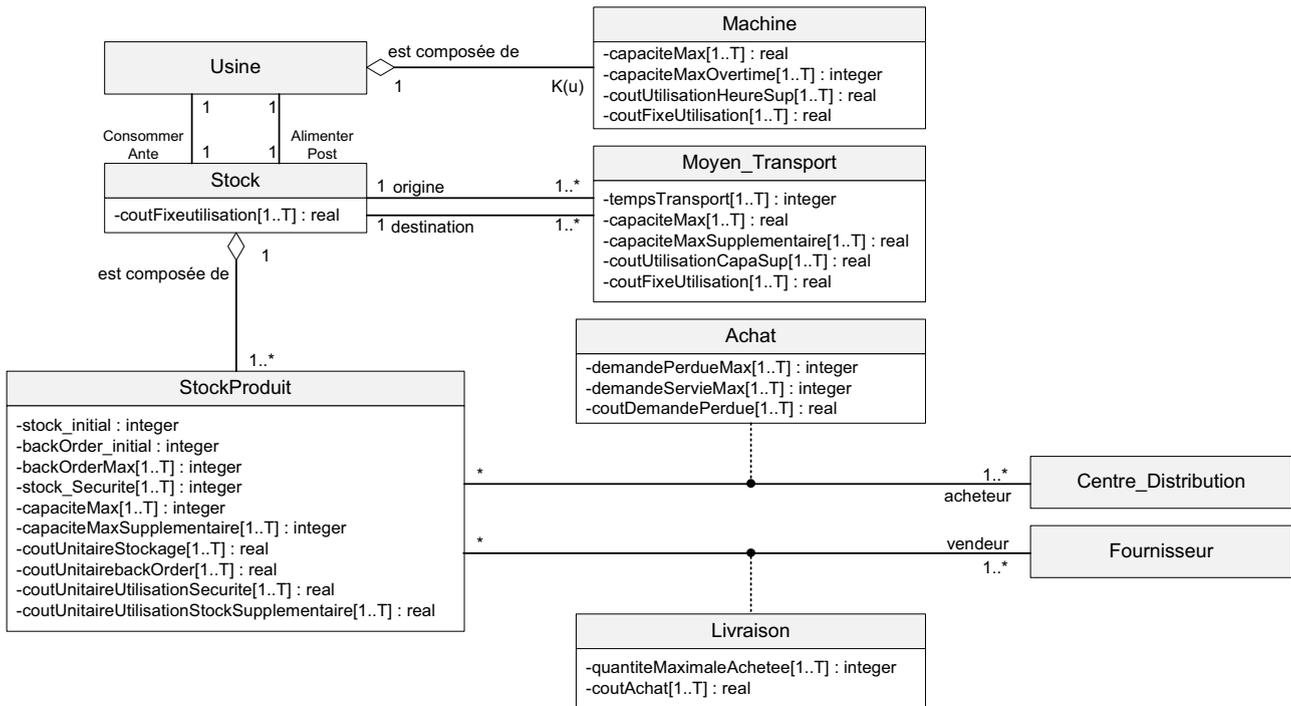


Figure 2.4 – Sous-système physique

1.1 Les usines

La chaîne logistique est constituée de plusieurs usines. Chaque usine est composée d'une ou plusieurs ressources de production (que nous appelons également Machine). Pour chaque ressource de production, nous avons :

- une capacité de production par période (pouvant être exprimée, par exemple, en nombre d'heures de production par période) : `capaciteMax[1..T]`,
- une capacité supplémentaire maximale par période : `capaciteMaxOvertime[1..T]` (par exemple, un volume maximum d'heures supplémentaires),
- un coût fixe d'utilisation de la ressource par période qui est engendré par l'utilisation de la ressource, indépendamment du produit fabriqué : `coutFixeUtilisation[1..T]`
- un coût d'utilisation d'une unité de capacité supplémentaire par période : `coutUtilisationHeureSup[1..T]`

La figure 2.5 donne la représentation UML correspondante.

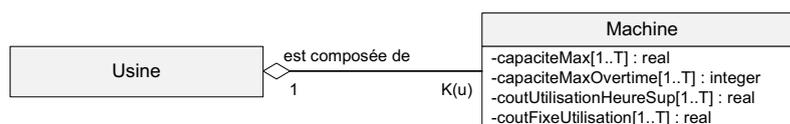


Figure 2.5 – Modélisation UML des usines et des machines qui leur sont associées

1.2 Les stocks

Nous appelons stock, un stock physique (un entrepôt par exemple). Chaque stock est constitué d'un ou plusieurs stocks logiques qui correspondent chacun à un type de produit stocké, c'est-à-dire qu'un stock logique est dédié à un type de produit. Nous appelons ces stocks logiques « StockProduit ».

Pour chaque StockProduit, nous avons :

- un stock initial : stock_Initial,
- un backorder initial : backOrder_Initial,
- un nombre de backorder maximal par période : backOrderMax[1..T],
- un niveau de stock de sécurité par période : stock_Securite[1..T],
- une capacité maximale de stockage (en nombre de produits par exemple) par période : capaMax[1..T],
- une capacité supplémentaire maximale de stockage par période : capaMaxSupplementaire[1..T].

Nous retrouvons également les coûts associés à l'utilisation du stock :

- un coût unitaire de backorder par période : coutUnitaireBackorder[1..T],
- un coût unitaire de stockage par période : coutUnitaireStockage[1..T],
- un coût unitaire de stock supplémentaire : coutunitaireUtilisationStockSupplementaire[1..T],
- un coût unitaire d'utilisation du stock de sécurité : coutUnitaireUtilisationSécurité[1..T].

La figure 2.6 donne la modélisation UML des ressources de stockage.

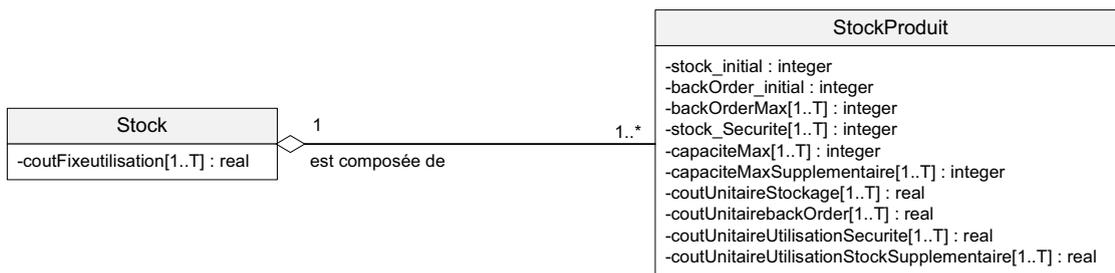


Figure 2.6 – Modélisation UML des ressources de stockage

1.3 Les moyens de transport

Nous définissons un moyen de transport comme la possibilité de transporter une certaine quantité de produit entre deux stocks de la chaîne logistique. Il est caractérisé, comme pour les ressources précédentes, par :

- une capacité maximale de transport par période (typiquement un certain volume) :
capaciteMax[1..T]
- une capacité supplémentaire maximale par période : capaciteMaxSupplementaire[1..T]
- un coût fixe d'utilisation de la ressource par période qui est engendré par l'utilisation de la ressource, indépendamment du produit fabriqué : coutFixeUtilisation[1..T]
- un coût d'utilisation d'une unité de capacité supplémentaire par période : coutUtilisationCapa-Sup[1..T]

On trouve également un temps de transport qui représente le temps qu'il faut pour joindre le stock de départ au stock de destination et qui dépend de la période de départ : TempsTransport[1..T].

La figure 2.7 donne la modélisation UML des moyens de transport.

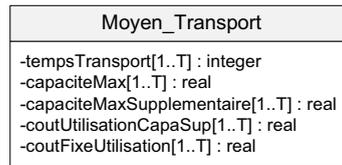


Figure 2.7 – Modélisation UML des moyens de transport

1.4 Les fournisseurs et les centres de distribution

Les fournisseurs et les centres de distribution sont les deux dernières ressources présentes dans le sous-système physique. En ce qui concerne les fournisseurs, ils sont susceptibles de livrer chaque produit dans n'importe quel stock physique à chaque période. Cependant dire qu'un fournisseur livre un certain produit dans un stock physique revient à dire qu'il livre cette quantité dans le StockProduit correspondant du stock physique. De la même façon, un centre de distribution est susceptible de s'approvisionner dans n'importe quel stock physique à chaque période, c'est-à-dire, suivant le produit demandé, dans n'importe quel StockProduit correspondant.

Les fournisseurs et les centres de distributions sont représentés par deux classes dans la modélisation UML. Nous les donnerons ultérieurement lors de la présentation du système physique dans sa globalité.

1.5 Les dépendances entre les différentes ressources composant le sous-système physique

Dans cette partie, nous montrons les dépendances qui existent entre les différentes ressources du sous-système physique décrites précédemment.

1.5.1 Dépendances Stock-Usine

A chaque usine composant la chaîne logistique sont associés deux stocks : un stock amont (dit stock « ante ») et un stock aval (dit stock « post »)

- le stock « ante » est le stock de produits dont l'usine a besoin pour fabriquer un produit donné (stock ante-fabrication)
- le stock « post » est le stock des produits fabriqués par l'usine (stock post-fabrication).

La figure 2.8 schématise cette association.

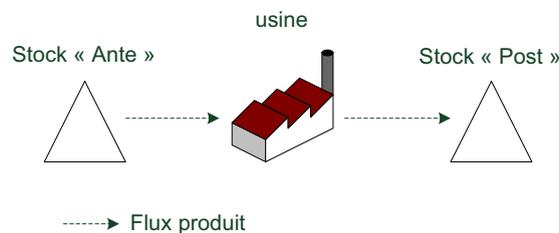


Figure 2.8 – positionnement des stocks

Les stocks étant à capacité finie, il sera possible d'entreposer dans chaque stock « ante » des produits autres que ceux dont l'usine qui lui est associé a besoin pour sa production, et dans le stock « post » des produits autres que ceux fabriqués par l'usine.

Dans la suite, nous utilisons la convention suivante :

- IA désigne un stock de type « ante »
- IP désigne un stock de type « post »
- $\Lambda \in \{IA, IP\}$ signifie que Λ est un stock de type « ante » ou un stock de type « post »

La figure 2.9 donne la modélisation UML de la relation unissant une usine et ses deux stocks.

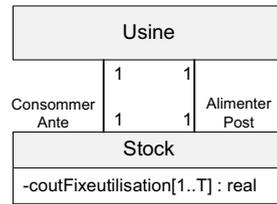


Figure 2.9 – Modélisation UML des dépendances Usine-Stocks

1.5.2 Dépendances Stock-Moyen de Transport

Nous avons défini un moyen de transport comme la possibilité de transporter une certaine quantité de produit entre deux stocks de la chaîne logistique. Un moyen de transport aura donc un stock d'origine et un stock de destination ce qui induira un temps de transport entre les deux stocks. La figure 2.10 illustre les 12 moyens de transport que nous définissons entre les stocks de deux usines.

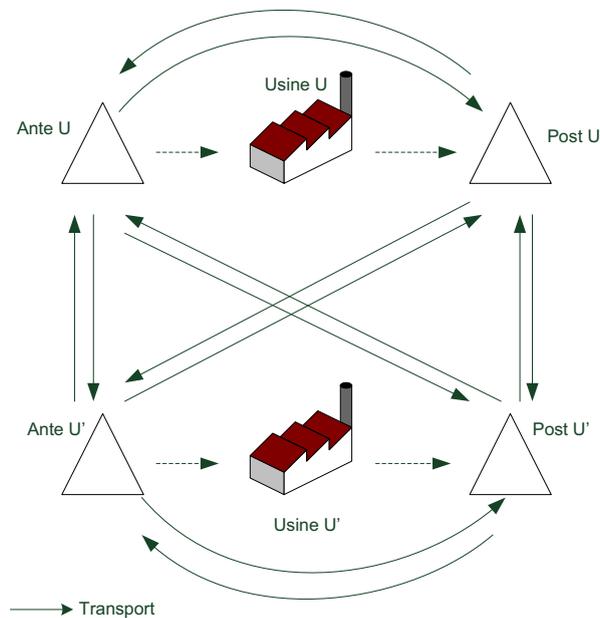


Figure 2.10 – Les transports entre stocks

Alors que nous ne considérons aucun transport entre une usine et un de ses stocks, il peut paraître paradoxal d'en considérer entre deux stocks rattachés à une même usine. Cependant, ce choix se justifie aisément. En effet, nous avons supposé qu'une usine ne pouvait s'approvisionner en composants nécessaires à sa production que dans son stock « ante ». Or, on peut être amené à modéliser le fait qu'une usine produise elle-même les composants dont elle a besoin. Il faut donc pouvoir transporter les composants fabriqués du stock « post » au stock « ante ». De plus, lorsque les stocks sont à capacité finie, on peut être amené à répartir les produits stockés par l'usine sur les deux stocks, d'où la nécessité de prévoir des moyens de transport entre ces deux stocks.

La figure 2.11 donne la modélisation UML de la relation unissant moyen de transport et stocks.

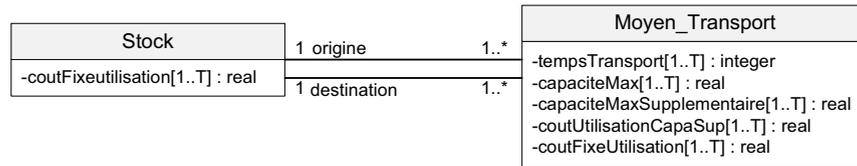


Figure 2.11 – Modélisation UML des dépendances Stock-Transport

1.5.3 Dépendances StockProduit-Fournisseur

Un fournisseur est susceptible de livrer un type de produits dans tout StockProduit correspondant. Plus précisément, tout StockProduit peut s’approvisionner chez n’importe quel fournisseur. Du coup, la relation entre un StockProduit et un Fournisseur est caractérisée par :

- une quantité maximale de produit pouvant être achetée au Fournisseur par période : quantitéMaximaleAchetee[1..T],
- un coût d’achat pour une unité de produit, défini par période : coutAchat[1..T].

D’où la modélisation UML donnée par la figure 2.12.

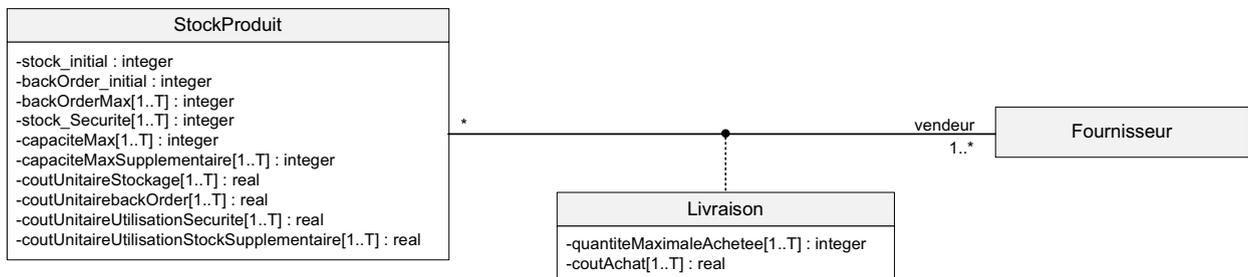


Figure 2.12 – Modélisation UML des dépendances StockProduit-Fournisseur

1.5.4 Dépendances StockProduit-Centre de distribution

Un centre de distribution est susceptible de s’approvisionner pour un produit dans tout StockProduit correspondant. La relation entre un StockProduit et un centre de distribution est caractérisée pour chaque période par :

- un coût unitaire de demande perdue : CoutDemandePerdue[1..T],
- une quantité maximale de demande servie : demandeServieMax[1..T],
- une quantité maximale de demande perdue : demandePerdueMax[1..T].

La figure 2.13 donne le modèle UML du lien de dépendance unissant StockProduit et centre de distribution.

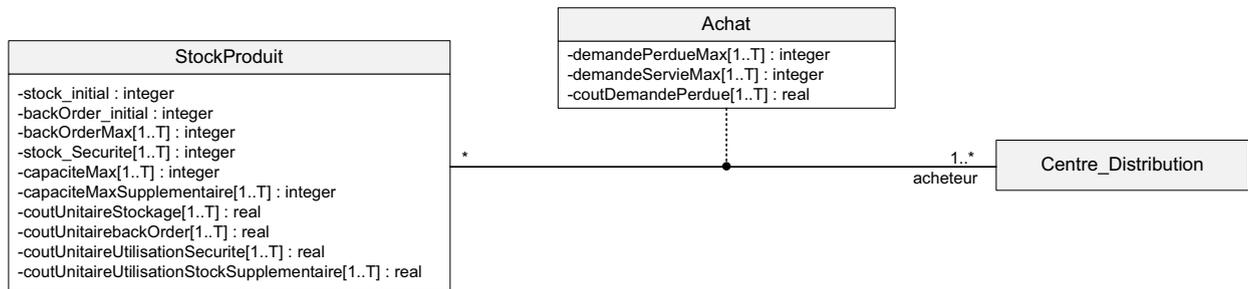


Figure 2.13 – Modélisation UML des dépendances StockProduit-Centre_Distribution

2 Le sous-système logique

Le sous-système logique est composé de l'ensemble des flux physiques traversant la chaîne logistique, c'est-à-dire de l'ensemble de tous les produits présents dans la chaîne logistique. On va donc retrouver les produits semi-finis, les produits-finis et les matières premières.

Dans notre modélisation, nous appelons produit soit un produit fini, soit un produit semi-fini. Un produit est caractérisé par :

- un numéro l'identifiant : numArticle
- pour chaque période, une capacité utilisée pour transporter l'article : capacite_Transport[1..T]
- un niveau de backorder maximal par période pour toute la chaîne logistique : back_order_max_SC[1..T]

Un produit fini possède une nomenclature qui définit les produits semi-finis et les matières premières ainsi que les quantités nécessaires à sa fabrication : ceux-ci ont un numéro de nomenclature correspondant à leur position dans la nomenclature du produit fini considéré (un produit fini a un numéro de nomenclature nul).

La figure 2.14 donne le modèle UML du sous-système logique.

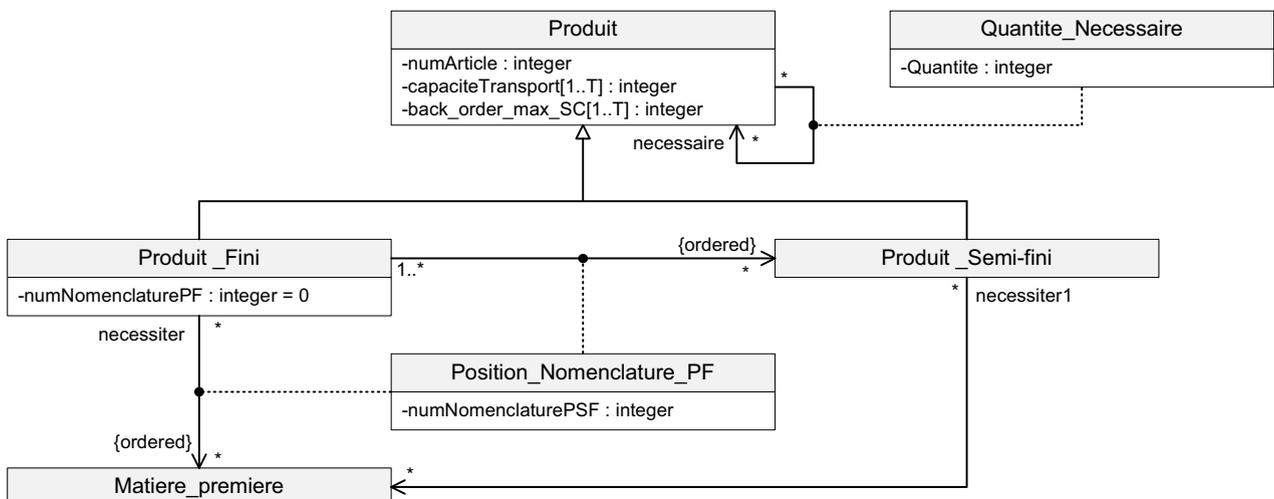


Figure 2.14 – Sous-système logique

3 Le sous-système décisionnel

Le sous-système décisionnel agit sur le sous-système logique et sur le sous-système physique. Dans celui-ci nous allons retrouver les contraintes liées aux ressources de production, de transport et de stockage, les règles de gestion (l'équilibre des stocks) ainsi que les coûts pris en compte. On trouve également les caractéristiques générales de la chaîne logistique.

3.1 Les contraintes du système de production

La figure 2.15 illustre les contraintes liées à la production dans leur globalité.

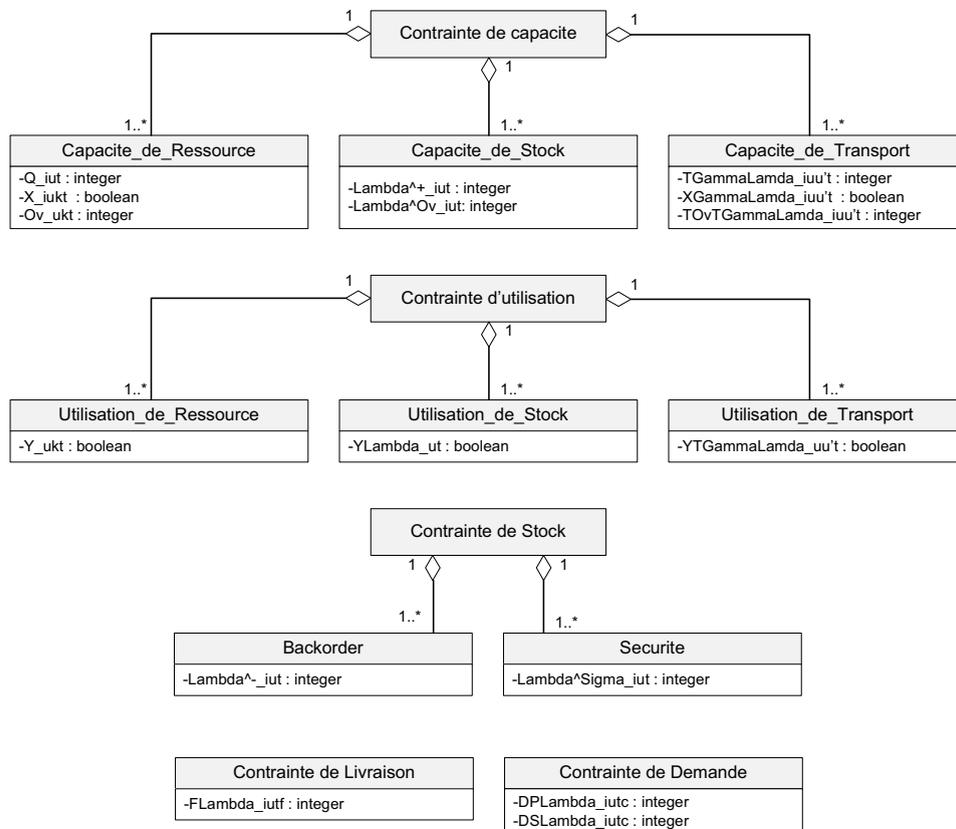


Figure 2.15 – Les contraintes du système de production

3.1.1 Les contraintes de capacité

Pour chaque ressource de la chaîne logistique, nous retrouvons les contraintes de capacité associées.

- pour les ressources de production : à chaque période t , la production effectuée par la ressource k ne doit pas dépasser sa capacité maximale. On va donc retrouver pour chaque période t :
 - . la quantité fabriquée de produit i par l'usine u à laquelle appartient la ressource k (Q_{iut})
 - . l'attribut qui détermine si la ressource a été configurée pour la fabrication du produit i (X_{iukt})
 - . la capacité supplémentaire utilisée (Ov_{ukt})
- pour les ressources de stockage : à chaque période t , la quantité stockée ne doit pas dépasser la capacité maximale du stock. On va donc retrouver pour chaque usine u et chaque période t :
 - . les quantités en stock dans le stock produit associé au produit i où Λ désigne le type de stock (Post ou Ante) (Λ^+_{iut})
 - . la capacité supplémentaire utilisée pour le stock produit associé au produit i (Λ^{Ov}_{iut})

- pour les ressources de transport : à chaque période t , la quantité transportée ne doit pas dépasser la capacité maximale du moyen de transport. On va donc retrouver pour chaque couple d'usines (u, u') , chaque période t et chaque produit i :
 - . les quantités transportées d'un stock Gamma à un stock Lambda ($T\Gamma\text{Gamma}\Lambda_{iuu't}$) où Gamma et Lambda représentent chacun un type de stock non nécessairement différents.
 - . l'attribut qui détermine si le moyen de transport d'un stock Gamma à un stock Lambda a été aménagé pour transporter du produit i ($X\Gamma\text{Gamma}\Lambda_{iuu't}$).
 - . la capacité supplémentaire éventuellement utilisée pour transporter du produit i ($TOv\Gamma\text{Gamma}\Lambda_{iuu't}$).

La figure 2.16 illustre les contraintes liées aux capacités des ressources du système de production.

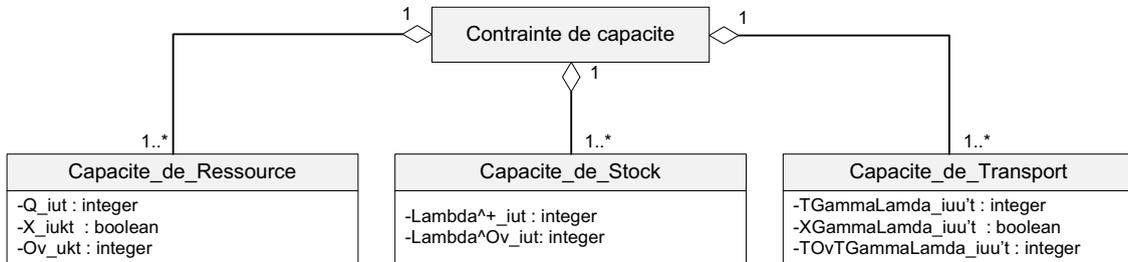


Figure 2.16 – Les contraintes de capacité

3.1.2 Les contraintes d'utilisation

Les contraintes d'utilisation permettent de savoir si une ressource a été utilisée. On retrouve donc des contraintes d'utilisation pour les ressources de production, de stockage et de transport.

- pour la ressource de production k de l'usine u , on retrouve l'attribut binaire Y_{ukt} qui prendra la valeur 1 si la ressource a été utilisée durant la période t , 0 sinon.
- pour le stock de type Lambda (post ou ante) de l'usine u , on retrouve l'attribut binaire $Y\Lambda_{ut}$ qui prendra la valeur 1 si le stock a été utilisé durant à la période t , 0 sinon.
- pour la ressource de transport reliant les stocks de type Gamma et Lambda de l'usine u et u' respectivement, on retrouve l'attribut binaire $Y\Gamma\text{Gamma}\Lambda_{uu't}$ qui prendra la valeur 1 si la ressource a été utilisée durant à la période t , 0 sinon.

La figure 2.17 illustre les contraintes liées à l'utilisation des ressources du système de production.

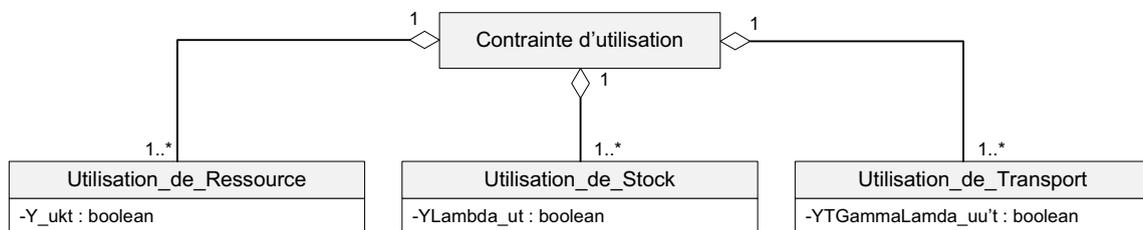


Figure 2.17 – Les contraintes d'utilisation

3.1.3 Les contraintes de stock

Au niveau des contraintes de stocks, nous allons retrouver celles liées au niveau de backorder (qui ne doit pas dépasser un certain niveau) et à l'utilisation du stock de sécurité. Ainsi, pour chaque usine u , chaque type de stock Lambda, chaque produit i et chaque période t nous aurons :

- le niveau de backorder Λ^-_{iut}
- la quantité prélevée dans le stock de sécurité Λ^{Σ}_{iut}

La figure 2.18 illustre les contraintes liées au stock du système de production.

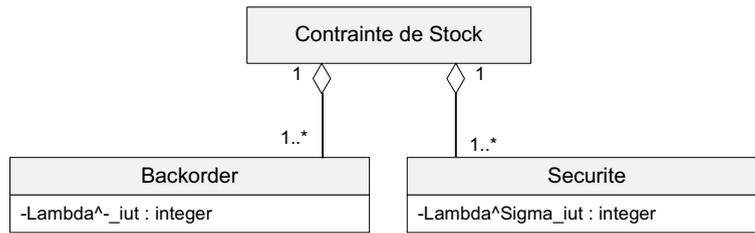


Figure 2.18 – Les contraintes de stock

3.1.4 Les contraintes de demande et de livraison

Les contraintes de livraison portent sur la quantité maximale que peut livrer un fournisseur à un stock de type Λ d'une usine u en produit i à la période t : on va retrouver les quantités livrées par le fournisseur f $FLambda_iutf$.

Les contraintes de demande portent sur les quantités de demande perdue et de demande à servir pour chaque centre de distribution. Elles déterminent d'une part les niveaux de demande servie et perdue mais aussi s'assurent que celles-ci ne dépassent pas une certaine quantité : on va donc retrouver la quantité de demande servie et perdue pour le stock de type Λ de l'usine u en produit i à la période t pour le centre de distribution c $DSLambda_iutc$ et $DPLambda_iutc$.

La figure 2.19 illustre les contraintes liées à la demande et à la livraison.



Figure 2.19 – Les contraintes demande et livraison

3.2 Les coûts pris en compte

Les coûts considérés sont de 4 ordres. En effet, nous avons les coûts liés :

- à la production qui dépendent des quantités fabriquées, des reconfigurations et utilisations de ressources, des capacités supplémentaires utilisées,
- au stockage qui sont fonction des quantités stockées, en bakorder, prélevées sur les stocks de sécurité, de l'utilisation des stocks, de la demande perdue etc.,
- au transport qui dépendent des quantités transportées et de l'utilisation des moyens de transport ainsi que de la capacité supplémentaire éventuellement allouée,
- à l'approvisionnement qui sont fonction des quantités livrées.

Chacun d'eux dépend de l'utilisation des ressources qui lui sont associées : c'est ce qu'illustre la figure 2.20

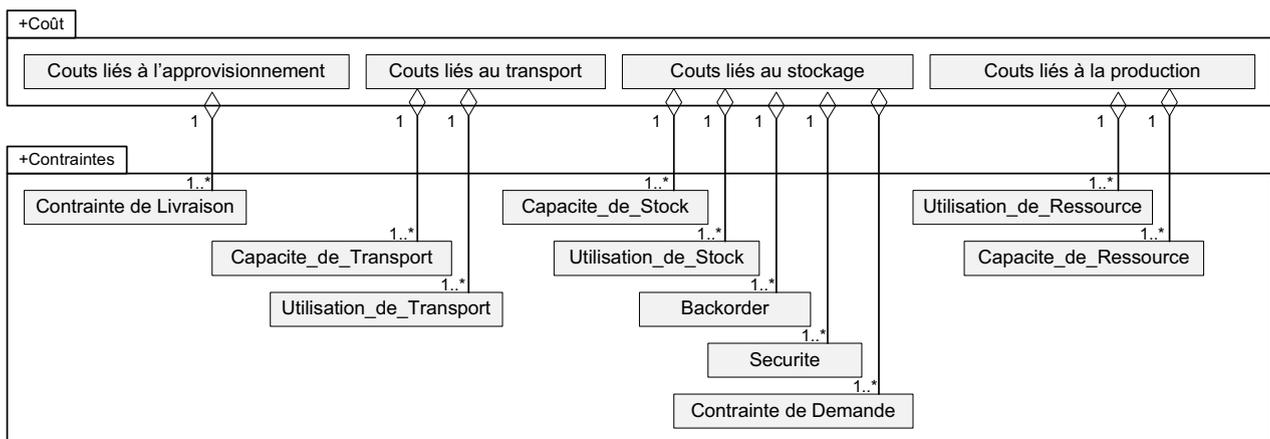


Figure 2.20 – Les coûts pris en compte au niveau de la chaîne logistique

3.3 La règle de gestion : équilibre des stocks

Cette règle permet de déterminer pour chaque produit, chaque période la quantité présente dans chaque stock : celle-ci est évidemment fonction des quantités fabriquées et mises dans ce stock, des quantités prélevées pour la fabrication, des backorders, de la demande perdue et de la demande à servir. Evidemment, les quantités arrivées en provenance ou expédiées vers d'autres stocks ainsi que les livraisons effectuées par les fournisseurs sont prises en compte. C'est ce qu'exprime la figure 2.21

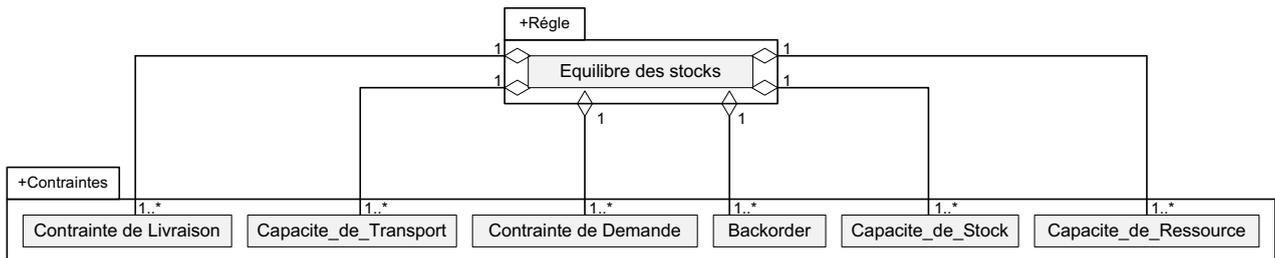


Figure 2.21 – La règle d'équilibre des stocks

3.4 Les caractéristiques générales de la chaîne logistique

Dans celles-ci, on va retrouver la taille de l'horizon de planification (T) ainsi que la capacité supplémentaire maximale en transport pour toute la chaîne logistique par période. C'est également dans les caractéristiques générales que vont se décider l'activation ou non des différentes contraintes. C'est ce qu'illustre la figure 2.22.

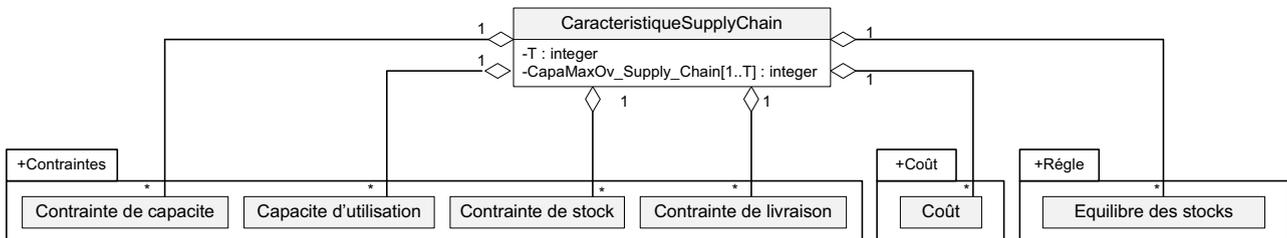


Figure 2.22 – Les caractéristiques générales de la chaîne logistique

4 Les dépendances entre les différents sous-systèmes

Dans cette partie, nous détaillons les relations qui unissent les sous-systèmes physique, logique et décisionnel.

4.1 Les dépendances entre le sous-système physique et le sous-système logique

4.1.1 La dépendance Usine-Produit

Le but d'un modèle de planification tactique multi-site est de déterminer des quantités de produits à fabriquer par usine et par période. Cela signifie donc qu'il existe bien une dépendance entre la classe Usine et la classe Produit sur lequel se trouvent le coût de fabrication du produit pour cette usine ainsi que son délai d'obtention. Nous trouvons donc une classe association `Caracteristique_Production` entre la classe Usine et la classe Produit contenant :

- les coûts de production d'une unité de produit pour chaque période : `cout_Production[1..T]`
- le lead time de production pour le produit dans cette usine : `delai_Obtention`.

La figure 2.23 donne le modèle UML du lien de dépendance unissant Usine et Produit.

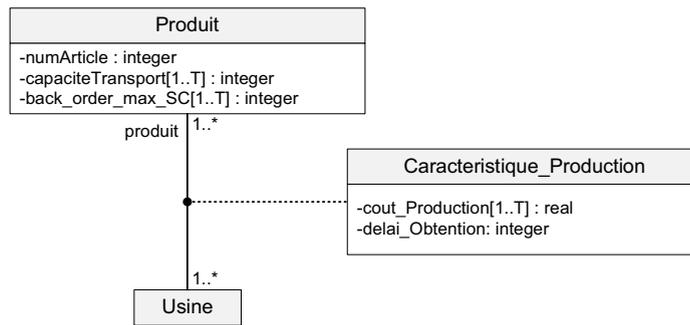


Figure 2.23 – Modélisation UML de la dépendance Usine-Produit

4.1.2 La dépendance Moyen_Transport-Produit

Nous avons vu qu'un moyen de transport relie deux stocks physiques et permet de véhiculer plusieurs types de produits. On a donc bien une dépendance entre la classe Produit et la classe Moyen_Transport. Nous trouvons donc une classe association Caracteristique_Produit_Transport entre Produit et Moyen_Transport dans laquelle on va retrouver pour chaque période :

- la capacité utilisée pour l'aménagement du moyen de transport en vue de transporter ce produit : capaciteConsommeeConfiguration[1..T],
- le coût unitaire de transport pour ce produit : cout_Transport[1..T].

La figure 2.24 donne le modèle UML du lien de dépendance unissant Moyen_Transport et Produit.

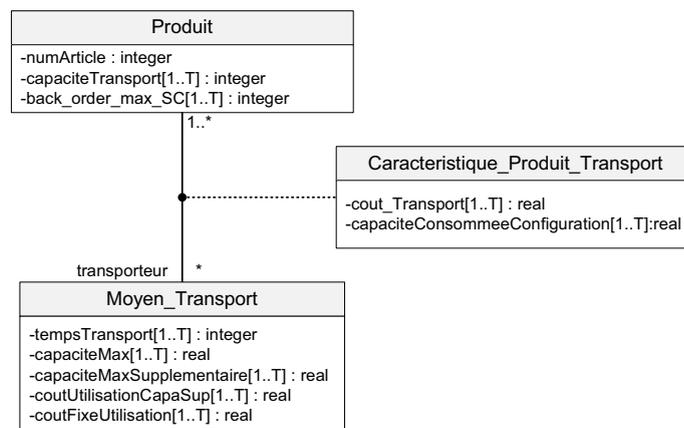


Figure 2.24 – Modélisation UML de la dépendance entre Moyen_Transport et Produit

4.1.3 La dépendance Centre_Distribution-Produit

A chaque période, chaque centre de distribution peut émettre une demande d'approvisionnement pour chaque produit. Il existe donc un lien de dépendance unissant les classes Centre_Distribution et Produit. Nous retrouvons donc une classe association Demande_Produit entre Centre_Distribution et Produit dans laquelle nous retrouvons pour chaque période :

- la demande d'approvisionnement émise par le centre de distribution : demande[1..T],
- la quantité maximale de demande perdue que le centre de distribution peut admettre pour ce produit : demande_perdue_max[1..T].

La figure 2.25 donne le modèle UML du lien de dépendance unissant Centre_Distribution et Produit.

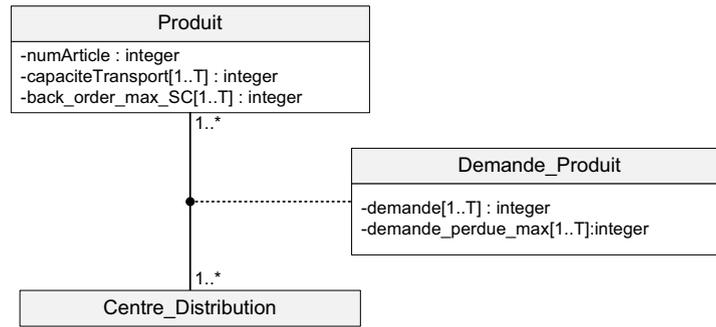


Figure 2.25 – Modélisation UML du lien entre Centre_Distribution et Produit

4.1.4 La dépendance Machine-Produit

A chaque période, chaque machine peut éventuellement entrer dans la fabrication d'un produit. Il y a donc bien une dépendance entre les classes Machine et Produit. Nous définissons donc une classe association entre ces deux classes, dans laquelle nous trouvons pour chaque période :

- la capacité consommée pour fabriquer le produit sur la machine : capacitéConsomméeFabrication[1..T],
- la capacité consommée pour configurer la machine en vue de fabriquer ce produit : capacitéConsomméeConfiguration[1..T],
- le coût engendré par la configuration de la machine : coutConfiguration[1..T]

La figure 2.26 donne le modèle UML du lien de dépendance unissant Machine et Produit.

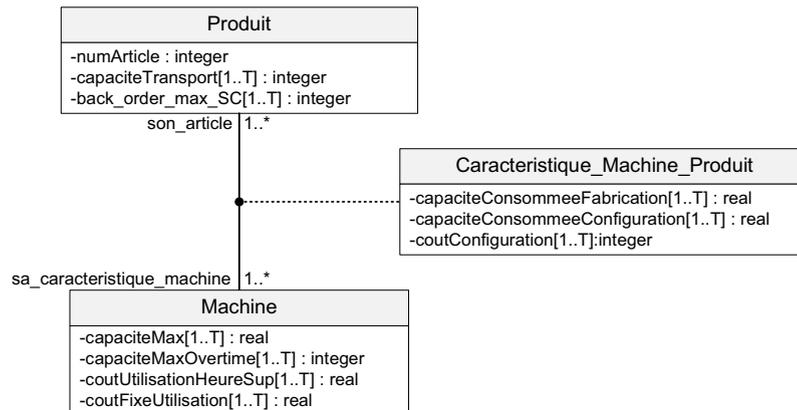


Figure 2.26 – Modélisation UML de la dépendance entre Machine et Produit

4.1.5 Le lien de dépendance Produit-StockProduit

Il existe une relation entre la classe Produit et la classe StockProduit, par définition même du StockProduit : en effet, pour un type produit donné, nous avons défini un StockProduit comme étant un stock logique ne contenant que des produits de ce type. En d'autres termes, un stock produit est relié à un produit mais un produit peut être relié à plusieurs StockProduit.

La formalisation UML de cette relation, donnée par la figure 2.27, est évidente.

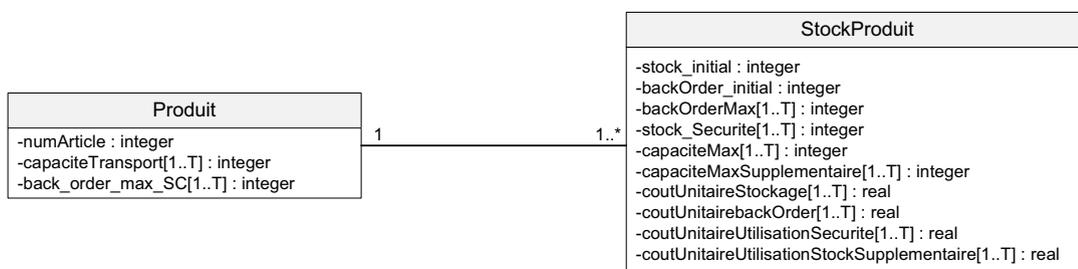


Figure 2.27 – Modélisation UML du lien de dépendance entre Produit et StockProduit

4.1.6 Le lien de dépendance Produit-Fournisseur

La dernière dépendance que nous mettons en exergue est celle reliant la classe Fournisseur et la classe Produit. Celle-ci est très naturelle car elle traduit juste le fait qu'un fournisseur livre des produits à la chaîne logistique. Nous allons donc pouvoir définir une classe association entre les classes Produit et Fournisseur : Achat_SC dans laquelle nous trouvons, pour chaque période, la quantité maximale de produit susceptible d'être achetée par la chaîne logistique à un fournisseur : quantité_achetee_maximale[1..T].

La figure 2.28 donne le modèle UML du lien unissant Fournisseur et Produit.

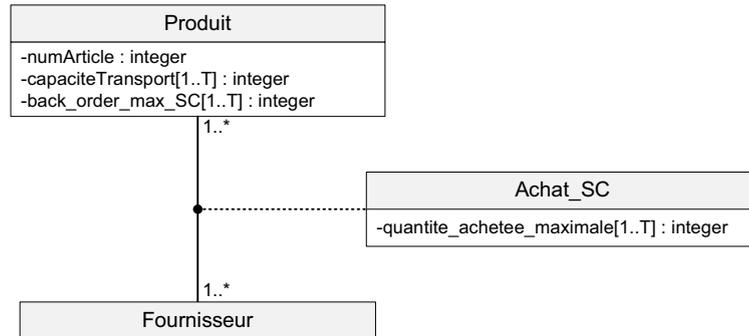


Figure 2.28 – Modélisation UML de la relation entre Fournisseur et Produit

4.2 Les dépendances entre le sous-système physique et le sous-système décisionnel

Les liens de dépendance entre les sous-systèmes physique et décisionnel sont très naturels. En effet, ceux-ci expriment juste sur quelles ressources sont appliquées les contraintes et règles du sous-système décisionnel. Ainsi, les contraintes de capacité et d'utilisation de ressource seront appliquées aux machines, les contraintes de stocks (capacité, utilisation, backorder, sécurité) et la règle d'équilibre des stocks au stock produit, les contraintes de livraison au fournisseur et les contraintes de demande au centre de distribution. La figure 2.29 illustre ces liens de dépendance

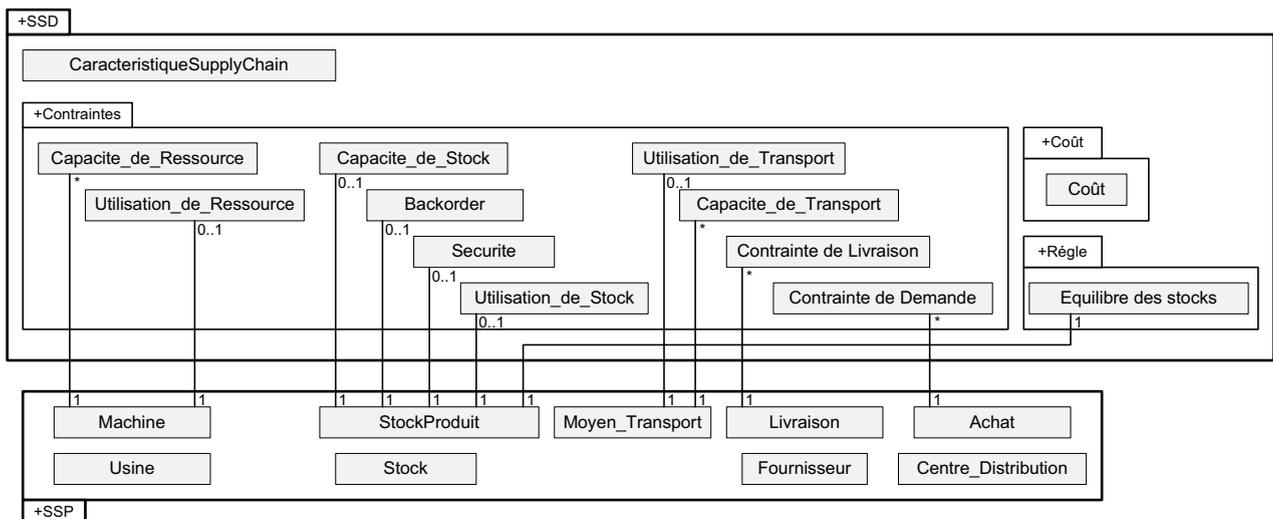


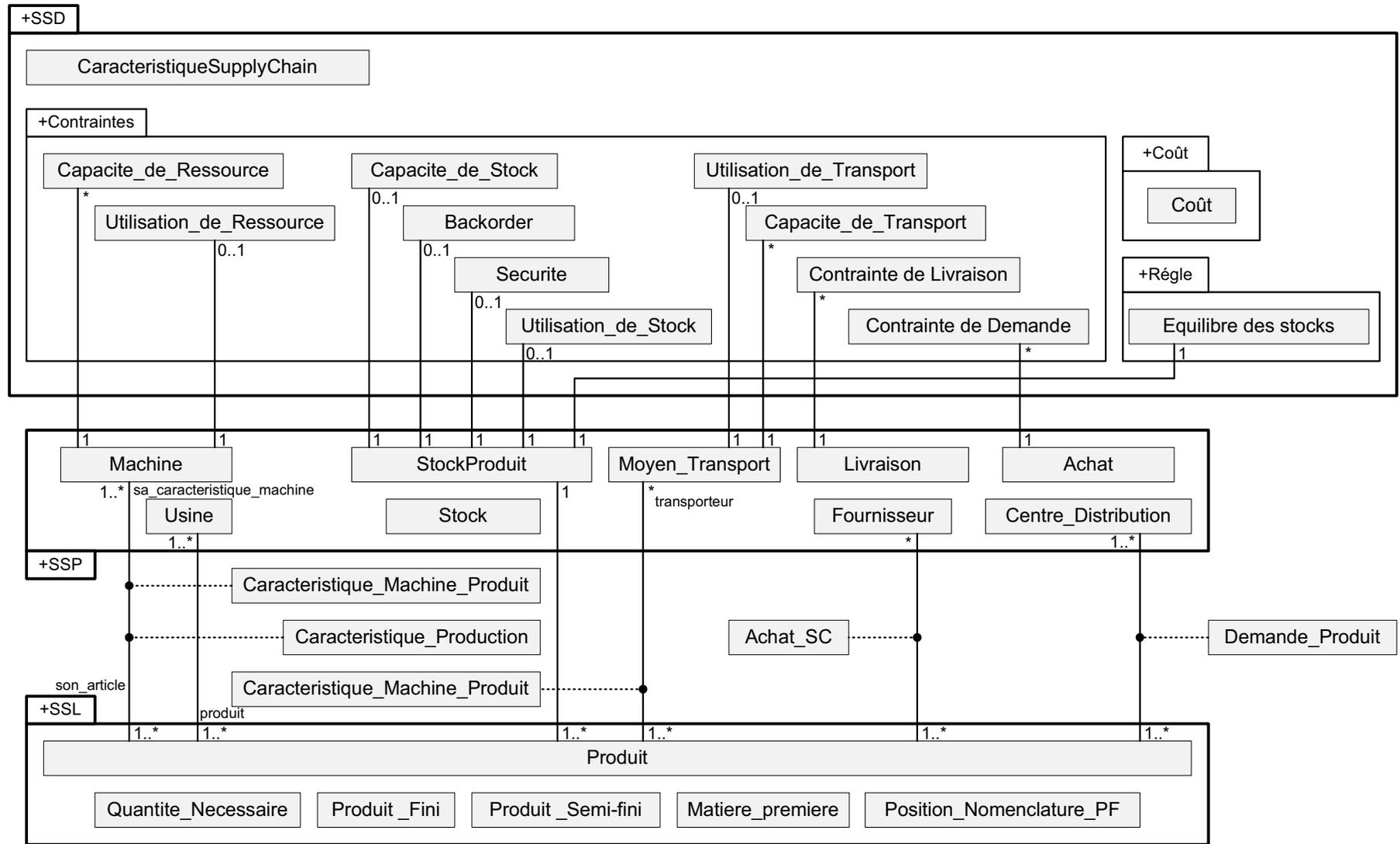
Figure 2.29 – Modélisation UML des liens entre le sous-système physique et le sous-système décisionnel

5 Le modèle UML représentant les trois sous-systèmes et les liens les unissant

La figure 2.30 donne une vue globale des trois sous-systèmes décrits et des dépendances les unissant.

Dans cette section nous avons donc spécifié tour à tour les sous-systèmes physique, logique et décisionnel. Nous avons décrit les liens les unissant. Nous avons donc donner le modèle de connaissance générique de notre problématique.

Figure 2.30 – Modélisation UML des liens entre les trois sous-systèmes

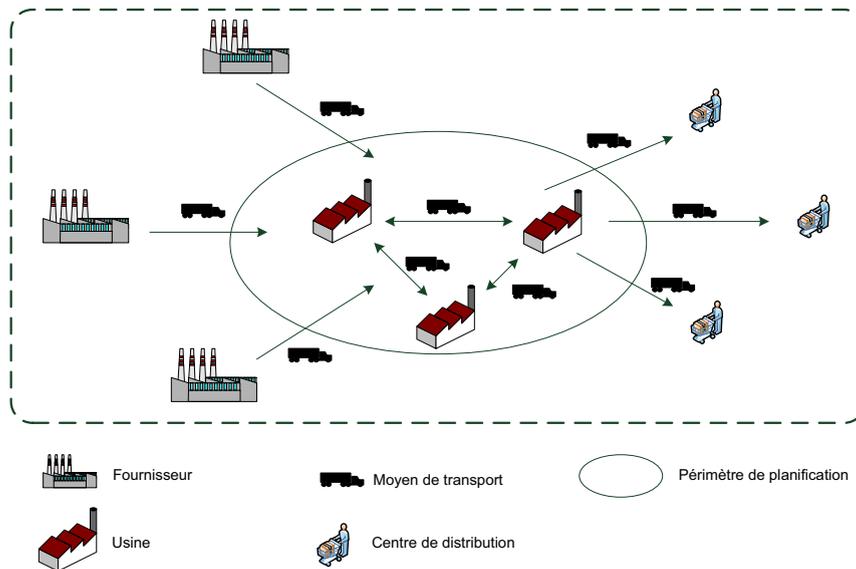


Dans la section suivante, nous donnons le modèle mathématique générique obtenu à partir du modèle de connaissance générique. Pour une question de lisibilité du modèle mathématique, nous avons préféré rédéfinir les noms des paramètres.

III Modèle mathématique générique pour la planification multi-site

Dans cette partie, nous proposons un modèle mathématique générique pour la planification tactique multi-site que nous obtenons à partir du modèle générique de connaissance. Le but de ce modèle est de planifier l'approvisionnement, la production, la distribution et le transport entre les diverses entités de la chaîne logistique sur un horizon moyen terme.

Nous rappelons la chaîne interne considérée :



la chaîne logistique considérée (Figure 2.1)

Celle-ci est composée de trois acteurs principaux : les fournisseurs, les usines et les centres de distribution.

Dans la suite, nous notons par :

- F : le nombre de fournisseurs
- C : le nombre de centres de distribution
- U : le nombre d'usines
- N : Le nombre de produits dont on planifie la fabrication
- T : Le nombre de périodes composant l'horizon de planification.

Chaque produit peut nécessiter, pour sa fabrication, un certain nombre d'autres produits (dits composants du dit produit) : on parle de lien de nomenclature entre un produit A et un produit B si, pour fabriquer A, il est nécessaire d'utiliser une certaine quantité de produit B. Ainsi, on peut définir entre les produits à planifier un graphe orienté acyclique dit graphe « gozinto » où chaque sommet représente un produit et où un arc entre deux sommets A et B représente un lien de nomenclature entre A et B : le poids associé à cet arc donne la quantité de produit B nécessaire pour fabriquer un produit A. Il est possible de représenter ce graphe par une matrice dite « gozinto » que nous notons ainsi :

a_{ij} : Quantité de produit i nécessaire à la fabrication d'un produit j ,
 $(i, j) \in \llbracket 1, N \rrbracket^2$

Nous commencerons par décrire chaque ressource prise en compte puis nous finirons par la description des flux de produits au travers de la formalisation mathématique des contraintes les régissant. Afin de faciliter la lecture de cette partie, le tableau 2.1 représente une grille de lecture de celle-ci en donnant, pour chaque type de ressource et pour la fonction objectif, les paragraphes où sont décrits variables et contraintes les concernant.

Flux Produits	Variables	Contraintes	Fonction Objectif
Production	§ III.1	§ III.4.1	§ III.4.4
Stock	§ III.2	§ III.4.2	
Transport	§ III.3	§ III.4.3	

Tableau 2.1 – grille de lecture du chapitre

Dans la suite de ce chapitre, les variables définies sont entières, sauf mention contraire.

1 Notations mathématiques des ressources « production »

Lorsqu'un produit est fabriqué dans une usine, toutes les ressources ne sont pas nécessairement utilisées. En effet, selon la gamme de fabrication du produit considéré, les moyens mis en œuvre diffèrent tant en termes de ressources qu'en termes de temps d'utilisation de ces dernières. De plus, pour fabriquer un produit donné, il faut éventuellement reconfigurer les ressources utilisées ce qui induit, d'une part, un temps de reconfiguration (qui peut être modélisé par une perte de capacité) et d'autre part, un certain coût. La figure 2.31 schématise ceci.

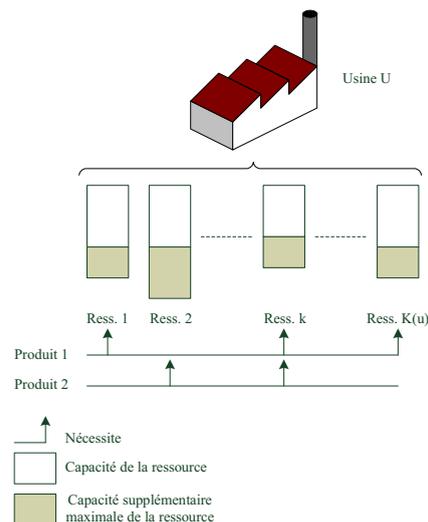


Figure 2.31 – Les ressources de production

Les paramètres du modèle liés à la fabrication des produits et à l'utilisation des ressources sont notés par :

- $K(u)$: Nombre de ressources dans l'usine u , $u \in \llbracket 1, U \rrbracket$,
- Z_{iuk} : $\begin{cases} 1 & \text{si la ressource } k \text{ de l'usine } u \text{ entre dans la fabrication du} \\ & \text{produit } i \\ 0 & \text{sinon} \end{cases}$
 $(i, u) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket$, $k \in \llbracket 1, K(u) \rrbracket$
- lp_{iu} : Délai d'obtention (lead time) d'un produit i dans l'usine u
 $(i, u) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket$
- $CapaR_{ukt}$: Capacité maximale de fabrication de la ressource k de l'usine u ,
 durant la période t , $(u, t) \in \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$, $k \in \llbracket 1, K(u) \rrbracket$
- $CapaR_{ukt}^{Ov}$: Nombre maximal d'unités de capacité supplémentaires pour la
 ressource k de l'usine u à la période t ,
 $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$, $k \in \llbracket 1, K(u) \rrbracket$
- $CapaR_t^{Ovsc}$: Nombre maximal d'unités de capacité supplémentaires défini par
 la chaîne logistique pour l'ensemble des ressources à la période t ,
 $t \in \llbracket 1, T \rrbracket$
- $Capa_{iukt}$: Capacité consommée pour la fabrication d'un produit i par la
 ressource k de l'usine u durant la période t ,
 $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$, $k \in \llbracket 1, K(u) \rrbracket$
- $SetupR_{iukt}$: Capacité consommée pour la reconfiguration de la ressource k
 de l'usine u pour fabriquer des produits i à la période t ,
 $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$, $k \in \llbracket 1, K(u) \rrbracket$
- $F\Lambda_{iutf}^{max}$: Quantité maximale de produit i achetée au fournisseur f par le
 stock Λ de l'usine u durant la période t
 $(i, u, t, f) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, F \rrbracket$
- F_{itf}^{maxsc} : Quantité maximale de produit i achetée au fournisseur f par
 l'ensemble de la chaîne logistique durant la période t .
 $(i, t, f) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, F \rrbracket$
- CP_{iut} : Coût de production d'une unité de produit i par l'usine u à la
 période t , $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
- CL_{iukt} : Coût de reconfiguration de la ressource k de l'usine u pour
 fabriquer des produits i à la période t ,
 $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$, $k \in \llbracket 1, K(u) \rrbracket$
- CO_{vukt} : Coût d'utilisation d'une unité de capacité supplémentaire pour
 la ressource k de l'usine u à la période t ,
 $(u, t) \in \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$, $k \in \llbracket 1, K(u) \rrbracket$
- CFR_{ukt} : Coût fixe d'utilisation de la ressource k de l'usine u à la période t ,
 $(u, t) \in \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$, $k \in \llbracket 1, K(u) \rrbracket$
- $CF\Lambda_{iutf}$: Coût d'achat d'une unité de produit i au fournisseur f pour le
 stock Λ de l'usine u durant la période t
 $(i, u, t, f) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, F \rrbracket$

Nous notons les variables de décision liées à la fabrication des produits et à l'utilisation des ressources par :

- Q_{iut} : Quantité de produit i fabriquée durant la période t par l'usine u ,
 $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
- X_{iukt} : $\begin{cases} 1 & \text{si la ressource } k \text{ de l'usine } u \text{ a été configurée pour fabriquer} \\ & \text{des produits } i \text{ durant la période } t \\ 0 & \text{sinon} \end{cases}$
 $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket, k \in \llbracket 1, K(u) \rrbracket$
- Y_{ukt} : $\begin{cases} 1 & \text{si la ressource } k \text{ de l'usine } u \text{ a été utilisée durant la période } t \\ 0 & \text{sinon} \end{cases}$
 $(u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket, k \in \llbracket 1, K(u) \rrbracket$
- OV_{ukt} : Nombre d'unités de capacité supplémentaire utilisés par la ressource k de l'usine u à la période t , $(u, t) \in \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket, k \in \llbracket 1, K(u) \rrbracket$
- $F\Lambda_{iutf}$: Quantité de produit i achetée au fournisseur f pour le stock Λ de l'usine u durant la période t , $(i, u, t, f) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, F \rrbracket$

2 Notations mathématiques pour les ressources « stock »

En ce qui concerne les stocks, nous utilisons les notations suivantes ($\Lambda \in \{IA, IP\}$) :

- $CapaS\Lambda_{iut}^+$: Stock maximal en produit i dans le stock Λ de l'usine u à la période t , $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
- $CapaS\Lambda_{iut}^-$: Backorder maximal en produit i dans le stock Λ de l'usine u à la période t , $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
- $CapaS_{it}^{-sc}$: Backorder maximal en produit i pour toute la chaîne logistique à la période t , $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
- $CapaS\Lambda_{iut}^{Ov}$: Quantité maximale d'unités de stock supplémentaires utilisables pour entreposer des produits i dans le stock Λ de l'usine u à la période t , $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
- $CapaS_{it}^{Ovsc}$: Quantité maximale d'unités de stock supplémentaires utilisables pour entreposer des produits i pour toute la chaîne logistique à la période t , $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
- $Se\Lambda_{iut}$: Stock de sécurité en produit i dans le stock Λ de l'usine u à la période t , $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
- Λ_{iu0}^+ : Stock initial en produit i dans le stock Λ de l'usine u
 $(i, u) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket$
- Λ_{iu0}^- : Backorder initial en produit i dans le stock Λ de l'usine u
 $(i, u) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket$

- $DS\Lambda_{iut}^{max}$: Demande servie maximale en produit i à la période t par le stock Λ de l'usine u pour le centre de distribution c
 $(i, u, t, c) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, C \rrbracket$
- $DP\Lambda_{iut}^{max}$: Demande perdue maximale en produit i à la période t par le stock Λ de l'usine u pour le centre de distribution c
 $(i, u, t, c) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, C \rrbracket$
- DP_{itc}^{maxsc} : Demande perdue maximale en produit i à la période t par l'ensemble des stocks de la chaîne logistique à la période t pour le centre de distribution c , $(i, t, c) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, C \rrbracket$
- $C\Lambda_{iut}^+$: Coût de stockage d'une unité de produit i dans le stock Λ de l'usine u pendant la période t , $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
- $C\Lambda_{iut}^-$: Coût de backorder pour une unité de produit i dans le stock Λ de l'usine u pendant la période t , $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
- $C\Lambda_{iut}^{Ov}$: Coût d'utilisation d'une unité de stock supplémentaire en vue de stocker des produits i dans le stock Λ de l'usine u pendant la période t , $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
- $CSe\Lambda_{iut}$: Coût d'utilisation d'une unité du stock de sécurité du produit i dans le stock Λ de l'usine u pendant la période t
 $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
- $CDP\Lambda_{iutc}$: Coût d'une unité de demande perdue en produit i pour le centre de distribution c par le stock Λ de l'usine u à la période t
 $(i, u, t, c) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, C \rrbracket$
- $CFS\Lambda_{ut}$: Coût fixe d'utilisation du stock Λ de l'usine u à période t
 $(u, t) \in \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$

Les variables relatives aux stocks que nous utilisons dans le modèle sont listées ci-après :

- Λ_{iut}^+ : Quantité de produit i dans le stock Λ de l'usine u à la fin de la période t , $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
- Λ_{iut}^- : Quantité de produit i en backorder dans le stock Λ de l'usine u à la fin de la période t , $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
- Λ_{iut}^{Ov} : Quantité d'unités de stock supplémentaires utilisée pour stocker des produits i dans le stock Λ de l'usine u à la fin de la période t ,
 $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
- Λ_{iut}^Σ : Quantité de produit i ponctionnée dans le stock de sécurité Λ de l'usine u à la fin de la période t , $(i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
- $Y\Lambda_{ut}$: $\begin{cases} 1 & \text{si le stock } \Lambda \text{ de l'usine } u \text{ a été utilisé durant la période } t \\ 0 & \text{sinon} \end{cases}$
 $(u, t) \in \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
- $DS\Lambda_{iutc}$: Demande à servir en produit i à la période t par le stock Λ de l'usine u

pour le centre de distribution c

$$(i, u, t, c) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, C \rrbracket$$

DPA_{iutc} : Demande perdue en produit i à la période t par le stock Λ de l'usine u

pour le centre de distribution c

$$(i, u, t, c) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, C \rrbracket$$

3 Modélisation des ressources « transport »

Dans le modèle de connaissance, il est stipulé que deux stocks A et B quelconques étaient reliés par deux ressources de transport : la première reliant A à B et la seconde B à A. Ainsi, on peut définir un graphe orienté complet dont les sommets sont les stocks et où un arc reliant un sommet A et un sommet B représente la ressource de transport reliant A à B. Le poids associé à cet arc représente la capacité de cette dernière.

Les paramètres liés au transport sont donnés ci-dessous :

$$((\Gamma, \Lambda) \in \{IA, IP\} \times \{IA, IP\})$$

$CapaT\Gamma\Lambda_{uu't}$: Capacité maximale de transport partant du stock Γ de l'usine u vers le stock Λ de l'usine u' durant la période t ,
 $(u, u', t) \in \llbracket 1, U \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$

$CapaT\Gamma\Lambda_{uu't}^{Ov}$: Nombre maximal d'unités de capacité de transport supplémentaires pour le transport du stock Γ de l'usine u vers le stock Λ de l'usine u' durant la période t ,
 $(u, u', t) \in \llbracket 1, U \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$

$SetupT\Gamma\Lambda_{iiu't}$: Capacité de transport perdue pour aménager le transport en vue de transporter le produit i du stock Γ de l'usine u vers le stock Λ de l'usine u' durant la période t ,
 $(u, u', t) \in \llbracket 1, U \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$

$CapaT_t^{Ovsc}$: Nombre maximal d'unités de capacité de transport supplémentaires pour toute la chaîne logistique durant la période t , $t \in \llbracket 1, T \rrbracket$

$CapaT_{it}$: Capacité de transport utilisée pour transporter une unité de produit i durant la période t , $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$

$lt_{iiu'}$: temps de transport pour un produit i de l'usine u vers l'usine u' , $(i, u, u') \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, U \rrbracket$

$CT\Gamma\Lambda_{iiu't}$: Coût de transport d'un produit i du stock Γ de l'usine u vers le stock Λ de l'usine u' durant la période t ,
 $(i, u, u', t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$

$CTOv\Gamma\Lambda_{uu't}$: Coût d'utilisation d'une unité de capacité de transport supplémentaire du stock Γ de l'usine u vers le stock Λ de l'usine u' durant la période t , $(u, u', t) \in \llbracket 1, U \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$

$CFT\Gamma\Lambda_{uu't}$: Coût fixe d'utilisation du moyen de transport du stock Γ de

l'usine u vers le stock Λ de l'usine u' durant la période t ,

$$(u, u', t) \in \llbracket 1, U \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$$

Les variables relatives aux transports que nous utilisons dans le modèle sont listées ci-après :

$T\Gamma\Lambda_{uu't}$:	Quantité de produit i transporté du stock Γ de l'usine u vers le stock Λ de l'usine u' durant la période t ,
		$(i, u, u', t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
$XT\Gamma\Lambda_{uu't}$:	$\begin{cases} 1 & \text{si le moyen de transport allant du stock } \Gamma \text{ de l'usine } u \\ & \text{au stock } \Lambda \text{ de l'usine } u' \text{ a été aménagé pour transporter} \\ & \text{des produits } i \text{ durant la période } t \\ 0 & \text{sinon} \end{cases}$
		$(i, u, u', t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket, k \in \llbracket 1, K(u) \rrbracket$
$YT\Gamma\Lambda_{uu't}$:	$\begin{cases} 1 & \text{si le moyen de transport allant du stock } \Gamma \text{ de l'usine } u \\ & \text{au stock } \Lambda \text{ de l'usine } u' \text{ a été utilisé durant la période } t \\ 0 & \text{sinon} \end{cases}$
		$(u, u', t) \in \llbracket 1, U \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket, k \in \llbracket 1, K(u) \rrbracket$
$TOv\Gamma\Lambda_{uu't}$:	Nombre d'unités de capacité de transport supplémentaires pour le transport du stock Γ de l'usine u vers le stock Λ de l'usine u' durant la période t ,
		$(u, u', t) \in \llbracket 1, U \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$

4 Modélisation des flux de produits

Dans cette partie, nous allons présenter les contraintes régissant les flux de produits. Nous allons commencer par présenter celles liées aux ressources de production.

4.1 Les contraintes liées aux ressources « production »

Nous avons vu que chaque usine u était composée de $K(u)$ ressources possédant chacune une certaine capacité de production. Leurs capacités peuvent être ponctuellement augmentées grâce à l'utilisation d'une capacité supplémentaire donnant lieu à un surcoût proportionnel à la quantité supplémentaire utilisée. Cette dernière peut être bornée au niveau de chaque ressource ou par une capacité maximale globale définie au niveau de la chaîne logistique. De plus, nous avons vu que l'utilisation d'une ressource d'une usine pour la fabrication d'un produit dépendait de la gamme de celui-ci et que, dans ce cas, il pouvait être nécessaire de reconfigurer cette dernière, consommant ainsi une certaine capacité de production.

La contrainte (2.1) est une contrainte de capacité pour une ressource de production de l'usine u :

$$\sum_{i=1}^N (Capa_{iukt}Q_{iut} + SetupR_{iukt}X_{iukt}) \times Z_{iuk} \leq CapaR_{ukt} + Ov_{ukt} \quad (2.1)$$

$$\forall (u, t) \in \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket, k \in \llbracket 1, K(u) \rrbracket$$

La quantité $(Capa_{iukt}Q_{iut} + SetupR_{iukt}X_{iukt}) \times Z_{iuk}$ exprime la capacité consommée pour la fabrication d'un produit i par la ressource k si cette dernière appartient à la gamme de ce produit. La somme de toutes ces quantités ne doit pas excéder la capacité maximale de la ressource, éventuellement

augmentée d'une certaine capacité supplémentaire. Cette capacité supplémentaire peut être bornée par une capacité supplémentaire maximale par ressource, ce qu'exprime la contrainte (2.2) :

$$Ov_{ukt} \leq CapaR_{ukt}^{Ov} \quad \forall (u, t) \in \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket, k \in \llbracket 1, K(u) \rrbracket \quad (2.2)$$

La contrainte (2.3) borne les capacités supplémentaires allouables aux ressources au niveau de la chaîne logistique :

$$\sum_{u=1}^U \sum_{k=1}^{K(u)} Ov_{ukt} \leq CapaR_t^{Ovsc} \quad \forall t \in \llbracket 1, T \rrbracket \quad (2.3)$$

La contrainte (2.4) permet de modéliser le fait qu'un produit ne sera fabriqué dans l'usine u que s'il existe dans celle-ci les ressources entrant dans sa gamme de fabrication :

$$\sum_{k=1}^{K(u)} Capa_{iukt} Q_{iut} \leq \sum_{k=1}^{K(u)} Z_{iuk} (CapaR_{ukt} + CapaR_{ukt}^{Ov}) \quad (2.4)$$

$$\forall (i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket, k \in \llbracket 1, K(u) \rrbracket$$

La contrainte (2.5) indique si la ressource a été reconfigurée en vue de fabriquer des produits i , ce qui permettra d'imputer à la contrainte de capacité le temps consommé à cette reconfiguration :

$$Capa_{iukt} Z_{iuk} Q_{iut} \leq (CapaR_{ukt} + CapaR_{ukt}^{Ov}) X_{iukt} \quad (2.5)$$

$$\forall (i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket, k \in \llbracket 1, K(u) \rrbracket$$

La contrainte (2.6) permet de déterminer si la ressource k de l'usine u a été utilisée durant la période t , ceci permettra de prendre en compte un coût fixe d'utilisation de celle-ci :

$$\sum_{i=1}^N X_{iukt} \leq N \times Y_{ukt} \quad \forall (u, t) \in \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket, k \in \llbracket 1, K(u) \rrbracket \quad (2.6)$$

4.2 Les contraintes liées aux ressources « stock »

Traditionnellement, on désigne par « équations d'équilibre des stocks » les contraintes liant les produits aux stocks. Celles-ci traduisent la dynamique de ces derniers de la façon suivante : le niveau du stock en produit i à la période t est égal au niveau de stock en produit i à la période $t - 1$ auquel on ajoute les produits i qui sont entrés en stock à la période t et on soustrait ceux qui en sont sortis.

Dans notre formalisation, nous avons vu qu'à chaque usine sont associés deux stocks dont les rôles diffèrent. Commençons par examiner la contrainte liée au stock « ante ». La figure 2.32 illustre les flux produits entrant et sortant de ce dernier.

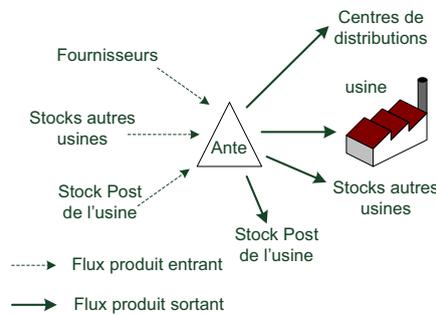


Figure 2.32 – Flux produit lié au stock « ante »

La contrainte s'exprime mathématiquement de la façon suivante :

$$\begin{aligned}
IA_{iut}^+ - \left(IA_{iut}^- + \sum_{c=1}^C DPIA_{iutc} \right) &= IA_{iu(t-1)}^+ - IA_{iu(t-1)}^- + \sum_{f=1}^F FIA_{iutf} \\
&+ \sum_{\substack{u'=1 \\ u' \neq u}}^U \sum_{\Lambda \in \{IA, IP\}} T\Lambda IA_{iu'u(t-lt_{iu'u})} + TIPIA_{i uut} \\
- \sum_{j=1}^N a_{ij} Q_{jut} - \sum_{c=1}^C DSIA_{iutc} - \sum_{\substack{u'=1 \\ u' \neq u}}^U \sum_{\Lambda \in \{IA, IP\}} TIA\Lambda_{i u u' t} \\
&- TIAIP_{i uut}
\end{aligned} \tag{2.7}$$

$$\forall (i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$$

Dans cette contrainte (2.7), on voit apparaître les flux décrits par la figure 2.32 :

- la quantité en produit i livrée par les fournisseurs est donnée par :

$$\sum_{f=1}^F FIA_{iutf}$$

- la quantité en produit i livrée à partir des stocks des autres usines à la période t est donnée par les quantités transportées à partir de ceux-ci. Les temps de transport étant pris en compte, les quantités livrées au stock « ante » de l'usine u à la période t sont celles expédiées à la période t moins le temps de transport. Cette quantité livrée par les stocks des autres usines est donnée par :

$$\sum_{\substack{u'=1 \\ u' \neq u}}^U \sum_{\Lambda \in \{IA, IP\}} T\Lambda IA_{iu'u(t-lt_{iu'u})}$$

- la quantité de produit i livrée à partir du stock « post » de l'usine est donnée par :

$$TIPIA_{i uut}$$

- la quantité de produit i utilisée par l'usine pour fabriquer des produits nécessitant le produit i comme composant est donnée par :

$$\sum_{j=1}^N a_{ij} Q_{jut}$$

- la demande en produit i à servir aux centres de distributions s'exprime par :

$$\sum_{c=1}^C DSIA_{iutc}$$

- les quantités à livrer aux stocks des autres usines sont données par :

$$\sum_{\substack{u'=1 \\ u' \neq u}}^U \sum_{\Lambda \in \{IA, IP\}} TIA\Lambda_{i u u' t}$$

- les quantités à livrer au stock «post» de l'usine s'expriment par :

$$TIAPA_{i uut}$$

On voit également que dans cette contrainte sont prises en compte la demande perdue et les backorders.

Les flux produits entrant et sortant du stock «post» sont illustrés par la figure 2.33.

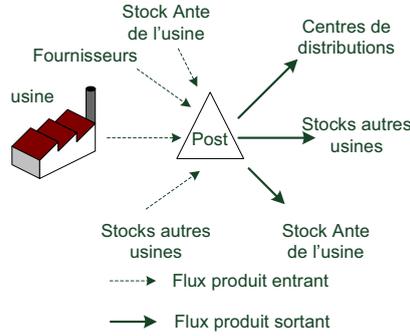


Figure 2.33 – Flux produit lié au stock « post »

La contrainte s'exprime mathématiquement de la façon suivante :

$$\begin{aligned}
 IP_{iut}^+ - \left(IP_{iut}^- + \sum_{c=1}^C DP_{iut}^c \right) &= IP_{iu(t-1)}^+ - IP_{iu(t-1)}^- + Q_{iu(t-l_{piu})} \\
 &+ \sum_{f=1}^F FIP_{iutf} + \sum_{\substack{u'=1 \\ u' \neq u}}^U \sum_{\Lambda \in \{IA, IP\}} T\Lambda IP_{iu'u(t-l_{iu'u})} + TIAIP_{iuit} \\
 &- \sum_{c=1}^C DSIP_{iut}^c - \sum_{\substack{u'=1 \\ u' \neq u}}^U \sum_{\Lambda \in \{IA, IP\}} TIP\Lambda_{iuu't} - TIPIA_{iuit}
 \end{aligned} \tag{2.8}$$

$$\forall (i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$$

On peut voir que cette contrainte (2.8) est très proche de la précédente. Cependant, la différence fondamentale réside dans le fait que l'usine auquel le stock est rattaché livre des produits i mais n'en consomme pas.

Les contraintes suivantes régissent les capacités de chaque stock ($\Lambda \in \{IA, IP\}$)

La contrainte sur la capacité du stock s'écrit :

$$\Lambda_{iut}^+ \leq CapaS\Lambda_{iut}^+ + \Lambda_{iut}^{Ov} \quad \forall (i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \tag{2.9}$$

Pour chaque stock, la capacité supplémentaire peut être bornée par une certaine quantité :

$$\Lambda_{iut}^{Ov} \leq CapaS\Lambda_{iut}^{Ov} \quad \forall (i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \tag{2.10}$$

Nous avons indiqué que nous pouvions borner la quantité globale des capacités supplémentaires en stock pour toute la chaîne logistique. Cela s'écrit :

$$\sum_{u=1}^U \sum_{\Lambda \in \{IA, IP\}} \Lambda_{iut}^{Ov} \leq CapaS_{it}^{Ovsc} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \tag{2.11}$$

Nous prenons également en compte les stocks de sécurité. Descendre en dessous du niveau du stock de sécurité engendre un certain coût qui est fonction de la quantité prise sur celui-ci. Cette quantité se mesure grâce à la contrainte suivante :

$$Se\Lambda_{iut} - \Lambda_{iut}^+ \leq \Lambda_{iut}^\Sigma \leq Se\Lambda_{iut} \quad \forall (i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \quad (2.12)$$

Nous avons également vu que nous prenions en compte les retards de livraison (backorder) et les livraisons annulées (demande perdue). Il est possible de limiter ces quantités au niveau de chaque stock et globalement au niveau de la chaîne logistique. Cela se traduit par les contraintes suivantes :

$$\Lambda_{iut}^- \leq CapaS\Lambda_{iut}^- \quad \forall (i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \quad (2.13)$$

$$\sum_{u=1}^U \sum_{\Lambda \in \{IA, IP\}} \Lambda_{iut}^- \leq CapaS_{it}^{-sc} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (2.14)$$

$$DP\Lambda_{iutc} \leq DP\Lambda_{iutc}^{max} \quad \forall (i, u, t, c) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, C \rrbracket \quad (2.15)$$

$$\sum_{u=1}^U \sum_{\Lambda \in \{IA, IP\}} DP\Lambda_{iutc} \leq DP_{itc}^{maxsc} \quad \forall (i, t, c) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, C \rrbracket \quad (2.16)$$

Dans le modèle proposé, chaque centre de distribution c émet une demande en produit i (éventuellement nulle) D_{itc} . Celle-ci définit pour chaque stock une demande à servir ($DS\Lambda_{iutc}$) et l'ensemble de toutes ces demandes à servir est égale à la demande du centre de distribution. Cela se traduit par la contrainte suivante :

$$\sum_{u=1}^U \sum_{\Lambda \in \{IA, IP\}} DS\Lambda_{iutc} = D_{itc} \quad \forall (i, t, c) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, C \rrbracket \quad (2.17)$$

Cependant, il est possible de limiter, pour un stock donné, la quantité de demande à servir pour le client c . Cela s'écrit :

$$DS\Lambda_{iutc} \leq DS\Lambda_{iutc}^{max} \quad \forall (i, u, t, c) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, C \rrbracket \quad (2.18)$$

Nous avons pris en compte la possibilité qu'à toute période t , tout fournisseur puisse livrer chaque stock en produits i . Cependant, il est possible de limiter ces livraisons d'une part au niveau de chaque stock et d'autre part au niveau de la chaîne logistique. Cela nous donne les contraintes suivantes :

$$F\Lambda_{iutf} \leq F\Lambda_{iutf}^{max} \quad \forall (i, u, t, f) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, F \rrbracket \quad (2.19)$$

$$\sum_{u=1}^U \sum_{\Lambda \in \{IA, IP\}} F\Lambda_{iutf} \leq F_{itf}^{maxsc} \quad \forall (i, t, f) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, F \rrbracket \quad (2.20)$$

Nous avons pris en compte la possibilité d'imputer un coût fixe pour l'utilisation d'un stock : dès qu'un produit i au moins est entreposé dans le stock, cela engendre un coût fixe (indépendant du nombre de produits entreposés). Il faut donc être capable de détecter lorsque le stock est utilisé ou non. C'est le rôle de la contrainte suivante :

$$\sum_{i=1}^N \Lambda_{iut}^+ \leq \sum_{i=1}^N (CapaS\Lambda_{iut}^+ + CapaS\Lambda_{iut}^{Ov}) Y\Lambda_{ut} \quad \forall (u, t) \in \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \quad (2.21)$$

4.3 Les contraintes liées aux ressources « transport »

Les ressources « transport » permettent de véhiculer des produits d'un stock vers un autre. Dans notre formalisation, nous considérons que ces ressources peuvent être à capacité finie et que celle-ci peut être ponctuellement augmentée moyennant un certain coût. Nous rappelons que cette augmentation peut être bornée d'une part par une capacité maximale supplémentaire par ressource de transport, d'autre part par une capacité maximale supplémentaire globale allouable à l'ensemble des ressources de transport de la chaîne logistique. Dans notre modélisation, nous avons distingué deux types de stock : les stocks « ante » et « post ». Nous avons donc 4 types de ressources de transport dépendant du type de stock de départ et du type de stock d'arrivée :

- le transport « post » - « post »
- le transport « post » - « ante »
- le transport « ante » - « post »
- le transport « ante » - « ante »

Les contraintes suivantes régissent les capacités de chaque transport :

$$((\Gamma, \Lambda) \in \{IA, IP\} \times \{IA, IP\})$$

$$\begin{aligned} \sum_{i=1}^N (T\Gamma\Lambda_{iuv't} \text{Capa}T_{it} + \text{Setup}T\Gamma\Lambda_{iuv't}XT\Gamma\Lambda_{iuv't}) \\ \leq \text{Capa}T\Gamma\Lambda_{uv't} + \text{TOv}\Gamma\Lambda_{uv't} \end{aligned} \quad (2.22)$$

$$\forall (u, u', t) \in \llbracket 1, U \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$$

La quantité $T\Gamma\Lambda_{iuv't}\text{Capa}T_{it} + \text{Setup}T\Gamma\Lambda_{iuv't}XT\Gamma\Lambda_{iuv't}$ correspond à la capacité consommée pour transporter des produits de type i (capacité consommée effectivement plus la capacité perdue pour aménager le transport en vue de transporter le produit i). La somme de toutes ces quantités doit être inférieure à la capacité du moyen de transport éventuellement augmentée d'une capacité supplémentaire. Cependant cette capacité peut être limitée, c'est ce qu'exprime les contraintes (2.23) et (2.24) :

$$\text{TOv}\Gamma\Lambda_{uv't} \leq \text{Capa}T\Gamma\Lambda_{uv't}^{Ov}, \forall (u, u', t) \in \llbracket 1, U \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \quad (2.23)$$

$$\sum_{u=1}^U \sum_{u'=1}^U \sum_{(\Gamma, \Lambda) \in \{IA, IP\}^2} \text{TOv}\Gamma\Lambda_{uv't} \leq \text{Capa}T_t^{Ovsc} \quad \forall t \in \llbracket 1, T \rrbracket \quad (2.24)$$

Nous trouvons également une contrainte de « setup » qui indique si le moyen de transport a dû être aménagé pour transporter des produits i :

$$T\Gamma\Lambda_{iuv't}\text{Capa}T_{it} \leq (\text{Capa}T\Gamma\Lambda_{uv't} + \text{Capa}T\Gamma\Lambda_{uv't}^{Ov}) XT\Gamma\Lambda_{iuv't} \quad (2.25)$$

$$\forall (i, u, u', t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$$

La contrainte (2.26) indique si le moyen de transport a été utilisé. Cela permettra d'imputer un coût fixe d'utilisation :

$$\sum_{i=1}^N XT\Gamma\Lambda_{iuv't} \leq N \times YT\Gamma\Lambda_{uv't} \quad \forall (u, u', t) \in \llbracket 1, U \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \quad (2.26)$$

4.4 Fonction objectif

Le but d'un modèle de planification tactique est de déterminer des plans de production répondant au mieux à la demande client à un coût minimum. La fonction objectif de ce modèle représente donc les coûts engendrés que l'on cherche à minimiser, c'est-à-dire les coûts liés à la production, au stockage, au transport et à l'approvisionnement.

$$\begin{aligned}
 \text{Minimiser } Z &= \text{coûts liés à la production} \\
 &+ \text{coûts liés au stockage} \\
 &+ \text{coûts liés à l'achat de produits chez les fournisseurs} \\
 &+ \text{coûts liés au transport}
 \end{aligned}$$

Dans les coûts liés à la production, on retrouve les coûts de fabrication proprement dit, les coûts de reconfiguration des ressources, les coûts fixes d'utilisation des ressources et, bien entendu les coûts d'utilisation de capacités supplémentaires.

$$\begin{aligned}
 \text{Coûts liés à la production} &= \text{coûts de fabrication} \\
 &+ \text{coûts de reconfiguration des ressources} \\
 &+ \text{coûts fixes d'utilisation des ressources} \\
 &+ \text{coûts d'utilisation de capacités supplémentaires}
 \end{aligned}$$

Cela s'exprime mathématiquement de la façon suivante :

$$\begin{aligned}
 \text{Coûts liés à la production} &= \sum_{t=1}^T \sum_{u=1}^U \sum_{i=1}^N \left(CP_{iut} Q_{iut} + \sum_{k=1}^{K(u)} CL_{iukt} X_{iukt} \right) \\
 &+ \sum_{t=1}^T \sum_{u=1}^U \sum_{k=1}^{K(u)} (CFR_{ukt} Y_{ukt} + COv_{ukt} Ov_{ukt})
 \end{aligned} \tag{2.27}$$

En ce qui concerne les coûts liés aux stocks, on va retrouver non seulement les coûts de stockage, les coûts de backorder et d'utilisation des stocks de sécurité mais aussi les coûts de demande perdue, d'utilisation d'unités de stocks supplémentaires et les coûts fixes d'utilisation des stocks.

$$\begin{aligned}
 \text{Coûts liés au stockage} &= \text{coûts de stockage} \\
 &+ \text{coûts de backorder} \\
 &+ \text{coûts d'utilisation d'unités de stocks supplémentaires} \\
 &+ \text{coûts d'utilisation des stocks de sécurité} \\
 &+ \text{coûts fixes d'utilisation des stocks} \\
 &+ \text{coûts de demande perdue}
 \end{aligned}$$

Mathématiquement, cela s'exprime de la façon suivante :

Coûts liés au stockage=

$$\begin{aligned} & \sum_{t=1}^T \sum_{u=1}^U \sum_{i=1}^N \sum_{\Lambda \in \{IA, IP\}} (C\Lambda_{iut}^+ \Lambda_{iut}^+ + C\Lambda_{iut}^- \Lambda_{iut}^- + C\Lambda_{iut}^{Ov} \Lambda_{iut}^{Ov} + CSe\Lambda_{iut} \Lambda_{iut}^{\Sigma}) \\ & + \sum_{t=1}^T \sum_{u=1}^U \sum_{\Lambda \in \{IA, IP\}} \left(CFS\Lambda_{ut} Y\Lambda_{ut} + \sum_{i=1}^N \sum_{c=1}^C CDP\Lambda_{iutc} DP\Lambda_{iutc} \right) \end{aligned} \quad (2.28)$$

Les coûts liés aux achats chez les fournisseurs sont calculés de la façon suivante :

$$\text{Coûts liés aux achats} = \sum_{t=1}^T \sum_{u=1}^U \sum_{i=1}^N \sum_{f=1}^F \sum_{\Lambda \in \{IA, IP\}} C F \Lambda_{iut f} F \Lambda_{iut f} \quad (2.29)$$

Quant aux coûts liés aux transports, on retrouve les coûts de transports qui sont fonction des quantités transportées mais aussi les coûts fixes d'utilisation de ceux-ci et aussi les coûts d'utilisation de capacités supplémentaires de transport :

$$\begin{aligned} \text{Coûts liés aux transports} &= \text{coûts de transports} \\ &+ \text{coûts fixes d'utilisation des transports} \\ &+ \text{coûts d'utilisation de capacités supplémentaires de transport} \end{aligned}$$

$$\begin{aligned} \text{Coûts liés aux Transport} &= \sum_{t=1}^T \sum_{i=1}^N \sum_{u=1}^U \sum_{\substack{u'=1 \\ u' \neq u}}^U \sum_{(\Lambda, \Gamma) \in \{IA, IP\}^2} C T \Gamma \Lambda_{i u u' t} T \Gamma \Lambda_{i u u' t} \\ &+ \sum_{t=1}^T \sum_{u=1}^U \sum_{\substack{u'=1 \\ u' \neq u}}^U \sum_{(\Lambda, \Gamma) \in \{IA, IP\}^2} C F T \Gamma \Lambda_{u u' t} Y T \Gamma \Lambda_{u u' t} \\ &+ \sum_{t=1}^T \sum_{u=1}^U \sum_{\substack{u'=1 \\ u' \neq u}}^U \sum_{(\Lambda, \Gamma) \in \{IA, IP\}^2} C T O v \Gamma \Lambda_{u u' t} T O v \Gamma \Lambda_{u u' t} \\ &+ \sum_{t=1}^T \sum_{i=1}^N \sum_{u=1}^U (C T I P I A_{i u u t} T I P I A_{i u u t} + C T I A I P_{i u u t} T I A I P_{i u u t}) \\ &+ \sum_{t=1}^T \sum_{i=1}^N \sum_{u=1}^U (C F T I P I A_{i u u t} Y T I P I A_{i u u t} + C F T I A I P_{i u u t} Y T I A I P_{i u u t}) \\ &+ \sum_{t=1}^T \sum_{i=1}^N \sum_{u=1}^U (C T O v I P I A_{i u u t} T O v I P I A_{i u u t} + C T O v I A I P_{i u u t} T O v I A I P_{i u u t}) \end{aligned} \quad (2.30)$$

IV Deux exemples d'instanciation du modèle générique

Dans cette partie, nous allons montrer comment instancier le modèle proposé afin de retrouver les modèles de planification tactique mono-site de référence ainsi que les modèles de planification multi-site rencontrés dans la littérature. Dans un premier temps, nous montrerons comment obtenir le MLCLSP qui est un modèle mono-site et multi-ressource puis, dans un second temps, nous nous intéresserons plus

précisément au modèle de planification multi-site donné par (Thierry 2003) et nous verrons comment il peut être facilement obtenu par instanciation du modèle générique. La figure 2.34 illustre la manière d’obtenir les principaux modèles de la littérature grâce au modèle présenté.

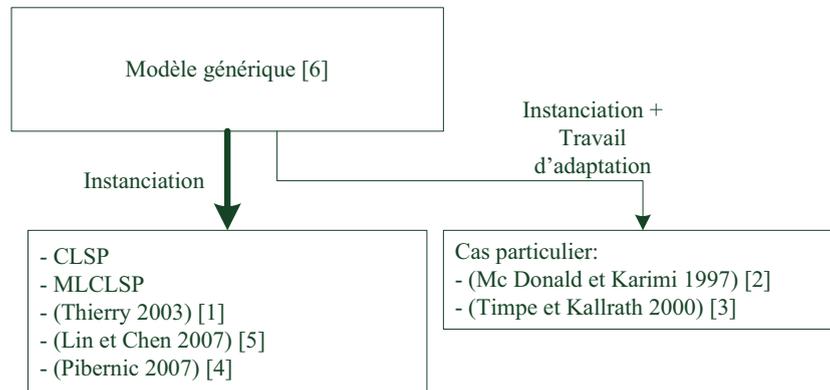


Figure 2.34 – Le modèle générique et les principaux modèles de la littérature

Comme mentionnée sur la figure 2.34, un travail d’adaptation est requis pour les modèles de (Mc Donald et Karimi 1997) et (Timpe et Kallrath 2000). Pour le modèle de (Mc Donald et Karimi 1997), les quantités fabriquées sont fonction d’un taux de fabrication, il suffira donc de les écrire en conséquence. Pour le modèle de (Timpe et Kallrath 2000), l’adaptation se fera au niveau des contraintes de transport : dans ce dernier, les transports considérés sont effectués par containers (la capacité de transport est donc « fractionnée » en plusieurs parties correspondant, pour chacune d’elles, à la capacité d’un container). Nous n’avons pas intégré ce type de transport dans notre modèle générique car il est vraiment dédié au cas d’étude de (Timpe et Kallrath 2000) et, donc, ne revêt pas un caractère générique.

1 Le MLCLSP

Le MLCLSP est un modèle mono-site multi-ressource traitant de la problématique MRP et MRPII : il planifie la fabrication des produits finis mais aussi tous les composants entrant dans leur fabrication en prenant en compte des contraintes de capacité au niveau de la production.

Nous donnons ici la formulation du MLCLSP donnée par (França et al. 1997).

Les paramètres du modèle sont :

- N : nombre de produits dont on planifie la fabrication
- T : nombre de périodes composant l’horizon de planification.
- K : nombre de ressources utilisées pour la production.
- a_{ij} : nombre de composants i utilisés pour fabriquer j ,
 $(i, j) \in \llbracket 1, N \rrbracket \times \llbracket 1, N \rrbracket$
- d_{it} : demande en produit i à la fin de la période t ,
 $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
- c_{it} : coût de production pour un produit i à la période t ,
 $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
- s_{it} : coût de lancement du produit i durant la période t

- $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
 h_{it} : coût de stockage du produit i durant la période t
 $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
 b_{kt} : capacité de la ressource k disponible à la période t
 $(k, t) \in \llbracket 1, K \rrbracket \times \llbracket 1, T \rrbracket$
 f_{ikt} : temps de reconfiguration de la ressource k pour produire i à la
période t $(i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, K \rrbracket \times \llbracket 1, T \rrbracket$
 v_{ikt} : capacité de la ressource k consommée pour produire i à la
période t $(i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, K \rrbracket \times \llbracket 1, T \rrbracket$
 M_{it} : une borne supérieure pour Q_{it} , $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$

Les variables de décisions du modèle sont :

- Q_{it} : Quantité de produit i fabriquée à la période t $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
 I_{it} : Niveau de stock en produit i à la période t $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
 Y_{it} : $\begin{cases} 1 & \text{si on produit } i \text{ à la période } t, \\ 0 & \text{sinon.} \end{cases}$ $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$

En utilisant ces notations, le MLCLSP s'écrit :

$$\text{Minimiser } \sum_{i=1}^N \sum_{t=1}^T (s_{it}Y_{it} + c_{it}Q_{it} + h_{it}I_{it}) \quad (2.31)$$

$$I_{it} = I_{i(t-1)} + Q_{it} - d_{it} - \sum_{j=1}^N a_{ij}Q_{jt} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (2.32)$$

$$\sum_{i=1}^N (f_{ikt}Y_{it} + v_{ikt}Q_{it}) \leq b_{kt} \quad \forall (k, t) \in \llbracket 1, K \rrbracket \times \llbracket 1, T \rrbracket \quad (2.33)$$

$$Q_{it} \leq M_{it}Y_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (2.34)$$

$$Y_{it} \in \{0, 1\} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (2.35)$$

$$Q_{it}, I_{it} \geq 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (2.36)$$

Dans la fonction objectif (2.31), on retrouve les cours de production, de stockage et de reconfiguration.

La contrainte (2.32) est la contrainte d'équilibre des stocks.

Les contraintes de capacités des ressources de production sont modélisées grâce à l'inégalité (2.33). Les temps de reconfiguration sont pris en compte.

La contrainte (2.34) permet de déterminer correctement les périodes où il est nécessaire de reconfigurer le système de production afin de fabriquer des produits i .

Dans ce modèle n'est considéré qu'un seul stock dans lequel l'usine puisera les composants nécessaires à la production et entreposera les produits qu'elle aura fabriqués. Dans le modèle de base, on ne considère qu'un seul client et on ne fait pas appel à des fournisseurs.

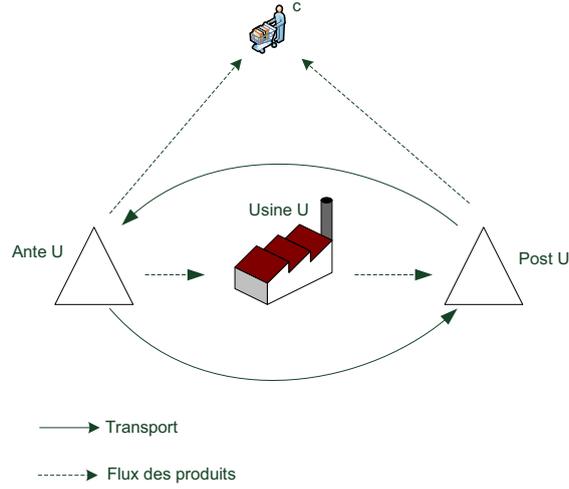


Figure 2.35 – Le modèle générique instancié avec une usine, un client et sans fournisseur

La figure 2.35 illustre l’instanciation du modèle générique dans le cas d’une seule usine u , sans fournisseur et avec un unique client c .

Dans un premier temps, intéressons-nous aux contraintes liées aux ressources « stock » : dans le modèle de base, les backorders et la demande perdue sont interdits et il n’y a pas de stock de sécurité. Nous avons donc les égalités suivants :

$$CapaSIP_{iut}^- = CapaSIA_{iut}^- = 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (2.37)$$

$$DPPIP_{iutc}^{max} = DPIA_{iutc}^{max} = 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (2.38)$$

$$SeIP_{iutc} = SeIA_{iutc} = 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (2.39)$$

Les relations (2.37) et (2.38) impliquent, via les contraintes (2.13) et (2.15) que

$$\begin{aligned} IP_{iut}^- &= IA_{iut}^- = 0 & \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \\ DPIP_{iutc}^- &= DPIA_{iutc}^- = 0 & \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \end{aligned}$$

Par conséquent, la contrainte (2.7) s’écrit sous la forme :

$$IA_{iut}^+ = IA_{iu(t-1)}^+ + TPIA_{iuut} - \sum_{j=1}^N a_{ij} Q_{jut} - DSIA_{iutc} - TIAIP_{iuut} \quad (2.40)$$

$$\forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$$

Et, compte tenu de l’absence de lead time ($lp_{iu} = 0$), la contrainte (2.8) devient :

$$IP_{iut}^+ = IP_{iu(t-1)}^+ + Q_{iut} + TIAIP_{iuut} - DSIP_{iutc} - TPIA_{iuut} \quad (2.41)$$

$$\forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$$

Dans le modèle du MLCLSP, nous n’avons qu’un seul stock. Pour modéliser cela, il suffit de transporter systématiquement tout ce qui arrive dans le stock «post» dans le stock «ante», de désactiver le transport du stock «ante» vers le stock «post», d’empêcher le client de se servir dans le stock «post» et de mettre la capacité du stock «post» à 0. Dans cette instanciation, la ressource de transport allant

du stock «post» au stock «ante» ne sert qu'à rapatrier les produits : nous supposons qu'elle est donc à capacité infinie et qu'elle n'engendre aucun coût. La figure 2.36 illustre cette modélisation.

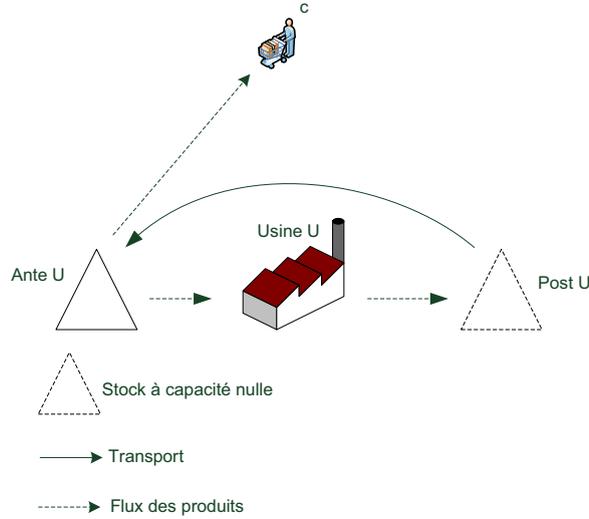


Figure 2.36 – Le modèle générique instancié pour le MLCLSP

Au niveau du paramétrage du modèle, cela nous donne :

$$CapaSIP_{iut}^+ = CapaSIP_{iut}^{Ov} = 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (2.42)$$

$$CapaSIA_{iut}^+ = +\infty \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (2.43)$$

$$DSIP_{iutc}^{max} = 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (2.44)$$

$$CapaTIPIA_{iut} = +\infty \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (2.45)$$

$$CapaTIAIP_{iut}^{Ov} = CapaTIAIP_{iut} = 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (2.46)$$

La relation (2.42) implique, via les contraintes (2.9) et (2.10), que :

$$IP_{iut}^+ = 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$$

La relation (2.44) montre qu'en utilisant la contrainte (2.18) on obtient :

$$DSIP_{iutc} = 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$$

La relation (2.46) conjuguée aux contraintes (2.22) et (2.23) implique que :

$$TIAIP_{iut} = 0 \quad \forall t \llbracket 1, T \rrbracket$$

La contrainte (2.41) s'écrit donc :

$$TIPIA_{iut} = Q_{iut} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$$

Il suffit juste alors de remarquer que la contrainte (2.17) nous donne :

$$DSIA_{iutc} = D_{itc} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$$

et, par suite la contrainte (2.40) devient :

$$IA_{iut}^+ = IA_{iu(t-1)}^+ + Q_{iut} - \sum_{j=1}^N a_{ij} Q_{jut} - D_{itc} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$$

ce qui correspond à la contrainte (2.32) du MLCLSP.

En ce qui concerne les ressources de production, le MLCLSP de base ne permet pas l'utilisation de capacité supplémentaire. Cela se traduit dans le modèle proposé par :

$$CapaR_{ukt}^{Ov} = 0 \quad \forall (k, t) \in \llbracket 1, K(u) \rrbracket \times \llbracket 1, T \rrbracket \quad (2.47)$$

En utilisant la contrainte (2.2), on obtient alors :

$$Ov_{ukt} = 0 \quad \forall (k, t) \in \llbracket 1, K(u) \rrbracket \times \llbracket 1, T \rrbracket$$

Ce qui entraîne que la contrainte (2.1) devient :

$$\sum_{i=1}^N (Capa_{iukt} Q_{iut} + SetupR_{iukt} X_{iukt}) \times Z_{iuk} \leq CapaR_{ukt} \quad (2.48)$$

Il suffit donc d'utiliser le fait que $Z_{iuk} = 1, \forall (i, u, k) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, K(u) \rrbracket$ et nous obtenons la contrainte (2.33).

La contrainte (2.5) alliée à la relation (2.47) donne :

$$Capa_{iukt} Q_{iut} \leq CapaR_{ukt} X_{iukt}$$

ce qui s'écrit encore :

$$Q_{iut} \leq \frac{CapaR_{ukt}}{Capa_{iukt}} X_{iukt}$$

ce qui donne la contrainte 2.34, en posant $M_{iut} = \sup_{k \in K(u)} \frac{CapaR_{ukt}}{Capa_{iukt}}$.

Quant à la fonction objectif, dans le MLCLSP ne sont pris en compte que les coûts de stockage, les coûts de production et les coûts de reconfiguration : il suffit donc de fixer les autres coûts à 0.

2 Un modèle de planification multi-site avec prise en compte des transports

Nous rappelons le modèle présenté par (Thierry 2003). C'est un modèle multi-site où chaque usine est mono-ressource. Les paramètres de celui-ci sont :

- N : nombre de produits dont on planifie la fabrication
- T : nombre de périodes composant l'horizon de planification
- U : nombre d'usines
- C : nombre de clients
- L : ensemble des liaisons définies entre les sites (usines, clients)
- d_{itc} : quantité de produit i demandée par le client c pour la période t
(i, t, c) $\in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, C \rrbracket$
- a_{ij} : quantité de produit i nécessaire pour fabriquer une unité de produit j (nomenclature) (i, j) $\in \llbracket 1, N \rrbracket \times \llbracket 1, N \rrbracket$
- b_{iu} : quantité de ressource nécessaire pour la réalisation d'une unité de produit i par l'usine u (i, u) $\in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket$

- C_{tu} : quantité maximale de ressource disponible à l'usine u durant la période t (t, u) $\in \llbracket 1, T \rrbracket \times \llbracket 1, U \rrbracket$
 $LArt_{iu}$: liste des produits pouvant être fabriqués par l'usine u et qui nécessitent directement le composant i (i, u) $\in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket$
 TT_{il} : durée de transport de l'article i par la liaison l (i, l) $\in \llbracket 1, N \rrbracket \times \llbracket 1, L \rrbracket$
 $depuis_u$: ensemble des liaisons dont l'origine est l'usine u , $u \in \llbracket 1, U \rrbracket$
 $vers_u$: ensemble des liaisons dont la destination est l'usine u , $u \in \llbracket 1, U \rrbracket$
 $vers_c$: ensemble des liaisons dont la destination est le client c , $u \in \llbracket 1, C \rrbracket$
 CP_{iut} : coût de production d'un produit i dans l'usine u durant la période t (i, u, t) $\in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
 CS_{iut} : coût de stockage d'un produit i dans l'usine u durant la période t (i, u, t) $\in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
 CT_{itl} : coût de transport d'un produit i sur la liaison l débutant à la période t (i, t, l) $\in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, L \rrbracket$

Les variables du modèle sont :

- Q_{iut} : quantité de produit i fabriquée par l'usine u durant la période t (i, u, t) $\in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$
 Tr_{itl} : quantité de produit i transportée sur la liaison l au cours de la période t , (i, t, l) $\in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, L \rrbracket$
 I_{iut} : quantité de produit i stockée par l'usine u à la fin de la période t (i, u, t) $\in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$

En utilisant ces notations, le modèle de (Thierry 2003) s'écrit :

$$\text{Minimiser } \sum_{i=1}^N \sum_{u=1}^U \sum_{t=1}^T \left(CP_{iut} Q_{iut} + CS_{iut} I_{iut} + \sum_{l \in \text{depuis}_u} CT_{itl} Tr_{itl} \right) \quad (2.49)$$

$$I_{iu(t+1)} = I_{iut} + Q_{iut+1} + \sum_{l \in \text{vers}_u} Tr_{i(t+1)-TT_{il}} - \sum_{j \in Lart_{iu}} a_{ij} Q_{ju(t+1)} - \sum_{l \in \text{depuis}_u} Tr_{i(t+1)l} \quad (2.50)$$

$$\forall (i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$$

$$\sum_{l \in \text{vers}_c} Tr_{itl} = d_{itc}, \forall (i, t, c) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, C \rrbracket \quad (2.51)$$

$$\sum_{i=1}^N b_{iu} Q_{iut} \leq C_{iut}, \forall (i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \quad (2.52)$$

La fonction objectif (2.49) correspond à une minimisation des coûts de transport, des coûts de production et des coûts de stockage.

La contrainte (2.50) est l'équation d'équilibre des stocks. On peut voir au travers de celle-ci, que chaque usine ne possède qu'un seul stock dans lequel elle puise les composants nécessaires à la production et entreposera les produits qu'elle a fabriqués. Il n'y a ni backorder, ni demande perdue, ni stock de sécurité. On remarquera également que les lead time de production ne sont pas pris en compte, contrairement à ceux de transport.

La contrainte (2.51) nous dit que les quantités transportées à chaque client c durant la période t correspondent à sa demande.

Enfin la contrainte (2.52) est une contrainte de capacité. Cette contrainte nous montre que chaque usine est considérée comme mono-ressource. Pour finir, remarquons qu'il n'y a pas de contrainte de capacité sur les transports.

La figure 2.37 donne la configuration de la chaîne logistique considérée dans la formalisation de (Thierry 2003).

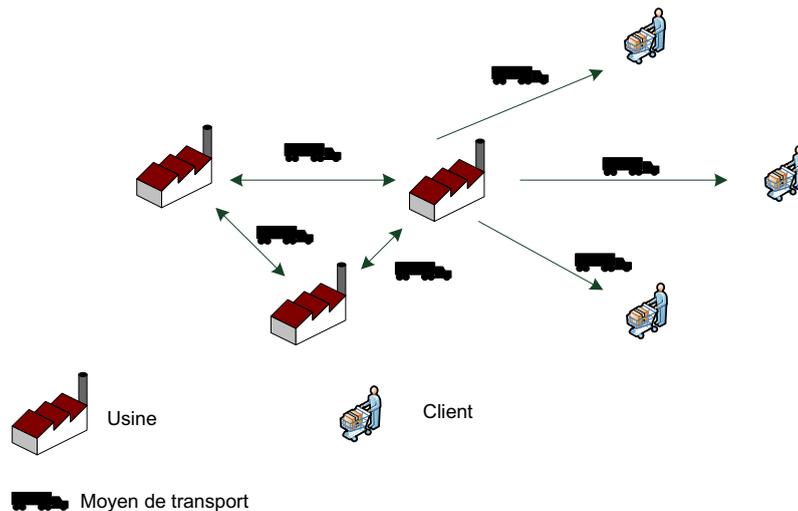


Figure 2.37 – La chaîne logistique
(Thierry 2003)

Dans ce modèle, les coûts de transport entre les usines et les clients sont pris en compte. De prime abord, ce cas de figure ne semble pas envisagé dans le modèle que nous proposons : en effet, seuls les transports inter-usines sont considérés. Cependant, il est tout à fait possible de modéliser cela : pour ce faire, il suffit d'associer bijectivement à chaque client c une usine fictive u_c possédant les propriétés suivantes :

- cette usine aura une capacité de production nulle,
- son stock « post » ne pourra livrer aucun client et ne pourra être livré par aucun autre stock,
- son stock « ante » sera le seul stock de la chaîne logistique à pouvoir livrer le client auquel il est rattaché et ne pourra livrer aucun autre stock. De plus, il devra recevoir pour chaque période t exactement la demande de ce client : on va donc lui affecter une capacité nulle.

Par facilité d'écriture, nous notons C l'ensemble des usines fictives et $U \cup C$ l'ensemble de toutes les usines : cet ensemble comprend donc les usines fictives. En utilisant cet artifice, on pourra donc imputer des coûts de transport entre les stocks de la chaîne logistique et ses clients. La figure 2.38 schématise cette association.

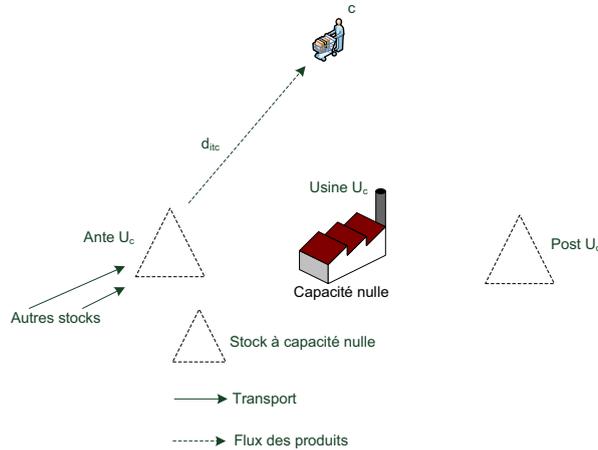


Figure 2.38 – Association usine fictive, client

Intéressons-nous aux stocks des usines (fictives ou non) de la chaîne logistique. A l'instar du ML-CLSP, les backorders et la demande perdue sont interdits, il n'y a pas de stock de sécurité et nous n'avons qu'un seul stock par usine. En utilisant le même paramétrage que pour le MLCLSP, nous obtenons la contrainte suivante, valable pour les usines de la chaîne logistique et les usines fictives :

$$\begin{aligned}
 IA_{iut}^+ &= IA_{iu(t-1)}^+ + Q_{iut} - \sum_{j=1}^N a_{ij} Q_{jut} - \sum_{c=1}^C DSIA_{iutc} \\
 &\quad + \sum_{\substack{u'=1 \\ u' \neq u}}^U \sum_{\Lambda \in \{IA, IP\}} T \Lambda IA_{iu'u(t-t_{iu'u})} \\
 &\quad - \sum_{\substack{u'=1 \\ u' \neq u}}^U \sum_{\Lambda \in \{IA, IP\}} T \Lambda \Lambda_{iuu't}
 \end{aligned} \tag{2.53}$$

$$\forall (i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \cup C \rrbracket \times \llbracket 1, T \rrbracket$$

Maintenant, intéressons-nous uniquement aux usines non fictives de la chaîne logistique.

Seuls les stocks « ante » des usines fictives peuvent livrer des clients, nous pouvons écrire :

$$DSIA_{iutc}^{max} = 0 \quad \forall (i, u, t, c) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, C \rrbracket \tag{2.54}$$

La contrainte (2.18) alliée à la relation (2.54) donne donc :

$$DSIA_{iutc} = 0 \quad \forall (i, u, t, c) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, C \rrbracket$$

Nous interdisons les stocks « post » de livrer tout autre stock que le stock « ante » de l'usine à laquelle il est rattaché. Cela va s'écrire :

$$CapaTIPIA_{u't}^{Ov} = CapaTIPIA_{u't} = 0 \tag{2.55}$$

$$\forall (u, u', t) \in \llbracket 1, U \cup C \rrbracket \times \llbracket 1, U \cup C - \{u\} \rrbracket \times \llbracket 1, T \rrbracket$$

Nous interdisons les stocks « ante » de livrer tout stock « post ». Ainsi :

$$CapaTIAIP_{uu't}^{Ov} = CapaTIAIP_{uu't} = 0 \quad (2.56)$$

$$\forall (u, u', t) \in \llbracket 1, U \cup C \rrbracket \times \llbracket 1, U \cup C \rrbracket \times \llbracket 1, T \rrbracket$$

Ensuite, nous interdisons les stocks « ante » des usines fictives de livrer un quelconque stock. Cela nous donne :

$$CapaTIA\Lambda_{uu't}^{Ov} = CapaTIA\Lambda_{uu't} = 0 \quad (2.57)$$

$$\forall (\Lambda, u, u', t) \in \{IA, IP\} \times \llbracket 1, C \rrbracket \times \llbracket 1, U \cup C \rrbracket \times \llbracket 1, T \rrbracket$$

Les relations (2.55), (2.56) et (2.57) impliquent via les contraintes (2.22) et (2.23), que

$$\begin{aligned} TIPIA_{uu't} &= 0 & \forall (u, u', t) \in \llbracket 1, U \cup C \rrbracket \times \llbracket 1, U \cup C - \{u\} \rrbracket \times \llbracket 1, T \rrbracket \\ TIAIP_{uu't} &= 0 & \forall (u, u', t) \in \llbracket 1, U \cup C \rrbracket \times \llbracket 1, U \cup C \rrbracket \times \llbracket 1, T \rrbracket \\ TIA\Lambda_{uu't} &= 0 & \forall (\Lambda, u, u', t) \in \{IA, IP\} \times \llbracket 1, C \rrbracket \times \llbracket 1, U \cup C \rrbracket \times \llbracket 1, T \rrbracket \end{aligned}$$

La contrainte (2.53) va donc s'écrire :

$$\begin{aligned} IA_{iut}^+ &= IA_{iu(t-1)}^+ + Q_{iut} - \sum_{j=1}^N a_{ij} Q_{jut} + \sum_{\substack{u'=1 \\ u' \neq u}}^U TIAIA_{iu'u(t-lt_{iu'u})} \\ &\quad - \sum_{\substack{u' \in U \cup C \\ u' \neq u}} TIAIA_{iuu't} \end{aligned} \quad (2.58)$$

$$\forall (i, u, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket$$

Nous reconnaissons la contrainte (2.50).

Intéressons-nous maintenant aux usines fictives.

En reprenant la contrainte (2.53) et en utilisant les relations précédentes, on obtient directement :

$$\begin{aligned} IA_{iu_{ct}}^+ &= IA_{iu_c(t-1)}^+ + Q_{iu_{ct}} - \sum_{j=1}^N a_{ij} Q_{ju_{ct}} - \sum_{c'=1}^C DSIA_{iu_{ct}c'} \\ &\quad + \sum_{\substack{u'=1 \\ u' \neq u}}^U TIAIA_{iu'u_c(t-lt_{iu'u_c})} \end{aligned} \quad (2.59)$$

$$\forall (i, u_c, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, C \rrbracket \times \llbracket 1, T \rrbracket$$

Chaque usine fictive est mono-ressource (la ressource sera noté k) et a une capacité de production nulle, nous obtenons donc :

$$CapaR_{u_c kt}^{Ov} = CapaR_{u_c kt} = 0 \quad \forall (u_c, t) \in \llbracket 1, C \rrbracket \times \llbracket 1, T \rrbracket \quad (2.60)$$

En utilisant les contraintes (2.1) et (2.2), nous obtenons :

$$Q_{iu_{ct}} = 0 \quad \forall (i, u_c, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, C \rrbracket \times \llbracket 1, T \rrbracket$$

De plus, les stocks de l'usine sont à capacité nulle. Cela donne :

$$CapaSIA_{iu_{ct}}^+ = CapaSIA_{iu_{ct}}^{Ov} = 0 \quad \forall (i, u_c, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, C \rrbracket \times \llbracket 1, T \rrbracket \quad (2.61)$$

En alliant cette relation à la contrainte (2.9), nous avons :

$$IA_{iu_c t}^+ = 0 \quad \forall (i, u_c, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, C \rrbracket \times \llbracket 1, T \rrbracket$$

Le stock « ante » de l'usine u_c ne peut servir que le client c , nous avons donc :

$$DSIA_{iu_c t c'}^{max} = 0 \quad \forall (i, u_c, t, c') \in \llbracket 1, N \rrbracket \times \llbracket 1, C \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, C - \{c\} \rrbracket \quad (2.62)$$

La contrainte (2.18) alliée à la relation (2.62) donne donc :

$$DSIA_{iu_c t c'} = 0 \quad \forall (i, u_c, t, c') \in \llbracket 1, N \rrbracket \times \llbracket 1, U \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, C - \{c\} \rrbracket$$

En utilisant la contrainte (2.17) nous obtenons :

$$DSIA_{iu_c t c} = D_{itc} \quad \forall (i, u_c, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, C \rrbracket \times \llbracket 1, T \rrbracket$$

et c est le client associé à l'usine fictive u_c .

La contrainte (2.59) s'écrit donc :

$$\sum_{\substack{u'=1 \\ u' \neq u_c}}^U TIAIA_{iu' u_c (t - t_{iu' u_c})} = D_{itc} \quad (2.63)$$

$\forall (i, u_c, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, C \rrbracket \times \llbracket 1, T \rrbracket$, c est le client associé à l'usine fictive u_c . On reconnaît la contrainte (2.51)

Dans le modèle de (Thierry 2003), les transports sont à capacité infinie mais, cependant, il est possible d'interdire le transport entre deux usines ou entre une usine et un client : pour cela, il suffit de ne pas créer la « liaison » adéquate. Dans le modèle que nous proposons, il suffit de fixer la capacité de la ressource de transport à 0 si on ne veut pas créer la liaison correspondante, sinon on la fixe à $+\infty$.

La contrainte (2.52) s'obtient immédiatement de la contrainte (2.1) : en effet, les usines sont mono-ressource, les temps de reconfiguration de la ressource ne sont pas pris en compte et il n'y a pas de capacité de production supplémentaire.

Quant à la fonction objectif ne sont pris en compte que les coûts de stockage, les coûts de production et les coûts de reconfiguration : il suffit donc de fixer les autres coûts à 0.

Nous venons donc d'instancier le modèle générique pour retrouver le modèle proposé par (Thierry 2003).

Conclusion

Dans ce chapitre, nous avons présenté un modèle de connaissance générique pour le domaine de la planification tactique des chaînes logistiques dont nous rappelons la définition donnée par (Lee et Billington 1993) : « un réseau de sites de production et de sites de distribution qui se procurent des matières premières, les transforment et les distribuent au consommateur ». Ce modèle permet d'obtenir facilement un modèle mathématique qui, comme nous l'avons montré, est également générique et permet de répondre aux problématiques du Master Planning, PIC, PDP et MRP. Nous présentons une synthèse du travail effectué dans ce chapitre sous forme de 3 tableaux qui comparent le modèle générique et les modèles multi-site de la littérature :

- le premier concerne les différentes hypothèses de modélisation,
- le second s'intéresse aux différents coûts pris en compte,
- le troisième récapitule les différentes contraintes modélisées.

En complétant le tableau 1.1 et les notations de la figure 2.34, nous obtenons le tableau 2.2 qui situe le modèle de planification proposé par rapport aux modèles multi-site de la littérature.

Hypothèses	[1]	[2]	[3]	[4]	[5]	[6]
Approvisionnement			X		X	X
Production	X	X	X	X	X	X
Distribution	X	X	X	X		X
Backorder					X	X
Annulation de commande		X				X
Stock de sécurité		X	X			X
Stock à Capacité finie			X			X
Mono-Ressource	X		X	X	X	X
Multi-Ressource		X				X
Ressource à capacité finie	X	X	X	X	X	X
Lead Time de production					X	X
Temps de reconfiguration					X	X
Nomenclature	Générale	Générale	Générale	Générale	Série	Générale
Capacité de transport	infinie	infinie	finie	infinie	infinie	finie
Lead Time de transport	X		X			X
Lead Time d'approvisionnement					X	X

Tableau 2.2 – Le modèle générique proposé et les modèles de la littérature

Nous constatons à la lecture du tableau 2.2 que le modèle générique que nous proposons permet de prendre en compte toutes les hypothèses rencontrées dans la littérature. Le tableau 2.3 reprend les différents coûts présents dans la fonction objectif du modèle mathématique générique et montre comment ceux-ci ont été pris en compte dans les modèles multi-site de la littérature.

	Modèle générique	[1]	[2]	[3]	[4]	[5]
Fonction Objectif	coûts liés à la production					
	coût de fabrication	X	X	X	X	X
	coût de reconfiguration				X	X
	coût fixe d'utilisation		X			
	coût d'utilisation de capacités supplémentaires					
	coûts liés au stockage					
	coût de stockage	X	X	X	X	X
	coût de backorder					X
	coût d'utilisation de stocks supplémentaires					
	coût d'utilisation de stocks de sécurité		X			
	coût fixe d'utilisation de stocks					
	coût de demande perdue		X			
	coûts liés à l'approvisionnement					
	coût d'approvisionnement			X		X
	coûts liés au transport					
	coût de transports	X	X	X	X	X
coût fixe d'utilisation des transports						
coût d'utilisation de capacités de transport supplémentaires						

Tableau 2.3 – Prise en compte des différents coûts dans les modèles multi-site de la littérature

Le tableau 2.4 reprend les différentes contraintes du modèle mathématique générique et montre celles qui ont été prises en compte dans les modèles multi-site de la littérature.

	Modèle générique	[1]	[2]	[3]	[4]	[5]
Contraintes	contraintes liées aux ressources de production					
	(2.1) : capacité de la ressource	X	X	X	X	X
	(2.2) : capacité supplémentaire maximale de la ressource					
	(2.3) : capacité supplémentaire maximale pour la chaîne log.					
	(2.4) : présence des ressources pour la fabrication désirée					
	(2.5) : configuration de la ressource				X	X
	(2.6) : utilisation de la ressource					
	contraintes liées aux ressources de stockage					
	(2.7) : équilibre des stocks «amont»	X	X	X	X	X
	(2.8) : équilibre des stocks «aval»					
	(2.9) : capacité du stock			X		
	(2.10) : capacité supplémentaire maximale pour le stock					
	(2.11) : capacité supplémentaire maximale de stockage pour la chaîne log.					
	(2.12) : utilisation du stock de sécurité		X	X		
	(2.13) : backorder maximal pour le stock					
	(2.14) : backorder maximal pour la chaîne log.					
	(2.15) : demande perdue maximale pour le stock					
	(2.16) : demande perdue maximale pour la chaîne log.					
	(2.17) : répartition de la demande sur les stocks	X				
	(2.18) : demande à servir maximale pour le stock					
	(2.19) : quantité livrée maximale pour un stock					
	(2.20) : quantité livrée maximale pour la chaîne log.					
	(2.21) : utilisation du stock					
	contraintes liées aux ressources de transport					
	(2.22) : capacité pour la ressource de transport			X		
	(2.23) : capacité supplémentaire maximale pour la ressource de transport					
(2.24) : capacité supplémentaire maximale de transport pour la chaîne log.						
(2.25) : configuration de la ressource de transport						
(2.26) : utilisation de la ressource						

Tableau 2.4 – Prise en compte des différentes contraintes dans les modèles multi-site de la littérature

Il convient de remarquer que l'objectif initial de ce modèle se limitait à la planification de la production et des transports au sein de la chaîne logistique interne. Cependant, en adoptant l'artifice utilisé pour retrouver le modèle de (Thierry 2003) par instantiation du modèle générique (partie IV.3), nous avons montré qu'il était possible d'intégrer la planification des moyens de transport pour toute la chaîne logistique (entre les fournisseurs et les usines d'une part, et entre les usines et les centres de distribution d'autre part).

Dans ce chapitre, nous avons procédé en traduisant mathématiquement le modèle de connaissance générique en modèle mathématique générique puis en instanciant ce dernier pour retrouver les différents modèles de la littérature. Cependant, on aurait pu instancier directement le modèle générique de connaissance pour obtenir ces derniers. De plus, ce modèle générique de connaissance peut être utilisé pour d'autres types de modèle (simulation, algorithmique etc.). C'est ce qu'illustre la figure 2.39

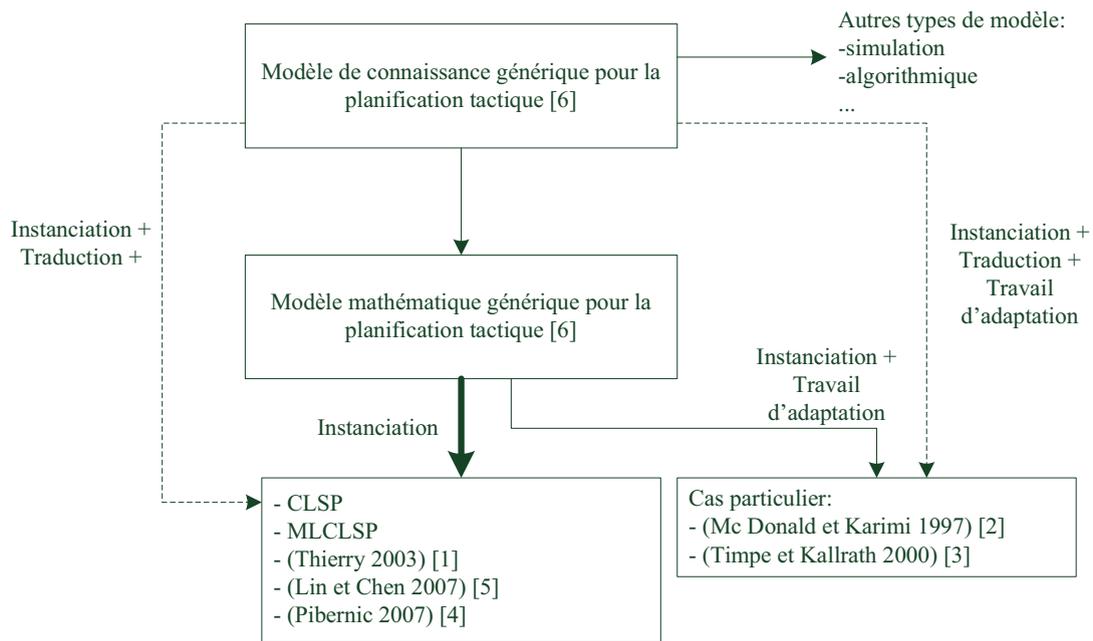


Figure 2.39 – Les possibilités d’instanciation du modèle générique de connaissance

Outre les modèles multi-site de la littérature, nous retrouvons, par instanciation de ce modèle générique, les modèles de référence pour la planification tactique mono-site (CLSP et MLCLSP) qui feront l’objet des deux chapitres suivants. Ce modèle générique sera également utilisé dans le chapitre IV pour déduire un modèle mathématique générique d’une chaîne logistique de type Flow-Shop Hybride.

Chapitre 3

Etat de l'art et proposition de méthodes pour le Capacitated Lot Sizing Problem

Sommaire

Introduction	111
I Le modèle mathématique	111
1 Le modèle de base	111
2 La reformulation en problème de plus court chemin	112
3 Le modèle approximatif de Manne	114
4 Les extensions du CLSP	115
4.1 Le CLSP avec Backlogging	115
4.2 Le CLSP avec Overtime	115
4.3 Le CLSP avec Setup Time	116
4.4 Le CLSP à capacité infinie	116
II Etat de l'art sur les méthodes de résolution	117
1 Calcul de bornes inférieures	117
1.1 Relaxation linéaire	117
1.2 Relaxation lagrangienne	118
2 Méthodes pour le modèle du CLSP	119
2.1 Les méthodes exactes	119
2.2 Les méthodes approchées	120
3 Méthodes pour le modèle du CLSP avec Setup Time	123
3.1 Les méthodes exactes	123
3.2 Les méthodes approchées	124
4 Conclusion sur l'état de l'art	128
III Proposition de méthodes pour le CLSP	128
1 Calcul d'une borne inférieure	129
1.1 Méthode de descente par gradient	129
1.2 Méthode de sous-gradient	131
2 Chaînage de deux métaheuristiques	132
2.1 La métaheuristique « optimisation »	134
2.2 La métaheuristique « solution initiale »	144
3 La Métaheuristique « bi-objectif »	147
4 Mise en œuvre et résultats	148
4.1 Comparaison des deux méthodes de calcul de borne inférieure	148
4.2 Mise en œuvre et résultats pour le chaînage de métaheuristique	149
4.3 Mise en œuvre et résultats pour la métaheuristique « bi-objectif »	150
4.4 Tests sur les instances de (Eppen et Martin 1987)	151

4.5	Conclusion	154
IV	Adaptation au CLSP avec Setup Time	154
1	Prise en compte des temps de lancements	154
1.1	Le calcul de la borne inférieure	155
1.2	La fonction bi-objectif	156
1.3	Le système de voisinage	156
2	Mise en œuvre et résultats	157
	Conclusion	158

Introduction

Au début de nos travaux, nous nous sommes intéressés à des problématiques de planification mono-niveau : l'élaboration du PIC et du PDP. Dans le premier chapitre, nous avons vu qu'il existait un modèle de base pour l'élaboration de ces deux plans : le « Capacitated Lot-Sizing Problem » (Drexel et Kimms 1997). D'après notre classification (Figure 1.13), celui-ci est un modèle mono-niveau, « big bucket », à demande variable et à plusieurs produits. C'est également un modèle mono-ressource. Il s'obtient très facilement grâce au modèle générique en faisant la remarque suivante : le CLSP est un MLCLSP mono-ressource dont la matrice Gozinto est la matrice nulle (on ne fabrique que des produits finis).

Dans ce chapitre, nous commençons par présenter le modèle du CLSP ainsi que ses principales extensions. Parmi celles-ci, la prise en compte de temps de reconfiguration de l'atelier (« setup time ») tient une place tout à fait à part puisqu'elle induit une difficulté supplémentaire liée à l'obtention d'une solution réalisable. Ensuite, nous donnons un état de l'art des méthodes de résolution rencontrées dans la littérature : un extrait de celui-ci a été publié dans (Comelli et al. 2008). La section suivante est consacrée à la présentation de différentes méthodes d'optimisation. Celles-ci sont basées sur l'utilisation d'une métaheuristique de type recuit simulé. La première d'entre elles est en fait un chaînage de deux métaheuristicques dont la première vise à obtenir une solution réalisable au problème et la seconde à optimiser cette solution. La seconde méthode est une métaheuristique qui utilise une fonction bi-objectif permettant de s'affranchir des contraintes de capacité du système de production, assurant ainsi la connexité par arc de l'espace des solutions. Nous testons ces méthodes sur des instances de la littérature et forts des résultats obtenus, nous montrons comment adapter cette dernière au cas où les temps de reconfiguration (setup time) sont pris en compte.

I Le modèle mathématique

Dans ce paragraphe, nous présentons le modèle du CLSP ainsi que certaines de ces reformulations sur lesquelles se sont appuyées quelques méthodes de résolution.

1 Le modèle de base

Le CLSP est défini de la façon suivante :

Les données du modèle sont :

T	: Longueur (en périodes) de l'horizon de la planification,	
N	: Nombre de produits,	
C_t	: Capacité disponible à la période t ,	$t \in \llbracket 1, T \rrbracket$
d_{it}	: Demande en produit i à la fin de la période t ,	$i \in \llbracket 1, N \rrbracket, t \in \llbracket 1, T \rrbracket$
I_{i0}	: Stock initial en produit i ,	$i \in \llbracket 1, N \rrbracket$
p_i	: Capacité nécessaire pour produire une unité de produit i ,	$i \in \llbracket 1, N \rrbracket$
h_i	: Coût de stockage du produit i durant une période,	$i \in \llbracket 1, N \rrbracket$
s_i	: Coût de lancement du produit i .	$i \in \llbracket 1, N \rrbracket$
b_i	: Coût de production pour une unité de produit i .	$i \in \llbracket 1, N \rrbracket$

Les variables de décision du modèle sont :

$$\begin{aligned}
Q_{it} & : \text{Quantité de produit } i \text{ fabriquée à la période } t, & i \in \llbracket 1, N \rrbracket, t \in \llbracket 1, T \rrbracket \\
I_{it} & : \text{Niveau de stock en produit } i \text{ à la fin de la période } t, & i \in \llbracket 1, N \rrbracket, t \in \llbracket 1, T \rrbracket \\
X_{it} & : \begin{cases} 1 & \text{s'il y a lancement pour le produit } i \text{ à la période } t, \\ 0 & \text{sinon.} \end{cases} & i \in \llbracket 1, N \rrbracket, t \in \llbracket 1, T \rrbracket
\end{aligned}$$

En utilisant ces notations, on peut écrire le problème du CLSP sous forme d'un programme linéaire mixte :

$$\text{Minimiser } \sum_{i=1}^N \left[\sum_{t=1}^T (s_i X_{it} + b_i Q_{it} + h_i I_{it}) \right] \quad (3.1)$$

Sous les contraintes :

$$I_{it} = I_{i(t-1)} + Q_{it} - d_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.2)$$

$$p_i Q_{it} \leq C_t X_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.3)$$

$$\sum_{i=1}^N p_i Q_{it} \leq C_t \quad \forall t \in \llbracket 1, T \rrbracket \quad (3.4)$$

$$X_{it} \in \{0, 1\} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.5)$$

$$I_{it}, Q_{it} \geq 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.6)$$

- L'objectif (3.1) est de minimiser la somme des coûts de lancement, de stockage et de production de tous les produits pour toutes les périodes.
- La contrainte (3.2) représente l'équilibre des stocks.
- La contrainte (3.3) impose qu'un lancement pour le produit i soit effectué à la période t si on a une fabrication de celui-ci durant cette période.
- La contrainte (3.4) est une contrainte de capacité.
- La contrainte (3.5) définit les variables de lancement comme étant binaires et (3.6) est une condition de non-négativité.

Si on se réfère au principe des trois phases décrit dans l'approche MRPII (on commence par planifier les produits niveau par niveau en ignorant les contraintes de capacité, ensuite on tient compte des contraintes capacitaires en décalant et pour finir on séquence la production) : ce modèle répond donc aux problématiques relatives aux phases I et II.

2 La reformulation en problème de plus court chemin

(Eppen et Martin 1987) ont proposé une reformulation du problème du CLSP en généralisant la reformulation en problème de plus court chemin du CLSP à capacité infinie utilisée dans (Wagner et Whitin 1958) pour leur résolution par programmation dynamique.

La formulation utilisée par (Wagner et Whitin 1958) s'appuie sur un graphe où chaque nœud correspond au début d'une période et où chaque arc entre chaque paire de sommets (t, τ) représente la décision de produire durant la période t la quantité nécessaire pour satisfaire l'intégralité de la demande entre la période t et la période τ (figure 3.1) (on utilise la propriété zéro-stock). Si on affecte un coût de production à cet arc, on obtient une formulation en problème de plus court chemin qui permettra d'obtenir la solution optimale.

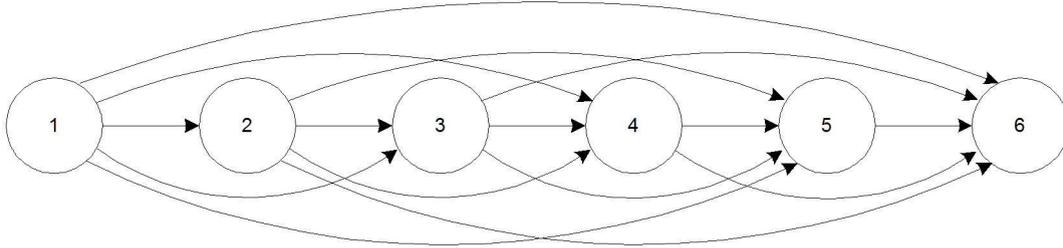


Figure 3.1 – Graphe pour un problème de « lot-sizing » à 6 périodes pour 1 produit

En reprenant les notations du CLSP, auxquelles on ajoute :

- $q_{it\tau}$: variable binaire qui vaut 1 si on prend la décision de produire l'intégralité de la demande
: comprise entre la période t et τ à la période t , 0 sinon
- $f_{it\tau}$: représente les coûts de production et de stockage engendrés par la décision de fabriquer
le produit i pour satisfaire la demande de la période t jusqu'à la période τ comprise.

Ce modèle s'écrit donc :

$$\text{Minimiser } \sum_{i=1}^N \sum_{t=1}^T \sum_{\tau=t}^T f_{it\tau} q_{it\tau} + \sum_{i=1}^N \sum_{t=1}^T s_i X_{it} \quad (3.7)$$

Sous les contraintes :

$$\sum_{\tau=1}^T q_{i1\tau} = 1 \quad \forall i \in \llbracket 1, N \rrbracket \quad (3.8)$$

$$-\sum_{\tau=1}^{t-1} q_{i\tau(t-1)} + \sum_{\tau=t}^T q_{it\tau} = 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 2, T \rrbracket \quad (3.9)$$

$$\sum_{\tau=t}^T q_{it\tau} \leq X_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.10)$$

$$q_{it\tau} \in \{0, 1\} \quad \forall (i, t, \tau) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, T \rrbracket \quad (3.11)$$

$$X_{it} \in \{0, 1\} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.12)$$

- L'objectif (3.7) est de minimiser la somme des coûts de production, de lancement et de stockage de tous les produits pour toutes les périodes.
- Les contraintes (3.8) indiquent qu'il ne peut y avoir plus d'un arc emprunté à partir de la première période.
- Les contraintes (3.9) sont les contraintes de conservation des flux
- Les contraintes (3.10) sont les contraintes de lancement.
- Les contraintes (3.11) et (3.12) sont des contraintes d'intégrité.

La propriété zéro-stock n'étant pas valide dans le cas du CLSP, la formulation de (Eppen et Martin 1987) relaxe la contrainte d'intégrité de la variable $q_{it\tau}$ ($0 \leq q_{it\tau} \leq 1$) et celle-ci s'interprète alors comme le pourcentage de la demande en produit i de la période t à la période τ qui sera produite durant la période t . La contrainte (3.11) devient :

$$0 \leq q_{it\tau} \leq 1 \quad \forall (i, t, \tau) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, T \rrbracket \quad (3.13)$$

Pour tenir compte de la capacité, la contrainte 3.14 est rajoutée :

$$\sum_{i=1}^N \sum_{\tau=t}^T (p_i d_{it\tau} q_{it\tau}) \leq C_t \quad \forall t \in \llbracket 1, T \rrbracket \quad (3.14)$$

où $d_{it\tau}$ est égale à la somme de la demande en produit i de la période t à la période τ . Comme nous le verrons dans la section dédiée au calcul de la borne inférieure, l'intérêt de ce modèle réside dans le fait que sa relaxation linéaire fournit une borne inférieure comparable à la relaxation lagrangienne des contraintes de capacités du CLSP (3.4).

3 Le modèle approximatif de Manne

(Manne 1958) a proposé une autre formulation pour le problème du CLSP. Cette formulation est dite approximative dans le sens où, pour chaque produit i , on ne considère que les séquences de production (Q_{i1}, \dots, Q_{iT}) qui satisfont la propriété du zéro-stock (Wagner Within), c'est-à-dire qu'on ne pourra produire à une période donnée que si le stock à la fin de la période précédente est nul. De tels plans de production sont appelés des *plans dominants*. Pour les problèmes de lot-sizing sans capacité, il existe une solution optimale qui possède cette propriété (la propriété de WW (Wagner et Whitin 1958)). *Mais ceci n'est pas nécessairement vrai pour le CLSP.*

Les données de base du problème restent les mêmes que pour le CLSP mais on va réécrire le problème différemment en considérant l'ensemble des plans dominants, de cardinal K :

Soient $X_{it}^{(k)}$, $Q_{it}^{(k)}$ et $I_{it}^{(k)}$ les variables de lancement, de production et de stock à la période t du plan dominant k pour le produit i . Soit $f_i^{(k)}$ le coût total du plan dominant k pour le produit i , c'est-à-dire :

$$f_i^{(k)} = \sum_{t=1}^T (s_i X_{it}^{(k)} + b_i Q_{it}^{(k)} + h_i I_{it}^{(k)}) \quad \forall (i, k) \in \llbracket 1, N \rrbracket \times \llbracket 1, K \rrbracket$$

Soit $\alpha_{it}^{(k)}$ la consommation de la ressource à la période t dans le plan k pour le produit i :

$$\alpha_{it}^{(k)} = p_i Q_{it}^{(k)} \quad \forall (i, t, k) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \times \llbracket 1, K \rrbracket$$

Soit $\theta_i^{(k)}$ la variable de décision de notre modèle. $\theta_i^{(k)}$ vaut 1 si nous avons choisi le k^{eme} plan dominant de production pour le produit i , et 0 sinon.

$$\text{Minimiser } \sum_{i=1}^N \left(\sum_k f_i^{(k)} \theta_i^{(k)} \right) \quad (3.15)$$

Sous les contraintes :

$$\sum_{i=1}^N \sum_{k=1}^K \alpha_{it}^{(k)} \theta_i^{(k)} \leq C_t \quad \forall t \in \llbracket 1, T \rrbracket \quad (3.16)$$

$$\sum_{k=1}^K \theta_i^{(k)} = 1 \quad \forall i \in \llbracket 1, N \rrbracket \quad (3.17)$$

$$\theta_i^{(k)} \in \{0, 1\} \quad \forall (i, k) \in \llbracket 1, N \rrbracket \times \llbracket 1, K \rrbracket \quad (3.18)$$

- L'objectif (3.15) est de minimiser la somme des coûts de lancement, de stockage et de production de tous les produits pour toutes les périodes au travers de la minimisation des coûts des plans dominants pour chaque produit.

- La contrainte (3.16) représente la contrainte de capacité : elle nous dit que, pour une période t , la somme des capacités requises par les plans dominants de chaque produit doit être inférieure à la capacité globale du système.
- La contrainte (3.17) implique qu'un seul plan de production est choisi pour le produit i .
- La contrainte (3.18) impose à la variable de décision $\theta_i^{(k)}$ d'être binaire.

(Bitran et Matsuo 1986) ont aussi montré que l'ensemble des solutions optimales de la formulation approximative de Manne ne fournit qu'un sous-ensemble des solutions du CLSP : en effet, une solution optimale pour le CLSP ne satisfait pas nécessairement à la propriété du zéro-stock.

Nous venons d'exposer le modèle de base du CLSP ainsi que certaines de ses reformulations. Mais il existe de nombreuses extensions à celui-ci. Ces dernières permettent de répondre à des problématiques plus spécifiques comme nous allons le voir dans le prochain paragraphe.

4 Les extensions du CLSP

Le CLSP a été conçu pour répondre à la problématique de l'élaboration de PDP sous contraintes. Or, de par sa conception (obligation de répondre à la demande sans retard, capacité fixe...), il ne permet pas de modéliser une certaine flexibilité au sein du système de production. C'est pour cela que de nombreuses extensions ont été proposées dans la littérature.

4.1 Le CLSP avec Backlogging

Le modèle initial du CLSP impose que la demande soit toujours satisfaite à temps. Or cette approche peut être remise en cause essentiellement pour deux raisons :

1. Il n'existe pas nécessairement de planning de production solution de ce problème (en particulier si nous sommes dans le cas d'un système de production sous-capacitaire).
2. Il se peut également qu'il soit plus économique de ne pas satisfaire toute la demande client (pour éviter, par exemple, des coûts de configuration du système de production).

Certains auteurs (Zangwill 1969) ont considéré les ruptures de stock (pénuries). Dans ce cas, un coût de rupture (h_i^-) est engendré pour chaque unité de demande de produit i non satisfaite avant sa période échue. Si on note par I_{it}^- la quantité de produit i en retard à la fin de la période t alors la contrainte (3.2) devient :

$$I_{it} - I_{it}^- = I_{i(t-1)} - I_{i(t-1)}^- + Q_{it} - d_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.19)$$

Les contraintes de non-négativité (3.6) intègrent la nouvelle variable I_{it}^- :

$$I_{it}, I_{it}^-, Q_{it}, \geq 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.20)$$

et la fonction objectif (3.1) est modifiée afin de prendre en compte les coûts de pénurie :

$$\text{Minimiser } \sum_{t=1}^T \left[\sum_{i=1}^N (s_i X_{it} + b_i Q_{it} + h_i I_{it} + h_i^- I_{it}^-) \right] \quad (3.21)$$

4.2 Le CLSP avec Overtime

Plutôt que de permettre une baisse de la satisfaction client (retard), certains gestionnaires préfèrent donner une plus grande flexibilité au système de production en permettant l'accroissement temporaire de ses capacités au moyen, par exemple, d'heures supplémentaires. Ceci a été modélisé en modifiant les contraintes de capacité (3.4) comme suit :

$$\sum_{i=1}^N p_i Q_{it} \leq C_t + O_t \quad \forall t \in \llbracket 1, T \rrbracket \quad (3.22)$$

où O_t représente l'accroissement temporaire de la capacité. Naturellement, faire appel à des heures supplémentaires, à de la soustraction ... engendre un surcoût de production. La fonction objectif (3.1) devient :

$$\text{Minimiser } \sum_{t=1}^T \left[\sum_{i=1}^N (s_i X_{it} + b_i Q_{it} + h_i I_{it}) + m_t O_t \right] \quad (3.23)$$

où m_t représente le coût unitaire de l'accroissement de la capacité.

Bien évidemment, les contraintes de non-négativité (3.6) intègrent la nouvelle variable O_t :

$$I_{it}, Q_{it}, O_t \geq 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.24)$$

4.3 Le CLSP avec Setup Time

Dans certains cas industriels, le temps de lancement d'une campagne de production peut être important. C'est pourquoi il est nécessaire de modéliser ce temps par une perte de capacité. Ainsi, dans la littérature, plusieurs auteurs considèrent des temps de lancement (setup time). Ils les intègrent en les assimilant à une consommation de capacité. La contrainte (3.4) s'exprimera donc de la manière suivante :

$$\sum_{i=1}^N (\tau_{it} X_{it} + p_i Q_{it}) \leq C_t \quad \forall t \in \llbracket 1, T \rrbracket \quad (3.25)$$

où τ_{it} représente le temps de lancement de la campagne de production modélisé sous forme d'une perte de capacité de production.

Il existe une autre extension très importante, non pas industriellement comme nous allons nous en rendre compte, mais par les applications en termes de méthodes de résolution qu'elle apporte.

4.4 Le CLSP à capacité infinie

Lorsqu'on considère le cas où la capacité est infinie : on parle alors du **Lot Sizing Problem** (LSP). On obtient une formalisation de celui-ci en relaxant la contrainte (3.4) et en modifiant légèrement (3.3) de la manière suivante : la capacité étant infinie, le paramètre p_i devient donc superflu et il est clair qu'on ne veut pas produire plus de produit i à la période t que la demande restant à satisfaire sur l'horizon T .

Donc, si on note $d_{itq} = \sum_{k=t}^q d_{ik}$ la demande à satisfaire entre les périodes t et q comprises, la contrainte (3.3) devient pour le LSP :

$$Q_{it} \leq d_{itT} X_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.26)$$

Ce modèle n'est pas d'une grande utilité industriellement car une capacité infinie est irréaliste mais il est utilisé pour élaborer des heuristiques pour la résolution du CLSP, ceci pour deux raisons :

- Il existe un algorithme polynomial (en $O(T^2)$) qui permet de trouver la solution optimale à ce problème (l'algorithme de Wagner Within (Wagner et Whitin 1958)). Cette solution vérifie la propriété du zéro-stock.

- La relaxation des contraintes de capacité du CLSP aboutit à ce problème.

Nous venons de présenter les principales extensions du CLSP. Le prochain paragraphe s'attache à montrer les principales approches de résolution rencontrées dans la littérature.

II Etat de l'art sur les méthodes de résolution

(Bitran et Yanasse 1982) ont prouvé que le problème du CLSP est \mathcal{NP} -difficile dans le cas où la capacité est finie. De plus, en considérant les temps de lancement (setup time), (Trigeiro et al. 1989) ont montré que le problème de faisabilité est \mathcal{NP} -complet.

De nombreuses approches de résolution ont été élaborées. Le problème étant \mathcal{NP} -difficile, les méthodes exactes se sont vite butées à la taille des instances, c'est pour cela qu'une large partie des recherches s'est tournée sur l'élaboration d'heuristiques et dans une moindre proportion sur l'utilisation de métaheuristiques.

Le calcul de bornes inférieures s'avère très utile dans le cas de la résolution d'un problème \mathcal{NP} -difficile pour plusieurs raisons dont les principales sont :

- elles sont utilisées dans des procédures énumératives (*Branch & Bound*, *Branch & Cut...*),
- elles permettent de mesurer la qualité des méthodes approchées.

Dans cette partie nous commençons par présenter les différentes relaxations de contraintes rencontrées dans la littérature pour ce problème en donnant les résultats obtenus en termes de qualité de borne inférieure. Puis, nous présentons un état de l'art des méthodes de résolution pour le CLSP dans lequel nous distinguons le CLSP avec setup time du fait de la complexité supplémentaire qu'il engendre.

1 Calcul de bornes inférieures

Différentes bornes inférieures pour le CLSP ont été obtenues en utilisant diverses relaxations.

1.1 Relaxation linéaire

La relaxation linéaire du CLSP est obtenue en relaxant la contrainte (3.5). Elle s'écrit donc :

$$0 \leq X_{it} \leq 1 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.27)$$

Il existe un résultat dû à (Chen et Thizy 1990) concernant la relaxation linéaire du problème :

Théorème 3.1 *En appelant Q^* la solution optimale du problème et Q_{Li}^* la solution du problème relaxé, alors :*

$$Q_{Li}^* \geq \sum_{i=1}^N s_i \geq N \times \min_{i \in \llbracket 1, N \rrbracket} s_i \quad \text{et} \quad Q^* - Q_{Li}^* \leq (T - 1) \times \sum_{i=1}^N s_i$$

(Chen et Thizy 1990) ont montré que généralement cette relaxation donne une borne inférieure d'assez mauvaise qualité et lui préfèrent la relaxation lagrangienne.

1.2 Relaxation lagrangienne

Dans ce paragraphe, nous noterons par $Q_{La-(eq)}^*$ la solution optimale obtenue par la relaxation lagrangienne des contraintes (eq).

Dans la littérature, on rencontre plusieurs relaxations lagrangiennes pour le CLSP dans lesquelles on dualise :

- les contraintes de lancement (3.3),
- les contraintes d'équilibre des stocks (3.2),
- les contraintes de capacité (3.4).

Là encore, (Chen et Thizy 1990) nous fournissent des résultats très intéressants :

Contraintes de lancement :

Théorème 3.2 *Dans le cas de la relaxation lagrangienne des contraintes de lancement (3.3), nous avons :*

$$Q_{Li}^* = Q_{La-3.3}^*$$

Ce théorème permet d'affirmer que la relaxation lagrangienne des contraintes de lancement ne nous fournira pas de résultats meilleurs qu'une relaxation linéaire.

Contraintes d'équilibre des stocks :

Théorème 3.3 *Dans le cas de la relaxation lagrangienne des contraintes d'équilibre des stocks (3.2), nous avons :*

$$Q_{Li}^* \leq Q_{La-3.2}^* \leq Q_{Li}^* + T \times \max_{i \in [1, N]} s_i$$

Ce théorème nous montre que la relaxation des contraintes d'équilibre des stocks nous fournira, en général, une meilleure borne que la relaxation linéaire du problème mais, en remarquant que :

$$\begin{aligned} Q_{La-3.2}^* &\leq Q_{Li}^* + T \times \max_{i \in [1, N]} s_i && \text{équivalent à} \\ Q_{La-3.2}^* - Q_{Li}^* &\leq T \times \max_{i \in [1, N]} s_i && \text{équivalent à} \quad (\text{en supposant } Q_{Li}^* \neq 0) \\ \frac{Q_{La-3.2}^* - Q_{Li}^*}{Q_{Li}^*} &\leq \frac{T \times \max_{i \in [1, N]} s_i}{Q_{Li}^*} && \text{et par le théorème 3.1 (en supposant } \min_{i \in [1, N]} s_i > 0) \\ \frac{Q_{La-3.2}^* - Q_{Li}^*}{Q_{Li}^*} &\leq \frac{T}{N} \times \frac{\max_{i \in [1, N]} s_i}{\min_{i \in [1, N]} s_i} \end{aligned}$$

Cela signifie que plus le nombre de produits sera grand, plus la relaxation lagrangienne des contraintes d'équilibre des stocks sera proche de la relaxation linéaire, donc sera de mauvaise qualité.

Contraintes de capacité :

Théorème 3.4 *Dans le cas de la relaxation lagrangienne des contraintes de capacité (3.4), nous obtenons :*

$$Q^* - Q_{La-3.4}^* \leq (T-1)^2 \times \max_{i \in [1, N]} s_i$$

En corollaire à ce théorème, (Bitran et Matsuo 1986) ont montré que :

$$\frac{Q^* - Q_{La-3.4}^*}{Q_{La-3.4}^*} \leq \frac{(T-1)^2}{N} \times \frac{\max_{i \in [1, N]} s_i}{\min_{i \in [1, N]} s_i}$$

ce qui montre que plus le nombre de produits est grand, plus la relaxation lagrangienne des contraintes de capacité fournit une borne d'excellente qualité. Autrement dit, cette borne inférieure tend asymptotiquement vers l'optimum lorsque le nombre de produits tend vers l'infini.

En ce qui concerne les relaxations pour la reformulation du modèle en plus court chemin, (Chen et Thizy 1990) énoncent le théorème suivant qui fournit, en conservant les notations précédentes, une classification des bornes inférieures obtenues en termes de qualité :

Théorème 3.5

$$Q_{Li}^* = Q_{La-3.3}^* \leq Q_{La-3.4}^* = (Q')_{La-3.8}^* = (Q')_{La-3.11}^* = (Q')_{Li}^* \leq (Q')_{La-3.9-3.10}^* \leq Q^* = (Q')^* \text{ et} \\ Q_{Li}^* = Q_{La-3.3}^* \leq Q_{La-3.2}^* \leq (Q')_{La-3.9-3.10}^* \leq Q^* = (Q')^*$$

où Q' se rapporte à la reformulation du problème en plus court chemin. Ce théorème nous montre que la reformulation du problème sous cette forme fournit de meilleures bornes inférieures que le modèle initial. C'est pour cela qu'elle est souvent utilisée dans des procédures de *Branch & Bound* pour l'élagage de l'arbre de recherche.

En ce qui concerne le modèle approximatif, (Manne 1958) a prouvé que sa relaxation linéaire admet une grande majorité de variables en 0-1 dans sa solution optimale si le nombre de produits est beaucoup plus élevé que le nombre de périodes dans l'horizon de planification, ce qui permet de donner une bonne solution approchée du problème original.

2 Méthodes pour le modèle du CLSP

Comme dit précédemment, deux grandes familles d'approches ont été développées pour le CLSP : les méthodes exactes et les méthodes approchées.

2.1 Les méthodes exactes

En général, les méthodes exactes pour le CLSP sont des méthodes énumératives mais elles se heurtent systématiquement à la taille des instances des problèmes. Pour résoudre ces problèmes, on peut trouver :

- des méthodes à base de *Branch & Bound* qui ont été développées par (Gelders et al. 1986) et (Hindi 1995),
- des méthodes à base de *Branch & Cut* qui ont été proposées par (Leung et al. 1989), (Pochet et Wolsey 1991), (Barany et al. 1984) et (Belvaux et Wolsey 2000),
- des méthodes à base de reformulation du modèle en plus court chemin : (Eppen et Martin 1987) utilisent une méthode à base de *Branch & Bound* pour résoudre ce problème.

On remarquera que (Eppen et Martin 1987) utilisent comme benchmark des instances de 200 produits sur 10 périodes. Ils arrivent à atteindre, en 171 minutes, un Gap (écart relatif) de 0.18% par rapport à la relaxation linéaire du problème qui est de très bonne qualité en vertu du Théorème 3.5.

Le CLSP étant \mathcal{NP} -difficile, beaucoup de méthodes approchées ont été développées pour ce problème.

2.2 Les méthodes approchées

Des heuristiques pour le problèmes du CSLP ont été développées par beaucoup d'auteurs. (Maes et Van Wassenhove 1988) ont proposé une classification des heuristiques en deux catégories :

- les heuristiques de bon sens,
- les heuristiques à base de programmation mathématique.

Les heuristiques de bon sens sont généralement des algorithmes gloutons. Elles abordent le problème de trois façons, soit période par période, soit de façon itérative, soit produit par produit :

- *période par période* : la construction du plan s'effectue séquentiellement de la période 1 à la période T . Elles sont basées sur le critère de Silver-Meal du coût moyen par période ((Lambrecht et Vanderveken 1979),(Dixon et Silver 1981), (Maes et Van Wassenhove 1986b)).
- *itérativement* : ces heuristiques partent d'une solution (non nécessairement réalisable) pour l'horizon total et génèrent des solutions réalisables grâce à de simples décalages. C'est ainsi que procède l'heuristique de (Dogramaci et al. 1981).
- *produit par produit* : A la première itération, un produit est planifié sur tout l'horizon. Aux itérations suivantes, un nouveau produit est planifié sur tout l'horizon en utilisant les capacités restantes et en utilisant des contraintes supplémentaires sur la production et le stockage pour assurer la réalisabilité du résultat. La procédure s'arrête lorsque tous les produits sont planifiés. C'est le principe de l'heuristique proposée par (Kirca et Kökten 1994).

(Maes et Van Wassenhove 1986a) ont comparé les heuristiques de (Dogramaci et al. 1981), (Dixon et Silver 1981), (Lambrecht et Vanderveken 1979) et ont montré que celle de (Dogramaci et al. 1981) était bien meilleure que les deux autres mais plus gourmande en temps CPU.

Les heuristiques à base de programmation mathématique ont été classées par (Salomon 1991) en trois catégories :

- les heuristiques à base de relaxation,
- les heuristiques à base de programmation linéaire,
- les heuristiques à base de génération de colonnes.

Suivant cette classification :

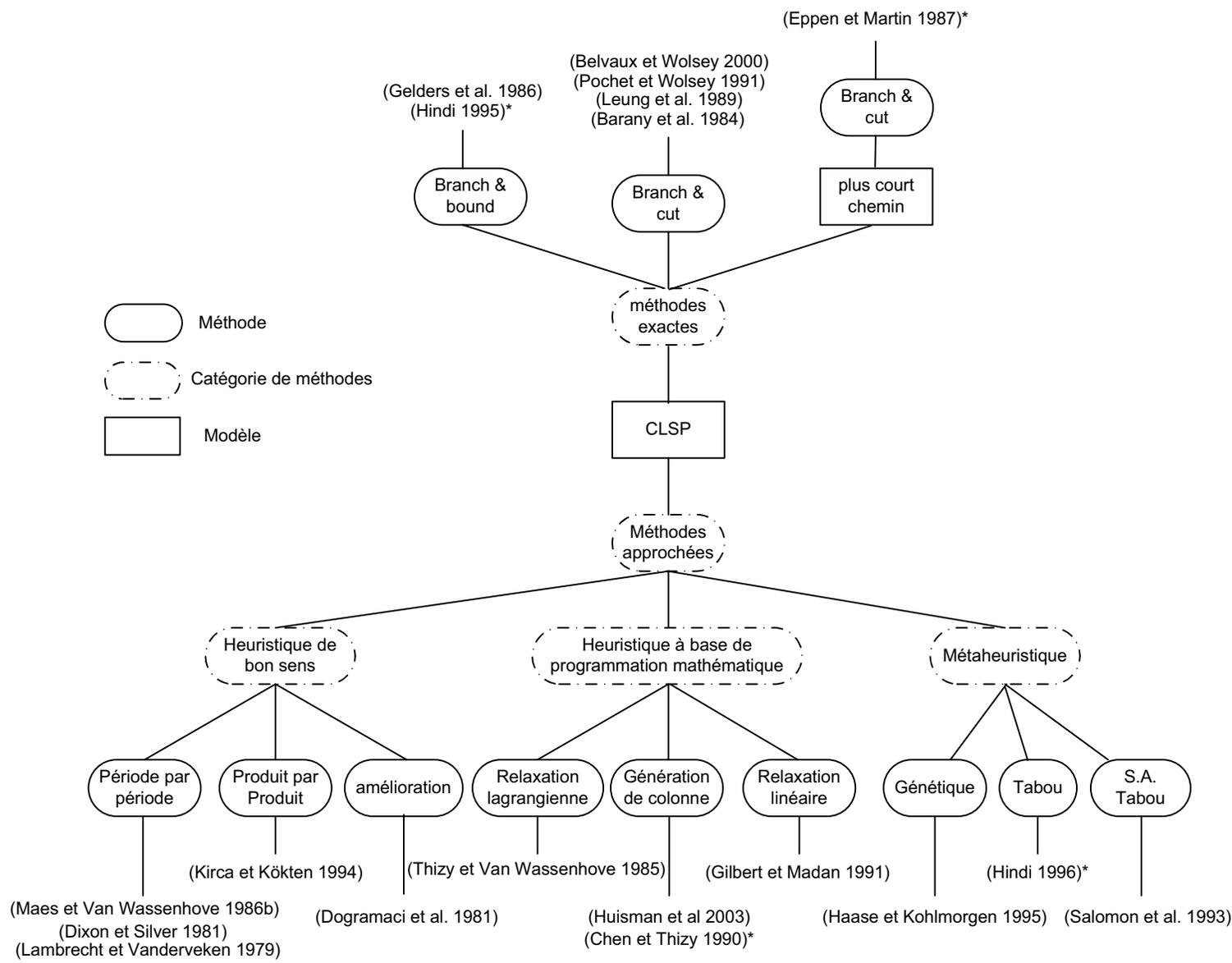
- Les heuristiques à base de relaxation s'appuient sur la relaxation des contraintes couplantes (les contraintes de capacités) : on les dualise dans la fonction objectif (relaxation lagrangienne). C'est la méthode qu'utilise l'heuristique (Thizy et Van Wassenhove 1985). Le problème initial se décompose alors en plusieurs problèmes sans capacité avec un seul produit qu'il peut résoudre avec l'algorithme de (Wagner et Whitin 1958),
- Les heuristiques à base de programmation linéaire sont basées sur des formulations alternatives du problème (formulation sous forme de problème de plus court chemin) dont les relaxations linéaires fournissent de meilleures bornes ou présentent une meilleure structure. Si les solutions obtenues ne sont pas réalisables, on les perturbe de différentes manières pour trouver des solutions réalisables (Gilbert et Madan 1991).
- Les heuristiques à base de génération de colonnes sont basées sur le modèle approximatif de Manne ((Chen et Thizy 1990),(Huisman et al. 2003)).

Les métaheuristiques ont été utilisées plus récemment pour résoudre ce problème :

- (Salomon et al. 1993) ont utilisé la recherche Tabou et le recuit simulé en codant leur solution grâce à un voisinage basé sur une matrice de setup (binaire) et en utilisant un algorithme glouton pour lisser la production sur le planning déterminé.
- (Haase et Kohlmorgen 1995) ont utilisé les algorithmes génétiques, ils génèrent leur population initiale grâce à une version modifiée de l'algorithme de Dixon-Silver (Dixon et Silver 1981) en

introduisant un caractère aléatoire au choix des produits à planifier pour une période.
 - (Hindi 1996) propose une recherche Tabou pour résoudre ce problème.

La figure 3.2 résume l'état de l'art présenté :



* References utilisant les benchmarks définis par (Eppen et Martin 1987)
 (200 produits X 10 périodes)

Figure 3.2 – Les méthodes de résolution pour le CLSP

Référence	Méthodes Exactes	Méthodes Approchées	Taille maximale des instances testées	Gap obtenu	Valeur de comparaison
(Hindi 1995)	<i>B&B</i>		200×10 *	0,36	(Eppen et Martin 1987)
(Eppen et Martin 1987)	<i>B&B</i>		200×10 *	0,18	BIP
(Gelders et al. 1986)	<i>B&B</i>		12×12	0,36	(Dogramaci et al. 1981)
(Pochet et Wolsey 1991)	<i>B&C</i>		30×10	0,90	LB
(Belvaux et Wolsey 2000)	<i>B&C</i>		20×10	0	BIP
(Barany et al. 1984)	<i>B&C</i>		20×13	0	BIP
(Leung et al. 1989)	<i>B&C</i>		8×8		
(Lambrecht et Vanderveken 1979)		période/période	4×6	2,64	BIP
(Dixon et Silver 1981)		période/période	20×13	×	×
(Maes et Van Wassenhove 1986b)		période/période	200×10		
(Dogramaci et al. 1981)		amélioration	10×48	1,04	BIP
(Kirca et Kökten 1994)		produit/produit	20×20	1,41	B.H.
(Thizy et Van Wassenhove 1985)		Relaxation Lagrangienne	8×8	0,17	BIP
(Gilbert et Madan 1991)		Relaxation linéaire	9×36	2,93	L.B
(Chen et Thizy 1990)		Gen. Colonnes	200×10 *	×	×
(Huisman et al. 2003)		Gen. Colonnes	×	×	×
(Salomon et al. 1993)		Recuit simulé, Tabou	×	×	×
(Haase et Kohlmorgen 1995)		Algorithmme génétique	12 × 12	1,44	(Kirca et Kökten 1994)
(Hindi 1996)		Tabou	200 × 10 *	0,23	(Eppen et Martin 1987)

Tableau 3.1 – Etat de l’art pour le CLSP

Légende :

* Instances de (Eppen et Martin 1987)

× Pas de résultat numérique donné.

Le tableau 3.1 synthétise les différents articles répertoriés. On y retrouve la référence de l'article, le type de méthode utilisée, la taille des instances traitées et le GAP trouvé calculé par rapport à la valeur de comparaison (BIP : meilleure solution trouvée à la date de parution de l'article, B.H. : solution trouvée par la meilleure heuristique, LB : borne inférieure).

3 Méthodes pour le modèle du CLSP avec Setup Time

Nous avons vu que le modèle du CLSP avec Setup Time comportait une difficulté supplémentaire dans le sens où le problème de faisabilité est \mathcal{NP} -complet (Trigeiro et al. 1989). Il n'est donc pas étonnant que peu de méthodes exactes aient été développées pour résoudre ce problème. Le problème de faisabilité, donc de détermination d'une solution admissible, étant très difficile, nous allons voir que la majorité des auteurs a contourné cette difficulté en modifiant certaines contraintes, notamment en assurant une certaine flexibilité au niveau de la capacité du système de production. Pour ce problème précis, il existe un ensemble d'instances de référence qui a été défini par (Trigeiro et al. 1989) et qui sert communément de Benchmark pour la communauté scientifique. Ces instances considèrent des problèmes pouvant aller jusqu'à 24 produits sur 30 périodes. Mais il est néanmoins très difficile de pouvoir comparer les multiples méthodes rencontrées dans la littérature car, comme dit précédemment, elles intègrent une certaine flexibilité au modèle pour pallier la difficulté de la faisabilité.

3.1 Les méthodes exactes

Il existe peu de méthodes exactes pour le problème du CLSP avec Setup Time.

(Miller et al. 2000) a proposé un algorithme à base de *Branch & Cut*. Il effectue ses tests sur les instances de Trigeiro et obtient de très bons résultats en moins d'une heure de calcul (il arrive à 0.41% de GAP par rapport à la meilleure solution connue sur certaines instances réputées difficiles).

(Belvaux et Wolsey 2000) ont développé un outil logiciel « BC-Prod » à base de *Branch & Cut* qui permet de résoudre plusieurs types de problèmes de lot-sizing. Pour le modèle qui nous intéresse, ils obtiennent un GAP pouvant aller jusqu'à 0,87% par rapport à la meilleure solution connue sur certaines instances très difficiles de Trigeiro.

(Absi et Kedad-Sidhoum 2006) proposent quant à eux un algorithme de *Branch & Cut* basé sur une extension du modèle qui permet de contourner la difficulté de la faisabilité. En effet, ils permettent de ne pas satisfaire la demande. Cela se traduit par un coût de perte qui est imputé dans la fonction objectif. Cette extension porte le nom de « Multi-item capacitated lot-sizing problem with setup times and shortage costs ». Cette extension leur a permis de proposer un algorithme beaucoup plus rapide que ceux proposés précédemment et d'obtenir des GAPs de l'ordre de 1% par rapport à la borne inférieure proposée par leur solveur (Cplex 9.0) en moins de 600 secondes sur les instances de Trigeiro, remaniées par leurs soins pour intégrer les coûts de perte.

Une autre façon de contourner la difficulté de l'élaboration d'une solution admissible est de rendre flexible la capacité de production. Pour ce faire, certains auteurs ont introduit la possibilité d'accroître temporairement les capacités au moyen, par exemple, d'heures supplémentaires. (Diaby et al. 1992a) utilisent cette extension et ont développé un *Branch & Bound* où la détermination d'une borne inférieure à chaque nœud se fait grâce à une relaxation lagrangienne des contraintes d'équilibre des stocks (ce qui débouche sur un problème de sac à dos). Par cette technique, ils arrivent à traiter jusqu'à 99 produits sur 8 périodes avec un Gap de 0.65% par rapport à la borne inférieure mais les solutions obtenues se trouvent être au-delà de la capacité initiale. Ils remarquent également que la relaxation lagrangienne de la contrainte d'équilibre des stocks ne fournit pas de très bonnes bornes inférieures et concluent qu'il faut lui préférer celle des contraintes de capacité.

Ces résultats, bien que très impressionnants, ne doivent pas nous faire perdre de vue que nous ne considérons pas des instances de type industriel. En effet, bien que très difficiles, les instances de Trigeiro ne définissent, somme toute, que des benchmarks de taille modeste en comparaison des problèmes de planification que nous pouvons trouver dans un contexte industriel. Et c'est ce qui constitue la limite des méthodes exactes et la force des méthodes approchées qui, bien que fournissant des résultats plus

modestes sur de petites instances, permettront d'obtenir des résultats acceptables sur des problèmes de grande taille, et cela en un temps raisonnable.

3.2 Les méthodes approchées

Beaucoup de méthodes approchées ont été développées pour ce problème. Mais peu d'entre-elles tentent de résoudre le problème dans sa forme initiale comme nous allons le voir dans la suite.

3.2.1 Le modèle initial

Les méthodes approchées que nous avons pu recenser dans notre état de l'art concernant le modèle dans sa forme originale, se basent toutes sur une relaxation lagrangienne des contraintes de capacité couplée avec une heuristique de lissage qui reconstruit une solution admissible à partir de celle obtenue par la relaxation lagrangienne. Plus précisément, toutes les heuristiques que nous avons rencontrées s'appuient sur l'heuristique de (Trigeiro et al. 1989) en tentant d'en améliorer quelques aspects.

La méthode de base est donc due à (Trigeiro et al. 1989). Dans celle-ci, les auteurs utilisent une relaxation lagrangienne des contraintes de capacité. Le problème se décompose donc en sous-problèmes sans capacité qu'ils ont pu résoudre optimalement grâce à l'algorithme de Wagner-Within (Wagner et Whitin 1958). Ensuite, ils utilisent une heuristique de lissage qui tente d'éliminer les dépassements de capacité en favorisant les alternatives de moindre coût (en suivant le critère de Silver-Meal : minimiser le coût moyen par période) à chaque itération de leur relaxation. Il est à noter qu'ils utilisent un algorithme de sous-gradient pour guider leur relaxation lagrangienne.

Plus récemment, (Du Merle et al. 1997) ont amélioré l'heuristique de (Trigeiro et al. 1989), non pas au niveau de l'heuristique de lissage, mais au niveau de la relaxation lagrangienne, en employant une méthode par point intérieur ou la méthode des coupes de Kelley. Ils remarquent que cette modification a un effet bénéfique en terme de rapidité pour cette heuristique.

(Hindi et al. 2003) améliorent l'heuristique de (Trigeiro et al. 1989) en modifiant l'heuristique de lissage présente dans celle-ci. En fait, à chaque itération de la relaxation lagrangienne, ils utilisent cette heuristique une première fois. S'ils obtiennent un résultat alors ils appliquent une seconde phase qu'ils appellent « recherche locale » : ils perturbent un peu la solution courante en tâchant encore de minimiser les coûts mais en ne tenant pas nécessairement compte de la capacité. Si celle-ci se trouve dépassée, ils réutilisent l'heuristique de lissage pour tâcher d'éliminer ce dépassement de capacité. Ils recommencent jusqu'à ce qu'ils ne trouvent pas une meilleure solution après k_{max} lissages. Avec ce principe, ils observent une nette amélioration des résultats obtenus par l'heuristique de (Trigeiro et al. 1989), puisqu'ils passent d'un GAP de 0.55% à un GAP de 0.28% par rapport à la borne inférieure obtenue par (Trigeiro et al. 1989) sur son instance la plus grande.

(Süral et al. 2008) proposent une heuristique lagrangienne pour le problème original mais en ne tenant pas compte des coûts de setup. Ils ont remarqué que l'heuristique de (Trigeiro et al. 1989) obtenait un GAP moyen de 34,29 % pour ce nouveau problème. En utilisant leur heuristique lagrangienne (SDW), ils arrivent à atteindre un GAP moyen de 27,03 %.

3.2.2 Avec Pricing

Pour contourner le problème de faisabilité du problème, (Haugen et al. 2006) ont eu l'idée de baser leur optimisation non plus sur une minimisation des coûts de production mais plutôt sur une maximisation du profit engendré (prix de vente - coût de production). Dans leur modèle, ils utilisent le prix de chaque produit comme variable de décision, ce qui leur permet de réguler la demande en fonction de leur capacité de production : les prix de chaque produit augmentant en fonction de la capacité consommée. En effet, ils partent du principe que plus le produit sera cher, moins la demande sera élevée. Ils obtiennent ainsi un nouveau modèle qui se trouve être quadratique et qu'ils optimisent par une relaxation lagrangienne. Comme benchmark, ils utilisent les instances de (Trigeiro et al. 1989) et arrivent à un GAP moyen de 0,58% de leurs bornes inférieures en seulement une itération. Bien que

ce résultat semble prometteur, il ne faut pas perdre de vue que, malgré le fait que cette formulation s'appuie sur le CLSP avec Setup time, le modèle utilisé est différent et par sa nature, et par son objectif.

3.2.3 Avec Overtime

S'appuyant sur la conclusion qu'ils avaient faite dans (Diaby et al. 1992a), (Diaby et al. 1992b) ont proposé une heuristique basée sur une relaxation lagrangienne des contraintes de capacité avec overtime. Ils peuvent ainsi résoudre les sous-problèmes engendrés par l'algorithme de Wagner-Within (Wagner et Whitin 1958) ce qui leur permet de déterminer des périodes de production. Ensuite, leur heuristique consiste, au cas où il y aurait des dépassements de capacité excédant la taille maximale d'« overtime » consentie, à résoudre un problème de transport quitte à perturber un peu les périodes de production. Par cette technique, ils arrivent à résoudre des problèmes de très grande taille (en excédant la capacité initiale) de 5000 produits sur 30 périodes avec un GAP de 0.59% par rapport à leur borne inférieure.

Plus récemment, ((Ozdamar et Bozyel 2000), (Ozdamar et al. 2002)) utilisent une technique basée sur un algorithme génétique qu'ils couplent avec une méthode de recherche locale s'appuyant sur une variante de la méthode Tabou intégrant une technique de recuit simulé dans l'acceptation d'une solution courante dégradée. Ils nomment cette heuristique G.A.T.A. (Genetic Algorithm With Tabu and Annealing). Ils testent leur méthode sur leurs propres instances (14 produits sur 6 périodes) et obtiennent un GAP de 0.17% par rapport aux meilleures solutions obtenues sur ces instances (fournies dans (Ozdamar et Bozyel 2000)).

3.2.4 Avec Stocks initiaux

Une autre façon de contourner la faisabilité du problème a été de considérer des stocks initiaux non nuls qui seront également utilisés comme variables de décision et pénalisés dans la fonction objectif. En effet, déterminer une solution réalisable peut du coup se faire facilement, ne serait-ce qu'en mettant en stock initial l'intégralité de la demande sur l'horizon. C'est cette extension qu'utilisent (Degraeve et Jans 2003). Ils mettent en œuvre un algorithme basé sur une décomposition de Dantzig-Wolfe. Pour ce faire, ils n'utilisent pas la formulation approximative de Manne (Manne 1958), mais proposent leur propre modèle. Ils testent leur méthode sur les instances de Trigeiro et obtiennent un GAP de 0.2 % par rapport à la borne inférieure donnée par ce dernier sur les mêmes instances.

3.2.5 Avec Setup Carryover

Plusieurs auteurs ont fait le constat qu'en utilisant le modèle avec Setup Time, on ne tenait pas compte du fait que le système de production pouvait déjà être configuré pour une nouvelle campagne de fabrication d'un certain produit. En effet, le modèle du CLSP ne fournit pas de séquençement de la production. Pour ce faire, ils ont modifié ce modèle dans le but de pouvoir déterminer pour chaque période, les premiers et derniers produits à être fabriqués afin de pouvoir se dispenser d'éventuelles reconfigurations du système, le cas échéant. Ce modèle peut donc entraîner une baisse significative de la capacité utilisée globalement pour la reconfiguration du système.

C'est sur ce principe que s'appuie (Gopalakrishnan et al. 2001) en proposant une méthode Tabou pour résoudre ce problème. Dans leur algorithme, ils pénalisent dans la fonction objectif les dépassements de capacité ce qui leur permet d'élaborer facilement des solutions initiales. Ils testent leur heuristique sur les instances de (Trigeiro et al. 1989). Ils obtiennent sur ces instances des solutions réalisables mais avec un GAP d'en moyenne 12.4% par rapport à la borne inférieure fournie par l'heuristique de Trigeiro. Dans un second temps, ils modifient leur méthode Tabou pour le CLSP avec Setup Time et concluent qu'ils peuvent rivaliser avec l'heuristique de Trigeiro car ils obtiennent en moyenne un GAP de 4% par rapport à la borne inférieure fournie par l'heuristique de (Trigeiro et al. 1989) sur

les plus grandes instances de ces derniers. Remarquons tout de même, comme le concluent également les auteurs, que cette modélisation ne permet pas d'assurer la détermination d'une solution réalisable.

C'est pour cela que (Karimi et al. 2005) prennent en compte une possibilité de retard dans la satisfaction de la demande en intégrant du « Backlogging » dans ce modèle. Ils élaborent une heuristique à base de Tabou qu'ils testent sur leurs propres instances (12 produits X 6 périodes) et se comparent aux résultats obtenus par un solveur (LINGO) dont ils ont fixé le « time limit » à 300 minutes. Pour générer leur solution initiale, ils utilisent une heuristique de construction. Ils montrent ainsi que leur méthode Tabou fournit, en un meilleur délai, des résultats meilleurs que ceux obtenus par le solveur.

La figure 3.3 résume l'état de l'art présenté :

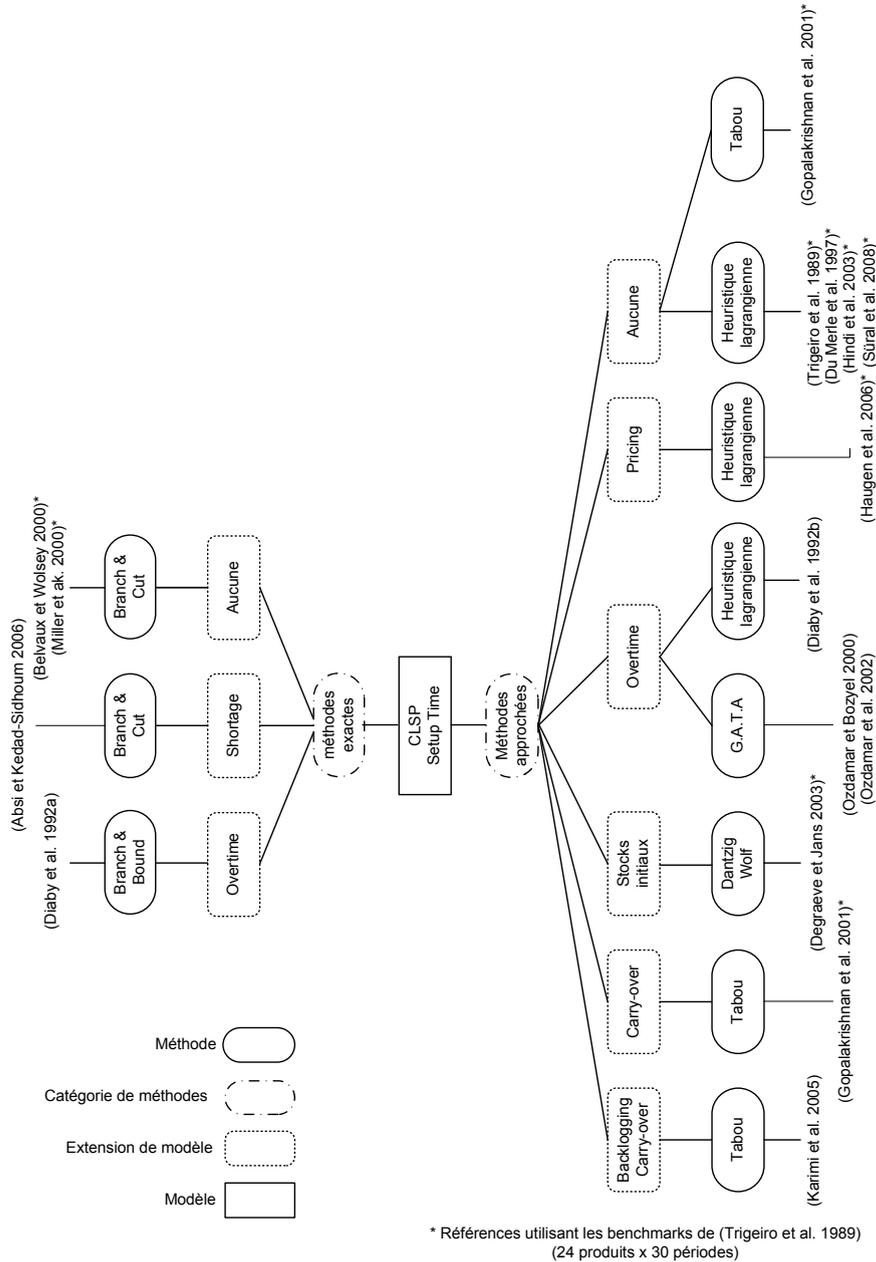


Figure 3.3 – Les méthodes de résolution pour le CLSP avec Setup-Time

Le tableau 3.2 synthétise les différents articles répertoriés :

Référence	Méthodes Exactes	Méthodes Approchées	Taille maximal des instances testées	GAP obtenu	Valeur de comparaison	Extension
(Miller et al. 2000)	$B\&C$		$30 \times 24^*$	0,41	BIP	
(Belvaux et Wolsey 2000)	$B\&C$		$30 \times 24^*$	0,87	BIP	
(Absi et Kedad-Sidhoum 2006)	$B\&C$		$30 \times 24^*$	1,00	LB	Shortage
(Diaby et al. 1992a)	$B\&B$		99×8	0,65	LB	Overtime
(Trigeiro et al. 1989)		H. lagrangienne	$30 \times 24^*$	×		
(Du Merle et al. 1997)		H. lagrangienne	$30 \times 24^*$	×		pt intérieur
(Hindi et al. 2003)		H. lagrangienne	$30 \times 24^*$	0,28	LB [@]	
(Haugen et al. 2006)		H. lagrangienne	$30 \times 24^*$	0,58	LB	Pricing (Quad)
(Diaby et al. 1992b)		H. lagrangienne	5000×30	0,59	LB	Overtime
(Ozdamar et Bozyel 2000)		G.A.T.A	14×6	×		Overtime
(Ozdamar et al. 2002)		G.A.T.A	14×6	0,17	BIP	Overtime
(Degraeve et Jans 2003)		Dantzig-Wolfe	$30 \times 24^*$	0,20	LB	Stocks Init.
(Gopalakrishnan et al. 2001)		Tabou	$30 \times 24^*$	12,4	LB [@]	Carryover
(Gopalakrishnan et al. 2001)		Tabou	$30 \times 24^*$	4,00	LB [@]	
(Karimi et al. 2005)		Tabou	12×6	×		Carryover+Backlog
(Süral et al. 2008)		H. lagrangienne	$30 \times 24^*$	27,03	LB ^s	no setup cost

Tableau 3.2 – Etat de l’art pour le CLSP avec setup time

Légende :

* Instances de (Trigeiro et al. 1989)

@ Borne inférieure donnée par l’heuristique de (Trigeiro et al. 1989)

^s Borne inférieure obtenue à partir de l’heuristique de (Trigeiro et al. 1989) sans les coûts de setup

Au vu de cet état de l'art, nous pouvons remarquer que peu de méthodes abordent le problème du CLSP avec Setup Time sous sa forme initiale. La plupart du temps, les auteurs utilisent des extensions de ce modèle pour faciliter l'obtention d'une solution admissible. Nous pouvons également remarquer que les heuristiques qui tentent de le résoudre s'appuient toutes sur une relaxation lagrangienne de la capacité. (Du Merle et al. 1997) suggèrent que c'est sans doute la meilleure façon d'aborder ce problème. On a vu également que seul (Gopalakrishnan et al. 2001) ont tenté d'apporter une réponse par une métaheuristique adaptée d'une extension « avec carryover ». C'est, à notre connaissance, la seule tentative qui a été faite sur ce modèle.

4 Conclusion sur l'état de l'art

L'état de l'art présenté permet de conclure sur l'importance des travaux portés sur ce modèle. Ceci s'explique par le fait que ces modèles sont à la base des moteurs de planification des APS ((Drexel et Kimms 1997),(Genin 2003)). Néanmoins, il est très difficile de pouvoir comparer les différentes méthodes proposées dans la littérature : en effet, le manque d'homogénéité des instances étudiées empêche toute comparaison des résultats donnés.

- Pour le CLSP, sur les 19 articles que nous avons répertoriés, seulement 5 partagent le même jeu d'instances.
- Pour le CLSP avec Setup Time, les instances de (Trigeiro et al. 1989) semblent faire référence (11 articles sur les 16 répertoriés les utilisent). Cependant, les auteurs usent de très nombreuses extensions permettant de réduire la complexité du problème de faisabilité, rendant ainsi impossible les comparaisons entre les différentes approches.

Il est donc difficile d'appréhender la performance des méthodes étudiées. C'est également ce que concluent (Jans et Degraeve 2007). Il serait donc intéressant de définir, pour chaque problème, quelles sont les instances à utiliser (les instances de référence) ce qui permettrait de pallier ce problème. Pour notre part, il semblerait logique d'utiliser les instances de (Eppen et Martin 1987) pour le CLSP et les instances de (Trigeiro et al. 1989) pour le CLSP avec setup time : en effet, ce sont les instances les plus communément utilisées. C'est ce que nous ferons par la suite.

Enfin, (Brahimi 2004) souligne le peu d'utilisation qui a été faite des métaheuristiques, malgré leurs performances prometteuses. C'est cette approche que nous allons développer par la suite.

III Proposition de méthodes pour le CLSP

Comme nous avons pu le remarquer au regard de notre état de l'art sur le CLSP (figure 3.2), peu de métaheuristiques ont été proposées pour tenter d'apporter une réponse satisfaisante au problème du CLSP. Nous nous proposons donc de montrer la pertinence de l'emploi de métaheuristiques à base de recuit simulé pour l'obtention de solutions de bonne facture pour ce problème.

Ce type de métaheuristique nécessite (Michalewicz et Fogel 2002) :

- un codage permettant de représenter les solutions du problème,
- une fonction d'évaluation de la solution,
- un système de voisinage (une application permettant de passer d'une solution à une autre).

Une condition nécessaire au bon fonctionnement de ce type de méthodes est que toute solution soit accessible et que le système de voisinage utilisé soit réversible :

- L'accessibilité est une propriété de connexité par arc : si on considère deux solutions quelconques, on peut passer de l'une à l'autre par une ou plusieurs applications du système de voisinage.
- La réversibilité du système de voisinage revient à parler de symétrie dans son application : Si une solution x peut être obtenue d'une solution y par application du système de voisinage alors la solution y peut être obtenue de la solution x de la même façon.

Pour pouvoir juger de la qualité des solutions obtenues par nos méthodes d'optimisation, nous avons également élaboré un calcul de borne inférieure basé sur une relaxation lagrangienne.

1 Calcul d'une borne inférieure

Pour déterminer une borne inférieure pour notre problème, nous allons utiliser une relaxation lagrangienne des contraintes de capacité pour deux raisons :

- (Chen et Thizy 1990) ont prouvé que c'était la relaxation susceptible de fournir la meilleure borne inférieure,
- Relaxer les contraintes de capacité aboutit à un problème polynomial que l'on sait résoudre par l'algorithme (Wagner et Whitin 1958).

Pour commencer, écrivons le problème relaxé du CLSP :

$$\text{Minimiser } \sum_{i=1}^N \left[\sum_{t=1}^T (s_i X_{it} + b_i Q_{it} + h_i I_{it}) \right] + \langle \lambda, \left(\sum_{i=1}^N p_i Q_{it} - C_t \right)_{t \in \llbracket 1, T \rrbracket} \rangle \quad (3.28)$$

Sous les contraintes :

$$I_{it} = I_{i(t-1)} + Q_{it} - d_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.29)$$

$$Q_{it} \leq M X_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.30)$$

$$X_{it} \in \{0, 1\} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.31)$$

$$I_{it}, Q_{it} \geq 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.32)$$

Nous noterons U l'ensemble défini par les contraintes (3.29), (3.30), (3.31) et (3.32).

Formons le lagrangien de ce problème :

$$\begin{aligned} \mathcal{L} : U \times (\mathbb{R}^T)^+ &\longrightarrow \mathbb{R} \\ (Q, \lambda) &\longmapsto \sum_{i=1}^N \left[\sum_{t=1}^T (s_i X_{it} + b_i Q_{it} + h_i I_{it}) \right] + \langle \lambda, \left(\sum_{i=1}^N p_i Q_{it} - C_t \right)_{t \in \llbracket 1, T \rrbracket} \rangle \end{aligned}$$

Définissons maintenant la fonction duale associée au problème :

$$\begin{aligned} h : W &\longrightarrow \mathbb{R} \\ \lambda &\longmapsto \min_{Q \in U} \mathcal{L}(Q, \lambda) \end{aligned}$$

où $W = \{\lambda \in (\mathbb{R}^T)^+ \text{ tel que } \min_{Q \in U} \mathcal{L}(Q, \lambda) > -\infty\}$.

La théorie liée à la relaxation lagrangienne nous apprend que trouver une bonne borne inférieure revient à maximiser la fonction duale associée au problème, c'est-à-dire résoudre le problème :

$$\max_{\lambda \in W} h(\lambda)$$

Cependant, (Lasdon 2002) a prouvé que pour tout $\lambda \geq 0$ la fonction h est différentiable en λ et les dérivées partielles sont $\frac{\partial h}{\partial \lambda_t}(\lambda) = \sum_{i=1}^N p_i Q_{it} - C_t$. Nous allons donc exploiter cette propriété pour résoudre le problème de maximisation lié à la fonction h et cela, par deux techniques différentes :

- en utilisant une méthode basée sur l'utilisation d'une méthode de descente par gradient,
- en utilisant une méthode basée sur l'utilisation d'une méthode de sous-gradient.

1.1 Méthode de descente par gradient

Pour maximiser la fonction h , nous utilisons une méthode de descente par gradient (Lasdon 2002) dont le synopsis est donné par la figure 3.4.

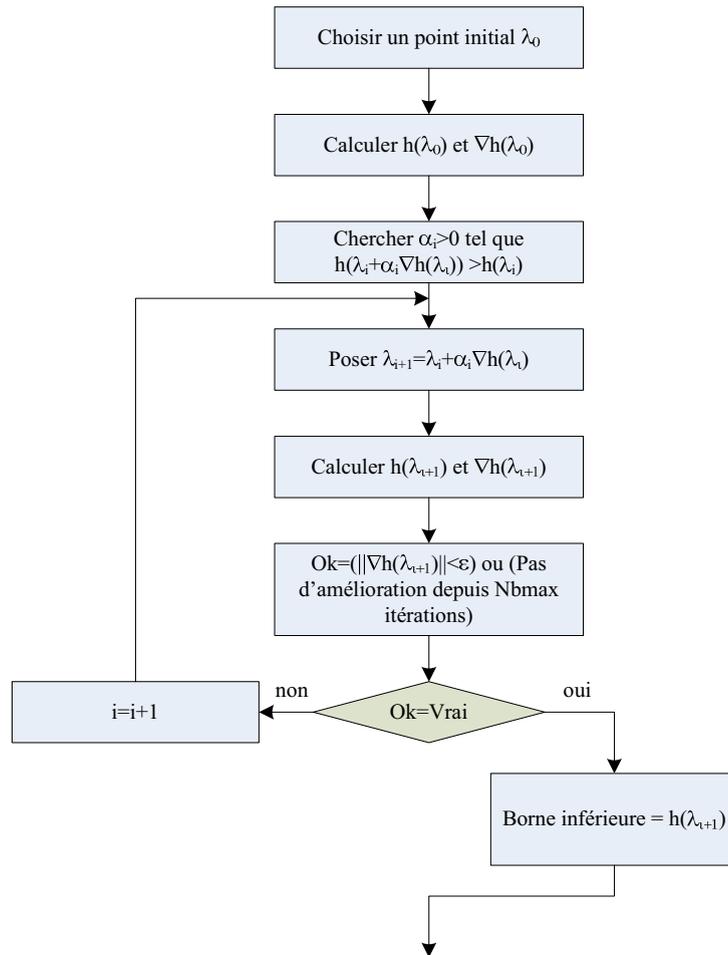


Figure 3.4 – Algorithme pour la descente par gradient

La principale difficulté réside dans la recherche de α_i . Pour pallier celle-ci, nous nous appuyons sur une interpolation quadratique. Nous procédons par étapes de la façon suivante :

1. Poser $s_t = \begin{cases} \nabla h(\lambda_i)_t & \text{si } (\lambda_i)_t > 0 \\ \max(0, \nabla h(\lambda_i)_t) & \text{sinon} \end{cases} \quad \forall t \in \llbracket 1, T \rrbracket$
2. Normaliser le vecteur s : $s_t = \frac{s_t}{\max_{k \in \llbracket 1, T \rrbracket} s_k}$
3. Calculer $h(\lambda_i + \alpha s)$ pour $\alpha = 0, 1, 2, 4, \dots, a, b, c$ ou c est la première valeur telle que h décroît : la valeur α_i que nous cherchons vérifie donc : $b \leq \alpha_i < c$
4. Si $0 \leq \alpha_i < 1$ (c'est à dire si $h(\lambda_i) > h(\lambda_i + s)$) alors diviser s par 2 et retourner à l'étape 3.
5. Calculer le polynôme d'interpolation quadratique passant par les points $(a, h(\lambda_i + as))$, $(b, h(\lambda_i + bs))$ et $(c, h(\lambda_i + cs))$ et son maximum α_{opt}
6. Si $h(\lambda_i + \alpha_{opt}s) > h(\lambda_i + bs)$ alors $\alpha_i = \alpha_{opt}$ sinon $\alpha_i = b$

Pour déterminer le polynôme d'interpolation quadratique passant par ces points, nous utilisons les polynômes de Lagrange qui leur sont associés :

- pour a , nous avons $L_a(x) = \frac{(x-b)(x-c)}{(a-b)(a-c)}$,
- pour b , nous avons $L_b(x) = \frac{(x-a)(x-c)}{(b-a)(b-c)}$,

- pour c , nous avons $L_c(x) = \frac{(x-a)(x-b)}{(c-a)(c-b)}$.

Le polynôme d'interpolation s'écrit donc :

$$P(x) = h(\lambda_i + as)L_a(x) + h(\lambda_i + bs)L_b(x) + h(\lambda_i + cs)L_c(x).$$

Ce polynôme atteint son optimum pour la valeur :

$$x^* = \frac{1}{2} \frac{h(\lambda_i + as)(c^2 - b^2) + h(\lambda_i + bs)(a^2 - c^2) + h(\lambda_i + cs)(b^2 - a^2)}{h(\lambda_i + as)(c - b) + h(\lambda_i + bs)(a - c) + h(\lambda_i + cs)(b - a)}.$$

1.2 Méthode de sous-gradient

La méthode du sous-gradient s'appuie sur l'élaboration d'une suite $(\lambda_k)_{k \in \mathbb{N}}$ vérifiant la relation de récurrence suivante :

$$\lambda_{k+1} = \lambda_k + \alpha_k \frac{g^k}{\|g^k\|}$$

où $g^k \in \partial h(\lambda_k)$ et $\alpha_k > 0$. ($\partial h(\lambda_k)$ est l'ensemble des sous-gradients de h en λ_k).

D'après la remarque sur la différentiabilité de h , on peut prendre pour g^k le gradient de h au point λ_k : $\nabla h(\lambda_k)$.

Le point délicat de la méthode est la détermination de la valeur α_k . Pour ce faire, (Held et Karp 1971) propose un schéma d'optimisation qui consiste à définir α_k de la façon suivante :

$$\alpha_k = \omega_k \frac{\underline{h} - h(\lambda_k)}{\|g^k\|}$$

où \underline{h} est une borne supérieure du problème $\max_{\lambda \in W} h(\lambda)$ et $0 < \omega_k < 2$.

Donc nous définissons la suite $(\lambda_k)_{k \in \mathbb{N}}$ de la façon suivante :

$$\lambda_{k+1} = \max \left[0, \lambda_k + \alpha_k \frac{\nabla h(\lambda_k)}{\|\nabla h(\lambda_k)\|} \right]$$

et

$$\alpha_k = \omega_k \frac{\underline{h} - h(\lambda_k)}{\|\nabla h(\lambda_k)\|}$$

Soit :

$$\lambda_{k+1} = \max \left[0, \lambda_k + \omega_k \frac{\nabla h(\lambda_k) (\underline{h} - h(\lambda_k))}{\|\nabla h(\lambda_k)\|^2} \right] \quad (3.33)$$

Astreindre la valeur de λ_k à être positif permet de préserver la différentiabilité de h en λ_k .

Pour déterminer nos multiplicateurs lagrangiens initiaux, nous avons utilisé la méthode proposée par Diaby *et al.* (Diaby et al. 1992b) en l'adaptant à notre problème : nous avons initialisé $(\lambda_0)_t$ pour une période $t \in \llbracket 1, T \rrbracket$ en lui affectant le rapport de la capacité totale utilisée pendant cette période si on produisait en lot pour lot par la capacité du système de production à cette période.

$$(\lambda_0)_t = \frac{\sum_{i=1}^N (s_i \delta(d_{it}) + p_i d_{it})}{C_t} \quad t \in \llbracket 1, T \rrbracket \quad \text{où } \delta(d_{it}) = \begin{cases} 0 & \text{si } d_{it} = 0 \\ 1 & \text{sinon} \end{cases}.$$

Après différents tests, nous avons construit la suite $(\omega_k)_{k \in \mathbb{N}}$ de la manière suivante :

$$\begin{cases} \omega_0 = 1,9 \\ \omega_k = \begin{cases} \omega_{k-1} \times 0,9 & \text{si il n'y a pas eu d'amélioration de plus de 0,1\% durant 1000 itérations} \\ \omega_{k-1} & \text{sinon.} \end{cases} \end{cases}$$

Si au cours de l'algorithme, la solution $h(\lambda_k)$ obtenue est meilleure que \underline{h} alors on met \underline{h} à jour en posant $\underline{h} = h(\lambda_k) \times 1,1$. L'algorithme s'arrêtera s'il n'y a pas eu d'amélioration de plus de 0,1% pendant 50000 itérations.

La méthode utilisée est schématisée dans la figure 3.5.

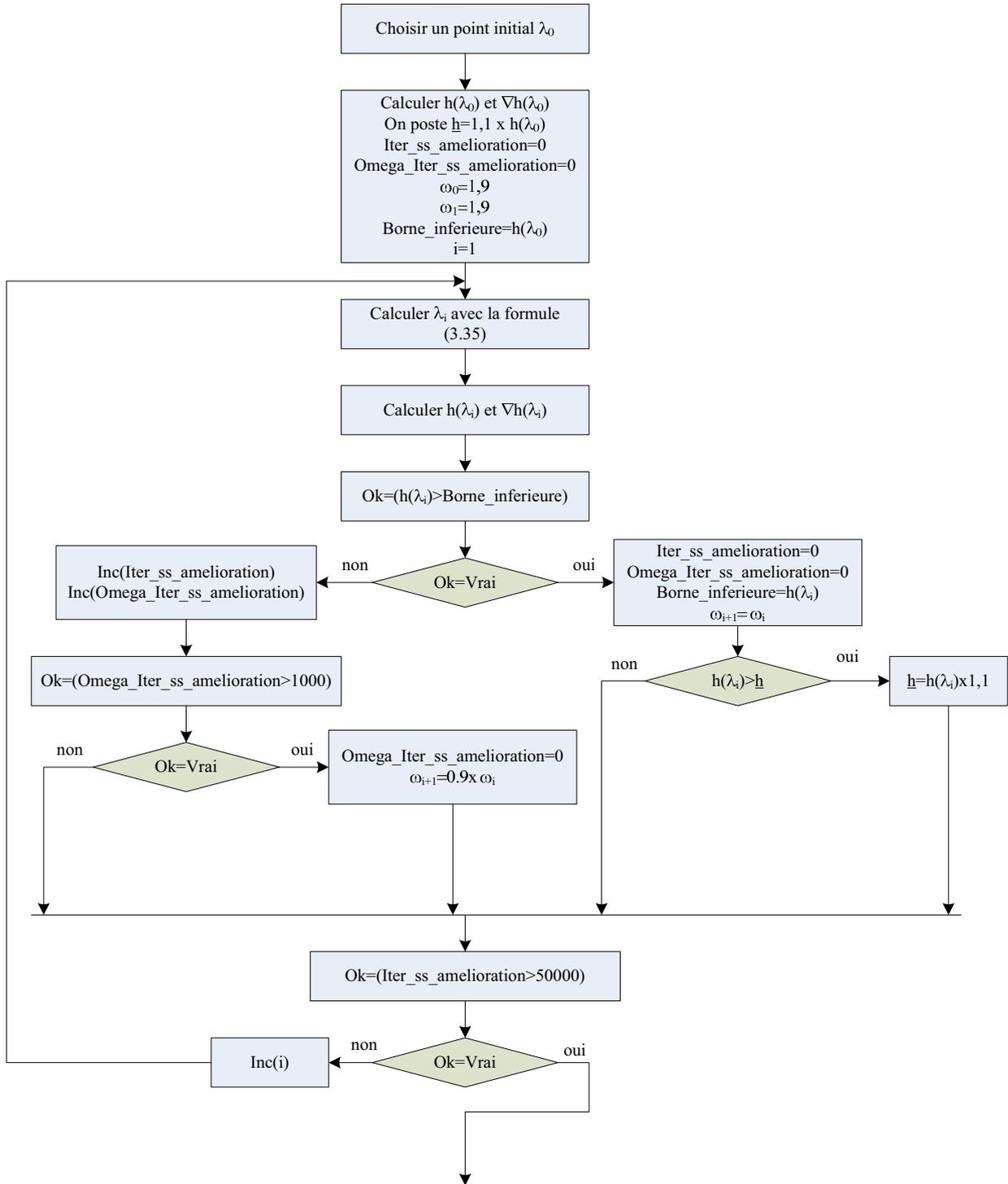


Figure 3.5 – Algorithme pour la méthode par sous-gradient

2 Chaînage de deux métaheuristiques

Dans cette partie, nous présentons une méthode d'optimisation basée sur le chaînage de deux métaheuristiques.

- La première métaheuristique (métaheuristique « solution initiale ») permet de déterminer une solution initiale au problème de planification : partant d'une solution non admissible, elle tente de trouver une solution réalisable au problème en minimisant une fonction objectif quadratique pénalisant les violations des contraintes de capacité. La méthode d'optimisation commençant par le calcul d'une borne inférieure, la solution (non admissible en générale) donnée par cette dernière est donnée en entrée de cette métaheuristique.
- La seconde métaheuristique (métaheuristique « optimisation ») prend comme solution initiale la solution réalisable déterminée par méthode précédente et tente d'en minimiser les coûts.

Le synopsis de la méthode d'optimisation est donnée par la figure 3.6

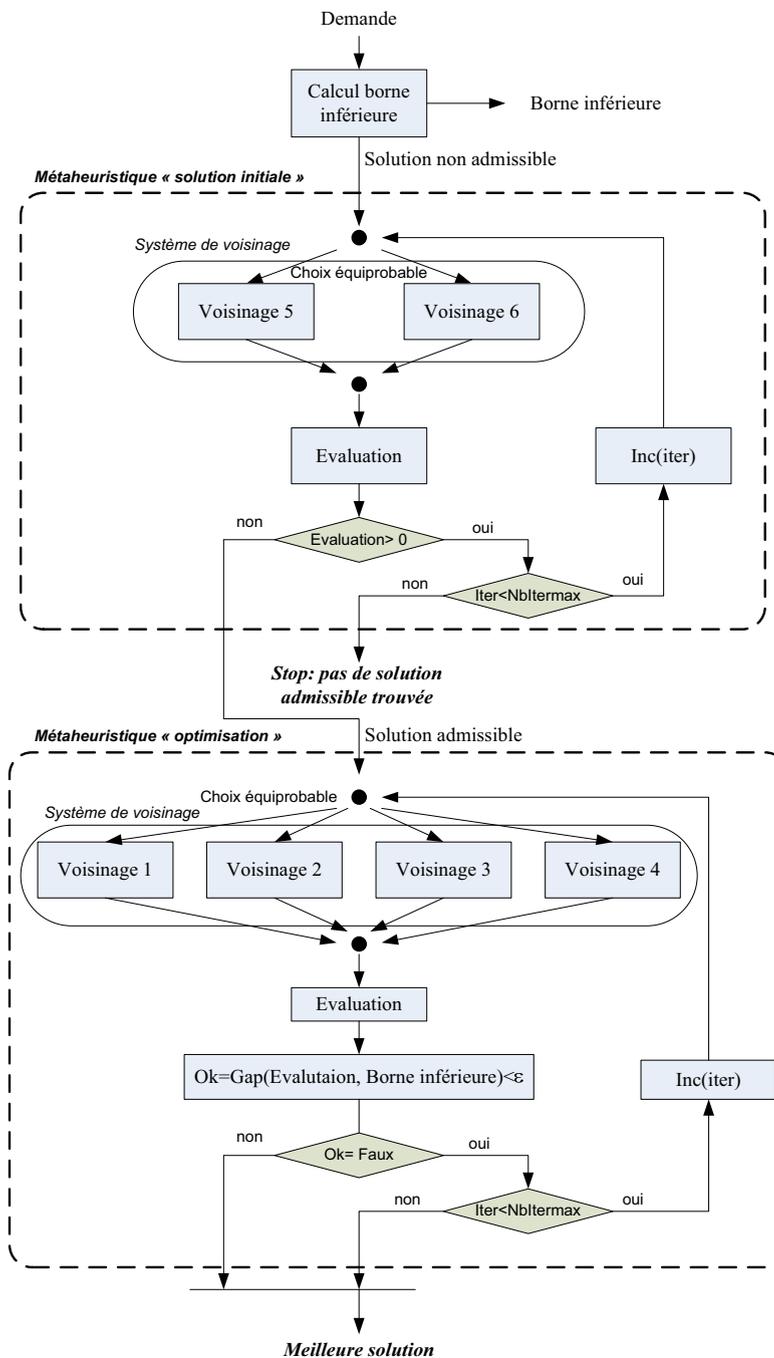


Figure 3.6 – Algorithme de notre méthode d'optimisation

2.1 La métaheuristique « optimisation »

Dans cette partie, nous décrivons la métaheuristique « optimisation ». Nous commençons par donner le codage de la solution puis les voisinages utilisés.

2.1.1 Le codage de la solution

Le CLSP utilise trois types de variables de décision : les quantités fabriquées par produit et par période (Q_{it}), les quantités en stock en fin de période pour chaque produit (I_{it}) et les variables binaires qui indiquent si il y a eu ou non un lancement de campagne pour chaque produit durant la période considérée (X_{it}). Il s'ensuit que pour coder une solution du CLSP, il faudrait tenir compte de l'ensemble de ces variables. Cependant, l'équation des stocks (3.2) montre une correspondance biunivoque entre les variables de stocks et les variables de quantités. En effet, elle peut également s'écrire sous la forme :

$$Q_{it} = I_{it} - I_{i(t-1)} + d_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$$

ce qui prouve que les quantités fabriquées peuvent être déterminées de façon unique par les quantités stockées. Réciproquement, en sommant de façon adéquate les différentes équations des stocks, on montre que :

$$I_{it} = I_{i0} + \sum_{k=1}^t (Q_{ik} - d_{ik}) \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$$

Cela démontre qu'on peut également déterminer de façon unique les quantités stockées en fonction des quantités fabriquées. Cela montre que représenter les stocks dans notre codage serait redondant. En ce qui concerne les variables de lancement de campagne, il est clair qu'elles sont fonction des quantités fabriquées. Plus précisément,

$$X_{it} = \chi_{]0, +\infty[}(Q_{it}) \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$$

où $\chi_{]0, +\infty[}$ est la fonction indicatrice de l'intervalle $]0, +\infty[$. Cette dernière n'étant pas bijective, nous pouvons conclure que connaissant les quantités produites il est possible d'en déduire de manière unique les lancements de campagne mais que la réciproque est fautive : en effet, connaissant les périodes où il y a lancement de campagne, il n'est en général pas possible d'en déduire de manière unique les quantités fabriquées par produit et par période.

Nous venons de voir qu'en définitive, il existe deux codages possibles pour une solution :

- soit une matrice représentant les lancements de production $X = (X_{it})_{(i,t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket}$
- soit une matrice représentant les quantités produites $Q = (Q_{it})_{(i,t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket}$

Utiliser le premier codage de façon efficace nécessite d'établir une correspondance entre la matrice binaire des lancements de production et un planning de production optimal en terme de coût respectant ces derniers. Ceci peut être réalisé en résolvant le programme linéaire initial dans lequel on aura fixé les valeurs des variables binaires X_{it} ((Hindi 1995), (Hindi 1996)). Comme le soulignent (Jans et Degraeve 2007), cela implique d'être capable de résoudre le programme linéaire de manière extrêmement efficace car il faudra le résoudre à chaque itération de la métaheuristique. Si ce n'est pas le cas, les quantités produites peuvent être déterminées heuristiquement : (Salomon et al. 1993) utilisent une heuristique gloutonne pour déterminer des quantités à fabriquer par produit et période qui soient les meilleures possibles mais ne peuvent assurer l'optimalité des quantités déterminées. Ainsi en restreignant l'espace de recherche de leur méthode d'optimisation, ils ne peuvent assurer l'appartenance de la solution optimale à cette restriction.

Dans notre cas, nous utilisons le second codage : en effet, pour le problème du CLSP, il est très simple à mettre en œuvre (n'ayant pas d'interdépendance entre les produits comme pour le MLCLSP)

et permet de nous passer de l'utilisation d'un solveur. Cependant, soulignons que dans le cas d'un modèle multi-niveau, l'utilisation d'un tel codage peut s'avérer d'une aide précieuse comme nous le verrons dans le chapitre suivant. Ainsi, si nous avons le planning de production suivant :

	période 1	période 2	...	période T
produit 1	Q_{11}	Q_{12}	...	Q_{1T}
produit 2	Q_{21}	Q_{22}	...	Q_{2T}
\vdots	\vdots	\vdots	...	\vdots
produit N	Q_{N1}	Q_{N2}	...	Q_{NT}

nous le coderons par la matrice :

$$\begin{pmatrix} Q_{11} & Q_{12} & \cdots & Q_{1T} \\ Q_{21} & Q_{22} & \cdots & Q_{2T} \\ \vdots & \vdots & & \vdots \\ Q_{N1} & Q_{N2} & \cdots & Q_{NT} \end{pmatrix}$$

Nous venons donc de voir comment nous allons coder nos plannings de production. Or, ceux-ci ne sont pas nécessairement des solutions admissibles pour notre problème du CLSP. Essayons de caractériser l'espace des solutions admissibles.

La contrainte (3.2) du modèle du CLSP nous dit que, moyennant la connaissance des stocks initiaux et de la demande, nous pouvons calculer les stocks pour chaque période et pour chaque produit dès que notre planning de production est déterminé. Nous noterons \mathcal{I}_Q la matrice des stocks (fonction de la matrice de production Q) obtenue par application de la contrainte (3.2). Bien évidemment, $\mathcal{I}_Q \in \mathcal{M}_{N,T}(\mathbb{Z})$ et les éléments $(I_{it})_{(i,t) \in \llbracket 1,N \rrbracket \times \llbracket 1,T \rrbracket}$ de la matrice représentent les quantités de produit i que nous avons en stock à la fin de la période t .

Le CLSP n'autorisant pas de retard dans la livraison de la demande, une condition nécessaire pour qu'une solution soit admissible est que les stocks soient positifs pour tous les produits sur toutes les périodes, ce qui peut s'énoncer formellement :

Proposition 3.1

$$Q \text{ est admissible} \Rightarrow \mathcal{I}_Q \geq 0.$$

De plus, si tous les stocks sont positifs, une condition suffisante pour que la solution soit admissible est qu'elle respecte les capacités du système, c'est à dire la contrainte (3.4). Nous pouvons donc énoncer :

Proposition 3.2

$$Q \text{ est admissible} \Leftrightarrow \begin{cases} \mathcal{I}_Q \geq 0. \\ \sum_{i=1}^N p_i Q_{it} \leq C_t \quad \forall t \in \llbracket 1, T \rrbracket \end{cases}$$

Cette proposition, qui caractérise les solutions admissibles de notre problème, va nous servir pour élaborer le mécanisme de visite des solutions admissibles pour le CLSP.

2.1.2 Le système de voisinage

Considérons une solution admissible Q de notre problème et soit \mathcal{I}_Q la matrice des stocks associée à ce planning. Le but est de déterminer une nouvelle solution admissible Q^* .

Le mécanisme de visite est basé sur l'algorithme suivant (figure 3.7) :

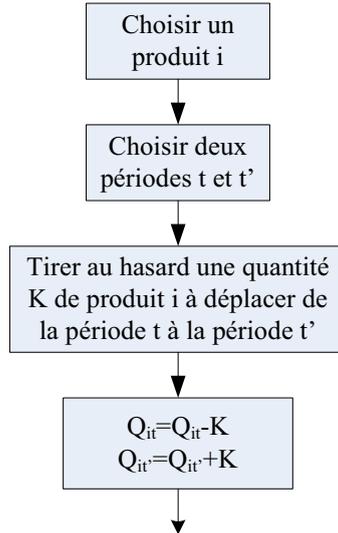


Figure 3.7 – Algorithme de base pour les voisinages

Il est clair qu'un tel algorithme a peu de chances, a priori, de fournir une solution admissible. En effet, il faut pouvoir déterminer l'ensemble des valeurs de K pour lequel \mathcal{Q}^* soit admissible, c'est à dire, par la proposition 3.2, tel que :

- 1) $\mathcal{I}_{\mathcal{Q}^*} \geq 0$
- 2) $\sum_{i=1}^N p_i Q_{it'}^* \leq C_{t'}$

Il est donc clair, d'après 2), que K doit être tel que :

$$\left(\sum_{j=1}^N p_j Q_{jt'} \right) + p_i K \leq C_{t'}$$

Soit, en supposant que p_i soit non nul,

$$K \leq \frac{C_{t'} - \left(\sum_{j=1}^N p_j Q_{jt'} \right)}{p_i}$$

Or $K \in \mathbb{N}$ et $K \leq Q_{it}$ car on ne peut pas déplacer plus que ce qui est produit, d'où :

$$0 \leq K \leq \min \left\{ E \left[\frac{C_{t'} - \left(\sum_{j=1}^N p_j Q_{jt'} \right)}{p_i} \right], Q_{it} \right\} \quad (3.34)$$

où E désigne la fonction partie entière.

La condition (3.34) nous assure le respect des capacités de production. Reste donc à s'assurer de la positivité des stocks.

Là, nous avons deux cas possibles :

- Ou bien $t' \leq t$ (figure 3.8). Dans ce cas, si $\mathcal{I}_{\mathcal{Q}} \geq 0$ alors nécessairement $\mathcal{I}_{\mathcal{Q}^*} \geq 0$: en effet, la solution \mathcal{Q} satisfaisant à la demande, la solution \mathcal{Q}^* fera de même (on ne fait que produire « encore plus en avance »). On remarquera que le cas $t' = t$ est un cas limite qui ne présente aucun intérêt ($\mathcal{Q}^* = \mathcal{Q}$).

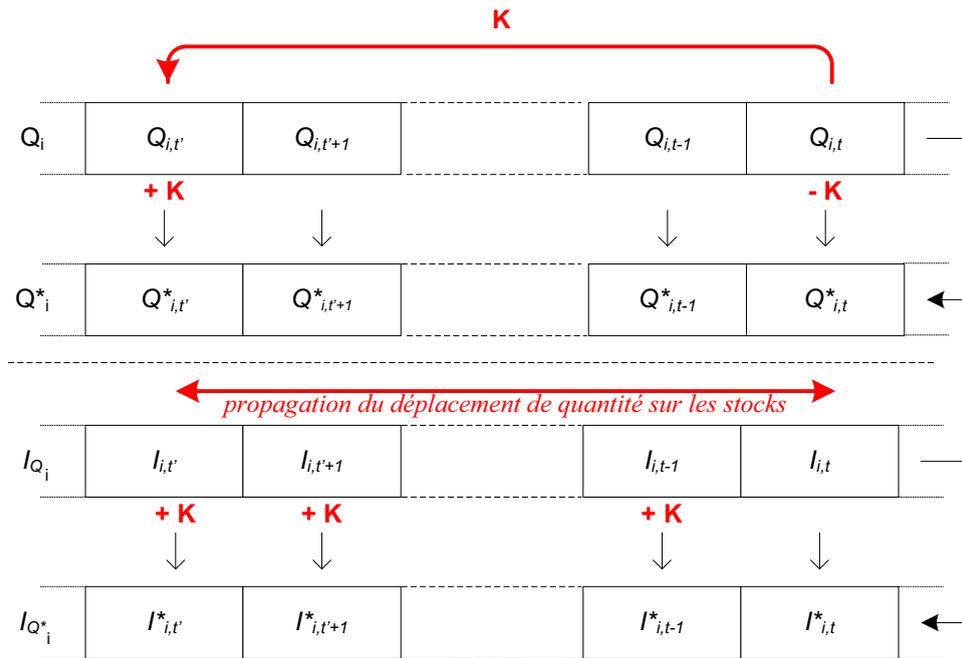


Figure 3.8 – Effet du déplacement de quantité sur les stocks (cas 1)

- Ou bien $t' > t$ (figure 3.9).

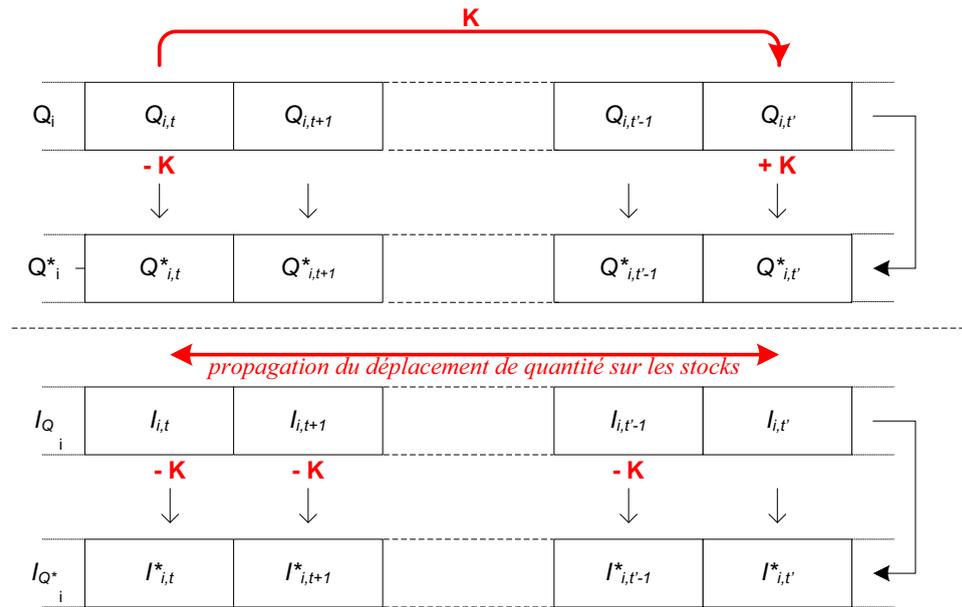


Figure 3.9 – Effet du déplacement de quantité sur les stocks (cas 2)

Dans ce cas, il faut préserver la positivité du stock, donc il faut que

$$0 \leq K \leq \min_{t \leq l < t'} (I_{il}) \tag{3.35}$$

Donc, nous pouvons énoncer la proposition suivante :

Proposition 3.3 Soit Q une solution admissible, pour que l'algorithme présenté dans la figure 3.7 fournisse une nouvelle solution Q^* admissible il faut et il suffit que :

- si $t' \leq t$, alors

$$0 \leq K \leq \min\left\{E \left[\frac{C_{t'} - \left(\sum_{j=1}^N p_j Q_{jt'} \right)}{p_i} \right], Q_{it}\right\}$$

- $t' > t$, alors

$$0 \leq K \leq \min\left\{E \left[\frac{C_{t'} - \left(\sum_{j=1}^N p_j Q_{jt'} \right)}{p_i} \right], Q_{it}, \min_{t \leq l < t'} (I_{il})\right\}$$

Cette proposition permet donc de caractériser l'obtention de solutions admissibles : il suffit donc, dans l'algorithme présenté dans la figure 3.7, de choisir aléatoirement une quantité K de produit i à déplacer vérifiant les relations définies dans celle-ci.

La proposition 3.3 va nous permettre de définir 4 voisinages différents pour notre métaheuristique, constituant ainsi notre système de voisinage.

2.1.2.1 Voisinage 1 : déplacement non guidé

Le but de ce voisinage est double :

- éliminer un maximum de lancements de campagne,
- tenter de réduire les stocks.

Pour éliminer un lancement de campagne d'un produit i pour une période t , il suffit de déplacer la totalité de sa production sur une autre période t' . Pour favoriser cela, il suffit de prendre K le plus grand possible (K_{max}) : ce choix sera appelée stratégie K_{max} . Pour éliminer une quantité en stock, le choix systématique de K_{max} n'est pas judicieux dans le sens où les quantités à déplacer pour diminuer le stock peuvent être moindres. C'est pour cela qu'une quantité inférieure ou égale à K_{max} sera choisie : on appellera ceci la stratégie d'ajustement.

Nous définissons ainsi deux politiques différentes que nous choisissons d'appliquer selon une certaine probabilité α . C'est-à-dire que nous avons cherché un nombre α tel qu'en choisissant aléatoirement un nombre $\beta \in [0, 1]$:

- si $\beta \leq \alpha$ alors la stratégie d'ajustement sera appliquée,
- sinon ce sera la stratégie du K_{max} .

Après plusieurs tests, nous avons choisi de fixer la valeur de α à $\frac{2}{3}$.

Cependant, nous avons remarqué qu'en début d'optimisation, la politique K_{max} est la plus efficace mais que, chemin faisant, son efficacité diminue au bénéfice de décalages de moindre envergure. Implicitement, cela suppose qu'au départ de l'algorithme, la solution admissible est « assez grossière » donc que nous allons pouvoir déplacer de grandes quantités pour améliorer la qualité de celle-ci alors qu'après un certain temps, il deviendra plus difficile de trouver une amélioration par ce principe et que de petits décalages deviendront bien plus efficaces.

α étant fixé, cela modélise mal le souhait que nous avons de privilégier en début d'algorithme l'application de la stratégie du K_{max} . Pour ce faire, nous avons décidé de remplacer α par une fonction dont la valeur dépend du nombre d'itérations. Bien évidemment, nous voulions que celle-ci converge vers α pour préserver les effets bénéfiques de ce dernier sur les performances de l'algorithme.

En récapitulant, nous voulions une fonction qui privilégie fortement le choix du K_{max} en début d'algorithme et qui converge vers α pour que les ajustements prennent le pas de façon efficace sur l'autre stratégie.

L'idée était donc de trouver une application f fonction du nombre x d'itérations telle que :

- tant que le nombre d'itérations est petit, elle force le choix de K_{max} et après, de façon très progressive, elle permet les ajustements,
- elle converge vers α .

Pour le problème de convergence vers α nous avons pris le parti de prendre une fonction de la forme :

$$f(x) = \alpha - g(x)$$

où $g(x)$ est une fonction qui tend vers 0 en tenant compte de la « progressivité » voulue.

Il ne restait donc plus qu'à déterminer la fonction g .

Cette fonction g devait admettre 0 comme limite en l'infini. L'idée a donc été de considérer une fonction du type $g(x) = \frac{1}{h(x)}$ telle que $\lim_{x \rightarrow +\infty} h(x) = +\infty$.

Parmi les fonctions usuelles, nous nous sommes tournés vers la fonction logarithme népérien (\ln) car elle traduit bien cette notion de croissance progressive que nous voulions obtenir. Nous avons donc défini $h(x)$ par :

$$h(x) = \ln(1 + 10^{-3} \times x)$$

Un tel choix a été guidé par les deux considérations suivantes :

- la fonction h doit être toujours strictement positive ce qui signifie que l'argument du logarithme népérien doit être strictement supérieur à 1,
- pendant un certain nombre d'itérations, on doit systématiquement choisir la stratégie K_{max} donc il faut que $g(x) > \alpha$. Nous avons choisi le coefficient 10^{-3} car, dans ce cas, $g(x) > \alpha$ pendant environ 3500 itérations.

La fonction f que nous proposons est :

$$f(x) = \frac{2}{3} - \frac{1}{\ln(1 + 10^{-3} \times x)} \quad (3.36)$$

où x représente le nombre d'itérations.

L'algorithme définissant le voisinage 1 est illustré par la figure 3.10 :

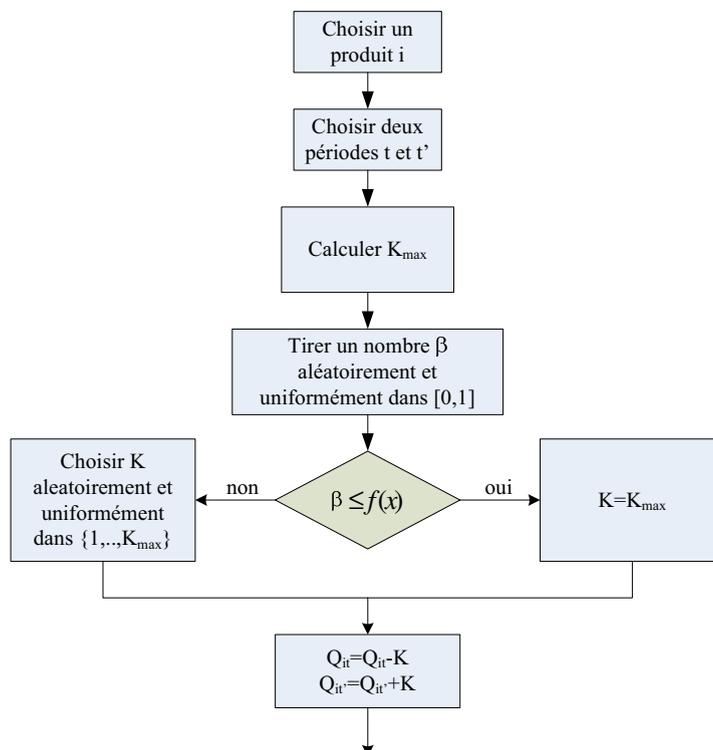


Figure 3.10 – Algorithme pour le voisinage 1

2.1.2.2 Voisinage 2

Le but de ce voisinage est de favoriser au maximum l'élimination des lancements de campagne les plus coûteux. Pour ce faire, les produits sont classés par ordre croissant de leur coût de lancement de campagne. Le choix du produit sur lequel le voisinage va agir est effectué aléatoirement. Cependant, on doit favoriser la désignation des plus coûteux : on va donc affecter à chaque produit, un intervalle dont la taille est fonction de son classement et qui permet ainsi de choisir prioritairement les plus coûteux. Ainsi, si le produit de type i est à la j^{eme} position du classement, on lui affectera l'intervalle entier $\llbracket 1 + \sum_{k=1}^{j-1} k, \sum_{k=1}^j k \rrbracket$ dont la taille est égale à son rang.

Le choix du produit sur lequel agit le voisinage s'effectue ainsi : un nombre α est tiré aléatoirement et uniformément dans l'intervalle $\llbracket 1, \frac{N(N+1)}{2} \rrbracket$ et appartiendra donc à l'intervalle d'un produit de type i . Une fois ce choix effectué, le voisinage procède de la façon suivante : on choisit deux périodes t et t' où il y a fabrication du produit i . Deux stratégies sont proposées :

- soit, à l'instar de la première politique du voisinage 1, on déplace la quantité maximale possible (K_{max}) de produit de type i de t à t' ,
- soit on essaie de déplacer l'intégralité de la fabrication de produit de type i de la période t jusqu'à la période $\max(t', t-1)$. Ce faisant, la capacité de cette période risque d'être dépassée, on applique donc un lissage de cette quantité sur des périodes où ce produit est également fabriqué : le choix de la période $\max(t', t-1)$ n'est pas anodin. En effet, si $t' > t$ alors on a tendance à éliminer du stock par contre si $t' < t-1$ alors déplacer la production en t' aurait tendance à générer un coût de stockage très important, d'où l'idée de déplacer la production en $t-1$.

Les deux stratégies seront choisies grâce à une probabilité : après plusieurs tests, nous avons choisi d'attribuer une probabilité de 0,8 pour la première d'entre elles.

Concernant le lissage, nous procédons de la façon suivante :

- si $t' < t$: pour l variant de $t-1$ à 1, on regarde si nous avons une quantité de produit de type i fabriquée durant la période l . Si oui, on calcule la quantité maximale déplaçable entre t et l : K_{max} et on déplace cette quantité de la période t à la période l . On s'arrête dès qu'on a pu éliminer toute la fabrication de produit de type i de la période t . Si on n'a pas pu déplacer l'intégralité de la fabrication à la période t , on laisse l'excédent de la production sur cette période.
- si $t < t'$, on lisse la production en partant de la période t' jusqu'à la première. On effectue les mêmes opérations que pour le cas $t' < t$, en faisant varier l de t' à 1, $l \neq t$.

L'algorithme définissant le voisinage 2 est illustré par la figure 3.11.

2.1.2.3 Voisinage 3

Le but de ce voisinage est de favoriser au maximum l'élimination des stocks les plus coûteux. Pour ce faire, à l'instar du voisinage 2, on va classer les produits par ordre croissant des coûts de stockage et on procédera au choix du produit en utilisant la même technique que dans le voisinage précédent. Par ailleurs, son principe de fonctionnement est pratiquement identique à ceci près qu'on va choisir dans ce cas, deux périodes t et t' telles que $t < t'$ et, dans l'application de la seconde stratégie, nous ferons varier l de t' à t , en prenant garde de ne pas rajouter de coût de lancement.

L'algorithme définissant le voisinage 3 est illustré par la figure 3.12.

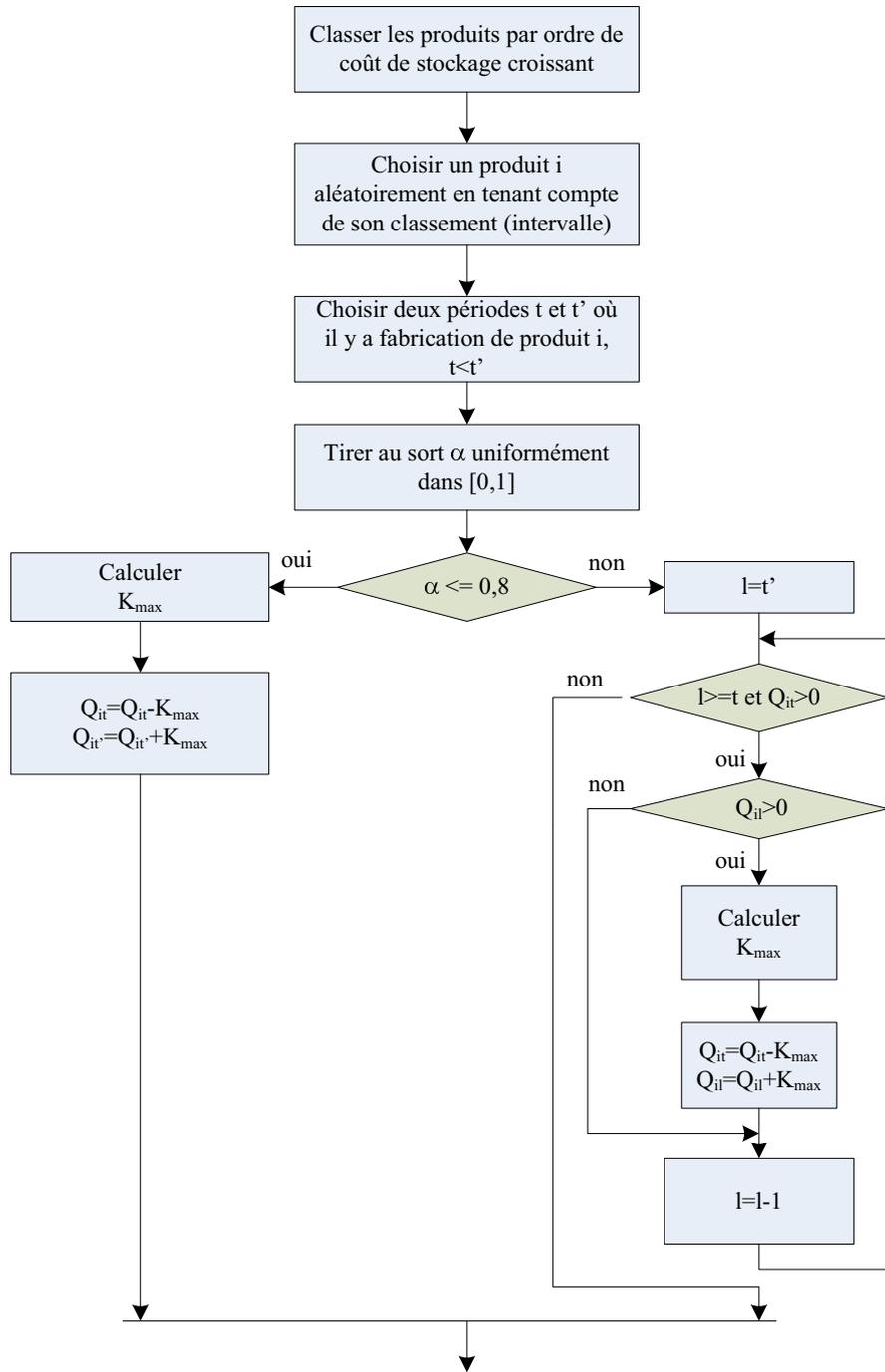


Figure 3.12 – Algorithme pour le voisinage 3

2.1.2.4 Voisinage 4

Ce voisinage va permettre d'échanger la fabrication de deux types de produits i et j entre deux périodes t et t' . Pour commencer, on choisit aléatoirement les deux produits i et j et les deux périodes t et t' , en prenant $t < t'$. Ensuite, en supposant qu'on ne fabrique plus de produit de type j durant la période t' , on calcule grâce à la proposition 3.3 la quantité maximale de produit de type i qu'on pourrait déplacer de t à t' (K_{max}). Ensuite, on suppose déplacée cette quantité et on regarde si la capacité du

système à la période t' est dépassée et si oui, de combien. Dans ce cas, on tente de rétablir le respect de la capacité en déplaçant une partie (la plus petite possible) de fabrication du produit de type j de la période t' à la période t . Si cet objectif n'est pas atteint, on diminue la valeur de K_{max} d'une unité et on réitère le processus jusqu'à ce que l'on ait déterminé la plus grande quantité possible déplaçable de produit de type i de la période t à la période t' pouvant être compensée par un déplacement d'une partie de la fabrication du produit de type j de la période t' à la période t . Ensuite, il ne reste plus qu'à procéder aux déplacements de production.

L'algorithme définissant le voisinage 4 est illustré par la figure 3.13.

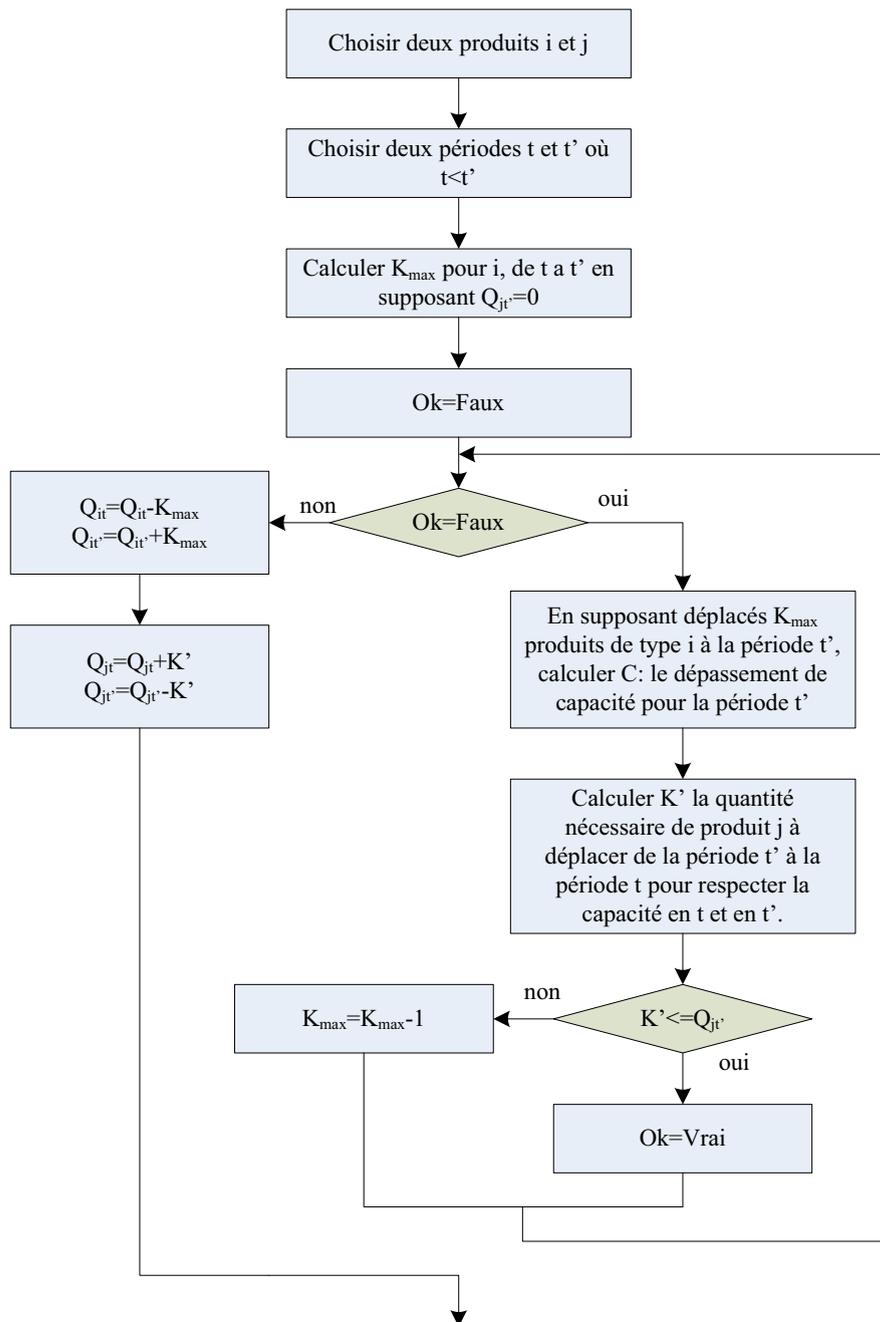


Figure 3.13 – Algorithme pour le voisinage 4

2.2 La métaheuristique « solution initiale »

Nous venons de décrire notre système de voisinage. Cependant, pour pouvoir l'utiliser, il faut pouvoir déterminer une solution initiale admissible, c'est à dire un planning faisable. Pour ce faire, il y a essentiellement deux manières différentes de rechercher cette solution admissible :

- on peut essayer de la construire,
- on peut, à partir d'une solution non admissible, essayer d'en déduire une par diverses techniques (perturbation, réparation...).

Les algorithmes basés sur une construction de la solution sont les plus nombreux ((Dixon et Silver 1981),(Lambrecht et Vanderveken 1979)...). Ils garantissent de trouver une solution initiale (s'il en existe une) : en effet, rappelons que le problème de faisabilité pour le CLSP est un problème de complexité polynômiale. D'autres, comme (Dogramaci et al. 1981) ont procédé par réparation. Nous nous situons dans cette seconde mouvance.

La question que nous nous sommes évidemment posée est de savoir si nous devons utiliser une heuristique connue qui nous assure de trouver une solution initiale. Nous avons pris le parti de ne rien en faire car, à l'instar de (Gopalakrishnan et al. 2001), nous voulions développer une approche facilement adaptable au CLSP avec Setup Time dont le problème de faisabilité est, quant à lui, NP-complet.

Dans notre état de l'art, nous avons remarqué qu'aucune tentative de résolution par une métaheuristique n'avait été tentée, c'est pourquoi nous avons décidé d'utiliser une métaheuristique à base de recuit simulé pour définir une nouvelle approche de résolution pour ce problème : une approche générique permettant de répondre au problème de faisabilité que ce soit au niveau du CLSP ou du CLSP avec setup time.

En reprenant le schéma défini par (Michalewicz et Fogel 2002), il nous faut :

- un codage d'une solution,
- une fonction objectif à optimiser qui traduise bien la notion de faisabilité,
- un système de voisinage.

Le codage que nous allons utiliser pour une solution est tout à fait indentique à celui proposé pour l'optimisation du CLSP : une matrice de production Q représentant le planning choisi. Une matrice Q sera considérée comme admissible pour le problème de faisabilité si elle respecte la demande donc, en conservant les notations, $I_Q \geq 0$.

2.2.1 La fonction objectif

Supposons que l'ensemble des solutions V que nous choisissons pour ce problème soit tel que toutes les solutions respectent la demande, c'est-à-dire que : $\forall Q \in V, I_Q \geq 0$. Pour qu'une solution soit faisable pour le CLSP, il suffit donc, d'après la proposition 3.2, qu'elle respecte la capacité du système de production.

Le but est donc de trouver une fonction objectif qui traduise le fait qu'il y ait ou non un dépassement de capacité. Si nous revenons à la proposition 3.2, une solution respecte la capacité du système de

production si et seulement si $\sum_{i=1}^N p_i Q_{it} \leq C_t, \forall t \in \llbracket 1, T \rrbracket$.

$$\begin{aligned} &\text{Or,} \\ &\sum_{i=1}^N p_i Q_{it} \leq C_t, \forall t \in \llbracket 1, T \rrbracket \text{ équivaut à} \\ &\sum_{i=1}^N p_i Q_{it} - C_t \leq 0, \forall t \in \llbracket 1, T \rrbracket \text{ ce qui équivaut à} \\ &\sum_{t=1}^T \text{Max} \left(0, \sum_{i=1}^N p_i Q_{it} - C_t \right) = 0. \end{aligned}$$

L'idée première fut donc de poser comme fonction objectif :

$$h(Q) = \sum_{t=1}^T \text{Max} \left(0, \sum_{i=1}^N p_i Q_{it} - C_t \right)$$

et de tenter de la minimiser.

Mais, le but étant de trouver une solution réalisable, nous voulions que cette fonction objectif force la métaheuristique à lisser la charge de production. Or, comme le montre la figure (3.14), notre fonction objectif peut nous donner la même valeur pour deux solutions qui ne sont pas équivalentes en termes de lissage de la charge.

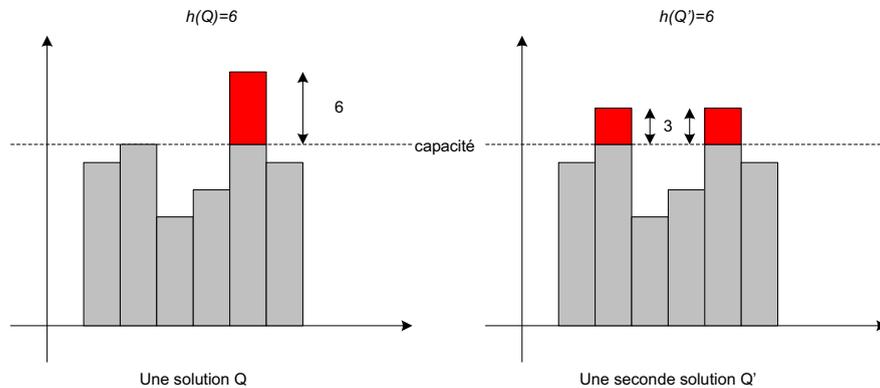


Figure 3.14 – Deux solutions de même coût

Nous avons donc modifié un peu notre fonction objectif initiale pour qu'elle prenne en compte cet aspect, en favorisant le lissage de la charge de production. Nous avons donc pris la fonction objectif suivante, qui traduit également le dépassement de capacité :

$$k(Q) = \sum_{t=1}^T \left[\text{Max} \left(0, \sum_{i=1}^N p_i Q_{it} - C_t \right) \right]^2$$

On peut voir maintenant que la seconde solution (figure 3.15) aura un coût plus faible que la première, c'était le résultat souhaité.

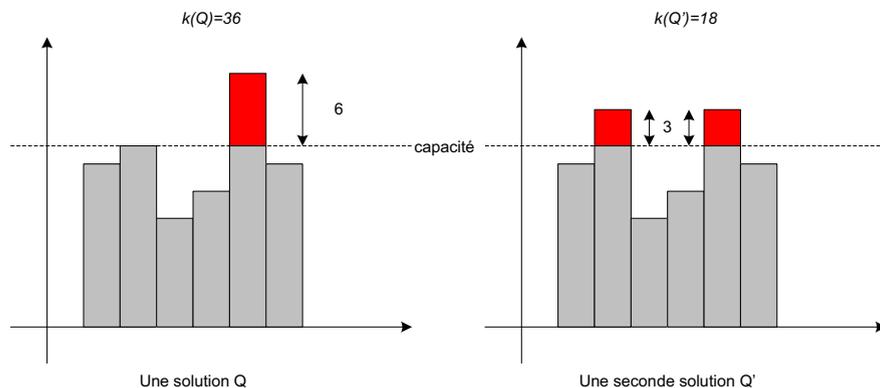


Figure 3.15 – Effet de la nouvelle fonction objectif sur les solutions précédentes

Il est clair que cette fonction objectif traduit la notion de faisabilité : en effet, lorsque celle-ci sera nulle, la solution sera faisable.

2.2.2 Le système de voisinage

Pour cette problématique, nous avons développé deux voisinages : les voisinages 5 et 6. Ces derniers diffèrent des précédents par leur non prise en compte des contraintes de capacité et s'appuie donc sur la proposition suivante qui est une paraphrase sans capacité de la proposition 3.3.

Proposition 3.4 *Soit Q une solution admissible pour le problème de faisabilité. Pour que le second système de voisinage fournisse une nouvelle solution Q^* admissible pour le problème de faisabilité, il faut et il suffit que :*

- si $t' \leq t$, alors

$$0 \leq K \leq Q_{it}$$

- $t' > t$, alors

$$0 \leq K \leq \min\{Q_{it}, \min_{t \leq l < t'} (I_{il})\}$$

2.2.2.1 Le voisinage 5 : non guidé sans capacité

Ce voisinage reprend le voisinage 1 (déplacement non guidé) mais ne tient plus compte de la capacité de la période t' , c'est-à-dire que K_{max} est calculé grâce à la proposition 3.4.

2.2.2.2 Le voisinage 6 : échange sans capacité

Ce voisinage est une version simplifiée du voisinage 4 (échange). Il permet d'échanger la fabrication de deux types de produits i et j entre deux périodes t et t' . Pour commencer, on choisit de façon aléatoire les deux produits i et j et les deux périodes t et t' , en prenant $t < t'$. On calcule grâce à la proposition 3.4 la quantité maximale de produit de type i qu'on pourrait déplacer de t à t' (K_{max}), ainsi que la quantité maximale de produit de type j qu'on pourrait déplacer de t' à t (K'_{max}). On déplace la quantité K de produit i choisie aléatoirement entre 1 et K_{max} et la quantité K' de produit i choisie aléatoirement entre 1 et K'_{max} .

L'algorithme définissant le voisinage 6 est illustré par la figure 3.16.

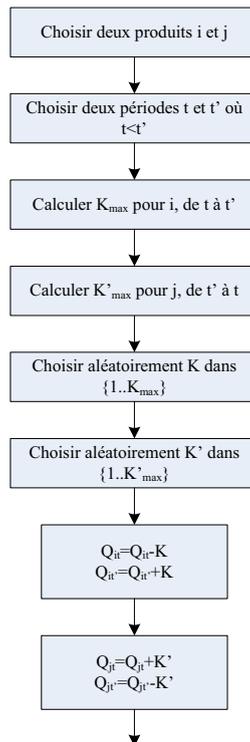


Figure 3.16 – Algorithme pour le voisinage 6

Après avoir mené quelques tests complémentaires, nous nous sommes aperçus que nous pouvions avoir des problèmes d'accessibilité dans notre système de voisinage. En effet, pour le problème du CLSP, une solution est admissible si et seulement si la demande est satisfaite sans retard et si la capacité du système est respectée. Le système de voisinage proposé permet de passer d'une solution admissible à une autre. Mais il se peut qu'il rencontre des problèmes dits d'accessibilité, c'est-à-dire qu'étant données deux solutions admissibles, il peut être très difficile, si ce n'est impossible, de passer de l'une à l'autre par une ou plusieurs applications du système de voisinage. Ce cas se rencontre surtout lorsque la capacité du système de production est à peine suffisante pour assurer la satisfaction de la demande.

C'est pour cela que nous avons développé une nouvelle stratégie pour pallier cette difficulté : l'utilisation d'une fonction bi-objectif.

3 La Métaheuristique « bi-objectif »

Pour assurer l'accessibilité, il faudrait pouvoir s'affranchir des contraintes de capacité, c'est-à-dire permettre au système de voisinage de s'extraire de l'espace des solutions admissibles pour le CLSP. Dans la suite, nous qualifierons de solution toute solution non admissible répondant à la demande (autrement dit, qui peut ne pas respecter la capacité du système). L'ensemble des solutions admissibles pour le CLSP est donc un sous-ensemble de ces solutions.

Le but étant de trouver une solution admissible pour le CLSP, il faut pouvoir en favoriser l'obtention. Pour ce faire, à chaque solution Q , est associée une évaluation formée par un couple de valeurs $(Cap(Q), f(Q))$, où :

- $Cap(Q)$ est fonction du dépassement de capacité engendré par la solution Q ($Cap(Q) = 0$ si les contraintes de capacité sont respectées, $Cap(Q) > 0$ sinon),
- $f(Q)$ le coût de celle-ci (déterminé par la fonction objectif (3.1)).

On va favoriser l'obtention d'une solution admissible en appliquant une relation d'ordre total sur l'ensemble des solutions de la manière suivante :

Définition 3.1 Soit deux solutions Q et Q' et soit $(Cap(Q), f(Q))$ et $(Cap(Q'), f(Q'))$ leurs évaluations respectives. On dira que Q est une meilleure solution que Q' si et seulement si :

- soit $Cap(Q) < Cap(Q')$
- soit $Cap(Q) = Cap(Q')$ et $f(Q) < f(Q')$.

Ainsi, toute solution admissible du CLSP sera meilleure que n'importe quelle solution non admissible pour le CLSP. Reste donc à définir la fonction $Cap(Q)$. Celle-ci doit vérifier la propriété suivante : $Cap(Q) = 0$ si la solution est admissible pour le CLSP, $Cap(Q) > 0$ sinon. Nous nous sommes donc tout naturellement tourné vers la fonction objectif quadratique proposée dans la métaheuristique « solution initiale ». La fonction bi-objectif est donc :

$$\left(\sum_{t=1}^T \left[\text{Max} \left(0, \sum_{i=1}^N p_i Q_{it} - C_t \right) \right]^2, \sum_{i=1}^N \left[\sum_{t=1}^T (s_i X_{it} + b_i Q_{it} + h_i I_{it}) \right] \right)$$

Comme nous permettons désormais de s'affranchir des contraintes de capacité, nous pouvons reprendre notre schéma d'optimisation initial et le modifier afin de supprimer la recherche de la solution initiale. Cependant, pour permettre de pleinement exploiter cette nouvelle fonction objectif et pour naviguer dans l'ensemble des solutions, nous incorporons les voisinages 5 et 6 à notre métaheuristique « optimisation ». La figure 3.17 donne l'algorithme de principe de notre métaheuristique munie de la fonction bi-objectif.

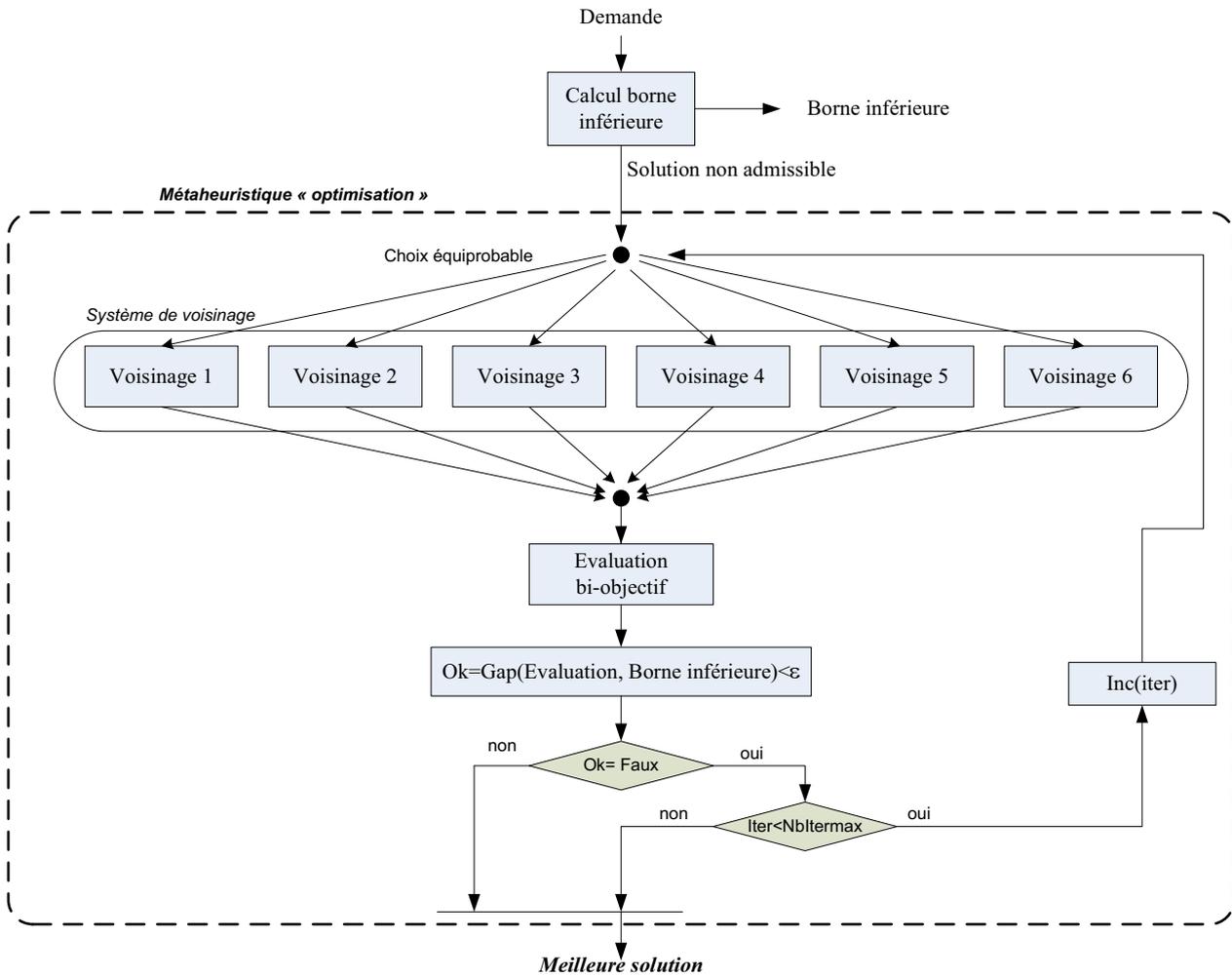


Figure 3.17 – Algorithme de principe de notre méthode d’optimisation avec fonction bi-objectif

4 Mise en œuvre et résultats

Dans cette partie, nous mettons en œuvre nos méthodes d’optimisation. Dans une première section, nous comparons les bornes inférieures obtenues par les deux méthodes envisagées. Puis, nous présentons les résultats obtenus par le chaînage des deux métaheuristiques. Ceux-ci sont comparés à ceux obtenus par la métaheuristique « bi-objectif ». Au vue de cette comparaison, nous utilisons la métaheuristique « bi-objectif » sur les instances de (Eppen et Martin 1987) et nous comparons nos résultats à ceux obtenus dans la littérature.

4.1 Comparaison des deux méthodes de calcul de borne inférieure

Pour comparer nos deux méthodes de calcul de borne inférieure, nous avons sélectionné quelques instances provenant de (Trigeiro et al. 1989) et notamment celles répertoriées par Wolsey (LotsizeLib 2006). Les instances considérées étant dédiées au problème du CLSP avec Setup Time, nous les avons modifiées afin de ne plus tenir compte des temps de lancement.

Nom	Nombre de produits	Nombre de périodes	gradient	sous-gradient
E1	6	15	34277,77	34277,78
E2	6	15	28784,99	28785,00
E5	6	15	40999,04	41003,67
E7	6	15	31456,69	31479,05
E8	6	15	37020,35	36792,01
E9	6	15	38298,92	38143,14
W14	4	15	49834,19	57091,04
W15	4	15	54456,62	56488,10
W19	8	15	137058,21	137595,38
W20	8	15	134503,60	136200,26
W22	8	15	84840,58	85518,07
G30	6	15	35512,42	36939,50
G53	12	15	70648,07	70717,77
G57	24	15	135914,08	135938,96
G62	6	30	60331,42	60627,16
G69	12	30	129537,63	129596,84
G72	24	30	287284,32	287390,89

Tableau 3.3 – Résultats du calcul de borne inférieure

Comme le montre le tableau 3.3, le calcul par la méthode de sous-gradient donne en général de meilleurs résultats que la méthode par gradient. Cependant, ce n'est pas toujours le cas. Dans le reste du chapitre, nous prendrons comme borne inférieure le maximum des bornes fournies par les deux méthodes.

4.2 Mise en œuvre et résultats pour le chaînage de métaheuristique

Celle-ci va se décomposer en trois étapes :

- la première étape consiste à calculer une borne inférieure au problème. Outre une borne inférieure, cela nous fournira une solution non admissible mais respectant les contraintes liées au stock ((3.2), (3.6)) que nous pourrions injecter dans la métaheuristique chargée de fournir une solution initiale,
- la seconde étape consiste à déterminer une solution admissible du problème. La métaheuristique utilisée s'initialise avec le plan déterminé par le calcul de la borne inférieure et utilise la fonction objectif quadratique et les voisinages 5 et 6. Dans le cas où une solution admissible est trouvée, elle servira de point de départ à la métaheuristique chargée de l'optimisation des coûts, sinon la procédure d'optimisation s'arrête.
- la dernière étape s'appuie également sur une métaheuristique basée sur le système de voisinage constitué des voisinages 1 à 4 dont la solution initiale sera la solution admissible déterminée par la seconde étape. Celle-ci s'arrêtera soit lorsque le nombre maximal d'itérations sera atteint, soit lorsque l'écart relatif (Gap) avec la borne inférieure sera inférieur à un certain ε .

Nous avons testé notre méthode d'optimisation sur le même jeu d'instances que pour la calcul de la borne inférieure. Parallèlement, nous avons utilisé un Solveur (Ilog Cplex 9.0) pour tenter de résoudre optimalement les instances traitées. Cependant, il est à noter qu'aucun paramétrage spécifique de Cplex n'a été effectué : le but n'étant pas d'obtenir une solution optimale à tout prix, mais uniquement de se servir du solveur comme référent. Pour chaque instance testée, nous avons effectué 5 répliquions de

notre méthode d'optimisation.

Les résultats obtenus sont synthétisés dans le tableau 3.4 et ont fait l'objet d'une publication dans (Gourgand et al. 2006). Pour le solveur, nous retrouvons la meilleure solution obtenue ainsi que son temps d'obtention. Pour la métaheuristique, nous avons référencé la meilleure solution obtenue après 5 réplifications de 40 millions itérations chacune, le temps et le nombre d'itérations nécessaires à son obtention. Nous trouvons ensuite le GAP qui est calculé :

- soit à partir de la solution optimale lorsqu'elle est donnée par le solveur :

$$GAP = \frac{SolutionObtenue - SolutionOptimale}{SolutionOptimale},$$

- sinon à partir de la borne inférieure :

$$GAP = \frac{SolutionObtenue - BorneInferieure}{BorneInferieure}.$$

Et pour finir, nous donnons la borne inférieure trouvée. Les résultats inscrits en caractère gras indiquent les meilleures solutions obtenues.

Nom	Taille	Solveur		chaînage de métaheuristiques				Borne inf
		Résultat	Temps	Résultat	Temps	Iter	Gap	Résultat
E1	6 × 15	34287,00	14,28 s	34287,00	11,0 s	200	0,00% ¹	34277,78
E2	6 × 15	28785,00	47,23 s	28785,00	10,0 s	1	0,00% ¹	28785,00
E5	6 × 15	41004,00	3167,91 s	41004,00	10,0 s	1	0,00% ¹	41003,67
E7	6 × 15	31485,00	4282,64 s	31493,00	334,8 s	1776023	0,02% ¹	31479,05
E8	6 × 15	37031,00	10,62 s	37047,00	597,5 s	4096886	0,04% ¹	37020,35
E9	6 × 15	38334,00	1929,91 s	38369,00	326,0 s	1015630	0,12% ¹	38298,92
W14	4 × 15	58864,75	55,6 s	59070,75	721,2 s	3061976	0,54% ¹	57091,04
W15	4 × 15	57257,50	21,7 s	57516,00	1451,8 s	2593705	1,06% ¹	56488,10
G30	6 × 15	37156,00	7540,6 s	37271,00	7087,1 s	17403985	1,26% ¹	36939,50
W19	8 × 15	140752,00 *	10349,00 s	140764,00	54273,8 s	13914637	2,52%	137595,38
W20	8 × 15	138709,00 *	8840,00 s	138242,00	3684,9 s	12708316	1,50%	136200,26
W22	8 × 15	86859,00 *	8509,50 s	87302,00	893,7 s	1791715	2,08%	85518,07
G53	12 × 15	72831,00 *	19020,00 s	71306,00	12115,8 s	30461122	0,84%	70717,77
G57	24 × 15	137762,00 *	3475,2 s	136877,00	3779,8 s	4741011	0,70%	135938,96
G62	6 × 30	62058,00 *	3671,1 s	61168,00	8270,3 s	8249727	1,46%	60627,16
G69	12 × 30	132097,00 *	3671,5 s	131280,00	573,8 s	395665	1,31%	129596,84
G72	24 × 30	295291,00 *	3201,3 s	289977,00	588,5	379541	0,9%	287390,89

(*) Solutions non optimales obtenues après overflow du solveur

⁽¹⁾ GAP calculé à partir de la solution optimale

Tableau 3.4 – Résultats de la méthode d'optimisation sur les instances répertoriées

Nous pouvons voir que les résultats obtenus sont plutôt encourageants. En effet, en se comparant au solveur, nous pouvons voir que nous surpassons régulièrement celui-ci dans les instances les plus grandes et que les Gaps obtenus par rapport à notre borne inférieure sont, somme toute, assez mineurs.

4.3 Mise en œuvre et résultats pour la métaheuristique « bi-objectif »

Dans cette partie, nous comparons les résultats obtenus avec la fonction bi-objectif à ceux obtenus précédemment (Tableau 3.4).

Pour chaque instance, nous avons référencé la meilleure solution obtenue après 5 réplifications de 40 millions itérations chacune et inscrit en caractère gras la meilleure solution obtenue. Le GAP est calculé par la formule suivante :

$$GAP = \frac{SolutionMonoObjectif - SolutionBiObjectif}{SolutionMonoObjectif}.$$

Le tableau 3.5 fournit les résultats obtenus.

Comme nous pouvons le remarquer, nous améliorons très sensiblement les résultats obtenus dans (Gourgand et al. 2006), ce qui n'est pas très étonnant car nous avons pu nous abstraire du problème d'accessibilité du système voisinage, grâce à la fonction bi-objectif.

Nom	Taille	chaînage de métaheuristiques		La métaheuristique bi-objectif		Gap
		Résultat obtenu	Itération	Résultat obtenu	Itération	
E1	6 × 15	34287,00	200	34287,00	1	0 %
E2	6 × 15	28785,00	1	28785,00	1	0 %
E5	6 × 15	41004,00	1	41004,00	1	0 %
E7	6 × 15	31493,00	1776023	31485,00	494	0,02 %
E8	6 × 15	37047,00	4096886	37031,00	1856	0,04 %
E9	6 × 15	38369,00	1015630	38334,00	769	0,09 %
W14	4 × 15	59070,75	3061976	58989	14236359	0,09 %
W15	4 × 15	57516,00	2593705	57264,5	17131745	0,43 %
G30	6 × 15	37271,00	17403985	37156,00	667982	0,31 %
W19	8 × 15	140764,00	13914637	140794,00	6977595	-0.02%
W20	8 × 15	138242,00	12708316	138065,00	19425013	0,13 %
W22	8 × 15	87302,00	1791715	86634,00	23632149	0,76 %
G53	12 × 15	71306,00	30461122	71016,00	3181148	0,41 %
G57	24 × 15	136877,00	4741011	135988,00	2489	0,65 %
G62	6 × 30	61168,00	8249727	60971,00	1009744	0,77 %
G69	12 × 30	131280,00	395665	130324,00	14028255	0,73 %
G72	24 × 30	289977,00	379541	288030,00	455854	0,67 %

Tableau 3.5 – Tableau comparatif des résultats pour les deux métaheuristiques

La métaheuristique utilisant la fonction bi-objectif obtenant les meilleurs résultats sur les instances de (Trigeiro et al. 1989) que nous avons retravaillées, nous allons l'appliquer aux instances de (Eppen et Martin 1987) que nous avons pris le parti de considérer comme les instances de référence pour le CLSP.

4.4 Tests sur les instances de (Eppen et Martin 1987)

Dans cette partie, nous présentons les résultats obtenus par notre métaheuristique munie de la fonction bi-objectif sur les instances de (Eppen et Martin 1987).

Pour chaque instance, nous avons effectué 5 réplifications de 40 millions itérations. Les résultats obtenus sont donnés dans le tableau 3.6.

Dans le tableau 3.7, nous nous comparons aux résultats de la littérature.

Nom	Taille	B.Inf	B.Sol	W.Sol	M.Sol	σ .Sol	B.Gap	W.Gap	M.Gap	σ .Gap
MARTIN1	8 × 8	7837,59	(0,8430)	(0,8870)	(0,8695,4)	472,759	7,5585	13,1724	10,9447	6,0319
MARTIN10	100 × 10	20899,27	(0,20913,2)	(0,20945)	(0,20923,16)	17,4515	0,0666	0,2187	0,1142	0,0835
MARTIN11	100 × 10	20950,39	(0,20956,4)	(0,20958,2)	(0,20957,68)	1,0264	0,0286	0,0372	0,0347	0,0048
MARTIN12	100 × 10	21033,47	(0,21076,2)	(0,21136,7)	(0,21098,892)	32,2888	0,2031	0,4907	0,311	0,1535
MARTIN13	100 × 10	21192,63	(0,21294)	(0,21375,2)	(0,21341,54)	50,5072	0,4783	0,8614	0,7026	0,2383
MARTIN14	100 × 10	21236,63	(0,21311,2)	(0,21420,4)	(0,21355,64)	58,1618	0,3511	0,8653	0,5603	0,2738
MARTIN15	100 × 10	21288,78	(0,21404)	(0,21504,6)	(0,21467,18)	58,231	0,5411	1,0137	0,8379	0,2735
MARTIN16	100 × 10	22916,95	(89,87,22567,82)	(89,87,22588,7)	NA	NA	NA	NA	NA	NA
MARTIN17	200 × 10	43631,94	(0,43812,54)	(0,43978,10)	(0,43914,24)	65,7409	0,4139	0,7933	0,6469	0,1506
MARTIN2	8 × 8	7722,27	(0,7918)	(0,7952)	(0,7940)	34,1872	2,5345	2,9748	2,8194	0,4427
MARTIN3	8 × 8	7534,16	(0,7610)	(0,7620)	(0,7616)	12,5524	1,0065	1,1392	1,0861	0,1666
MARTIN4	8 × 8	7464,16	(0,7520)	(0,7560)	(0,7528)	41,2349	0,748	1,2839	0,8551	0,5524
MARTIN5	50 × 8	48472,64	(0,48591)	(0,48831)	(0,48741,2)	86,9157	0,2441	0,7393	0,554	0,1793
MARTIN6	100 × 8	93991,21	(0,94234)	(0,94500)	(0,94380,2)	80,9489	0,2583	0,5413	0,4138	0,0861
MARTIN7	150 × 8	137361,14	(0,137537)	(0,138234)	(0,137906,8)	151,8985	0,128	0,6354	0,3972	0,1105
MARTIN8	200 × 8	187400,31	(0,187700)	(0,187860)	(0,187786)	33,2172	0,1599	0,2452	0,2058	0,0177
MARTIN9	20 × 13	5668,36	(0,5860,1)	(0,6058,3)	(0,5934,356)	196,3156	3,3825	6,8791	4,6925	3,4633

Tableau 3.6 – Résultats obtenus par la métaheuristique pour le CLSP sur les instances de (Eppen et Martin 1987)

NA : Non Applicable

Nombre d'instances traitées : 17

Gap Moyen par rapport à notre borne inférieure : 1,13 %

Fonction bi-objectif : $(\underbrace{Argument1}_{\text{Capacité}}, \underbrace{Argument2}_{\text{Coût}})$

Nom	Taille	B.Inf Epp	B.Inf	Gap	Epp	Hin D.W.	Hin H.1	Hin H.2	Hin OSL	Hin Tabou	Best Known	B.Sol	Gap
MARTIN1	8 × 8	7996,67	7837,59	1,98	8430	8430	9060	8670	8430	8730	8430	8430	0
MARTIN10	100 × 10	20899,28	20899,27	0	20923	20929,3	20929,3	20929,3	20911,2	20929,3	20911,2	20913,2	-0,01
MARTIN11	100 × 10	20950,39	20950,39	0	20962	21015,89	21069,85	21069,85	20990,07	21069,85	20962	20956,4	0,02
MARTIN12	100 × 10	21033,47	21033,47	0	21082,1	21121,83	21267,65	21267,65	21059,59	21267,65	21082,1	21076,2	0,02
MARTIN13	100 × 10	21192,63	21192,63	0	21336,2	21295,1	21319,82	21319,82	21257	21319,82	21257	21294	-0,18
MARTIN14	100 × 10	21236,63	21236,63	0	21381,43	21338,84	21368,47	21368,82	21320,11	21368,47	21338,84	21311,2	0,12
MARTIN15	100 × 10	21288,78	21288,78	0	21402,1	21400,92	21463,34	21463,34	21369,86	21463,34	21369,86	21404	-0,16
MARTIN16	100 × 10	Inf.	22916,95	NA	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.	(89,87,22567,82)	NA
MARTIN17	200 × 10	43631,64	43631,94	-0,01	43710,2	43775,2	43809,38	43809,38	43650	43809,36	43650	43812,54	-0,38
MARTIN2	8 × 8	7722,27	7722,27	0	7910	7910	8940	8230	7910	8250	7910	7918	-0,11
MARTIN3	8 × 8	7534,17	7534,16	0	7610	7610	7820	7610	7610	7690	7610	7610	0
MARTIN4	8 × 8	7464,17	7464,16	0	7520	7520	7520	7520	7520	7520	7520	7520	0
MARTIN5	50 × 8	48472,64	48472,64	0	49129,5	48686,2	49761	49178	48690	49001,9	48686,2	48591	0,19
MARTIN6	100 × 8	93991,22	93991,21	0	94379,7	94318,4	95941,4	95004,2	97130	95681,3	94318,4	94234	0,08
MARTIN7	150 × 8	137361,14	137361,14	0	137819	137829,1	140539,1	139949,1	138772,3	139755,2	137819	137537	0,2
MARTIN8	200 × 8	187503,25	187400,31	0,05	188122	188296,7	192234,2	190344,2	189646,3	191274,4	188122	187700	0,22
MARTIN9	20 × 13	5668,37	5668,36	0	5807,66	5814,52	5860,33	5834,3	5812	5860,31	5807,66	5860,1	-0,91

Tableau 3.7 – Comparaison des résultats obtenus avec ceux de la littérature

NA : Non Applicable

Inf. : infaisable

Nombre d'instances traitées : 17

Gap Moyen par rapport à la meilleure solution connue : -0,06 %

Les instances fournies par (Eppen et Martin 1987) ont été collectées ou générées par les auteurs à partir des instances données par (Dixon et Silver 1981) et (Thizy et Van Wassenhove 1983). Plus précisément, les instances Martin1-4 sont issues de (Thizy et Van Wassenhove 1983), Martin5-8 ont été générées par (Eppen et Martin 1987) à partir de Martin1, Martin9 est une instance de (Dixon et Silver 1981) et Martin10-17 ont été générées par les auteurs à partir de cette dernière.

Dans le tableau 3.6, nous trouvons le nom de l'instance ainsi que sa taille (produits×périodes), la borne inférieure obtenue (B.Inf), la meilleure solution obtenue (B.Sol), la pire (W.sol) et la moyenne (M.Sol) ainsi que l'écart type (σ .Sol) sur les 5 réplifications effectuées. Nous donnons ensuite le GAP obtenu par rapport à la borne inférieure : le meilleur (B.Gap), le pire (W.Gap), le moyen (M.Gap) ainsi que l'écart-type (σ .Gap). Le GAP est calculé par la formule :

$$GAP = \frac{\text{Solution obtenue} - \text{Borne inférieure}}{\text{Borne inférieure}}$$

Outre le nom et la taille de chaque instance, nous retrouvons dans le tableau 3.7, la borne inférieure donnée par (Eppen et Martin 1987) (B.inf Eppen), notre borne inférieure (B.Inf) ainsi que le GAP obtenu, puis les résultats fournis par (Eppen et Martin 1987), (Hindi 1995) (Hindi D.W. - un branch & Bound couplé avec une décomposition de Dantzig Wolfe-, Hindi H.1, Hindi H.2 - deux heuristiques-) et (Hindi 1996) (Hindi OSL, Hindi Tabou -une reformulation du modèle résolu avec le solveur OSL et une méthode Tabou-). Ensuite nous donnons la meilleure solution connue dans la littérature (Best Known -fournie dans (Hindi 1996)-) puis notre solution ainsi que son GAP par rapport à la meilleure solution connue. En caractère gras figure la meilleure solution obtenue.

Comme nous pouvons le constater, nous obtenons de très bons résultats grâce à notre métaheuristique. Nous améliorons même les meilleures solutions connues pour les instances Martin5, 6, 7, 8, 11 et 14. A l'exception de l'instance Martin17, nous surpassons systématiquement la solution obtenue par la méthode Tabou de (Hindi 1996). Pour l'instance Martin16 qui n'a pas de solution admissible, nous trouvons une solution non admissible dont la première coordonnée (la pénalisation de la capacité) avoisine les 89.

4.5 Conclusion

Dans cette partie, nous avons proposé une métaheuristique s'appuyant sur une fonction bi-objectif permettant de s'affranchir des contraintes de capacité permettant ainsi l'accessibilité de notre système de voisinage. Nous avons testé notre méthode sur les instances de référence de la littérature et nous avons obtenu de très bons résultats allant même jusqu'à l'amélioration des meilleures solutions de la littérature. Ces résultats ont fait l'objet d'une communication dans (Gourgand et al. 2008).

IV Adaptation au CLSP avec Setup Time

Nous avons vu que les résultats obtenus par notre métaheuristique munie de la fonction bi-objectif étaient plus que probants. A l'instar de (Gopalakrishnan et al. 2001), nous allons maintenant essayer de l'appliquer au CLSP avec Setup-Time. Cependant, certaines modifications mineures sont à effectuer.

1 Prise en compte des temps de lancements

Prendre en compte les temps de lancements va entraîner quelques adaptations de notre méthode d'optimisation.

Ecrivons le modèle du CLSP avec Setup Time en reprenant les notations du début de chapitre :

$$\text{Minimiser } \sum_{i=1}^N \left[\sum_{t=1}^T (s_i X_{it} + b_i Q_{it} + h_i I_{it}) \right] \quad (3.1)$$

Sous les contraintes :

$$I_{it} = I_{i(t-1)} + Q_{it} - d_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.2)$$

$$\sum_{i=1}^N (\tau_{it} X_{it} + p_i Q_{it}) \leq C_t \quad \forall t \in \llbracket 1, T \rrbracket \quad (3.25)$$

$$\sum_{i=1}^N p_i Q_{it} \leq C_t \quad \forall t \in \llbracket 1, T \rrbracket \quad (3.4)$$

$$X_{it} \in \{0, 1\} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.5)$$

$$I_{it}, Q_{it} \geq 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.6)$$

Nous voyons que c'est la contrainte de capacité (3.25) qui diffère. Or celle-ci a un impact au niveau de la relaxation lagrangienne (puisque sa violation est pénalisée dans la fonction objectif), au niveau de la fonction bi-objectif et dans les propositions 3.2 et 3.3 qui sous-tendent la validité du système de voisinage. Il faut donc inclure dans ces dernières cette nouvelle spécificité.

1.1 Le calcul de la borne inférieure

Pour commencer, écrivons le problème relaxé du CLSP :

$$\text{Minimiser } \sum_{i=1}^N \left[\sum_{t=1}^T (s_i X_{it} + b_i Q_{it} + h_i I_{it}) \right] + \langle \lambda, \left(\sum_{i=1}^N (\tau_{it} X_{it} + p_i Q_{it}) - C_t \right)_{t \in \llbracket 1, T \rrbracket} \rangle \quad (3.37)$$

Sous les contraintes :

$$I_{it} = I_{i(t-1)} + Q_{it} - d_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.38)$$

$$Q_{it} \leq M X_{it} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.39)$$

$$X_{it} \in \{0, 1\} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.40)$$

$$I_{it}, Q_{it} \geq 0 \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (3.41)$$

Nous noterons U l'ensemble défini par les contraintes (3.38), (3.39), (3.40) et (3.41).
Formons le lagrangien de ce problème :

$$\begin{aligned} \mathcal{L} : U \times (\mathbb{R}^T)^+ &\longrightarrow \mathbb{R} \\ (Q, \lambda) &\longmapsto \sum_{i=1}^N \left[\sum_{t=1}^T (s_i X_{it} + b_i Q_{it} + h_i I_{it}) \right] + \langle \lambda, \left(\sum_{i=1}^N (\tau_{it} X_{it} + p_i Q_{it}) - C_t \right)_{t \in \llbracket 1, T \rrbracket} \rangle \end{aligned}$$

Définissons maintenant la fonction duale associée au problème :

$$\begin{aligned} h : W &\longrightarrow \mathbb{R} \\ \lambda &\longmapsto \min_{Q \in U} \mathcal{L}(Q, \lambda) \end{aligned}$$

où $W = \{\lambda \in (\mathbb{R}^T)^+ \text{ tel que } \min_{Q \in U} \mathcal{L}(Q, \lambda) > -\infty\}$.

Il nous faut donc résoudre :

$$\max_{\lambda \in W} h(\lambda)$$

Les méthodes employées pour résoudre ce problème vont être identiques à celles vues précédemment à ceci près que pour tout $\lambda \geq 0$ les dérivées partielles sont $\frac{\partial h}{\partial \lambda_t}(\lambda) = \sum_{i=1}^N (\tau_{it} X_{it} + p_i Q_{it}) - C_t$.

1.2 La fonction bi-objectif

Comme nous l'avons vu, la fonction bi-objectif s'écrit comme un vecteur ayant comme première composante une fonction quadratique exprimant la faisabilité en termes de capacité, et comme seconde le coût engendré par la solution déterminée. Seule la première composante est impactée par la prise en compte des temps de lancement. Reprenons le raisonnement effectué auparavant pour établir la première composante : une solution respecte la capacité du système de production si et seulement si

$$\sum_{i=1}^N (\tau_{it} X_{it} + p_i Q_{it}) \leq C_t, \quad \forall t \in \llbracket 1, T \rrbracket.$$

Or,

$$\sum_{i=1}^N (\tau_{it} X_{it} + p_i Q_{it}) \leq C_t, \quad \forall t \in \llbracket 1, T \rrbracket \text{ équivaut à}$$

$$\sum_{i=1}^N (\tau_{it} X_{it} + p_i Q_{it}) - C_t \leq 0, \quad \forall t \in \llbracket 1, T \rrbracket \text{ ce qui équivaut à}$$

$$\sum_{t=1}^T \text{Max} \left(0, \sum_{i=1}^N (\tau_{it} X_{it} + p_i Q_{it}) - C_t \right) = 0.$$

Nous prendrons donc comme première composante :

$$\sum_{t=1}^T \left[\text{Max} \left(0, \sum_{i=1}^N (\tau_{it} X_{it} + p_i Q_{it}) - C_t \right) \right]^2$$

La fonction bi-objectif s'écrit donc :

$$\left(\sum_{t=1}^T \left[\text{Max} \left(0, \sum_{i=1}^N (\tau_{it} X_{it} + p_i Q_{it}) - C_t \right) \right]^2, \sum_{i=1}^N \left[\sum_{t=1}^T (s_i X_{it} + b_i Q_{it} + h_i I_{it}) \right] \right)$$

1.3 Le système de voisinage

Comme nous l'avons dit, le système de voisinage s'appuie sur les propositions 3.2 et 3.3. Leur réécriture pour la prise en compte des contraintes de setup est immédiate.

La proposition 3.2 devient :

Proposition 3.5

$$\mathcal{Q} \text{ est admissible} \Leftrightarrow \begin{cases} \mathcal{I}_{\mathcal{Q}} \geq 0. \\ \sum_{i=1}^N (\tau_{it} X_{it} + p_i Q_{it}) \leq C_t \quad \forall t \in \llbracket 1, T \rrbracket \end{cases}$$

La proposition 3.3 se transforme en :

Proposition 3.6 Soit \mathcal{Q} une solution admissible, pour que l'algorithme présenté dans la figure 3.7 fournisse une nouvelle solution \mathcal{Q}^* admissible il faut et il suffit que :

- si $t' \leq t$, alors

$$0 \leq K \leq \min \left\{ E \left[\frac{C_{t'} - \left(\sum_{j=1, j \neq i}^N (\tau_{jt'} X_{jt'} + p_j Q_{jt'}) \right) + \tau_{it'} + p_i Q_{it'}}{p_i} \right], Q_{it} \right\}$$

- $t' > t$, alors

$$0 \leq K \leq \min \left\{ E \left[\frac{C_{t'} - \left(\sum_{j=1, j \neq i}^N (\tau_{jt'} X_{jt'} + p_j Q_{jt'}) \right) + \tau_{it'} + p_i Q_{it'}}{p_i} \right], Q_{it}, \min_{t \leq l < t'} (I_{il}) \right\}$$

Il suffira donc de tenir compte de ces deux propositions dans le choix de K_{max} dans les voisinages 1, 2 3 et 4.

2 Mise en œuvre et résultats

Nous avons vu comment adapter simplement notre métaheuristique au CLSP avec Setup-Time. Nous allons maintenant pouvoir la tester sur les instances de référence que sont celles définies par (Trigeiro et al. 1989). A l'instar des tests menés avec les instances de (Maes et Van Wassenhove 1986b) pour le CLSP, nous avons fondé notre expérimentation sur 5 répliquions de notre méthode d'optimisation. Nous nous sommes comparés aux résultats fournis par l'heuristique de (Trigeiro et al. 1989) dont le code nous a été fourni par les auteurs. Les résultats obtenus sont extrêmement encourageants, et cela pour deux raisons :

- à la fin de l'optimisation, nous trouvons, pour chaque instance, une solution admissible alors que le problème de faisabilité est NP-complet : cela démontre l'efficacité de la fonction bi-objectif,
- nous améliorons sensiblement les résultats obtenus par l'heuristique de (Trigeiro et al. 1989) que ce soit au niveau de la borne inférieure ou du résultat de l'optimisation : sur les 581 instances actuellement testées, nous améliorons la solution sur 398 d'entre-elles.

Le tableau 3.8 synthétise les résultats obtenus

Taille des instances	Nombre	Nombre d'améliorations	GAP moyen / Borne inférieure
4 × 15	6	6	4,36
6 × 15	174	163	2,85
8 × 15	6	6	2,23
12 × 15	5	2	1,33
24 × 15	5	1	2,41
6 × 30	5	5	2,22
12 × 30	5	4	0,19
24 × 30	5	1	0,41
10 × 20	180	131	3,14
20 × 20	118	54	1,91
30 × 20	72	25	0,95
Récapitulatif	581	398	2,45

Tableau 3.8 – Résultats pour le CLSP avec setup time

On y retrouve la taille des instances traitées (nombre de produits × nombre de périodes), le nombre d'instances traitées de cette taille, le nombre de fois où nous améliorons la solution donnée par l'heuristique de (Trigeiro et al. 1989) et le GAP moyen obtenu pour chaque groupe d'instances par rapport à la borne inférieure donnée par (Trigeiro et al. 1989).

Avec sa méthode Tabou, (Gopalakrishnan et al. 2001) annonce un GAP moyen de 3% sur l'ensemble des instances de 15 et 30 périodes : notre méthode obtient un gap moyen de 2,69%. Sur les instances à 20 périodes, ils annoncent un GAP moyen de 4% en justifiant celui-ci par la présence d'instances de grande taille. Pour notre part, le GAP moyen obtenu est de 2,32% et c'est justement sur les instances de grande taille que les scores obtenus sont les meilleurs. Cependant, il faut reconnaître que c'est également sur ces instances-là que (Trigeiro et al. 1989) obtiennent les résultats les plus convaincants : cela tend à confirmer la conjecture de (Du Merle et al. 1997) quant à la pertinence des heuristiques

lagrangienne. Cependant, il ne faut pas perdre de vue que la méthode que nous proposons n'est pas dédiée au problème du CLSP avec Setup Time : l'élaboration d'un système de voisinage pour ce problème permettrait sans doute une amélioration des résultats obtenus. Au vue des résultats obtenus, on peut tout de même conclure que notre métaheuristique bi-objectif rivalise avec l'heuristique de (Trigeiro et al. 1989).

Conclusion

Dans ce chapitre, nous avons présenté deux méthodes d'optimisation pour le Capacitated lot Sizing Problem. L'une est basée sur le chaînage de deux métaheurstiques, l'autre sur une métaheuristique utilisant un fonction bi-objectif. Bien qu'efficace, la première méthode était confronté à des problèmes d'accessibilité du système de voisinage : la fonction bi-objectif de la seconde a permis, en s'affranchissant des contraintes de capacité, d'assurer cette accessibilité tout en favorisant l'obtention de solutions réalisables. Nous avons testé ces méthodes sur les instances de la littérature dont nous avons parfois amélioré les meilleures solutions connues.

Fort des bons résultats de la métaheuristique bi-objectif, nous l'avons adaptée pour prendre en compte les temps de lancement (setup time) et nous l'avons testée sur les instances de (Trigeiro et al. 1989) pour lesquelles nous avons obtenu de très bons résultats : en effet, outre le fait de trouver de meilleures résultats que l'heuristique de (Trigeiro et al. 1989) dans plus de 66% des cas, nous avons surpassé ceux trouvés par (Gopalakrishnan et al. 2001) et leur méthode Tabou.

Bien que cette fonction bi-objectif ait été employée pour le CLSP, son principe est facilement réemployable pour divers types de problèmes (pour peu que l'on sache hiérarchiser les contraintes relaxées par ordre d'importance) : elle est actuellement reprise avec succès par un élève ingénieur CNAM dans un problème de dimensionnement et d'ordonnancement de blocs opératoires, dans le cadre de son année à plein temps.

Chapitre 4

Méthodes approchées pour la planification d'une chaîne logistique de type Flow-Shop Hybride

Sommaire

Introduction	160
I Modèle mathématique pour une chaîne logistique de type flow-shop hybride . . .	161
1 Notations utilisées	162
1.1 Notations pour les ressources de production	162
1.2 Notations pour les ressources de stock	163
1.3 Notations pour les ressources de transport	164
2 Les contraintes du modèle mathématique	164
2.1 Les contraintes liées aux usines	164
2.2 Les contraintes liées aux stocks	165
3 La fonction objectif	168
4 Le modèle mathématique définitif	168
II Etat de l'art	170
1 Méthodes pour le MLLP	170
2 Méthodes pour le MLCLSP	174
2.1 MLCLSP sans setup time	174
2.2 MLCLSP avec setup time	175
3 Conclusion	181
III Méthodes d'optimisation pour la planification d'une chaîne logistique de type Flow-Shop Hybride	181
1 Résolution du problème de planification des « usines étages »	182
1.1 Codage de la solution	182
1.2 Système de voisinage	182
2 Résolution du problème d'affectation	191
3 La méthode d'optimisation pour le problème de planification de la chaîne logistique de type flow shop hybride	192
4 Mise en œuvre et résultats	192
4.1 Tests préliminaires	192
4.2 Résultats sur des instances de taille industrielle	194
IV Vers une nouvelle approche	195
1 Résolution du problème « usines étages »	195
2 Résolution du problème d'affectation	197
3 Une nouvelle méthode d'optimisation	198
Conclusion	198

Introduction

Ce chapitre s'appuie sur un cas d'étude concernant la planification d'une chaîne logistique interne constituée de six usines : U1, U2, U3, U4, U5 et U6. La figure 4.1 présente le synoptique des flux suivis par les produits au travers de cette chaîne logistique.

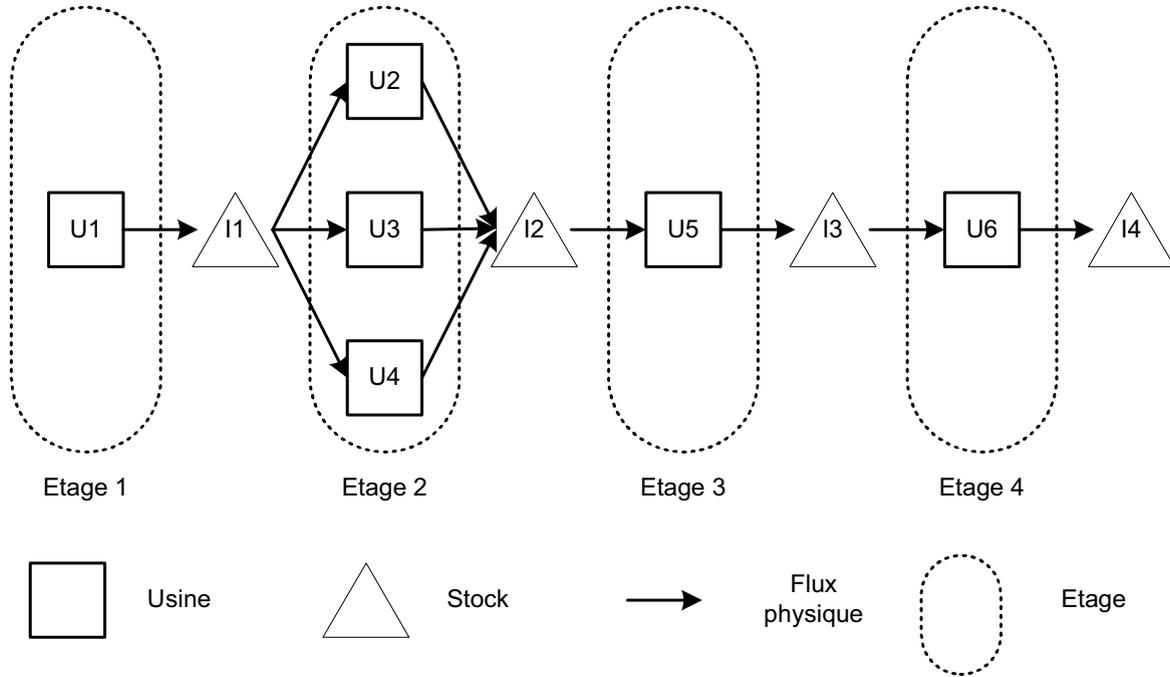


Figure 4.1 – Cas d'étude

La nomenclature des produits est considérée comme en série, bien que des composants soient adjoints à ceux-ci au fur et à mesure de leurs transformations : la problématique de planification se limite au produit principal. Une contrainte dite « de périodicité » est considérée : un produit obtenu en période t (donc stocké en t) n'est disponible pour un autre site de production ou pour un client qu'au début de la période $t + 1$.

La chaîne logistique étudiée est constituée de quatre étages. Au second étage, chaque produit peut être transformé indépendamment par les trois usines le composant. A chaque étage est assigné un stock où les produits qui auront été transformés par une usine le composant seront entreposés. Ainsi, à l'instar des structures d'ateliers étudiés par la communauté d'ordonnancement, nous appelons cette structure : chaîne logistique de type Flow Shop Hybride.

L'objectif de l'étude est de proposer une planification tactique pour l'ensemble de la chaîne logistique, consistant à générer pour chaque usine, un plan directeur de production sur un horizon de 12 mois à la maille semaine, ces différents plans devant être synchronisés sur l'ensemble de la chaîne.

Dans ce chapitre, nous commençons par donner un modèle mathématique pour une chaîne logistique de type Flow Shop Hybride. Celui-ci est obtenu par instantiation du modèle mathématique générique présenté au chapitre II. La relaxation de certaines contraintes de ce modèle permet de se ramener à un problème de MLCLSP à nomenclature en série : un état de l'art pour le MLCLSP est donc proposé dans une seconde partie. Ensuite, nous proposons des méthodes d'optimisation permettant de répondre à cette problématique.

I Modèle mathématique pour une chaîne logistique de type flow-shop hybride

Dans cette partie, nous donnons un modèle mathématique pour la planification tactique d'une chaîne logistique de type Flow Shop Hybride sous les hypothèses considérées dans le cas d'étude (contraintes de périodicité, nomenclature linéaire). La figure 4.2 représente une telle chaîne logistique.

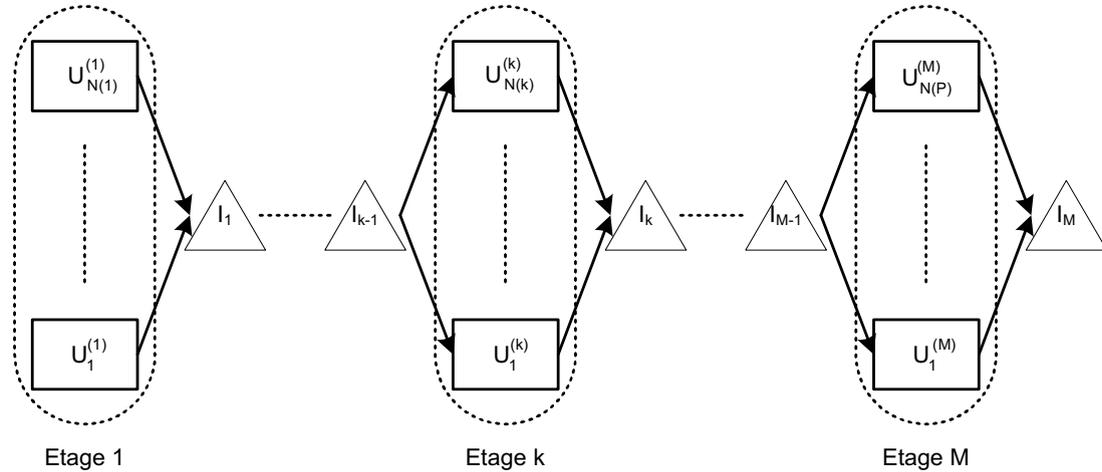


Figure 4.2 – Chaîne logistique de type Flow Shop Hybride

Pour obtenir ce modèle mathématique, nous allons instancier le modèle générique donné au chapitre II sur cette typologie de chaîne logistique.

Dans le contexte qui nous préoccupe, les fournisseurs et les centres de distribution ne sont pas pris en compte. Nous reprenons les notations du modèle générique en posant :

- N : Le nombre de produits finis dont on planifie la fabrication,
- T : Le nombre de périodes composant l'horizon de planification.

Pour respecter la structure de la chaîne logistique considérée, nous définissons :

- M : Le nombre d'étages de la chaîne logistique,
- $N(k)$: Le nombre d'usine composant l'étage k , $k \in \llbracket 1, M \rrbracket$.

Pour obtenir un produit fini i , celui-ci doit passer successivement sur chaque étage de la chaîne logistique pour y être transformé. Ainsi, nous pouvons noter par i_k la k^{eme} transformation du produit i , ce qui correspond à « l'état » du produit i après être passé sur l'étage k . Par soucis d'homogénéité, nous notons i_M le produit fini i . On notera donc :

- $D_{i_M,t}$: La demande client en produit i_M à la période t , $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$.

Pour chaque produit i_M , il est aisé d'écrire sa nomenclature au travers d'une matrice gozinto qui prend alors une forme particulièrement simple (la transposée d'un bloc de jordan nilpotent) :

$$A_{i_M} = \begin{array}{c|ccccccc} & i_M & i_{M-1} & i_{M-2} & \cdots & i_3 & i_2 & i_1 \\ \hline i_M & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ i_{M-1} & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ i_{M-2} & 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & 0 & 0 & \ddots & \cdots & \vdots & \vdots & \vdots \\ i_3 & 0 & 0 & \cdots & \ddots & 0 & 0 & 0 \\ i_2 & 0 & 0 & \cdots & \cdots & 1 & 0 & 0 \\ i_1 & 0 & 0 & \cdots & \cdots & 0 & 1 & 0 \end{array}$$

Dans la suite, toutes les variables sont supposées entières et positives sauf mention contraire.

1 Notations utilisées

Dans cette partie, nous présentons les notations utilisées pour élaborer le modèle mathématique

1.1 Notations pour les ressources de production

Dans la chaîne logistique considérée, chaque usine est caractérisée par :

- une unique ressource de production (mono-ressource),
- une capacité finie par période,
- l'étage auquel elle appartient : en effet, une usine appartenant à l'étage k ne pourra fabriquer que les produits i_k , $i \in \llbracket 1, N \rrbracket$.

Le temps de configuration et les délais d'obtention ne sont pas pris en compte et il n'y pas de possibilité d'ajouter de la capacité supplémentaire. Etant constants sur l'horizon, les coûts de production ne sont pas considérés.

En se conformant aux notations définies dans le modèle générique, les paramètres liés aux usines de production seront donc :

- $CapaR_{u_j^{(k)}, t}$: La capacité de l'usine $u_j^{(k)}$ de l'étage k à la période t
 $(i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket$, $j \in \llbracket 1, N(k) \rrbracket$
- $Capa_{i_k, u_j^{(k)}, t}$: La capacité consommée pour la fabrication du produit i_k sur l'usine $u_j^{(k)}$ de l'étage k à la période t , $(i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket$, $j \in \llbracket 1, N(k) \rrbracket$
- $CL_{i_k, u_j^{(k)}, t}$: Coût de configuration de l'usine $u_j^{(k)}$ de l'étage k pour fabriquer du produit i_k à la période t , $(i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket$, $j \in \llbracket 1, N(k) \rrbracket$

Nous notons les variables de décision liées à la fabrication des produits et la configuration des usines par :

- $Q_{i_k, u_j^{(k)}, t}$: La quantité de produits i_k fabriquée par l'usine $u_j^{(k)}$ de l'étage k à la période t
 $(i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket$, $j \in \llbracket 1, N(k) \rrbracket$
- $X_{i_k, u_j^{(k)}, t}$: $\begin{cases} 1 & \text{si l'usine } u_j^{(k)} \text{ de l'étage } k \text{ a été configurée pour fabriquer des produits} \\ & i_k \text{ durant la période } t \\ 0 & \text{sinon} \end{cases}$
 $(i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket$, $j \in \llbracket 1, N(k) \rrbracket$

1.2 Notations pour les ressources de stock

C'est au niveau de la définition des stocks que réside l'adaptation la plus importante. En effet, dans le modèle générique, à chaque usine est associée deux stocks : le stock « ante » et le stock « post ». Or, dans la typologie considérée, les stocks sont définis par étage : à chaque étage est associé un stock.

La figure 4.3 illustre la façon dont nous allons instancier le modèle pour les stocks.

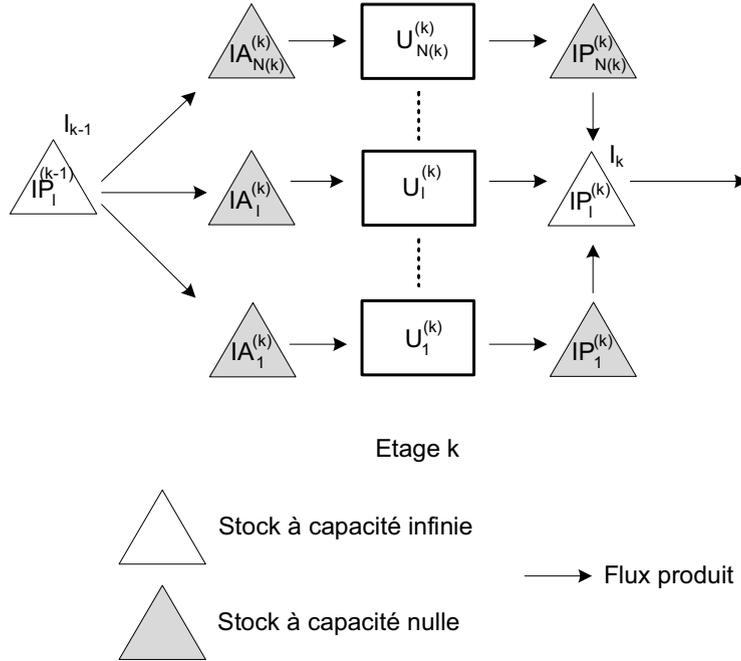


Figure 4.3 – Instanciation des stocks

À chaque étage k , nous choisissons une usine $u_l^{(k)}$ à laquelle nous affectons le stock de l'étage comme stock « post » qui est à capacité infinie. Pour les autres stocks, ils sont à capacité nulle. Les stocks de sécurité, les backorders et les coûts fixes d'utilisation des stocks ne sont pas pris en compte. La circulation entre les stocks s'effectuera grâce aux moyens de transport.

Les produits finis sont disponibles uniquement dans le stock I_M c'est à dire dans le stock $IP_l^{(M)}$: on aura donc de la demande servie et perdue uniquement pour ce stock. En accord avec les notations du modèle générique, nous définissons les paramètres suivants : ($\Lambda \in \{IA, IP\}$)

$\Lambda_{i_k, u_j^{(k)}, 0}^+$: stock initial de produit i_k dans le stock Λ de l'usine $u_j^{(k)}$ de l'étage k
 $(i, k) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket, j \in \llbracket 1, N(k) \rrbracket$

$CDPIP_{i_M, u_l^{(M)}, t}$: coût d'une unité de demande perdue en produit i_M pour le stock IP de l'usine $u_l^{(M)}$ de l'étage M , $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$ à la période t

Nous notons les variables de décision associées aux stocks par :

$\Lambda_{i_k, u_j^{(k)}, t}^+$: quantité de produit i_k dans le stock Λ de l'usine $u_j^{(k)}$ de l'étage k
à la période t , $(i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket, j \in \llbracket 1, N(k) \rrbracket$

$DSIP_{i_M, u_l^{(M)}, t}$: demande servie en produit i_M pour le stock IP de l'usine $u_l^{(M)}$ de l'étage M
 $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$ à la période t

$DPPIP_{i_M, u_l^{(M)}, t}$: demande perdue en produit i_M pour le stock IP de l'usine $u_l^{(M)}$ de l'étage M
 $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$ à la période t

Nous n'avons pas fait mention de coût de stockage. Ceci est dû aux contraintes de périodicité. En

effet, dire qu'un produit obtenu en période t (donc stocké en t) n'est disponible pour un autre site de production ou pour un client qu'au début de la période $t + 1$ implique nécessairement qu'il restera en stock au moins une période, ce qui rend constant une partie des coûts engendrés par le stockage. Cela nous amène donc à définir la notion de « surplus » comme étant la quantité de produit restant en stock au moins deux périodes et nous lui associons un coût (dit coût de surplus) : cette quantité sera donc totalement optimisable. Il est clair qu'on ne peut pas parler de surplus pour un stock à capacité nulle. Nous notons donc :

$$\begin{aligned}
 IP_{i_k, u_l^{(k)}, t}^S & : \text{ quantité de produits } i_k \text{ en surplus dans le stock «post» de l'usine } u_j^{(l)} \text{ de l'étage} \\
 & \quad k \text{ à la période } t, (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \\
 CIP_{i_k, u_l^{(k)}, t}^S & : \text{ coût d'une unité de produit } i_k \text{ en surplus dans le stock «post» de l'usine } u_l^{(k)} \\
 & \quad \text{de l'étage } k \text{ à la période } t, (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket
 \end{aligned}$$

1.3 Notations pour les ressources de transport

Comme illustré par la figure 4.3, les seuls moyens de transport admis sont ceux qui relient :

- pour chaque étage $k \in \llbracket 1, M \rrbracket$, les stocks « post » des usines d'un étage k , autre que l'usine $u_l^{(k)}$, au stock «post» de l'usine $u_l^{(k)}$,
- pour chaque étage $k \in \llbracket 2, M \rrbracket$, le stock « post » de l'usine $u_l^{(k-1)}$ aux stocks «ante» de chaque usine de l'étage k .

Ceux-ci sont à capacité infinie et leur utilisation n'engendre aucun coût.

Conformément au modèle générique, nous notons :

$$\begin{aligned}
 TIP_{i_k, u_j^{(k)}, u_l^{(k)}, t} & : \text{ quantité de produit } i_k \text{ transportée du stock «post» de l'usine } u_j^{(k)} \text{ de} \\
 & \quad \text{l'étage } k \text{ au stock «post» de l'usine } u_l^{(k)} \text{ de l'étage } k \text{ à la période } t, \\
 & \quad (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket, j \in \llbracket 1, N(k) \rrbracket - \{l\} \\
 TIPIA_{i_{k-1}, u_l^{(k-1)}, u_j^{(k)}, t} & : \text{ quantité de produit } i_{k-1} \text{ transportée du stock «post» de l'usine } u_l^{(k-1)} \\
 & \quad \text{de l'étage } k - 1 \text{ au stock «ante» de l'usine } u_j^{(k)} \text{ de l'étage } k \text{ à la période} \\
 & \quad t, (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 2, M \rrbracket \times \llbracket 1, T \rrbracket, j \in \llbracket 1, N(k) \rrbracket
 \end{aligned}$$

2 Les contraintes du modèle mathématique

Nous utilisons donc les notations définies précédemment, les notations du modèle mathématique générique, les contraintes du modèle générique et les spécificités du problème décrites précédemment.

2.1 Les contraintes liées aux usines

La contrainte de capacité (2.1) devient :

$$\sum_{i=1}^N \left(\text{Capa}_{i_k, u_j^{(k)}, t} Q_{i_k, u_j^{(k)}, t} \right) \leq \text{CapaR}_{u_j^{(k)}, t} \quad (4.1)$$

$$\forall (k, t) \in \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket, \forall j \in \llbracket 1, N(k) \rrbracket$$

Les contraintes (2.2), (2.3), (2.4) et (2.6) n'ont plus d'objet. La contrainte de lancement de production (2.5) devient :

$$\text{Capa}_{i_k, u_j^{(k)}, t} Q_{i_k, u_j^{(k)}, t} \leq \text{CapaR}_{u_j^{(k)}, t} X_{i_k, u_j^{(k)}, t} \quad (4.2)$$

$$\forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket, \forall j \in \llbracket 1, N(k) \rrbracket$$

2.2 Les contraintes liées aux stocks

Dans un premier temps, plaçons-nous à l'étage k , $k \in \llbracket 2, M \rrbracket$ et regardons ce que devient la contrainte (2.7) relative au stock « ante ». La figure 4.4 illustre le principe de fonctionnement de ces stocks à l'étage k .

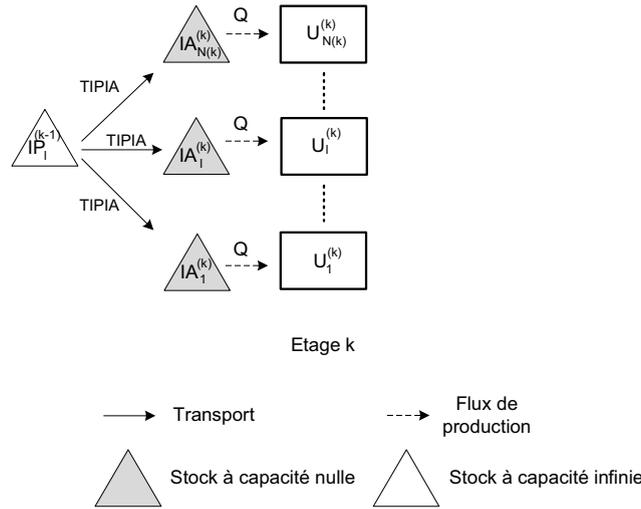


Figure 4.4 – Les stocks « ante » de l'étage k

En utilisant le fait que tous les éléments de la matrice « gozinto » sont nuls à l'exception de la sous-diagonale, celle-ci va donc s'écrire

$$IA_{i_k, u_j^{(k)}, t}^+ = IA_{i_k, u_j^{(k)}, t-1}^+ + TIPIA_{i_{k-1}, u_i^{(k-1)}, u_j^{(k)}, t} - a_{i_{k-1} i_k} Q_{i_k, u_j^{(k)}, t}$$

$\forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 2, M \rrbracket \times \llbracket 1, T \rrbracket, \forall j \in \llbracket 1, N(k) \rrbracket$

En utilisant la forme particulière de la matrice « gozinto » ($a_{i_{k-1} i_k} = 1 \forall k \in \llbracket 2, M \rrbracket$), et en utilisant le fait que tous les stocks « ante » sont à capacité nulle, nous obtenons :

$$TIPIA_{i_{k-1}, u_i^{(k-1)}, u_j^{(k)}, t} = Q_{i_k, u_j^{(k)}, t} \quad (4.3)$$

$\forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 2, M \rrbracket \times \llbracket 1, T \rrbracket, \forall j \in \llbracket 1, N(k) \rrbracket$

Pour ce qui est du premier étage ($k = 1$), cette contrainte n'existe évidemment pas.

Dans un second temps, plaçons-nous à l'étage k , $k \in \llbracket 1, M \rrbracket$ et regardons ce que devient la contrainte (2.8) relative au stock « post ». La figure 4.5 illustre le principe de fonctionnement de ces stocks à l'étage k , $k \in \llbracket 1, M - 1 \rrbracket$.

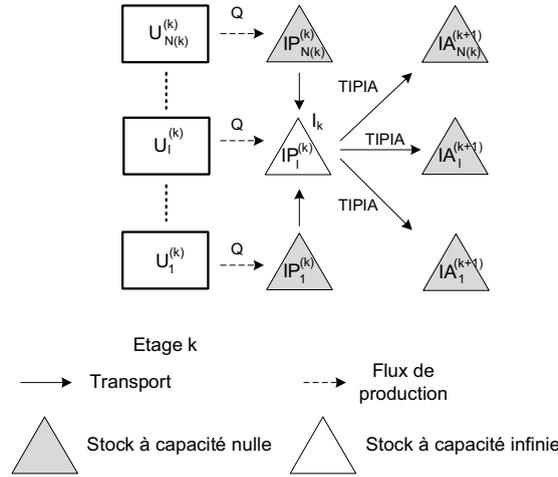


Figure 4.5 – Les stocks « post » de l'étage k

Deux cas sont à considérer :

- le cas de l'usine $u_l^{(k)}$,
- le cas des usines $u_j^{(k)}$, $j \in \llbracket 1, N(k) \rrbracket - \{l\}$.

Pour les usines $u_j^{(k)}$, $j \in \llbracket 1, N(k) \rrbracket - \{l\}$, la contrainte (2.8) devient

$$IP_{i_k, u_j^{(k)}, t}^+ = IP_{i_k, u_j^{(k)}, t-1}^+ - TIPIP_{i_k, u_j^{(k)}, u_l^{(k)}, t} + Q_{i_k, u_j^{(k)}, t}$$

$$\forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket, \forall j \in \llbracket 1, N(k) \rrbracket$$

Comme les stocks « post » de ces usines sont à capacité nulle, nous obtenons donc :

$$TIPIP_{i_k, u_j^{(k)}, u_l^{(k)}, t} = Q_{i_k, u_j^{(k)}, t} \tag{4.4}$$

$$\forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket, \forall j \in \llbracket 1, N(k) \rrbracket$$

Pour l'usine $u_l^{(k)}$, l'écriture de la contrainte diffère selon l'étage. La figure 4.6 illustre le principe de fonctionnement de ces stocks à l'étage M .

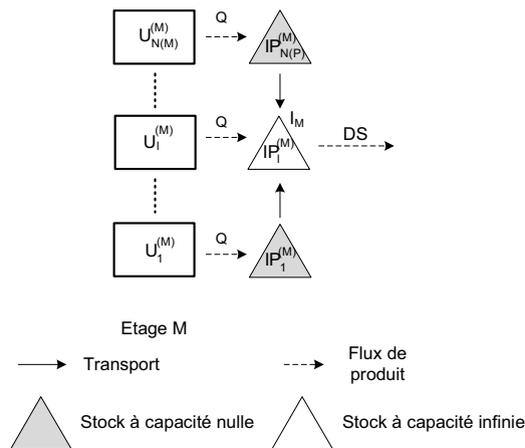


Figure 4.6 – Les stocks « post » de l'étage M

- sur l'étage $k \in \llbracket 1, M-1 \rrbracket$, la contrainte s'écrit :

$$IP^+_{i_k, u_l^{(k)}, t} = IP^+_{i_k, u_l^{(k)}, t-1} + Q_{i_k, u_l^{(k)}, t} + \sum_{j=1, j \neq l}^{N(k)} TIIPI_{i_k, u_j^{(k)}, u_l^{(k)}, t} - \sum_{j=1}^{N(k+1)} TIIPIA_{i_k, u_l^{(k)}, u_j^{(k+1)}, t}$$

En utilisant les contraintes (4.3) et (4.4), nous obtenons :

$$IP^+_{i_k, u_l^{(k)}, t} = IP^+_{i_k, u_l^{(k)}, t-1} + Q_{i_k, u_l^{(k)}, t} + \sum_{j=1, j \neq l}^{N(k)} Q_{i_k, u_j^{(k)}, t} - \sum_{j=1}^{N(k+1)} Q_{i_{k+1}, u_j^{(k+1)}, t}$$

Soit

$$IP^+_{i_k, u_l^{(k)}, t} = IP^+_{i_k, u_l^{(k)}, t-1} + \sum_{j=1}^{N(k)} Q_{i_k, u_j^{(k)}, t} - \sum_{j=1}^{N(k+1)} Q_{i_{k+1}, u_j^{(k+1)}, t} \quad (4.5)$$

$\forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M-1 \rrbracket \times \llbracket 1, T \rrbracket$

- sur l'étage M , la contrainte s'écrit :

$$IP^+_{i_M, u_l^{(M)}, t} = IP^+_{i_M, u_l^{(M)}, t-1} + Q_{i_M, u_l^{(M)}, t} + \sum_{j=1, j \neq l}^{N(M)} TIIPI_{i_M, u_j^{(M)}, u_l^{(M)}, t} - DSIP_{i_M, u_l^{(M)}, t}$$

Soit, en utilisant la contrainte (4.4)

$$IP^+_{i_M, u_l^{(M)}, t} = IP^+_{i_M, u_l^{(M)}, t-1} + \sum_{j=1}^{N(M)} Q_{i_M, u_j^{(M)}, t} - DSIP_{i_M, u_l^{(M)}, t} \quad (4.6)$$

$\forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$

Les autres contraintes du modèle générique liées aux stocks ((2.9) à (2.21)) ne sont pas actives si ce n'est la contrainte (2.17) qui lie les demandes clients, perdues et servies. Cela s'écrit donc :

$$DSIP_{i_M, u_l^{(M)}, t} - DPIP_{i_M, u_l^{(M)}, t} = D_{i_M, t} \quad (4.7)$$

$\forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$

Les contraintes de périodicité imposent qu'un produit obtenu en période t (donc stocké en t) n'est disponible pour un autre site de production ou pour un client qu'au début de la période $t+1$, cela signifie que :

- sur un étage k , $k \in \llbracket 2, M \rrbracket$, on ne peut pas fabriquer plus de produit i_k à la période $t+1$ que ce qu'il y avait en stock en produit i_{k-1} à l'étage $k-1$ à la période t . Cela s'écrit donc :

$$\sum_{j=1}^{N(k)} Q_{i_k, u_j^{(k)}, t+1} \leq IP^+_{i_{k-1}, u_l^{(k-1)}, t} \quad (4.8)$$

- pour la demande client, on ne peut pas donner au client plus de produit i_M à la période $t+1$ que ce qu'il y avait en stock en produit i_M à l'étage M à la période t :

$$DSIP_{i_M, u_l^{(M)}, t+1} \leq IP^+_{i_M, u_l^{(M)}, t} \quad (4.9)$$

Il ne reste plus qu'à évaluer les « surplus », cela s'écrit grâce aux contraintes suivantes :

$$IP_{i_k, u_l^{(k)}, t}^S = IP_{i_k, u_l^{(k)}, t}^+ - \sum_{j=1}^{N(k+1)} Q_{i_{k+1}, u_j^{(k+1)}, t+1} \quad (4.10)$$

$$\forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M-1 \rrbracket \times \llbracket 1, T-1 \rrbracket$$

$$IP_{i_k, u_l^{(k)}, T}^S = IP_{i_k, u_l^{(k)}, T}^+ \quad (4.11)$$

$$\forall (i, k) \in \llbracket 1, N \rrbracket \times \llbracket 1, M-1 \rrbracket$$

$$IP_{i_M, u_l^{(M)}, t}^S = IP_{i_M, u_l^{(M)}, t}^+ - DSIP_{i_M, u_l^{(M)}, t+1} \quad (4.12)$$

$$\forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T-1 \rrbracket$$

$$IP_{i_M, u_l^{(M)}, T}^S = IP_{i_M, u_l^{(M)}, T}^+ \quad (4.13)$$

$$\forall i \in \llbracket 1, N \rrbracket$$

Les contraintes de périodicité et de surplus n'apparaissent pas dans le modèle générique. Ces contraintes font partie du travail d'adaptation de ce dernier pour prendre en compte des spécificités particulières du problème considéré.

Concernant les moyens de transport, ils ne font l'objet d'aucune contrainte ni de coût inhérent à leur utilisation : les contraintes (2.22) à (2.26) ne sont donc pas activées.

3 La fonction objectif

Le but de ce modèle est de fournir, pour la chaîne logistique, le plan de production ayant le coût le plus faible possible. Dans les coûts, nous avons :

- les coûts de configuration des usines,
- les coûts de surplus,
- les coûts de demande perdue.

La fonction objectif du modèle s'écrit :

$$\begin{aligned} \text{Minimiser } & \sum_{i=1}^N \sum_{t=1}^T \sum_{k=1}^M \left[\sum_{j=1}^{N(k)} \left(CL_{i_k, u_j^{(k)}, t} X_{i_k, u_j^{(k)}, t} \right) + CIP_{i_k, u_l^{(k)}, t}^S IP_{i_k, u_l^{(k)}, t}^S \right] \\ & + \sum_{i=1}^N \sum_{t=1}^T CDPIP_{i_M, u_l^{(M)}, t} DPPIP_{i_M, u_l^{(M)}, t} \end{aligned} \quad (4.14)$$

4 Le modèle mathématique définitif

Dans les sections précédentes, nous avons présenté les notations et les contraintes du modèle mathématique pour la planification d'une chaîne logistique de type Flow Shop Hybride. Cependant, les contraintes (4.3) et (4.4) montrent :

- l'inutilité de définir les variables liées au transport,
- l'inutilité de définir, pour chaque étage k , les variables liés aux stocks « ante » et « post » de chaque usine si ce n'est le stock « post » de l'usine $u_l^{(k)}$ (qui est le stock rattaché à l'étage k).

En partant de ces constats, il est possible d'une part de supprimer toutes les notations liées au transport, liées aux stocks « ante » et « post » de chaque usine de chaque étage k à l'exception de celles relatives à l'usine $u_l^{(k)}$ et d'autre part d'alléger les notations liées aux stocks « post » des usines $u_l^{(k)}$ en les rattachant de nouveau à l'étage k , conformément à la figure 4.2.

Pour cela, nous utiliserons les notations suivantes :

- $CDP_{i_M,t}$: coût d'une unité de demande perdue en produit i_M pour le stock de l'étage M ,
 $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
- $I_{i_k,t}^+$: quantité de produit i_k dans le stock de l'étage k à la période t ,
 $(i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket$
- $DS_{i_M,t}$: demande servie en produit i_M pour le stock de l'étage M
 $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
- $DP_{i_M,t}$: demande perdue en produit i_M pour le stock de l'étage M
 $(i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket$
- $I_{i_k,t}^S$: quantité de produit i_k en surplus dans le stock de l'étage k à la période t ,
 $(i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket$
- $CI_{i_k,t}^S$: coût d'une unité de produit i_k en surplus dans le stock de l'étage k à la période t , $(i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket$

En utilisant ces nouvelles notations, le modèle mathématique s'écrit :

$$\text{Minimiser } \sum_{i=1}^N \sum_{t=1}^T \sum_{k=1}^M \left[\sum_{j=1}^{N(k)} \left(CL_{i_k, u_j^{(k)}, t} X_{i_k, u_j^{(k)}, t} \right) + CI_{i_k t}^S I_{i_k, t}^S \right] + \sum_{i=1}^N \sum_{t=1}^T CDP_{i_M, t} DP_{i_M, t} \quad (4.15)$$

Sous les contraintes :

$$\sum_{i=1}^N \left(Capa_{i_k, u_j^{(k)}, t} Q_{i_k, u_j^{(k)}, t} \right) \leq Capa R_{u_j^{(k)}, t} \quad \forall (k, t) \in \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket, \forall j \in \llbracket 1, N(k) \rrbracket \quad (4.16)$$

$$Capa_{i_k, u_j^{(k)}, t} Q_{i_k, u_j^{(k)}, t} \leq Capa R_{u_j^{(k)}, t} X_{i_k, u_j^{(k)}, t} \quad \forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket, \forall j \in \llbracket 1, N(k) \rrbracket \quad (4.17)$$

$$I_{i_k, t}^+ = I_{i_k, t-1}^+ + \sum_{j=1}^{N(k)} Q_{i_k, u_j^{(k)}, t} - \sum_{j=1}^{N(k+1)} Q_{i_{k+1}, u_j^{(k+1)}, t} \quad \forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M-1 \rrbracket \times \llbracket 1, T \rrbracket \quad (4.18)$$

$$I_{i_M, t}^+ = I_{i_M, t-1}^+ + \sum_{j=1}^{N(M)} Q_{i_M, u_j^{(M)}, t} - DS_{i_M, t} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (4.19)$$

$$DS_{i_M, t} - DP_{i_M, t} = D_{i_M, t} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (4.20)$$

$$\sum_{j=1}^{N(k)} Q_{i_k, u_j^{(k)}, t+1} \leq I_{i_{k-1}, t}^+ \quad \forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 2, M \rrbracket \times \llbracket 1, T \rrbracket \quad (4.21)$$

$$DS_{i_M, t+1} \leq I_{i_M, t}^+ \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (4.22)$$

$$I_{i_k,t}^S = I_{i_k,t}^+ - \sum_{j=1}^{N(k+1)} Q_{i_{k+1},u_j^{(k+1)},t+1} \quad \forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M-1 \rrbracket \times \llbracket 1, T-1 \rrbracket \quad (4.23)$$

$$I_{i_k,T}^S = I_{i_k,T}^+ \quad \forall (i, k) \in \llbracket 1, N \rrbracket \times \llbracket 1, M-1 \rrbracket \quad (4.24)$$

$$I_{i_M,t}^S = I_{i_M,t}^+ - DS_{i_M,t+1} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T-1 \rrbracket \quad (4.25)$$

$$I_{i_M,T}^S = I_{i_M,T}^+ \quad \forall i \in \llbracket 1, N \rrbracket \quad (4.26)$$

$$I_{i_k,t}^+, I_{i_k,t}^S \in \mathbb{N} \quad \forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket \quad (4.27)$$

$$X_{i_k,u_j^{(k)},t} \in \{0, 1\}, Q_{i_k,u_j^{(k)},t} \in \mathbb{N} \quad \forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket, \forall j \in \llbracket 1, N(k) \rrbracket \quad (4.28)$$

$$DS_{i_M,t}, DP_{i_M,t} \in \mathbb{N} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (4.29)$$

Si nous nous plaçons dans le cas où chaque étage n'est composé que d'une seule usine, il est important de remarquer qu'en relaxant les contraintes de périodicité (4.21), (4.22) et de capacité (4.16), nous obtenons le modèle de (Voros 2002) qui est un MLLP (un MLCLSP à capacité infinie) à nomenclature série, qui est un problème NP-difficile. Nous présentons dans la partie suivante un état de l'art sur les méthodes d'optimisation pour le MLLP et le MLCLSP.

II Etat de l'art

Dans cette section, nous présentons les contributions majeures pour les méthodes de résolution du MLLP et du MLCLSP. Cet état de l'art complète celui donné dans (Comelli et al. 2008).

1 Méthodes pour le MLLP

Contrairement à sa version mono-niveau, le MLLP est un problème NP-difficile (Arkin et al. 1989) même si sa solution optimale vérifie la propriété du zéro-stock (Dellaert et al. 2000).

Par suite, pour ce problème, peu de méthodes exactes ont été développées : on peut retrouver les contributions de (Zangwill 1966) qui propose un algorithme de programmation dynamique dans le cas d'une nomenclature série et celle de (Afentakis et Gavish 1986) qui s'intéressent à une nomenclature générale.

(Afentakis et Gavish 1986) reformulent le modèle du MLLP en utilisant la notion de « stock échelon » introduite par (Clark et Scarf 1960) qui permet de simplifier l'équation des stocks (1.25) du MLLP en ne faisant pas apparaître explicitement la demande interne : cela permet donc de reformuler la contrainte d'équilibre des stocks de manière quasi-similaire à celle des modèles mono-niveau. Cette transformation effectuée, les auteurs ont développé un Branch & Bound. Pour la phase « Bounding » de leur algorithme, ils utilisent une relaxation lagrangienne des contraintes portant sur les stocks échelons en s'appuyant sur une méthode par sous-gradient. Ils développent également deux heuristiques afin de générer des solutions réalisables (pour ce faire, ils se ramènent à des nomenclatures d'assemblage). Ils testent leurs méthodes sur leurs propres instances et comparent les résultats obtenus par les heuristiques et le Branch & Bound.

En ce qui concerne les méthodes approchées, on peut distinguer deux grands types d'approches :

- dans la première, on retrouve des techniques séquentielles planifiant niveau par niveau selon la logique MRP,
- dans la seconde, les niveaux sont pris en compte simultanément.

L'approche séquentielle : C'est historiquement la première approche qui a été développée. Cela s'explique facilement par les deux raisons suivantes :

- cette façon de faire est calquée sur la logique MRP,

- planifier niveau par niveau à capacité infinie est un problème polynômial (Wagner et Whitin 1958).

L'idée de base de cette technique est la suivante : lorsqu'on planifie à capacité infinie pour un niveau n , le planning obtenu devient la demande pour le niveau $n + 1$. La recherche s'est donc naturellement tournée vers l'adaptation de méthodes dédiés au mono-niveau à capacité infinie ou de simples règles de gestion (Veral et LaForge 1985). Un état de l'art sur les diverses méthodes employées peut être trouvé dans (Bahl et al. 1987). Cependant, planifier une production au niveau n engendrera des coûts au niveau $n + 1$: il est clair que cet aspect doit être pris en compte pour optimiser globalement le plan de production. C'est pour cela que les méthodes de résolution utilisant cette approche se sont ensuite axées sur des procédures de modification de coûts incluant pour partie ceux inhérents à la fabrication des composants (souvent une partie des coûts de configuration liés à ces derniers). Les premiers travaux significatifs sur ce sujet sont dus à (Blackburn et Millen 1982). Dans cet article, les auteurs empruntent quatre types de modification de coûts à la littérature portant sur le ELSP ((Schwarz et Schrage 1975), (McLaren 1977)) et les appliquent à l'algorithme de (Wagner et Whitin 1958) ainsi qu'aux heuristiques de (Silver et Meal 1973) et (Gorham 1968) en comparant les résultats obtenus sur leurs propres instances ainsi que sur celles données par (McLaren 1977) et (Graves 1981). Beaucoup d'auteurs ont enrichi cette approche en proposant d'autres procédures de modification de coût : on peut citer (Graves 1981), (Heinrich et Schneeweiss 1986), (Bookbinder et Koch 1990) et plus récemment (Dellaert et Jeunet 2002) qui reprennent les modifications de coûts proposées par les trois derniers en intégrant une dimension aléatoire à celles-ci (en fait, l'idée sous-jacente est de pondérer le coût engendré par la fabrication des composants par un coefficient aléatoirement et uniformément choisi dans $[0, 1]$).

L'approche simultanée : Ce type d'approche a vu le jour beaucoup plus récemment et il est intéressant de remarquer que les méthodes de résolution développées se sont tournées vers l'utilisation de méthodes approchées basées sur des métaheuristiques à solution unique ou à population.

Parmi les métaheuristiques à solution unique, seul le recuit simulé semble avoir été utilisé sur cette problématique. (Kuik et Solomon 1990) proposent d'utiliser comme codage de solution la matrice binaire représentant les setups et montrent qu'il est possible de déduire le meilleur planning respectant cette matrice grâce à la résolution du dual d'un programme linéaire donné par (McClain et al. 1987). Ils développent six systèmes de voisinage différents exploitant ce codage de solution et modifiant la solution soit ponctuellement pour un produit donné, soit sur un produit et tout ou partie des composants présents dans sa nomenclature. Les auteurs implémentent un recuit simulé pour chaque système de voisinage dont ils comparent ensuite les résultats pour jauger leur efficacité. Enfin, ils utilisent des heuristiques à approche séquentielle pour juger de la qualité des solutions qu'ils obtiennent. (Jeunet et Jonard 2005) proposent également un recuit simulé se basant sur le codage de solution et les deux systèmes de voisinage les plus performants donnés par (Kuik et Solomon 1990). Ils proposent également trois nouveaux systèmes de voisinage qui sont des extensions de ceux donnés par (Kuik et Solomon 1990). Après avoir combinés les 5 systèmes de voisinages obtenus dans leur métaheuristique, ils testent leur méthode sur des instances de très grande taille (allant jusqu'à considérer 200 produits sur un horizon de 24 périodes) et se comparent aux résultats des heuristiques développées dans (Dellaert et Jeunet 2002). Utilisant le même codage de solution, (Tang 2004) propose un nouveau système de voisinage s'appuyant sur une propriété appelée « Inner corner » démontrée par (Grubbstrom et Molinder 1996) et qui stipule que « dans la solution optimale, pour deux produits liés par une relation de demande interne, le réapprovisionnement du produit de plus bas niveau doit toujours intervenir à la même période qu'un réapprovisionnement du niveau supérieur, et à ce moment les niveaux de stocks cumulés coïncident ».

En ce qui concerne les métaheuristiques à population, on retrouve les travaux de (Dellaert et Jeunet 2000) qui proposent d'utiliser les algorithmes génétiques pour cette problématique : les chromosomes sont définis comme étant un ensemble de « vecteur chaîne ». Un « vecteur chaîne » est associé bijectivement à un produit et correspond à la ligne de la matrice binaire des setups associée à ce dernier. Ils développent ensuite des opérateurs de croisement et de mutation (croisement de deux «

vecteurs chaînes » associés au même produit de deux chromosomes différents, modification d'un setup dans un « vecteur chaîne », modification d'un ou plusieurs « vecteurs chaînes » en utilisant l'heuristique « RCWW » qui est une heuristique séquentielle utilisant une modification de coût et l'algorithme de (Wagner et Whitin 1958), etc.). En utilisant leurs propres instances, les auteurs comparent la solution obtenue par leur algorithme génétique à celles obtenues par les diverses heuristiques séquentielles. (Dellaert et al. 2000) reprennent ce travail et l'enrichissent de plusieurs opérateurs de croisement agissant, entre autres, sur plusieurs niveaux de nomenclature et se comparent à diverses heuristiques (notamment celles décrites dans (Dellaert et Jeunet 2002)) mais, paradoxalement pas à l'algorithme génétique développé dans (Dellaert et Jeunet 2000). Plus récemment, (Pitakaso et al. 2007) ont proposé l'utilisation de colonie de fourmis. Chaque fourmi est déterminée par une heuristique séquentielle à modification de coûts inspirée par l'heuristique « RCWW » proposée par (Dellaert et Jeunet 2002). Les auteurs testent leur méthode sur les instances données par (Dellaert et Jeunet 2000) et constatent une amélioration d'environ 1,37 % par rapport aux solutions données par l'algorithme génétique de (Dellaert et Jeunet 2000).

(Voros 2002) propose une reformulation du MLLP dans le cas d'une nomenclature en série et étudie la relaxation linéaire de ce modèle : il prouve que sur certains types d'instances (notamment celles à 5 périodes et avec une nomenclature à 2 niveaux) la relaxation linéaire fournit la solution optimale du problème.

La figure 4.7 résume l'état de l'art présenté :

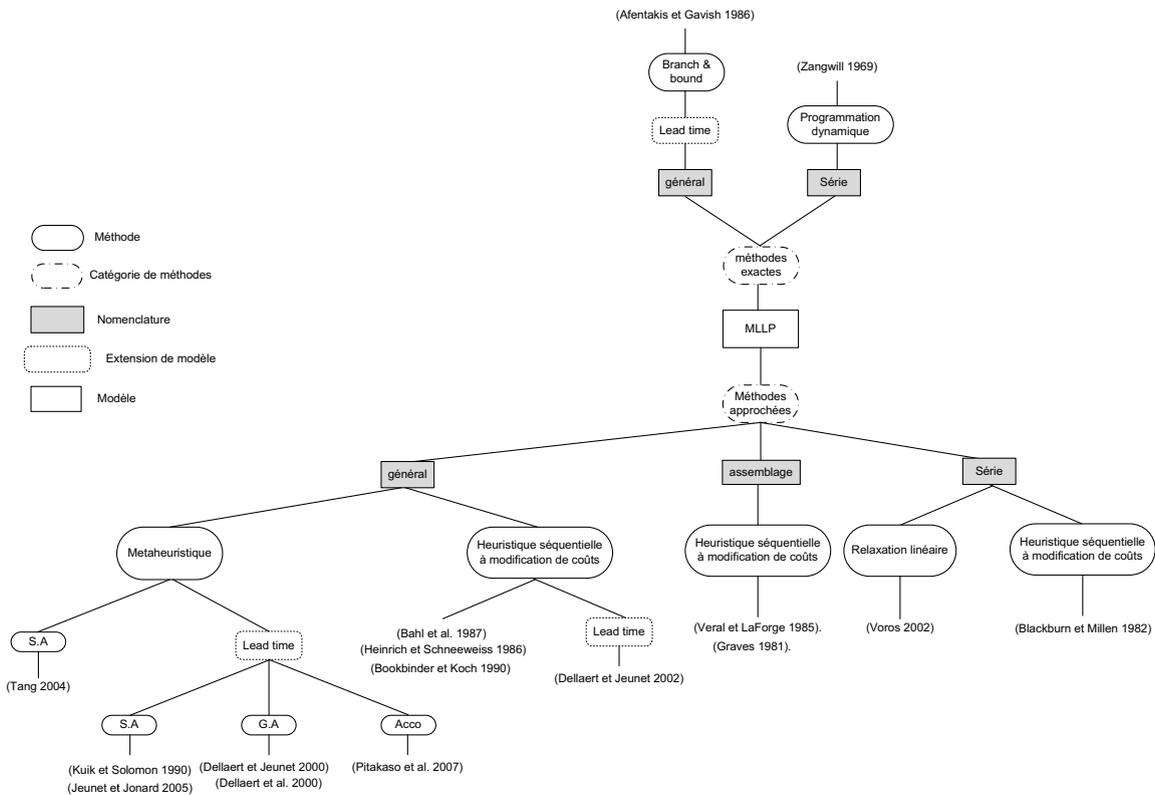


Figure 4.7 – Les méthodes de résolution pour le MLLP

Le tableau 4.1 synthétise les différents articles répertoriés. On y retrouve la référence de l'article, la nomenclature étudiée, le type de méthode utilisée, la taille des instances traitées et le GAP trouvé calculé par rapport à la valeur de comparaison (WW : Wagner-Within, LB : Borne Inférieure) et les extensions utilisées ($GAP = \frac{\text{Solution obtenue} - \text{valeur de comparaison}}{\text{valeur de comparaison}} * 100$).

Référence	Nomenclature	Méthodes Exactes	Méthodes Approchées	Taille maximale des instances testées	Gap	Valeur de comparaison	Extension
(Zangwill 1966)	série	D.P.		×	×	×	
(Afentakis et Gavish 1986)	générale	B & B		[3,15,6] × 12	0,45 %	LB	
(Veral et LaForge 1985)	assemblage		Règles	[1,9,6] × 12	-1,5 %	WW	
(Blackburn et Millen 1982)	série		HMC	[1,NC,5] × 12			
(Graves 1981)	assemblage		HMC	[1,5,5] × 12		WW	
(Heinrich et Schneeweiss 1986)	générale		HMC	[3,10,3] × 24			
(Bookbinder et Koch 1990)	générale		HMC	[1,12,7] × 12	-1,5 %	(Blackburn et Millen 1982)	
(Dellaert et Jeunet 2002)	générale		HMC	[15,500,18] × 52	×	HMC	Lead Time
(Kuik et Solomon 1990)	générale		SA	[5,57,10] × 12	-13,3 %	HMC	Lead Time
(Jeunet et Jonard 2005)	générale		SA	[1,200,15] × 24	×	HMC	Lead Time
(Tang 2004)	générale		SA	[1,9,9] × 12	×	×	
(Dellaert et Jeunet 2000)	générale		GA	[NC,500,18] × 52	-1,52 %	HMC	Lead Time
(Dellaert et al. 2000)	générale		GA	[1,50,10] × 36	-30,4 %	HMC	Lead Time
(Pitakaso et al. 2007)	générale		ACO	[NC,500,18] × 52	-0,18 %	(Dellaert et Jeunet 2000)	Lead Time
(Voros 2002)	série		R. Lin.	[1,2,5] × 5	×	×	

Tableau 4.1 – Etat de l’art pour le MLLP

Légende :

× : Pas de résultat numérique donné.

NC : Non Communiqué.

[a, b, c] × d : a : nombre de produits finis, b : nombre total de produits, c : nombre de niveaux, d : nombre de périodes.

HMC : Heuristique séquentielle à Modification de Coût.

SA : Simulated Annealing (recuit simulé)

GA : Genetic Algorithm (algorithme générique)

ACO : Ant Colony Optimization (colonie de fourmis)

R. Lin. : Relaxation Linéaire

2 Méthodes pour le MLCLSP

A l'instar du CLSP, le MLCLSP est un problème NP-Difficile (Maes et al. 1991). La prise en compte des capacités de production et des nomenclatures produits ont poussé les auteurs à prendre en considération la multiplicité des ressources de production : ainsi on retrouvera des modèles mono ou multi-ressource. Les modèles multi-ressource permettent une modélisation plus fine des gammes opératoires des divers produits ainsi qu'une meilleure estimation des capacités du système de production. La faisabilité du MLCLSP devient NP-Complet lorsqu'on y adjoint des temps de lancement (Maes et al. 1991). Dans leur état de l'art, (Rizk et Martel 2001) remarquent que très peu de méthodes exactes ont été proposées pour ce problème et supposent que c'est la grande difficulté du problème qui en est la cause.

2.1 MLCLSP sans setup time

Dans la littérature, on retrouve assez peu d'articles concernant la résolution du MLCLSP dans sa forme de base (sans setup time). En particulier, à notre connaissance, aucune méthode exacte n'a été développée pour ce problème.

(Roll et Karni 1991) proposent une heuristique en 6 étapes pour tenter de résoudre ce problème dans sa version mono-ressource. Partant d'une solution non nécessairement admissible (obtenue en appliquant niveau par niveau l'algorithme de (Wagner et Whitin 1958) ou une heuristique de lot pour lot), celle-ci tente d'améliorer la solution en la rendant faisable puis en optimisant son coût. Elle s'appuie sur un jeu de 8 règles de déplacement de quantités qui sont utilisées dans les diverses étapes de la méthode. Il est à noter que (Roll et Karni 1991) prennent en compte des contraintes de périodicité dans leur problème.

Pour le même problème, (Kuik et al. 1993) proposent un recuit simulé et une recherche Tabou en utilisant comme codage de solution une matrice de setup. Pour déduire les plannings correspondants, ils utilisent une heuristique gloutonne qui planifie la production en partant du produit ayant le coût de stockage le plus élevé jusqu'au produit ayant le coût de stockage le plus faible et en le planifiant au plus tard, en respectant les périodes de production prévues par la matrice de setup et les capacités du système de production : si cette procédure échoue, la solution est déclarée infaisable. Pour déterminer une solution initiale, ils utilisent une heuristique basée sur une reformulation du MLCLSP proposée par (Maes et al. 1991) dans le cas où la nomenclature est série, qu'ils étendent au cas d'une nomenclature d'assemblage. L'heuristique consiste à résoudre la relaxation linéaire de la reformulation et à fixer les variables de setup à 0 si elles sont inférieures à un certain ε , à 1 sinon. La solution obtenue est injectée dans l'une des métaheuristicues.

(Kim et Hosni 1998) se placent dans un cadre multi-ressource et proposent une méthode en deux étapes : partant d'une solution ne respectant en général pas les contraintes de capacité, la première étape vise à l'obtention d'une solution admissible par un lissage de la production. En démarrant de la dernière période, elle détecte les violations de capacité pour la période en cours et déplace les quantités en excès à la période précédente : la fabrication des composants utilisés pour produire les quantités déplacées sont éventuellement déplacées en tenant compte de leur lead-time de fabrication. A l'issue de la première étape, si la solution obtenue est admissible alors l'heuristique tente d'en améliorer le coût : si pour un produit i on a une fabrication à la période t et à la période $t + 1$ alors la totalité de la production est regroupée à la période t . Ce faisant, une infaisabilité due à une violation de la capacité risque d'apparaître : la première étape est donc de nouveau appliquée. L'heuristique s'arrêtera s'il n'y a plus d'amélioration de coût ou si une solution non admissible est générée sans qu'elle n'ait pu être réparée en appliquant la première étape.

Toujours dans un cadre multi-ressource, (Tempelmeier et Helber 1994) proposent une heuristique séquentielle (niveau par niveau) combinant une technique de modification de coût inspirée de celle développée par (Heinrich et Schneeweiss 1986) et l'heuristique de (Dixon et Silver 1981) afin de prendre en compte les capacités du système de production. Pour juger de la qualité de leur méthode, les auteurs réécrivent le MLCLSP en s'inspirant de la reformulation du CLSP en plus court chemin (Eppen et Martin 1987) et utilisent la relaxation linéaire du modèle obtenu pour en déduire une borne inférieure.

(Maes et al. 1991) proposent d'utiliser la reformulation du MLCLSP multi-ressource en un problème de localisation (Rosling 1986). Ils développent 3 heuristiques basées sur la relaxation linéaire de ce modèle : l'idée sous-jacente à celles-ci est de fixer définitivement, selon le résultat obtenu par la relaxation linéaire, des variables de setup à 1 et de résoudre à nouveau le programme linéaire relaxé jusqu'à obtention d'une solution admissible. Chaque heuristique implémente une façon particulière de fixer les variables de setup.

Sur le même problème que (Tempelmeier et Helber 1994), (Chen et Chu 2003) utilisent une heuristique lagrangienne basée sur la relaxation des contraintes d'intégrité portant sur les variables de setup. Cependant, le problème relaxé restant NP-difficile, ils ne peuvent utiliser une méthode par sous-gradient pour optimiser le dual lagrangien car ils ne peuvent résoudre le problème relaxé optimalement. C'est pour cela qu'ils utilisent une adaptation de cette dernière, appelée « surrogate subgradient method » (Zhao et al. 1999), qui permet de résoudre le dual lagrangien avec une méthode de résolution approchée du problème relaxé (ils utilisent la méthode du simplexe pour la résolution de celui-ci). A chaque itération de la relaxation lagrangienne, ils obtiennent une solution admissible en arrondissant chaque variable de setup à 1 si la quantité produite correspondante est strictement positive, à 0 sinon ; puis ils tentent d'optimiser cette solution en utilisant une heuristique basée sur une version modifiée de l'algorithme du simplexe.

La figure 4.8 résume l'état de l'art présenté.

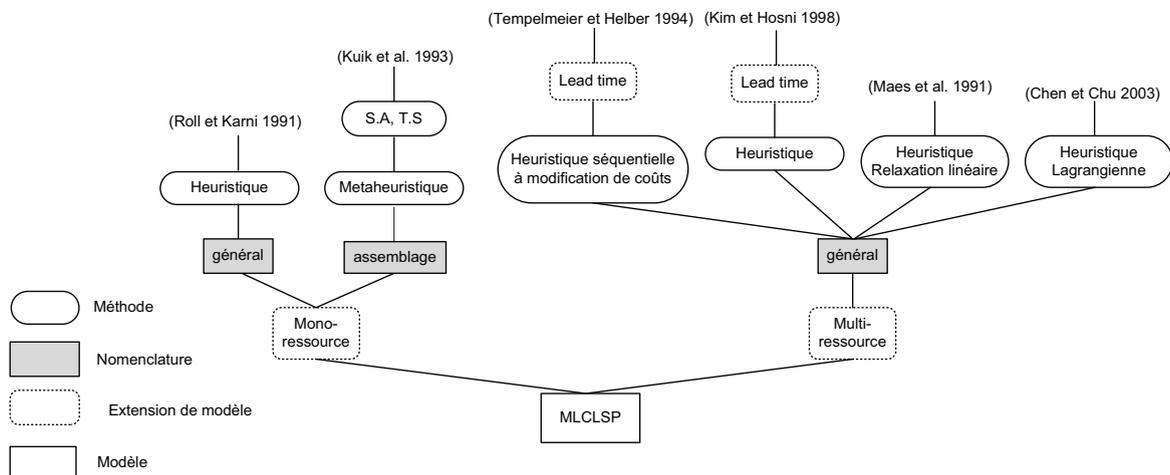


Figure 4.8 – Les méthodes de résolution pour le MLCLSP

2.2 MLCLSP avec setup time

La plupart des articles traitant du MLCLSP se sont focalisés sur son extension avec setup time. A l'instar du MLCLSP sans setup time, cette extension comporte une difficulté supplémentaire liée à sa faisabilité (Maes et al. 1991). C'est pour cette raison qu'on retrouve des extensions tels que l'overtime ou le backlogging pour lever cette difficulté. On notera que tous les papiers que nous avons référencés dans cette partie considèrent le problème dans sa version multi-ressource.

2.2.1 Le modèle sans extension

La majorité de la littérature s'est axée vers la résolution de ce problème.

Contrairement au MLCLSP, nous avons pu répertorier un article développant une méthode exacte : (Clark et Armentano 1995) proposent un ensemble d'inégalités valides en se basant sur la reformulation du MLCLSP avec des stocks échelon. En utilisant le solveur OSL, ils développent un Branch and Cut exploitant ces inégalités et se comparent avec un Branch and Bound disponible dans l'une des bibliothèques d'OSL, afin de juger de l'efficacité des coupes développées.

Certains auteurs s'appuient sur des méthodes exactes pour proposer des heuristiques permettant de répondre à cette problématique :

- Modélisant un atelier de production comportant des ressources goulets, (Billington et al. 1986) ont développé une méthode basée sur un Branch and Bound qu'ils couplent avec des heuristiques de la manière suivante : dans la phase «Bounding», ils proposent d'utiliser une heuristique lagrangienne basée sur la relaxation lagrangienne des contraintes de capacité et d'équilibre de stocks, guidée par une méthode de sous-gradient, pour se ramener à un problème mono-niveau et mono-produit à capacité infinie qu'ils résolvent par programmation dynamique. Ils utilisent une technique de modification de coût (Graves 1999) afin de tenir compte des liens de nomenclature : c'est pour cela que les auteurs préfèrent parler d'heuristique lagrangienne plutôt que de relaxation lagrangienne car ils ne peuvent prouver que la solution obtenue est une borne inférieure du problème. De plus, ils remarquent qu'en pratique il est très difficile avec cette technique d'obtenir une solution respectant les capacités de production de chaque ressource, c'est pour cela qu'ils intègrent à chaque itération de cette heuristique lagrangienne, deux heuristiques de lissage : toutes deux lissent la production en partant de la dernière période mais tandis que la première s'efforce de déplacer de la production d'une période où la capacité est dépassée à une autre où elle ne l'est pas, la seconde déplace systématiquement l'excès de production à la période précédente. Si une solution respectant les capacités n'est pas trouvée, tous les dépassements de capacité seront concentrés sur la première période. La stratégie de branchement s'effectue de la façon suivante :
 - . si un produit n'est pas fabriqué à une période t alors tous ces prédécesseurs ne seront pas fabriqués durant cette période mis à part s'ils utilisent une ressource goulet
 - . si un produit est fabriqué à une période t alors tous ces prédécesseurs pourront être produits à cette même période sauf éventuellement s'ils utilisent une ressource goulet.
 Dans leur stratégie de branchement, les auteurs décident de ne pas stériliser un nœud dès qu'une solution admissible est obtenue (cela est dû aux spécificités de l'heuristique lagrangienne) et considèrent cette stratégie comme «une heuristique permettant d'explorer une très grande variété de solutions» dont la sélection dépend du coût trouvé par l'heuristique lagrangienne.
- (Akartunali et Miller 2008) propose une heuristique utilisant une technique «Relax and Fix» (Belvaux et Wolsey 2000) : pour cela ils définissent ce qu'ils appellent une fenêtre dont la taille est exprimée en nombre de périodes. Partant du début de l'horizon, la fenêtre est déplacée en direction de la fin de l'horizon d'un certain nombre de périodes (appelé «overlapping») à chaque itération de l'heuristique. L'idée de l'heuristique est la suivante : en ajoutant des inégalités valides, la relaxation linéaire du problème est résolue par un solveur. Dans la solution trouvée, l'heuristique examine les périodes contenues dans la fenêtre et fixe définitivement les variables de setup à 1 dès qu'il y a une production. La fenêtre est ensuite avancée et la relaxation linéaire du programme est à nouveau résolu : cependant, les setups déjà fixés ne sont pas remis en cause. L'heuristique utilise ce schéma jusqu'à ce que la dernière période soit atteinte.

(Tempelmeier et Derstroff 1996) proposent une heuristique lagrangienne basée sur la relaxation lagrangienne des contraintes de capacité et d'équilibre de stock : les sous-problèmes obtenus sont résolus grâce à l'algorithme de (Wagner et Whitin 1958). A chaque itération de la relaxation lagrangienne, les dépassements de capacité et les violations de positivité des stocks sont calculés afin de mettre à jour les multiplicateurs lagrangiens lors de la prochaine itération. Ensuite une heuristique est utilisée afin de déterminer une solution admissible. La procédure s'arrête dès que l'un des critères suivants est satisfait : un nombre maximal d'itérations, le GAP entre la borne inférieure et la solution admissible déterminée est inférieur à un certain ε , les multiplicateurs lagrangiens sont inférieurs à un certain ε .

(França et al. 1997) proposent une heuristique en 4 étapes : la première étape consiste à appliquer l'algorithme de (Wagner et Whitin 1958) niveau par niveau. La solution ainsi obtenue risque de ne pas respecter les capacités du système de production. La seconde étape tente de réparer la solution en appliquant une heuristique en 2 phases : la première phase consiste à déplacer les quantités en excès vers une période ultérieure en respectant la positivité des stocks. Si la solution obtenue n'est toujours pas

admissible, alors la seconde phase est appliquée : un lissage de la production est appliqué. L'heuristique est ainsi appliquée jusqu'à ce qu'une solution admissible soit trouvée ou qu'un nombre maximum de passes soit atteint (dans ce cas la méthode s'arrête, faute de solution admissible). La troisième étape vise à améliorer la solution admissible trouvée à la seconde étape en déplaçant des quantités de manière similaire à l'heuristique présentée dans l'étape 2, mais en utilisant un argument basé sur des coûts afin de déterminer les produits les plus avantageux à déplacer. Cette étape s'arrête dès qu'il n'y a plus de déplacement améliorant la solution. La dernière étape essaie d'éliminer des coûts de setup en déplaçant l'intégralité de la production d'un produit d'une période sur une autre sans altérer l'admissibilité de la solution.

(Katok et al. 1998) ont développé une heuristique en trois étapes : la première étape vise à obtenir une solution admissible dont les coûts engendrés par les setups et les coûts de stockage sont très proches : pour ce faire, une formulation restreinte du programme linéaire initial est résolue plusieurs fois en modifiant les coûts de stockage et de setup afin d'obtenir le résultat recherché. La seconde étape utilise la solution précédemment déterminée et fixe séquentiellement chaque variable de setup à 0 et résout une restriction du programme linéaire précédent, sans toutefois modifier les coûts déterminés par l'heuristique précédente. Si la solution trouvée est meilleure, alors elle est conservée. L'étape 2 s'arrête dès que toutes les variables de setup ont été testées. La troisième et dernière étape est très similaire à la seconde et vise à déplacer la production de part et d'autre d'une certaine période en fixant la variable de setup correspondante à 0 et en relaxant les éventuelles contraintes fixant à 0 les variables de setup de part et d'autre de la période considérée.

Une partie des auteurs s'est également focalisée sur l'emploi de métaheuristiques pour tenter de résoudre ce problème.

(Barbarosoglu et Ozdamar 2000) proposent d'analyser l'effet des différentes relaxations de contraintes (équilibre des stocks, contraintes de capacité) sur les performances d'un recuit simulé. Les auteurs proposent 4 schémas d'optimisation selon les contraintes que l'on décide de relaxer et dont la violation est pénalisée dans la fonction objectif. On retrouve donc les possibilités de relaxation suivantes :

- 1- seules les contraintes d'équilibre des stocks,
- 2- seules les contraintes de capacités,
- 3- aucune contrainte,
- 4- les contraintes d'équilibre des stocks et de capacités.

Pour chacun des cas, un système de voisinage basé sur un déplacement de quantité est élaboré : la quantité est bornée de façon à ce que la solution obtenue après le déplacement respecte les contraintes non relaxées. Suivant le schéma dans lequel on se place, une méthode pour déterminer la solution initiale est donnée (l'utilisation de l'algorithme de (Wagner et Whitin 1958) niveau par niveau pour le schéma 2, une heuristique planifiant niveau par niveau en respectant les contraintes de capacité mais pas nécessairement les contraintes d'équilibre des stocks si la capacité n'est pas suffisante pour le schéma 1, une planification niveau par niveau grâce à l'algorithme de (Wagner et Whitin 1958) et une heuristique de lissage pour le schéma 3). (Ozdamar et Barbarosoglu 2000) exploitent ces schémas en les intégrant dans une heuristique lagrangienne. La méthodologie proposée est la suivante :

- Etape 1 : les sous-problèmes résultant de la relaxation sont résolus et la borne inférieure du problème original est mise à jour,
- Etape 2 : en utilisant la solution de l'étape 1, les multiplicateurs lagrangiens sont mis à jour,
- Etape 3 : la solution est injectée en entrée du recuit simulé et le schéma d'optimisation correspondant aux contraintes relaxées est utilisé pour déterminer une solution admissible, puis on retourne à l'étape 1.

(Berretta et Rodrigues 2007) proposent d'utiliser un algorithme mémétique s'appuyant sur l'heuristique de (França et al. 1997) : la population initiale est obtenue en appliquant les étapes 1 et 2 de l'heuristique de (França et al. 1997) en faisant varier les coûts pour obtenir des solutions différentes. Deux opérateurs de croisement sont proposés : l'un utilise une recombinaison aléatoire des quantités produites chez les deux parents, l'autre recalcule des coûts liés au setup suivant les quantités produites

chez les deux parents : une fois les coûts recalculés, la première étape de l'heuristique de (França et al. 1997) est utilisée. Un opérateur de mutation est également proposé : il s'agit d'une application de l'étape 4 de (França et al. 1997).

2.2.2 Le MLCLSP avec overtime

(Billington et al. 1983) proposent une méthode appelée «compression structure» permettant de simplifier la nomenclature des produits et donc de réduire la taille du problème à résoudre : ils montrent comment l'appliquer sur un MLCLSP avec Overtime.

(Stadler 1996) proposent deux nouvelles reformulations du modèle du MLCLSP avec overtime. En utilisant ses propres instances, il teste ses modèles en utilisant un solveur et compare les résultats obtenus à ceux fournis par les reformulations du MLCLSP avec overtime en plus court chemin inspirée de (Eppen et Martin 1987) et en problème de localisation (Rosling 1986). Après avoir remarqué que ces modèles génèrent moins de variables et de contraintes que les deux autres, il constate que ces deux reformulations fournissent de meilleurs résultats en des temps inférieurs.

(Xie et Dong 2002) proposent d'utiliser un algorithme génétique. Pour ce faire, ils utilisent le même codage de chromosome que (Dellaert et Jeunet 2000). Dans un premier temps, ils ne prennent pas en compte l'extension Overtime et génèrent une population de plannings ne tenant pas compte des contraintes de capacités. En partant de la dernière période, ils lissent la production en démarrant de la ressource la plus utilisée et du produit le plus bas dans la nomenclature (ainsi, ils sont sûrs que l'équation d'équilibre des stocks sera respectée). Un opérateur de croisement est proposé : une paire de chromosomes est sélectionnée de façon aléatoire mais en tenant compte de leur fitness, une position p est choisie aléatoirement dans le premier chromosome et on sectionne les deux chromosomes à la position p et les nouveaux chromosomes sont générés en prenant la première partie de chaque chromosome et en y rajoutant la seconde partie de l'autre. Un opérateur de mutation est également donné et consiste tout simplement à modifier la valeur d'un setup contenu dans le chromosome. Si les nouveaux chromosomes ne respectent pas les capacités de production, l'heuristique de lissage leur est appliqué. Cet algorithme génétique est ensuite appliqué dans le cas du MLCLSP avec overtime ce qui simplifie l'obtention d'une solution réalisable à l'issue de l'heuristique de lissage.

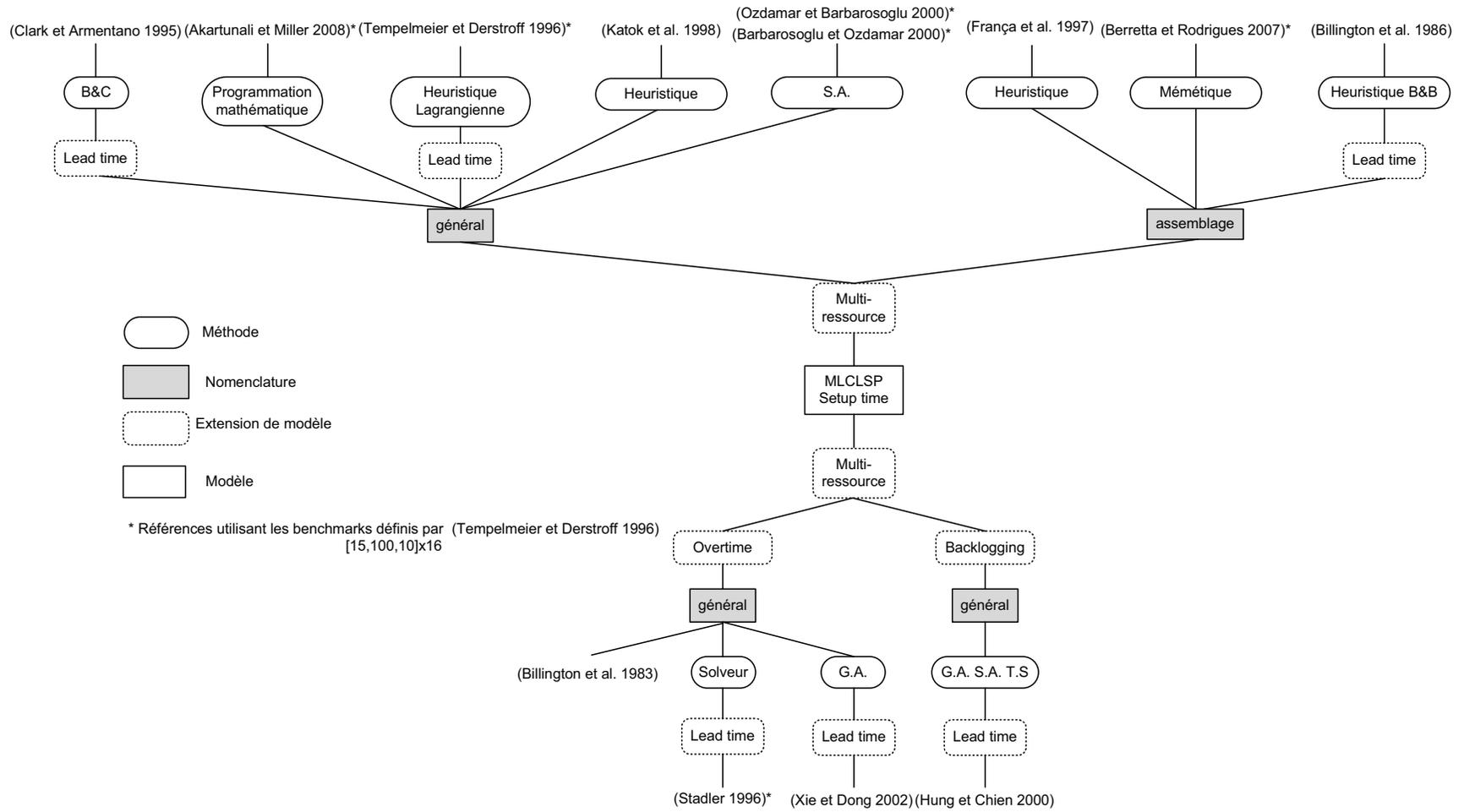
2.2.3 Le MLCLSP avec backlogging

(Hung et Chien 2000) proposent d'utiliser des métaheuristicues sur ce problème : on retrouve un recuit simulé et une méthode Tabou où le codage de la solution correspond à la matrice de setup. Pour déterminer une solution associée à cette matrice, un solveur résout le problème linéaire obtenu en fixant les variables de setup. Le système de voisinage utilisé consiste à choisir un produit, une période et à modifier la valeur du setup associé. Ils proposent aussi d'utiliser un algorithme génétique : l'opérateur de croisement utilisé est celui donné par (Hung et al. 1999) et l'opérateur de mutation correspond au système de voisinage de la méthode Tabou. Ils comparent ensuite les résultats obtenus par les trois méthodes.

La figure 4.9 résume l'état de l'art présenté. Le tableau 4.2 synthétise les différents articles répertoriés. On y retrouve la référence de l'article, le type de méthode utilisée, la taille des instances traitées et le GAP calculé par rapport à la valeur de comparaison (BIP : meilleure solution trouvée lors de la parution de l'article, LB : Borne Inférieure, Solveur : Solution donnée par un solveur, STD : solution donnée par (Tempelmeier et Derstroff 1996)) et les extensions utilisées ($GAP = \frac{\text{Solution obtenue} - \text{valeur de comparaison}}{\text{valeur de comparaison}} \times 100$).

Dans leur article, (Tempelmeier et Helber 1994) déplorent le manque d'instances de référence pour le problème du MLCLSP. Au vue de notre état de l'art, il semblerait intéressant de considérer celles définies par (Tempelmeier et Helber 1994) comme instances de référence pour le MLCLSP avec Setup time mais également dans sa version de base. En effet, parmi les instances de (Tempelmeier et Helber 1994) un sous-ensemble ne prend pas en compte les setup times.

Figure 4.9 – Les méthodes de résolution pour le MLCLSPST



Référence	Nomenclature	Méthodes Exactes	Méthodes Approchées	Taille maximale des instances testées	Gap	Valeur de comparaison	Ressource	Extension
MLCLSP sans Setup Time								
(Roll et Karni 1991)	général		heuristique	$[NC,8,NC] \times 15$	0,8 %	BIP	Mono	
(Kuik et al. 1993)	assemblage		S.A. T.S.	$[1,7,3] \times 12$	0,23 %	LB	Mono	
(Kim et Hosni 1998)	général		heuristique	\times	\times	\times	Multi	Lead Time
(Tempelmeier et Helber 1994)	général		HMC	$[6,40,5] \times 16$	2,0 %	LB	Multi	Lead Time
(Maes et al. 1991)	général		HRLi	$[3,9,3] \times 10$	\times	\times	Multi	
(Chen et Chu 2003)	général		HRLa	$[6,60,10] \times 12$	-18,23 %	Solveur	Multi	
MLCLSP avec Setup Time								
(Clark et Armentano 1995)	général	B&C		$[1,10,4] \times 12$	0,4 %	LB	Multi	Lead Time
(Billington et al. 1986)	assemblage		HB&B	\times	\times	\times	Multi	Lead Time
(Akartunali et Miller 2008)	général		HPM	$[15,100,10] \times 16$	1,12 %	LB	Multi	
(Tempelmeier et Derstroff 1996)	général		HRLa	$[15,100,10] \times 16$	1,4 %	LB	Multi	Lead Time
(França et al. 1997)	assemblage		heuristique	$[4,17,4] \times 10$	0,8 %	LB	Multi	
(Katok et al. 1998)	général		heuristique	$[12,72,5] \times 25$	-35,72 %	Solveur	Multi	
(Barbarosoglu et Ozdamar 2000)	général		S.A.	$[15,100,10] \times 16$	\times	\times	Multi	
(Ozdamar et Barbarosoglu 2000)	général		S.A.	$[15,100,10] \times 16$	4,01 %	LB	Multi	
(Berretta et Rodrigues 2007)	assemblage		Mémétique	$[15,100,10] \times 16$	-1,0 %	STD	Multi	
(Billington et al. 1983)	général			\times	\times	\times	Multi	Overtime
(Stadler 1996)	général	Solveur		$[15,100,10] \times 16$	1,6 %	BIP	Multi	Lead Time + Overtime
(Xie et Dong 2002)	général		G.A	$[2,21,6] \times 6$	\times	\times	Multi	Lead Time + Overtime
(Hung et Chien 2000)	général		G.A S.A T.S	$[5,20,NC] \times 10$	\times	\times	Multi	Lead Time + Backlogging

Tableau 4.2 – Etat de l’art pour le MLCLSP

Légende :

\times : Pas de résultat numérique donné.

NC : Non Communiqué.

$[a, b, c] \times d$: a : nombre de produits finis, b : nombre total de produits, c : nombre de niveaux, d : nombre de périodes.

HMC : Heuristique séquentielle à Modification de Coût.

SA, GA, TS : Simulated Annealing (recuit simulé), Genetic Algorithm (algorithme générique), Taboo Search (recherche tabou)

HRLi (resp HRLa) : Heuristique basée sur une relaxation linéaire (resp. lagrangienne)

B&C : Branch and Cut

HB&B, HPM : Heuristique basée sur un Branch and Bound, Heuristique basée sur une méthode de programmation mathématique

3 Conclusion

Nous avons présenté un état de l'art des méthodes de résolution pour le MLLP et le MLCLSP. Le modèle de planification de chaîne logistique pouvait être vu comme un MLCLSP particulier. En effet, dans le cas où chaque étage est réduit à une usine, il peut être obtenu du modèle de (Voros 2002) en y ajoutant des contraintes de capacité et des contraintes de périodicité. Dans la suite nous proposons de le résoudre en utilisant une méthode à base de recuit simulé qui s'appuie sur cette constatation. Cependant dans un premier temps, nous ne nous appuyerons pas sur un codage de solution inspiré par (Kuik et al. 1993) (utilisant les matrices de setup) mais nous reprenons un codage utilisant les quantités produites, à l'instar de ce qui a été fait dans le chapitre III de ce mémoire.

III Méthodes d'optimisation pour la planification d'une chaîne logistique de type Flow-Shop Hybride

Dans cette section, nous proposons une méthode d'optimisation pour planifier une chaîne logistique de type Flow-Shop Hybride. L'idée de base de cette méthode est de décomposer le problème en deux sous-problèmes : un problème de planification et un problème d'affectation. Le problème de planification s'obtient en agrégeant sur chaque étage l'ensemble des usines le composant pour obtenir ce que nous appelons une « usine étage ». La figure 4.10 illustre ce principe d'agrégation.

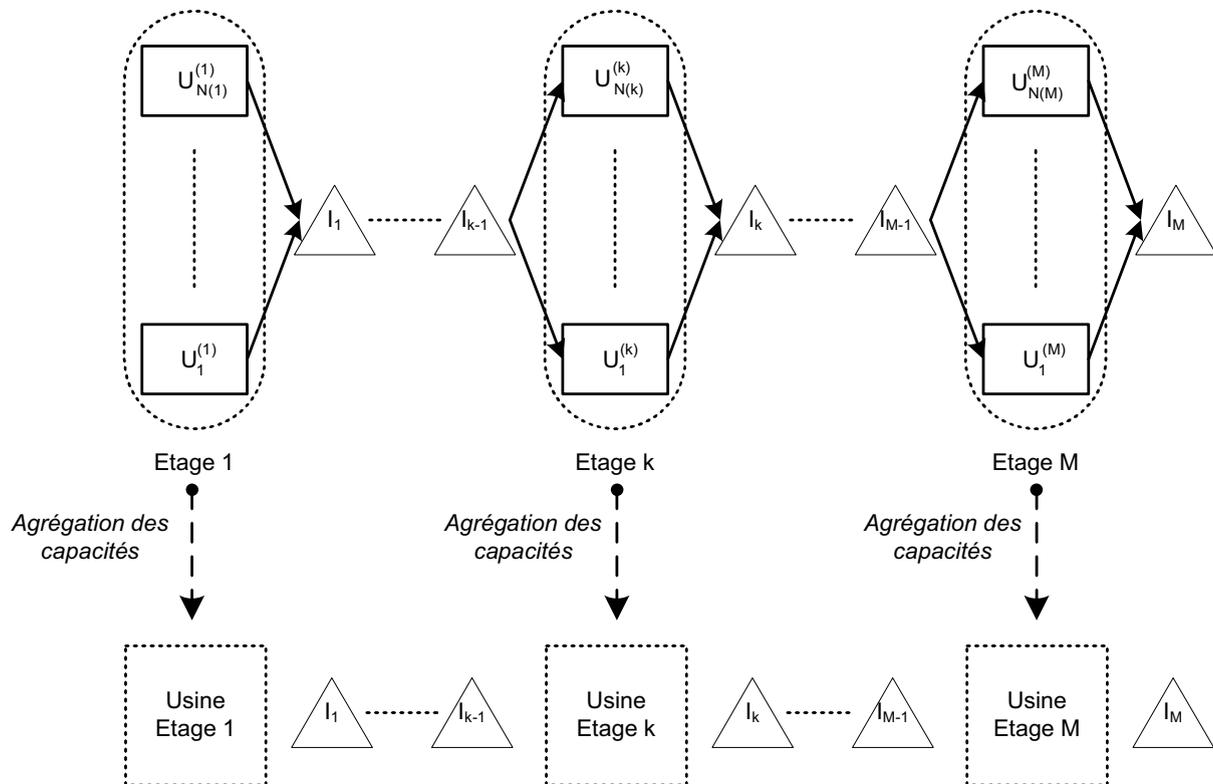


Figure 4.10 – Principe d'agrégation

Nous voyons que, par ce biais, nous nous ramenons à un MLCLSP à nomenclature série et, plus précisément, nous retrouvons le modèle de (Voros 2002) auquel on a adjoint des contraintes de capacité et de périodicité. Une fois une planification déterminée pour chaque « usine étage », nous nous ramenons au problème initial en répartissant la production de chaque « usine étage » sur chaque usine la composant : c'est le problème d'affectation que nous avons évoqué précédemment.

Dans un souci de simplification, nous appellerons désormais i le produit fini i_M ainsi que tous les produits i_k . Comme nous précisons systématiquement l'étage sur lequel nous agissons, il n'y a pas de

risque d'ambiguïté.

1 Résolution du problème de planification des « usines étages »

Pour résoudre ce problème, nous proposons une métaheuristique à base de recuit simulé. En reprenant le schéma donné par (Michalewicz et Fogel 2002), il faut que nous définissions :

- un codage permettant de représenter les solutions du problème,
- une fonction d'évaluation de la solution (qui sera la fonction objectif de notre modèle),
- un système de voisinage (une application permettant de passer d'une solution à une autre).

1.1 Codage de la solution

Comme pour la méthode proposée pour le CLSP, nous proposons un codage de la solution basée sur un planning de production et non sur une matrice de setup. Plus précisément, nous définissons pour chaque « usine étage » k une matrice $Q^{(k)}$ représentant le planning de production de cette dernière :

$$Q^{(k)} = \begin{pmatrix} Q_{11}^{(k)} & Q_{12}^{(k)} & \dots & Q_{1M}^{(k)} \\ Q_{21}^{(k)} & Q_{22}^{(k)} & \dots & Q_{2M}^{(k)} \\ \vdots & \vdots & & \vdots \\ Q_{N1}^{(k)} & Q_{N2}^{(k)} & \dots & Q_{NM}^{(k)} \end{pmatrix}$$

Nous coderons donc une solution de notre problème de planification pour la chaîne logistique sous la forme d'un vecteur de matrices représentant chacune le planning d'une « usine étage » :

$$Q = \left(\underbrace{Q^{(1)}, \dots, Q^{(k)}, \dots, Q^{(M)}}_{M \text{ étages}} \right)$$

1.2 Système de voisinage

Elaborer une « bonne » solution consiste à trouver un compromis entre un niveau de demande perdue, un niveau de stockage et le nombre de lancement de campagne en déterminant des quantités produites à chaque période sur chaque usine. Ce compromis peut être obtenu de deux manières :

- soit en augmentant ou en diminuant les quantités produites sur une ou plusieurs périodes,
- soit en regroupant les quantités de périodes consécutives.

Aussi, les voisinages que nous proposons sont les suivants : ils consistent à choisir aléatoirement une « usine étage » k , un produit i et une période t et à appliquer respectivement l'une des quatre stratégies suivantes :

- voisinage « augmentation » : augmenter la quantité produite en produit i à la période t sur l'« usine étage » k ,
- voisinage « diminution » : diminuer la quantité produite en produit i à la période t sur l'« usine étage » k ,
- voisinage « regroupement à gauche » : regrouper tout ou partie de la production du produit i des périodes t et $t - 1$ pour l'usine k , la quantité de la période t étant ajoutée à la période $t - 1$,
- voisinage « regroupement à droite » : regrouper tout ou partie de la production du produit i des périodes t et $t + 1$ pour l'usine k , la quantité de la période t étant ajoutée à la période $t + 1$,

1.2.1 Le voisinage « augmentation »

Pour ce voisinage, la fabrication de produit i pour la période t sur l'« usine étage » k est augmentée d'une quantité choisie aléatoirement : cela permet de modifier la demande satisfaite ainsi que les quantités en stock. La procédure utilisée va se dérouler en 3 étapes :

- la première étape consiste à déterminer la quantité « DP » de demande perdue en produit i que l'on va pouvoir faire éventuellement disparaître en augmentant la production en produit i à l'étage k : en effet, il n'est pas nécessaire de dépasser cette quantité car cela générerait du stock qui ne serait jamais consommé.
- la seconde étape consiste à choisir aléatoirement la quantité dont on va augmenter la fabrication en produit i sur l'« usine étage » k en tenant compte d'une part de la capacité de cette usine étage, d'autre part de la quantité DP déterminée précédemment.
- pour finir, si la capacité de l'« usine étage » k est dépassée pour la période t , nous rétablissons celle-ci en diminuant progressivement la fabrication des autres produits durant cette période jusqu'à rétablir le respect de la capacité.

Le calcul de la quantité « DP » doit considérer l'« usine étage » sur lequel est appliqué le voisinage car il doit tenir compte de la contrainte de périodicité. En effet, compte tenu des contraintes de périodicité, l'augmentation de la quantité fabriquée en produit i à l'étage k aura au mieux un impact à la période $t + (M - k)$ sur le planning de l'« usine étage » M donc à la période $t + (M - k + 1)$ sur la demande perdue. Par conséquent, le calcul de DP correspond à la somme des quantités de demande perdue de la période $t + (M - k + 1)$ à la fin de l'horizon. La figure 4.11 illustre l'effet de la périodicité sur la demande perdue.

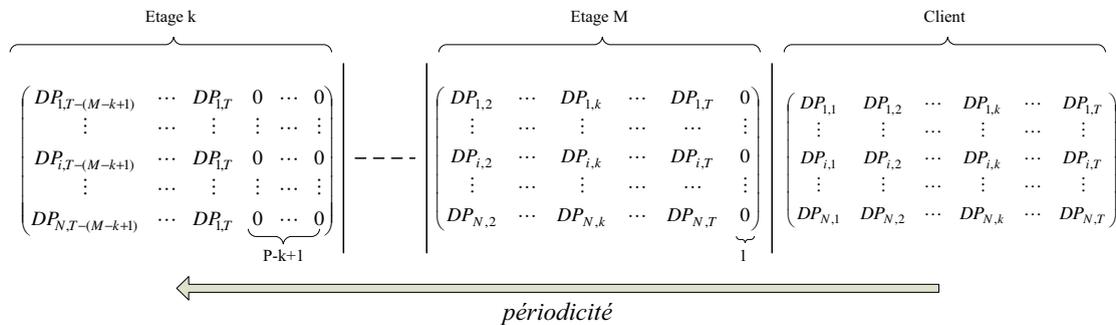


Figure 4.11 – prise en compte de la demande perdue sous l'effet des contraintes de périodicité

Dans la seconde étape, on commence par déterminer l'augmentation maximale de la fabrication du produit i sur l'étage k à la période t . Pour cela, on prend en compte uniquement la quantité fabriquée en produit i durant cette période et on calcule la capacité qu'il resterait si on avait uniquement fabriqué cela : on en déduit la quantité maximale de produit i ($QMax$) qu'on peut ajouter à cette période. L'augmentation maximale ($AjoutMax$) de la fabrication du produit i à la période t sera donc égale au minimum entre DP et $QMax$. La quantité effectivement ajoutée sera choisie aléatoirement et uniformément dans l'intervalle entier $\llbracket 0, AjoutMax \rrbracket$. Cependant, comme l'intégralité de la production de cette période n'a pas été prise en compte, il se peut qu'en rajoutant cette quantité la capacité de la période soit dépassée.

Le but de l'étape 3 est donc de rétablir le respect de la capacité de l'« usine étage » k à période t . Pour ce faire, on utilise la technique suivante : on choisit aléatoirement un produit j autre que i et on diminue sa production d'une unité et on réitère ce procédé jusqu'à ce que la capacité soit à nouveau respectée.

La figure 4.12 illustre le fonctionnement du voisinage augmentation.

1.2.2 Le voisinage « diminution »

Le fonctionnement de ce voisinage est beaucoup plus simple que le précédent : en effet, son but est de diminuer la fabrication en produit i de l'« usine étage » k à la période t . Nous choisissons donc une quantité à enlever, choisie aléatoirement et uniformément dans l'intervalle entier $\llbracket 0, Q_{it}^{(k)} \rrbracket$ et nous la retirons de la production : dans ce cas là, nous n'avons pas de problème de non respect de la capacité

de production.

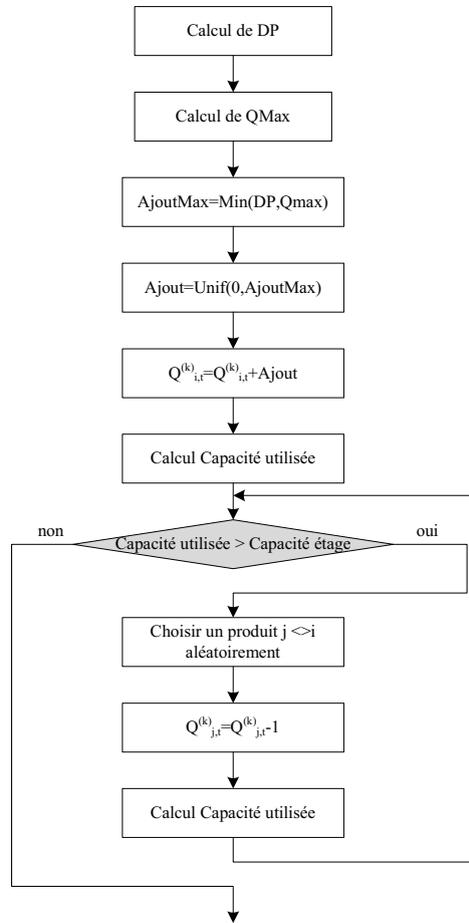


Figure 4.12 – Le voisinage « augmentation »

1.2.3 Le voisinage « regroupement à gauche »

Le but de ce voisinage est de regrouper tout ou partie de la production du produit i des périodes t et $t - 1$ pour l’usine k , la quantité de la période t étant ajoutée à la période $t - 1$. La procédure utilisée va se dérouler en 3 étapes :

- la première étape va consister à déterminer la quantité de produit i qu’on pourrait déplacer de la période t à la période $t - 1$ si on ne produisait que du produit i à la période $t - 1$,
- la seconde étape va consister à effectuer le déplacement de cette quantité de la période t à la période $t - 1$, ce faisant, on risque de dépasser la capacité disponible sur l’« usine étage » k à la période $t - 1$,
- la troisième étape consistera à rétablir le respect de la capacité à la période $t - 1$.

Dans la première étape, on commence par déterminer l’augmentation maximale de la fabrication du produit i sur l’étage k à la période $t - 1$. Pour cela, on utilise le même procédé que dans le voisinage « augmentation » : on prend en compte uniquement la quantité fabriquée en produit i durant cette période et on calcule la capacité qu’il resterait si on avait uniquement fabriqué cela. On en déduit la quantité maximale de produit i ($QMax$) qu’on peut ajouter à cette période.

Dans la seconde étape, on calcule la quantité Q qu’on va effectivement déplacer (en effet, on ne peut déplacer plus que la quantité produite à la période t) : $Q = Min(Q_{i,t}^{(k)}, QMax)$ et on déplace cette quantité de la période t à la période $t - 1$: $Q_{i,t}^{(k)} = Q_{i,t}^{(k)} - Q$ et $Q_{i,t-1}^{(k)} = Q_{i,t-1}^{(k)} + Q$. Cependant, comme

l'intégralité de la production de la période $t - 1$ n'a pas été prise en compte, il se peut qu'en rajoutant cette quantité la capacité de cette période soit dépassée.

Comme pour le voisinage « augmentation », le but de la dernière étape est donc de rétablir le respect de la capacité de l'« usine étage » k à période $t - 1$: on utilise donc la même technique.

La figure 4.13 illustre le fonctionnement du voisinage « regroupement à gauche ».

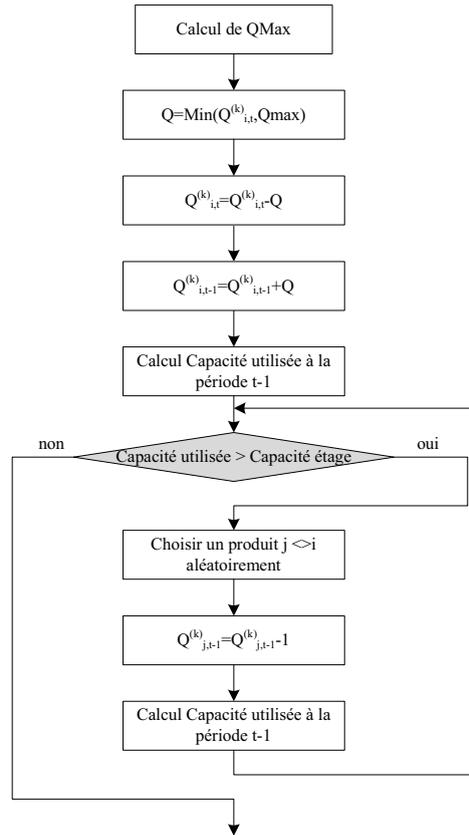


Figure 4.13 – Le voisinage « regroupement à gauche »

1.2.4 Le voisinage « regroupement à droite »

Le but de ce voisinage est de regrouper tout ou partie de la production du produit i des périodes t et $t + 1$ pour l'usine k , la quantité de la période t étant ajoutée à la période $t + 1$. Ce voisinage fonctionne de façon analogue au voisinage « regroupement à gauche », il suffit juste de remplacer $t - 1$ par $t + 1$ dans l'algorithme donné sur la figure 4.13.

1.2.5 Les procédures de réparation dédiées au produit i

Nous venons de présenter les voisinages utilisés. Cependant, agir sur une quantité pour un produit, une période et une « usine étage » impacte nécessairement les plans de production des autres « usines étage » risquant de générer du surplus négatif ou non consommé suivant le cas : en effet, ce qui est fabriqué par l'une est consommé par la suivante (ou, pour la dernière, est servi au client). Dans les voisinages utilisés, nous avons également vu que modifier la fabrication pour un produit i pouvait avoir un impact sur la production d'autres produits (en cas de dépassement de capacité par exemple). Pour rétablir l'admissibilité de la solution (tous les surplus doivent être positifs ou nuls sur tout l'horizon et nuls à la dernière période selon les contraintes (4.27) et (4.26)), il faut donc réparer la solution au niveau du produit i mais également au niveau des autres produits, d'où deux types de réparation :

- un dédié au produit i ,

- l'autre non dédié (donc pouvant également agir sur les autres produits).

Commençons par décrire les procédures dédiées au produit i .

La figure 4.14 illustre le processus de réparation dans le « cas favorable » (cette terminologie sera expliquée lors de la première procédure de réparation) après l'application du voisinage « augmentation ». Dans cet exemple, nous avons trois étages et la demande servie. Pour chaque étage, nous avons une matrice où la première ligne représente la production du produit i et la seconde le surplus qui lui est associé.

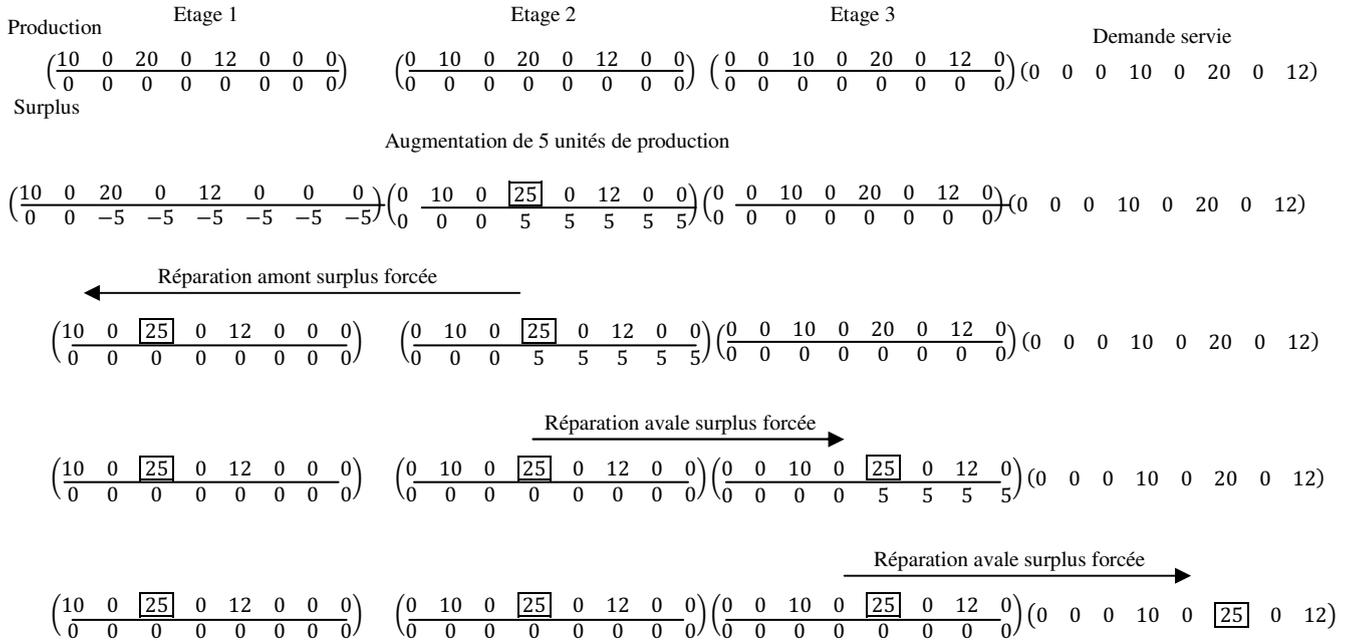


Figure 4.14 – Exemple d'utilisation des procédures de réparation sur le produit i

Dans l'exemple ci-dessus, le voisinage « augmentation » est utilisé sur l'« usine étage » 2 et on augmente de 5 la production. On génère donc du surplus négatif à l'étage inférieur et du surplus positif à la fin de l'horizon pour le stock de l'étage 2. Pour résoudre ce problème, deux procédures de réparation sont utilisées :

- La première procédure de réparation (procédure 1) appelée « réparation amont surplus forcée » va réparer la solution en augmentant la production de chaque étage amont jusqu'à la demande servie sur la période concernée en tenant compte de la contrainte de périodicité : ainsi nous n'avons plus de surplus négatif. Evidemment, en ajoutant de la fabrication en produit i sur chaque étage, nous pouvons éventuellement outrepasser la capacité disponible (d'où le qualificatif « forcée » de la procédure) : dans ce cas, on diminue la production des autres produits identiquement à la troisième étape du voisinage « augmentation ». Nous supposons ici qu'il est toujours possible de rétablir le respect de la capacité, c'est ce que nous appelons le « cas favorable ».
- De la même façon, la seconde procédure de réparation (procédure 2) appelée « réparation aval surplus forcée » vise à faire disparaître le surplus positif à la fin de l'horizon en le consommant dans les étages « aval » jusqu'à la demande servie. De la même façon, il est possible que la capacité des étages « aval » ne soit plus respectée : on utilise la même technique que pour la procédure précédente.

Comme nous l'avons précisé, nous nous sommes placés dans le cas où il était possible de réparer les dépassements de capacité ce qui n'est pas nécessairement le cas, même en retirant la production de tous les autres articles : dans ce cas, il va falloir retirer également un peu de la production que nous

avons rajoutée, régénérant par suite du surplus négatif et/ou positif. La figure 4.15 illustre l'effet des procédures 3 et 4 de réparation utilisées dans ce cas.

Dans cet exemple, nous supposons que les deux premières procédures de réparation ont été appliquées. Cependant, il reste du surplus positif en fin d'horizon pour le produit i à l'étage 2 et du surplus négatif à l'étage 1.

Résultat obtenu après les deux procédures de réparation

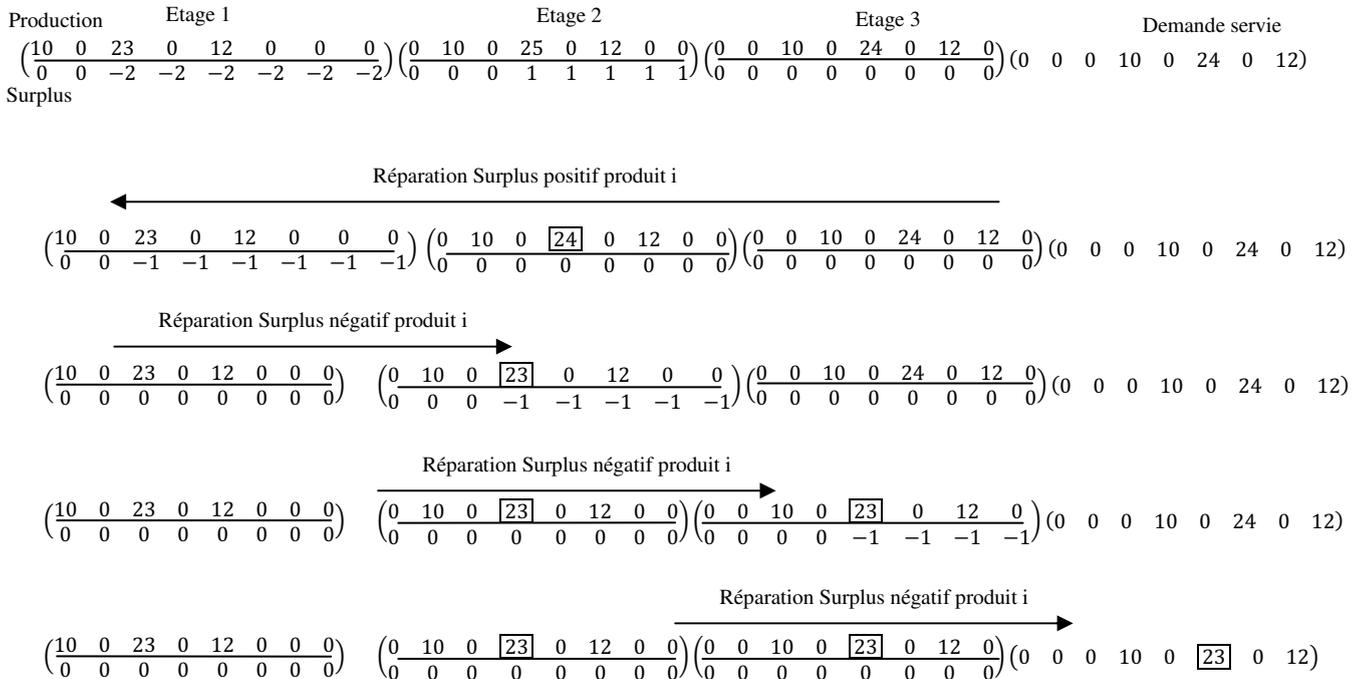


Figure 4.15 – Exemple d'utilisation des deux autres procédures de réparation sur le produit i

- la procédure 3 de réparation « réparation Surplus positif produit i » débute au dernier étage et détecte si il y a du surplus positif en fin d'horizon. Si c'est le cas, elle remonte l'horizon jusqu'à déterminer la première période où apparaît ce surplus non consommé et, dans ce cas, en tenant compte de la contrainte de périodicité, elle diminue la production à l'étage aval afin d'annuler ce surplus. Une fois le surplus corrigé à cet étage, elle recalcule les surplus pour tous les étages (car cela a une influence sur le surplus des étages inférieurs) et passe à l'étage précédent et ainsi de suite jusqu'au premier étage.
- la procédure 4 de réparation « réparation Surplus négatif produit i » débute au premier étage et recherche tout surplus négatif pour le produit i en partant de la première période jusqu'à la fin de l'horizon. Dès qu'une période est détectée, la production à l'étage aval est diminuée en tenant compte de la contrainte de périodicité et les surplus sont recalculés. Une fois tous les surplus éliminés pour cet étage, on passe à l'étage suivant jusqu'à ce qu'on arrive au dernier étage.

Dans le cas d'un voisinage « regroupement à gauche », les procédures de réparation utilisées diffèrent des précédentes. La figure 4.16 montre l'utilisation des procédures de réparation dans ce cas. La grande différence avec le voisinage « augmentation » réside dans le fait que pour ce voisinage là, on ne vise pas à augmenter la demande servie : c'est pour cela que la procédure 2 « réparation surplus avale forcée » n'est pas utilisée car elle répercute la modification jusqu'à la demande servie. Nous utilisons donc une cinquième procédure (procédure 5) dite « réparation avale surplus forcée sans demande servie » qui n'affecte pas la demande servie. Pour ce qui est de la seconde version de la procédure « réparation Surplus positif produit i » que nous appelons « réparation Surplus positif produit i V.2 » (procédure 6), elle commence par essayer de supprimer le surplus négatif en abaissant la production à la période

consécutives puis, si le surplus n'est pas totalement éliminé, elle diminue du surplus restant la quantité fabriquée à la période considérée. Evidemment, cela a pour conséquence de créer du surplus qui est consommé ultérieurement augmentant le coût dû au surplus, mais le but de ce voisinage est de donner une certaine flexibilité à la méthode d'optimisation, d'où ce choix.



Figure 4.16 – Exemple d'utilisation des procédures de réparation sur le produit *i* dans le cas d'un voisinage regroupement à gauche

1.2.6 Les procédures de réparation non dédiées

Ces procédures de réparation s'appliquent successivement à tous les produits et sont au nombre de 4.

- La première procédure de réparation non dédiée (procédure 7) vise à éliminer un maximum

de surplus négatif en partant du dernier étage et en remontant jusqu'au premier, et cela, pour chaque produit. Son fonctionnement se décompose de la façon suivante : pour le dernier étage, elle détermine pour un produit donné la présence de surplus négatif. Si c'est le cas, elle tente d'annuler le surplus négatif en augmentant, sans dépasser la capacité disponible, la fabrication du produit à la période où apparaît le surplus négatif puis passe au produit suivant (il se peut donc qu'il subsiste du surplus négatif). Une fois tous les produits examinés, la procédure passe à l'étage précédent et ainsi de suite jusqu'à avoir traité le premier étage.

- La deuxième procédure de réparation non dédiée (procédure 8) a pour objet l'élimination de tout surplus négatif qui n'aurait pas été éliminé par la procédure 7. Cette procédure balaye tous les étages du premier au dernier pour examiner la présence de surplus négatif. Si c'est le cas, elle se place au dernier étage (k) où il y a eu une détection, prend l'un des produits où il y a un surplus négatif et récupère la première période (t) où celui-ci est apparu. La procédure va diminuer la production de ce produit à l'étage suivant si $k < P$ ou la demande servie si on se trouve sur le dernier étage ($k = P$) : cela aura pour incidence de faire disparaître le surplus négatif. Pour ce faire, elle commence par abaisser la production du produit à l'étage $k + 1$ (ou le cas échéant, la demande servie) à la période $t + 1$. Si à l'issue de cette diminution, le surplus négatif considéré n'est pas annulé, alors on diminue la production à l'étage $k + 1$ (ou la demande servie) à la période t . Le surplus considéré sera donc forcément annulé car sinon il y aurait eu un surplus négatif à la période $t - 1$ de l'étage k pour ce produit, ce qui est contraire au choix de t . La procédure recommence à balayer tous les étages et ceci jusqu'à ce qu'il n'y ait plus de surplus négatif.
- La procédure de réparation non dédiée suivante (procédure 9) vise à éliminer un maximum de surplus positif et fonctionne de manière similaire à la procédure 7. Partant du dernier étage pour arriver au premier, elle détermine à l'étage courant k et pour un produit donné, la présence de surplus positif non consommé. Si c'est le cas, alors, partant de la dernière période, elle détermine la période t où est apparu cette quantité de surplus positif. Puis, si $k < P$, elle augmente la fabrication du produit à la période $t + 1$ à l'étage supérieur ($k < P$) tout en respectant la capacité de production restante sinon elle essaie de répartir ce surplus positif sur la demande servie en partant de la période $t + 1$ en prenant soin que la demande servie ne dépasse jamais la demande client. Une fois tous les produits examinés, elle passe à l'étage précédent. A l'instar de la procédure 7, il peut subsister du surplus positif non consommé à l'issue de cette procédure.
- La dernière procédure de réparation non dédiée (procédure 10) vise à éliminer tout surplus positif non consommé qui n'aurait pas été éliminé par la procédure 9. Cette procédure balaye tous les étages du premier au dernier pour examiner la présence de surplus positif non consommé. Si c'est le cas, elle se place au dernier étage (k) où il y a eu une détection, prend l'un des produits où il y a un surplus positif non consommé et récupère la première période (t) où celui-ci est apparu. Elle diminue la production de ce produit à l'étage k et à la période t de façon à annuler ce surplus positif, ce qui est possible car sinon il y aurait un surplus positif à la période précédente ce qui est contraire au choix de la période t .

Il n'est pas inutile de remarquer qu'en diminuant des quantités à fabriquer, les procédures 9 et 10 peuvent générer du surplus négatif, c'est pour cela qu'elles peuvent être suivies par l'application de la procédure 8.

Le lecteur pourra consulter le détail de ces procédures de réparation dans l'annexe 3 où figure l'ensemble des organigrammes les décrivant.

1.2.7 La méthode d'optimisation pour le problème de planification « usines etages »

La figure 4.17 donne le schéma d'optimisation que nous avons utilisé. Pour une question de lisibilité, nous désignons les procédures de réparation par leur numéro. A chaque voisinage utilisé, on peut voir quelles sont les procédures de réparation qui lui sont associées. On remarquera que le voisinage « diminution » n'utilise que les procédures 8 et 9 et ne nécessite pas de procédure de réparation dédiée

(en effet, ce voisinage n'impacte que la production du produit auquel il est appliqué et, comme il s'agit d'une diminution, ne risque pas d'engendrer un dépassement de capacité). En ce qui concerne le voisinage « regroupement à droite », il ne nécessite que l'emploi des procédures de réparation non dédiées, le problème de surplus négatif en amont de son étage ne se posant pas et, puisqu'il n'augmente pas la production globale du produit, il ne génère pas de surplus positif non consommé : il n'impacte donc que les autres produits d'où la seule utilisation des procédures de réparation non dédiées.

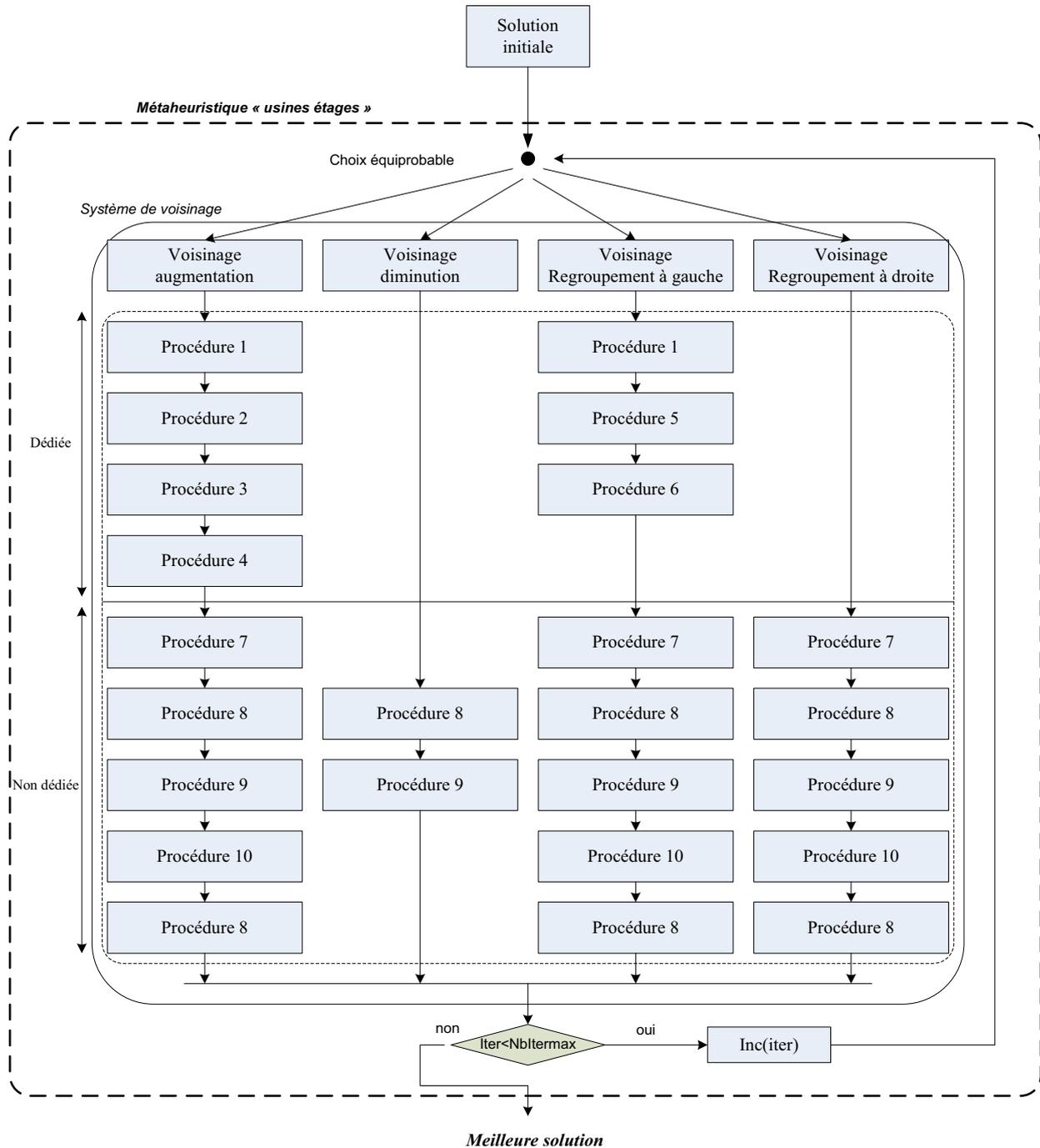


Figure 4.17 – Schéma d’optimisation pour la métaheuristique « usines étages »

Nous venons d’exposer une méthode de résolution pour le problème de planification des « usines étages ». Dans la section suivante, nous exposons la procédure d’affectation de la production de chaque « usine étage » k à l’ensemble des usines de l’étage k .

2 Résolution du problème d'affectation

Supposons que nous ayons obtenu une solution admissible au problème de planification des « usines étages », le but est donc d'affecter la production aux usines de chaque étage afin d'obtenir une solution à notre problème initial tout en minimisant les coûts de lancement générés (en effet, l'affectation aux usines n'influera ni sur les quantités stockées - qui sont rattachées aux étages - ni sur la demande perdue).

La technique que nous employons à cet effet s'appuie sur l'idée suivante : on doit affecter en priorité les produits dont le coût de lancement est le plus élevé aux usines possédant le plus de capacité restante afin d'éviter de scinder la production de ceux-ci, et donc d'en multiplier les coûts de lancement. La figure 4.18 illustre le fonctionnement de l'heuristique utilisée.

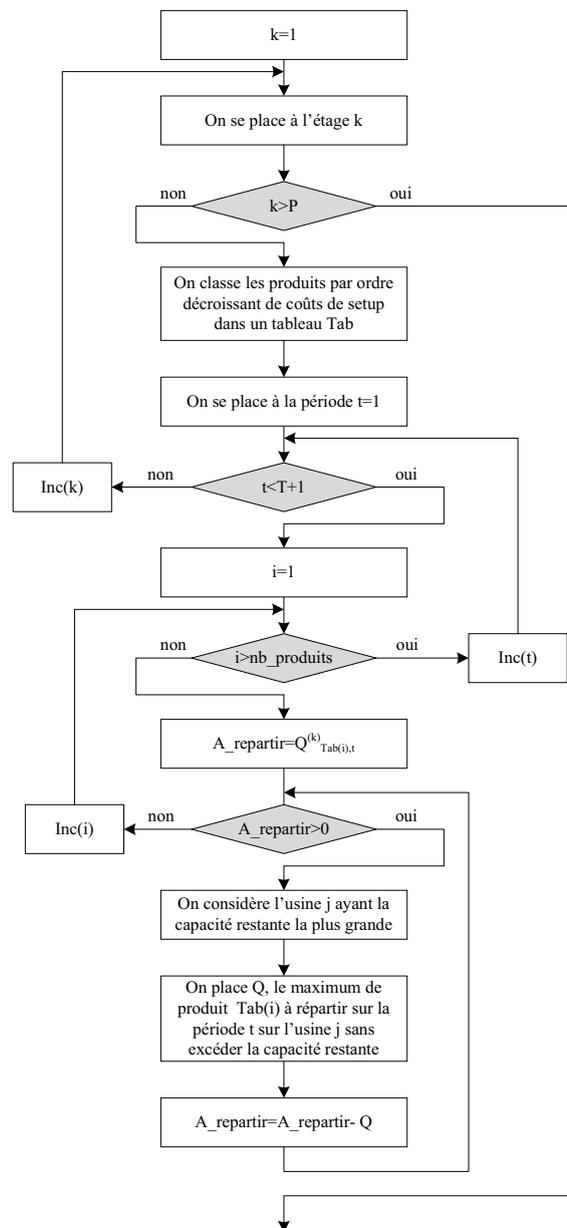


Figure 4.18 – heuristique d'affectation

3 La méthode d'optimisation pour le problème de planification de la chaîne logistique de type flow shop hybride

La méthode d'optimisation que nous présentons combine celle pour les « usines étages » et l'heuristique d'affectation suivant le schéma présenté dans la figure 4.19 :

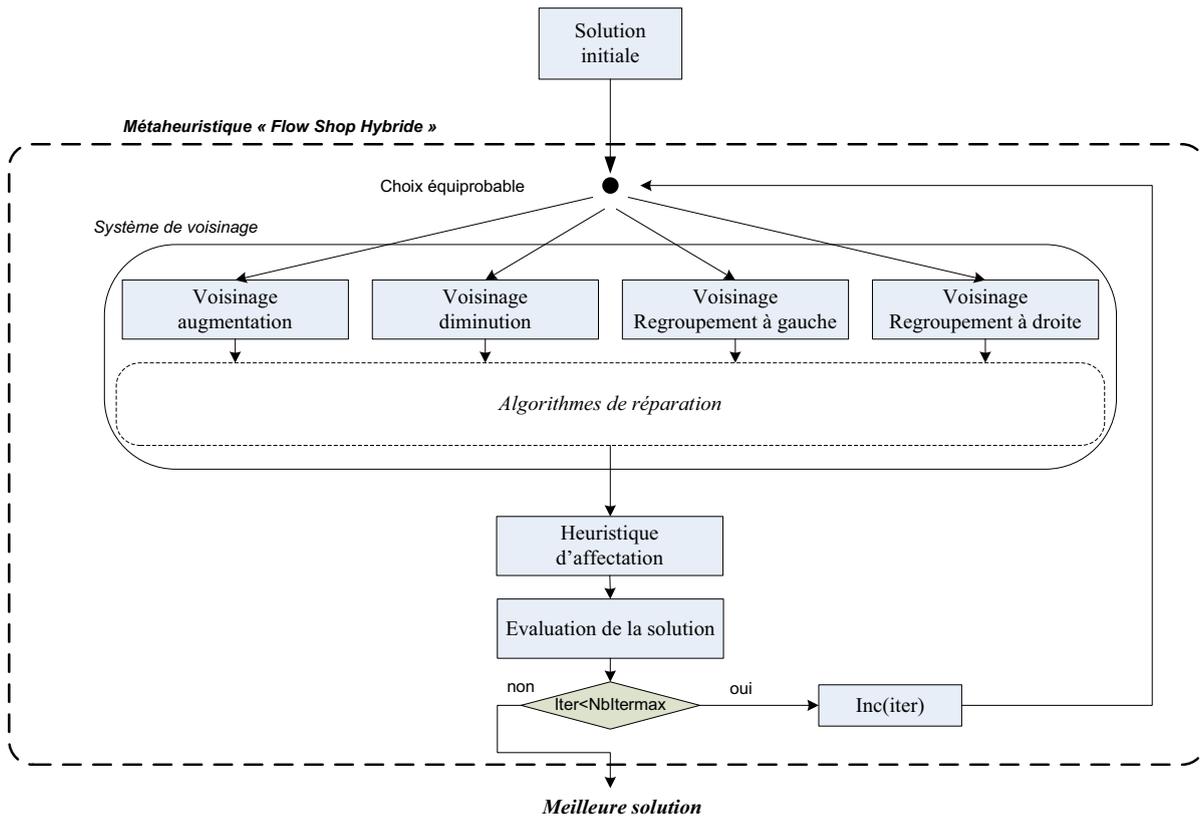


Figure 4.19 – Schéma d'optimisation pour la métaheuristique «Flow Shop Hybride»

4 Mise en œuvre et résultats

Dans cette section, nous présentons les résultats obtenus par notre méthode. Dans un premier temps, nous testons notre métaheuristique sur des instances de petite taille dont nous comparons les résultats avec ceux obtenus par un solveur (Cplex 11.0) afin de jauger la qualité de celle-ci. Dans un second temps, nous la mettons en œuvre sur notre cas d'étude où les instances sont de taille industrielle.

4.1 Tests préliminaires

Nous menons cette campagne de test en deux étapes :

- dans la première, nous utilisons uniquement la métaheuristique « usines étages ». Nous comparons les résultats obtenus avec ceux fournis par le solveur, en utilisant le modèle mathématique instancié sur la chaîne logistique agrégée (la chaîne logistique « usines étages »). Le but de cette étape est de juger de la pertinence de la méthode d'optimisation
- dans la seconde, nous testons la métaheuristique « Flow Shop Hybride » et nous comparons les résultats obtenus avec ceux fournis par le solveur, utilisant le modèle original.

La chaîne logistique que nous considérons est celle définie dans notre cas d'étude (figure 4.1). Nous considérons 6 produits finis et un horizon de planification constitué de 16 périodes.

Nous avons élaboré les instances de la façon suivante :

- la somme sur tout l'horizon des capacités de production sur chaque étage est sensiblement égale à la somme de la capacité nécessaire pour fabriquer l'intégralité de la demande (la capacité globale de la chaîne logistique permet de répondre globalement à la demande exprimée). Cependant, une certaine partie de la demande ne pourra pas être couverte pour des raisons liées aux contraintes de périodicité,
- les coûts de lancement sont volontairement faibles au regard des autres coûts exprimés (surplus et demande perdue),
- dans une première série d'instances, nous supposons les coûts de demande perdue supérieurs aux coûts de surplus et dans la seconde, l'inverse.

Pour ces tests, nous imposons un temps limité d'une heure pour le solveur et 1000000 itérations pour la méthode d'optimisation (ce qui correspond à approximativement 20 minutes de calcul).

4.1.1 Première étape

Le tableau 4.3 donne les résultats obtenus pour les instances testées.

Instances	Solveur	Metaheuristique			
		Descente Stochastique	Recuit Simulé	Kangourou	I.S.K.A
	Résultat	GAP	GAP	GAP	GAP
Coût de demande perdue supérieur au coût de surplus					
Instance 1	2169390*	-5.10^{-3}	-6.10^{-3}	-6.10^{-3}	-6.10^{-3}
Instance 2	927870	7.10^{-2}	$4,4.10^{-2}$	$4,8.10^{-2}$	4.10^{-2}
Instance 3	60800	1,5	0,9	1,8	2
Instance 4	138370	0,67	0,28	0,34	0,28
Coût de surplus supérieur au coût de demande perdue					
Instance 1	1739230*	$1,4.10^{-2}$	$5,7.10^{-3}$	$5,1.10^{-3}$	$5,7.10^{-3}$
Instance 2	1365780	$6,1.10^{-2}$	$3,4.10^{-2}$	$3,1.10^{-2}$	$2,4.10^{-2}$
Instance 3	299660	0,7	0,19	0,22	0,23
Instance 4	1686460*	$7,6.10^{-2}$	$2,6.10^{-2}$	$1,8.10^{-2}$	0,16

Tableau 4.3 – Résultats obtenus sur les instances de petite taille

Le GAP est calculé de manière suivante : $\frac{\text{Résultat Métaheuristique} - \text{Résultat Solveur}}{\text{Résultat Solveur}} \times 100$.

Les solutions du solveur marquées par * indiquent que le résultat a été obtenu suite à un overflow du solveur.

Le tableau 4.4 montre la répartition des coûts pour la solution donnée par le solveur et par le recuit simulé : nous retrouvons la quantité de demande perdue (D. perdue), la quantité de surplus (Surplus) et le nombre de lancement (Lanc.).

Les résultats consignés dans le tableau 4.3 montrent des GAP plutôt faibles, ce qui montre que sur des petites instances les résultats obtenus par nos métaheuristiques sont sensiblement équivalents à ceux fournis par le solveur. Le tableau 4.4 montre que les différences obtenues se situent surtout au niveau du nombre de lancement, ce qui n'est guère étonnant vu leur faible coût en comparaison de ceux du surplus et de la demande perdue.

Instances	Solveur				Recuit Simulé			
	Résultat	D. perdue	Surplus	Lanc.	Résultat	D. perdue	Surplus	Lanc.
Coût de demande perdue supérieur au coût de surplus								
Instance 1	2169390	2160	2340	47	2169260	2160	2810	43
Instance 2	927870	900	1350	87	928280	900	1351	126
Instance 3	60800	0	3000	80	61360	0	3000	136
Instance 4	138370	90	2370	97	138760	90	2370	136
Coût de surplus supérieur au coût de demande perdue								
Instance 1	1739230*	1590	1350	73	1739330	1590	1350	83
Instance 2	1365780	1200	1500	78	1366250	1200	1500	125
Instance 3	299660	150	1350	116	300230	150	1350	173
Instance 4	1686460*	1560	1140	106	1686900	1560	1142	128

Tableau 4.4 – Décomposition des coûts

4.1.2 Seconde étape

Nous reprenons les mêmes instances que nous appliquons désormais à la chaîne logistique de notre cas d'étude. Pour juger de l'influence de l'heuristique de répartition, nous reprenons la même graine pour le générateur aléatoire et nous appliquons la méthode en utilisant le même nombre d'itérations. Les résultats obtenus par les métaheuristiques et le solveur sont tout à fait identiques : cela démontre l'efficacité de l'heuristique de répartition.

4.2 Résultats sur des instances de taille industrielle

Ces instances que nous avons générées sont au nombre de 20 et répondent aux mêmes spécificités que les instances précédentes (en termes de capacité et de coût). Pour chacune d'elles, nous devons planifier 10 produits finis. Pour dix d'entre elles, l'horizon est de 30 périodes et pour les 10 suivantes de 100 périodes. Les conditions des tests sont sensiblement identiques : nous avons fixé un temps limite de 60 minutes pour le solveur et de 1500000 itérations pour les métaheuristiques (environ 40 à 60 mn de calcul).

Le tableau 4.5 donne les résultats obtenus pour les instances testées.

Le GAP est calculé de manière suivante :
$$\frac{\text{Résultat Métaheuristique} - \text{Résultat Solveur}}{\text{Résultat Solveur}} \times 100.$$

Les solutions du solveur marquées par * indiquent que le résultat a été obtenu suite à un overflow du solveur.

Le tableau 4.5 montre l'efficacité de notre métaheuristique sur les instances de très grande taille. En effet, pour les instances les plus grandes, notre méthode d'optimisation supplante assez largement le solveur.

Instances	Taille	Solveur	Metaheuristique			
			Descente Stochastique	Recuit Simulé	Kangourou	I.S.K.A
		Résultat	GAP	GAP	GAP	GAP
Coût de demande perdue supérieur au coût de surplus						
Instance 1	10 × 30	2050400*	-6,37	-6,289	-6,533	-6,538
Instance 2	10 × 30	1552852*	0,172	0,412	0,104	0,049
Instance 3	10 × 30	2330303*	0,139	0,266	0,081	0,06
Instance 4	10 × 30	3584830*	0,22	0,013	0,1	0,108
Instance 5	10 × 30	4254382*	0,191	0,005	0,083	0,088
Instance 6	10 × 100	2049601*	-0,488	-0,528	-0,628	-0,613
Instance 7	10 × 100	1686652*	-0,522	-0,562	-0,574	-0,574
Instance 8	10 × 100	2463838*	-0,497	-0,501	-0,523	-0,516
Instance 9	10 × 100	4462684*	0,919	0,492	0,655	0,663
Instance 10	10 × 100	2086272*	-0,399	-0,415	-0,478	-0,572
Coût de surplus supérieur au coût de demande perdue						
Instance 1	10 × 30	44175*	5,582	8,688	4,079	2,904
Instance 2	10 × 30	44031*	0,613	1,632	-2,855	-4,406
Instance 3	10 × 30	49264*	2,494	7,37	2,165	0,184
Instance 4	10 × 30	60208*	13,592	1,639	6,381	7,502
Instance 5	10 × 30	66016*	13,133	0,936	7,467	6,278
Instance 6	10 × 100	177499*	-5,031	-5,412	-5,611	-5,585
Instance 7	10 × 100	175032*	-4,253	-5,016	-7,081	-6,985
Instance 8	10 × 100	181329*	-2,638	-4,656	-4,014	-6,173
Instance 9	10 × 100	223273*	15,623	-3,399	12,423	12,698
Instance 10	10 × 100	177840*	-3,835	-4,924	-6,366	-4,911

Tableau 4.5 – Résultats obtenus sur les instances de taille industrielle

Cependant, devant la complexité des algorithmes mis en œuvre dans notre métaheuristique, une modification de notre approche pourrait sans doute la rendre plus efficace. C'est ce que nous allons faire dans le paragraphe suivant.

IV Vers une nouvelle approche

Dans cette section, nous présentons des travaux en cours de réalisation : aucun résultat ne sera donc donné. Cependant, nous décrivons le principe de la méthode que nous sommes en train d'implémenter. Comme la méthode précédente, nous reprenons le principe de décomposition en deux sous-problèmes.

1 Résolution du problème « usines étages »

L'idée que nous allons mettre en œuvre est inspirée des travaux de (Kuik et Solomon 1990). Nous allons utiliser à nouveau une métaheuristique mais, pour coder une solution, nous nous appuyons pour chaque « usine étage » k sur une matrice $X^{(k)}$ binaire représentant les setups :

$$X^{(k)} = \begin{pmatrix} X_{11}^{(k)} & X_{12}^{(k)} & \cdots & X_{1T}^{(k)} \\ X_{21}^{(k)} & X_{22}^{(k)} & \cdots & X_{2T}^{(k)} \\ \vdots & \vdots & & \vdots \\ X_{N1}^{(k)} & X_{N2}^{(k)} & \cdots & X_{NT}^{(k)} \end{pmatrix}$$

Nous coderons donc une solution de notre problème de planification pour la chaîne logistique sous la forme d'un vecteur de matrices représentant chacune le planning des setups d'une « usine étage » :

$$X = \left(\underbrace{X^{(1)}, \dots, X^{(k)}, \dots, X^{(M)}}_{M \text{ étages}} \right)$$

Regardons maintenant comment adapter les voisinages proposés précédemment.

- Le voisinage « augmentation » vise à augmenter la fabrication d'un produit i à la période t sur l'« usine étage » k . L'adaptation est immédiate : nous fixons $X_{i,t}^{(k)}$ à 1.
- Le voisinage « diminution » vise à diminuer la fabrication d'un produit i à la période t sur l'« usine étage » k , nous fixons donc $X_{i,t}^{(k)}$ à 0.
- Le voisinage « regroupement à gauche » opérera de la façon suivante : ayant choisi un étage k , un produit i et une période t , nous fixons $X_{i,t-1}^{(k)}$ à 1 si $X_{i,t}^{(k)} = 1$ sinon $X_{i,t-1}^{(k)}$ n'est pas modifié. Ensuite, nous fixons $X_{i,t}^{(k)} = 0$.
- Le voisinage « regroupement à droite » agira ainsi : ayant choisi un étage k , un produit i et une période t , nous fixons $X_{i,t+1}^{(k)}$ à 1 si $X_{i,t}^{(k)} = 1$ sinon $X_{i,t+1}^{(k)}$ n'est pas modifié. Ensuite, nous fixons $X_{i,t}^{(k)} = 0$.

Pour déduire le planning de production associé, nous utilisons le solveur en affectant aux variables de setup du modèle les valeurs données dans la solution fournie par le système de voisinage. Bien évidemment, les setups étant fixés, ils ne sont pas optimisables. C'est pour cela que le solveur devra uniquement minimiser les coûts liés au surplus et à la demande perdue, simplifiant d'autant la résolution du problème. Ecrivons donc le modèle que le solveur devra résoudre (la capacité de l'« usine étage » k à la période t est la somme des capacités des usines composant l'étage k à cette même période) et qui s'obtient facilement à partir de celui de la chaîne logistique de type « Flow Shop Hybride » :

$$\text{Minimiser } \sum_{i=1}^N \sum_{t=1}^T \sum_{k=1}^M CI_{i_k t}^S I_{i_k, t}^S + \sum_{i=1}^N \sum_{t=1}^T CDP_{i_M, t} DP_{i_M, t} \quad (4.30)$$

Sous les contraintes :

$$\sum_{i=1}^N (Capa_{i_k, t} Q_{i_k, t}) \leq \sum_{j=1}^{N(k)} Capa R_{u_j^{(k)}, t} \quad \forall (k, t) \in \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket \quad (4.31)$$

$$Capa_{i_k, t} Q_{i_k, t} \leq \left(\sum_{j=1}^{N(k)} Capa R_{u_j^{(k)}, t} \right) X_{i, t}^{(k)} \quad \forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket \quad (4.32)$$

$$I_{i_k, t}^+ = I_{i_k, t-1}^+ + Q_{i_k, t} - Q_{i_{k+1}, t} \quad \forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M-1 \rrbracket \times \llbracket 1, T \rrbracket \quad (4.33)$$

$$I_{i_M, t}^+ = I_{i_M, t-1}^+ + Q_{i_M, t} - DS_{i_M, t} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (4.34)$$

$$DS_{i_M,t} - DP_{i_M,t} = D_{i_M,t} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (4.35)$$

$$Q_{i_k,t+1} \leq I_{i_k-1,t}^+ \quad \forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 2, M \rrbracket \times \llbracket 1, T \rrbracket \quad (4.36)$$

$$DS_{i_M,t+1} \leq I_{i_M,t}^+ \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (4.37)$$

$$I_{i_k,t}^S = I_{i_k,t}^+ - Q_{i_{k+1},t+1} \quad \forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M-1 \rrbracket \times \llbracket 1, T-1 \rrbracket \quad (4.38)$$

$$I_{i_k,T}^S = I_{i_k,T}^+ \quad \forall (i, k) \in \llbracket 1, N \rrbracket \times \llbracket 1, M-1 \rrbracket \quad (4.39)$$

$$I_{i_M,t}^S = I_{i_M,t}^+ - DS_{i_M,t+1} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T-1 \rrbracket \quad (4.40)$$

$$I_{i_M,T}^S = I_{i_M,T}^+ \quad \forall i \in \llbracket 1, N \rrbracket \quad (4.41)$$

$$I_{i_k,t}^+, I_{i_k,t}^S \in \mathbb{N} \quad \forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket \quad (4.42)$$

$$Q_{i_k,t} \in \mathbb{N} \quad \forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket, \forall j \in \llbracket 1, N(k) \rrbracket \quad (4.43)$$

$$DS_{i_M,t}, DP_{i_M,t} \in \mathbb{N} \quad \forall (i, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, T \rrbracket \quad (4.44)$$

On voit que c'est sur la contrainte (4.32) que porte le paramétrage du modèle en fonction de la solution de la métaheuristique.

La résolution du programme linéaire achevée, nous obtenons donc les plannings de production de chaque « usine étage » : nous pouvons donc en déduire une nouvelle solution en annulant tout setup où il n'y a pas de production correspondante. En quelque sorte, nous pouvons interpréter la résolution du programme linéaire comme une procédure de réparation.

Nous avons donc mis en place tous les ingrédients pour la résolution du problème de planification « usines étages ». Regardons comment résoudre maintenant le problème d'affectation.

2 Résolution du problème d'affectation

Dans cette section, nous remplaçons l'heuristique d'affectation par la résolution d'un programme linéaire.

Le problème d'affectation s'énonce de la façon suivante : pour chaque étage k , nous devons répartir la production de l'« usine étage » k sur l'ensemble des usines composant l'étage k , le tout en minimisant les coûts de lancement. En utilisant les notations du modèle de planification de type « Flow Shop Hybride », nous voyons que ce problème s'écrit sous la forme du programme linéaire suivant :

$$\text{Minimiser } \sum_{k=1}^M \sum_{i=1}^N \sum_{t=1}^T \sum_{j=1}^{N(k)} \left(CL_{i_k, u_j^{(k)}, t} X_{i_k, u_j^{(k)}, t} \right) \quad (4.45)$$

Sous les contraintes :

$$\sum_{j=1}^{N(k)} Q_{i_k, u_j^{(k)}, t} = Q_{i_k, t} \quad \forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket \quad (4.46)$$

$$\sum_{i=1}^N \left(\text{Capa}_{i_k, u_j^{(k)}, t} Q_{i_k, u_j^{(k)}, t} \right) \leq \text{Capa} R_{u_j^{(k)}, t} \quad \forall (k, t) \in \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket, \forall j \in \llbracket 1, N(k) \rrbracket \quad (4.47)$$

$$\text{Capa}_{i_k, u_j^{(k)}, t} Q_{i_k, u_j^{(k)}, t} \leq \text{Capa} R_{u_j^{(k)}, t} X_{i_k, u_j^{(k)}, t} \quad \forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket, \forall j \in \llbracket 1, N(k) \rrbracket \quad (4.48)$$

$$Q_{i_k, u_j^{(k)}, t} \in \mathbb{N}, X_{i_k, u_j^{(k)}, t} \in \{0, 1\} \quad \forall (i, k, t) \in \llbracket 1, N \rrbracket \times \llbracket 1, M \rrbracket \times \llbracket 1, T \rrbracket \quad (4.49)$$

La fonction objectif (4.45) consiste à minimiser les coûts de lancement pour chaque étage.

La contrainte (4.46) est la contrainte d'affectation de la production : toute la production de l'étage k doit être répartie sur les usines le composant.

La contrainte (4.47) est une contrainte de capacité : à chaque période, la capacité utilisée par l'usine pour fabriquer la production qui lui est dévolue ne doit pas excéder sa capacité.

La contrainte (4.48) permet de déterminer les lancements de production pour chaque usine.

Les contraintes (4.49) sont les contraintes de positivité et d'intégrité.

3 Une nouvelle méthode d'optimisation

La figure 4.20 illustre le fonctionnement de la nouvelle approche.

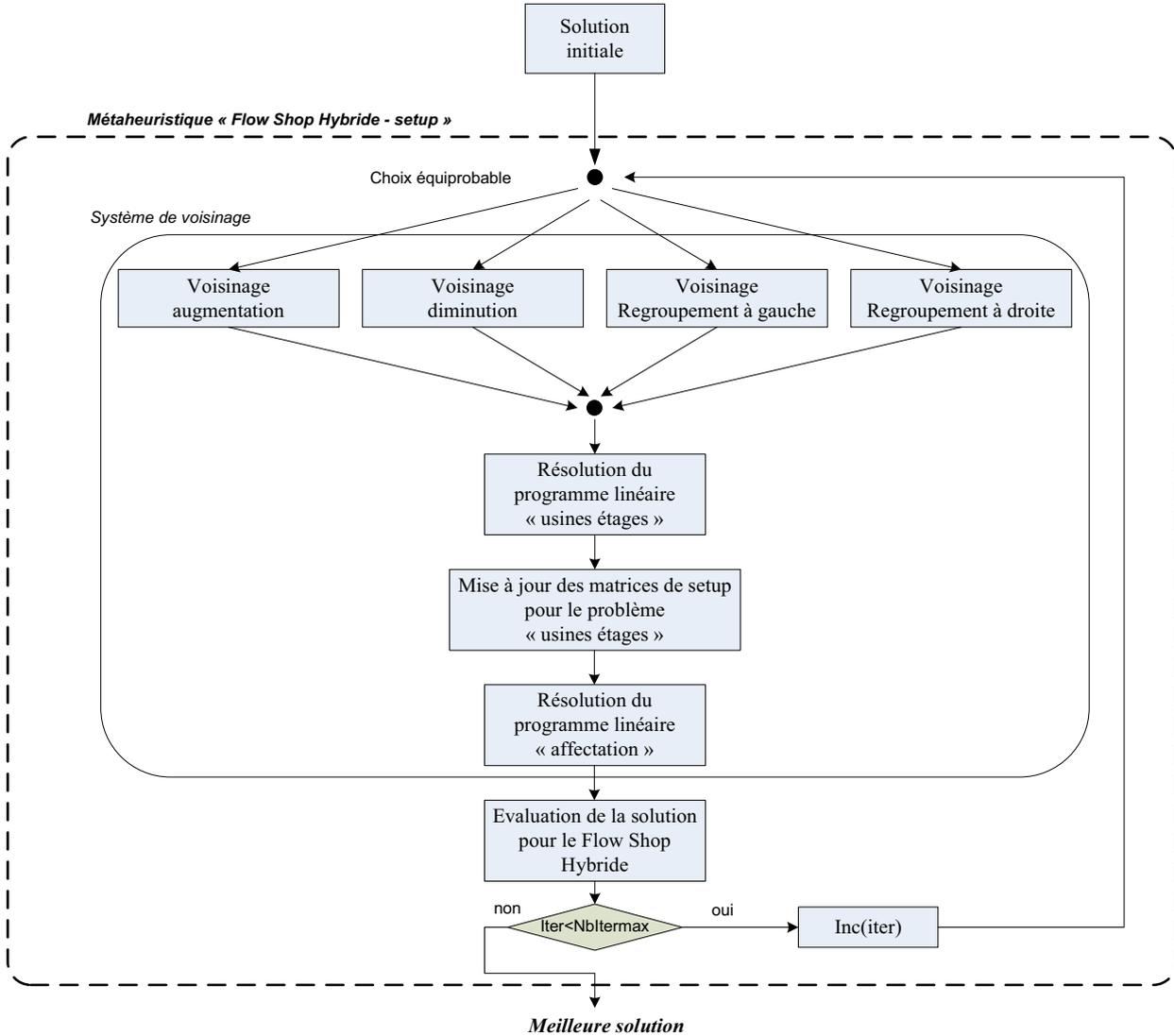


Figure 4.20 – Nouvelle approche d'optimisation

Conclusion

Dans ce chapitre, nous nous sommes intéressés à une problématique issue d'un cas d'étude industriel : la planification d'une chaîne logistique de type Flow-Shop Hybride. Grâce à une instanciation et une adaptation du modèle générique, nous avons proposé un modèle mathématique pour la planification de ce type de chaîne.

Les méthodes de résolution que nous avons proposées s'appuyant sur une décomposition du problème en deux sous-problèmes : un problème de planification à capacité finie avec un nomenclature série et un problème d'affectation. Le premier sous-problème s'apparentant à un MLCLSP à nomenclature série, nous avons proposé un état de l'art des méthodes de résolution pour le MLLP et le MLCLSP.

La première méthode de résolution proposée fait intervenir une métaheuristique de type recuit simulé dont les systèmes de voisinage sont basés sur du déplacement de quantités pour le problème de planification et une heuristique de répartition de la production pour le problème d'affectation. Les systèmes de voisinage rendant la solution infaisable, des heuristiques de réparation ont été proposées afin de restaurer son admissibilité. Cette méthode de résolution a été testée sur des instances de taille industrielle et les résultats obtenus sont plutôt convaincants.

Cependant, devant la complexité de mis en œuvre d'une telle méthode, nous proposons une piste d'amélioration utilisant d'une part un codage de solution basée sur des matrices de setup (Kuik et al. 1993) et, d'autre part, des modèles linéaires permettant de générer des solutions faisables pour chaque sous-problème permettant ainsi de se passer des algorithmes de réparation. Le premier modèle linéaire génère à partir de la matrice de setup fournie par le système de voisinage de la métaheuristique, un plan de production pour le problème de planification minimisant les coûts de stockage et de demande perdue. Le second modèle linéaire résout le problème d'affectation.

Chapitre 5

Pistes de recherche pour une synchronisation des plans tactique et opérationnel

Sommaire

Introduction	202
I Etat de l'art	202
1 Intégration de contraintes opérationnelles	202
2 Estimation des capacités du système de production	203
II Approche par couplage de modèles mathématiques	206
III Approche par couplage d'une méthode approchée et d'un modèle de simulation .	209
IV Illustration de l'approche utilisant la simulation sur la planification d'un atelier de type Flow-Shop hybride	211
Conclusion	219

Introduction

Dans notre premier chapitre, nous avons vu les difficultés liées au synchronisation besoin-matière dans un système MRPII, nous avons ensuite présenté les modèles mathématiques de « lot sizing » développés afin de répondre aux problématiques liées à l'élaboration des différents plans tactiques. Ces modèles prennent en paramètres les capacités estimées du système de production. Cependant, comme le soulignent divers auteurs ((Byrne et Bakir 1999), (Kim et Kim 2001)), ces évaluations s'avèrent souvent peu réalistes car elles ne reflètent pas correctement l'utilisation complexe des ressources d'un système de production réel ; complexité engendrée par la configuration des ateliers de production, la non-linéarité de la charge de travail sur les machines, la survenue d'événements aléatoires (pannes ...) etc. Ceci peut donc entraîner l'élaboration d'un plan de production dont on ne pourra assurer la mise en œuvre opérationnellement. Ainsi, il est légitime de se demander comment affiner l'estimation de ces capacités afin d'assurer la réalisabilité du plan de production déterminé et donc d'assurer cette synchronisation besoin-matière. Comme le souligne dans ses travaux de thèse (Wolosewicz 2008), les recherches menées sur cette problématique sont encore rares : beaucoup d'auteurs, tels que (Cossard 2004), se limitent à vérifier la faisabilité de leur planification au niveau opérationnel.

Dans ce chapitre nous commençons par brosser un bref état de l'art sur les méthodes développées pour tenter d'apporter une réponse satisfaisante à cette problématique puis nous proposons deux nouvelles approches s'appuyant pour la première sur le couplage de deux modèles mathématiques et pour la seconde sur un couplage entre une méthode approchée et plusieurs modèles de simulation qui permettent de décrire le système de production et ses diverses règles de gestion pour l'ordonnancement de la production.

I Etat de l'art

Traditionnellement, la détermination d'un plan de production se fait au travers de modèles mathématiques prenant en paramètres des capacités estimées du système de production. On vérifie ensuite la faisabilité du plan élaboré en tentant d'ordonnancer la production déterminée. Par suite, la planification et l'ordonnancement sont traités séquentiellement et indépendamment l'un de l'autre (Buxey 1989). Or, il est très difficile d'assurer la cohérence entre ces deux niveaux : les capacités du système sont évaluées beaucoup trop grossièrement et, comme le soulignent (Dauzère Pérès et Lasserre 2002), dépendent également de l'ordonnancement des opérations et des ressources. C'est ainsi que diverses stratégies ont été développées pour tenter de pallier cette carence. Dans la littérature, nous pouvons distinguer deux types d'approche. Le premier consiste à intégrer des contraintes opérationnelles dans le modèle de planification tandis que le second vise à estimer les capacités réelles du système de production en utilisant un modèle de simulation afin de décrire au mieux le comportement du système.

1 Intégration de contraintes opérationnelles

La première approche a été initiée par (Dauzère Pérès et Lasserre 1994) et consiste à intégrer des contraintes opérationnelles au modèle de planification. Plus précisément, ils proposent une approche intégrée de planification et d'ordonnancement s'appuyant sur un couplage de deux blocs, l'un déterminant un plan de production $X(y)$ pour un ordonnancement y fixé, l'autre déterminant un ordonnancement $Y(x)$ pour un plan de production x fixé. Le principe de fonctionnement est le suivant : pour un ordonnancement fixé, le module de planification construit un plan de production optimal respectant ce séquençement. Le plan déterminé est envoyé dans le module d'ordonnancement afin d'obtenir le meilleur ordonnancement possible permettant de le réaliser : le temps ainsi libéré permettra de planifier la production d'autres articles au niveau du module de planification. Cette procédure s'arrêtera soit après un certain nombre d'itérations, soit lorsque l'ordonnancement obtenu dans le module d'ordonnancement n'évoluera plus (on ne pourra plus gagner de temps). Les auteurs soulignent la modularité de leur

méthode : chaque bloc agit comme une « boîte noire » vis-à-vis de l'autre. Ils sont indépendants et communiquent par fichier. La figure 5.1 illustre cette approche.

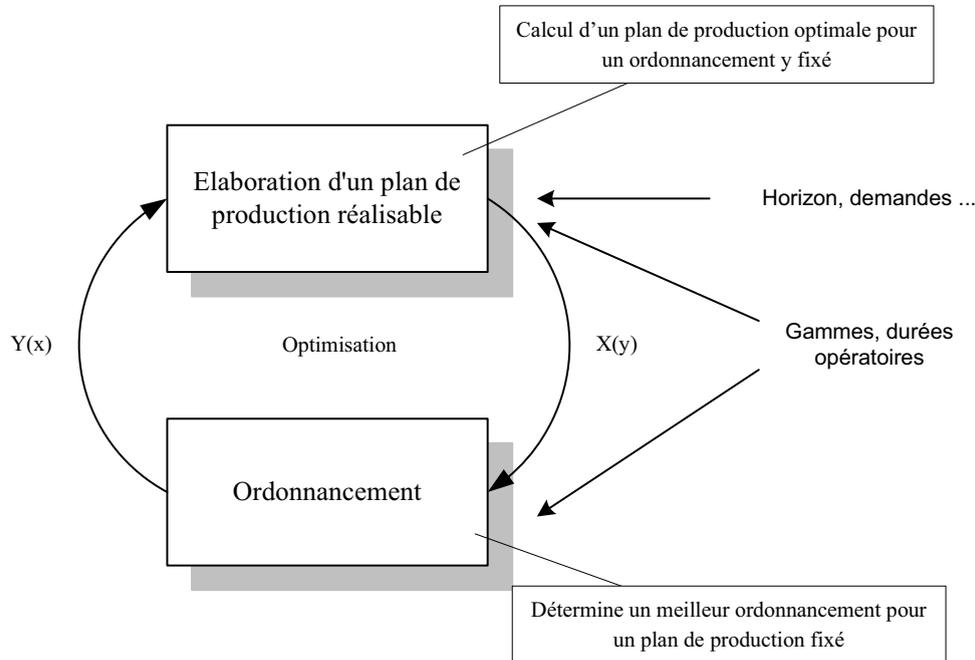


Figure 5.1 – Approche par intégration de contraintes opérationnelles (Dauzère Pérès et Lasserre 1994)

Les auteurs ont montré qu’une fois la séquence fixée, le modèle de planification obtenu pouvait être résolu assez facilement grâce à l’utilisation d’un solveur : la difficulté réside donc dans la résolution du problème d’ordonnancement. Or la convergence de la procédure vers un optimum local a été démontrée dans le cas idéal où le module d’ordonnancement calcule une solution optimale, ce qui n’est pas le cas en pratique. Néanmoins, testée sur un échantillon de problèmes académiques, leur méthode a donné de très bons résultats en peu d’itérations.

(Wolosewicz et al. 2006) ont étendu cette approche pour prendre en compte les temps de reconfiguration du système de production, complexifiant de fait la résolution du problème de planification. C’est pour cela qu’ils ont développé des heuristiques lagrangiennes afin de résoudre efficacement le problème de planification.

Soulignons tout de même que, dans cette approche, les auteurs n’estiment pas les capacités du système mais assurent la faisabilité de la planification par l’intégration du séquençage de la production au cœur même du bloc de planification. De plus, les interactions entre les périodes sont permises : en effet, des produits dont la fabrication a été planifiée sur une certaine période peuvent être ordonnancés pour partie sur une période connexe. Cependant, il n’est pas inutile de remarquer que le plan déterminé ne sera réalisable au niveau ordonnancement qu’à la condition que l’ordonnancement déterminé soit scrupuleusement suivi, ce qui semble peu probable dans la pratique. De plus, l’utilisation d’un modèle mathématique pour l’ordonnancement ne permet pas une modélisation fine du système de production (par exemple, on ne prend pas en compte les phénomènes aléatoires tel que les pannes etc.) ce qui peut déboucher sur une infaisabilité du plan de production déterminé par l’approche intégrée.

2 Estimation des capacités du système de production

La seconde approche vise à déterminer pour chaque période les capacités du système. Contrairement à l’approche précédente, il n’y a pas d’interaction entre les périodes.

(Sikora et al. 1996) proposent une procédure dédiée à un atelier de type Flow-Shop où pour chaque période, est résolu itérativement le problème de séquençement des opérations sur les ressources et le problème de la planification en utilisant respectivement l'heuristique de (Palmer 1965) et une variante de l'heuristique de Silver-Meal prenant en compte les capacités du système de production (au travers d'un critère qu'ils nomment « utilité »).

Le principe de fonctionnement de cette approche est le suivant :

1. On se place à la première période,
2. On utilise le critère « utilité » pour déterminer quel produit va être ajouté à la production de la période courante et quelle va en être la taille de lot, en fonction de la capacité restante sur la machine goulet courante.
3. Le nouveau planning de production de la période est envoyé au module de séquençement pour en déduire le nouveau séquençement de la production pour cette période et envoie le résultat au module de test. Le module de séquençement fournit également au module de planification le nouveau séquençement pour une meilleure prise en compte des temps de setup, qui, dans leur cas d'étude, dépendent précisément du séquençement proposé.
4. Le module de test simule l'atelier de production et renvoie d'une part le makespan au module de séquençement, et d'autre part la machine goulet courante ainsi que sa capacité restante au module de planification.
5. S'il reste assez de capacité pour séquençer un autre produit alors on retourne à 2.
6. Si on n'est pas à la dernière période alors on passe à la période suivante et on retourne à 2.
7. Fin de la procédure

La figure 5.2 illustre le fonctionnement de cette procédure.

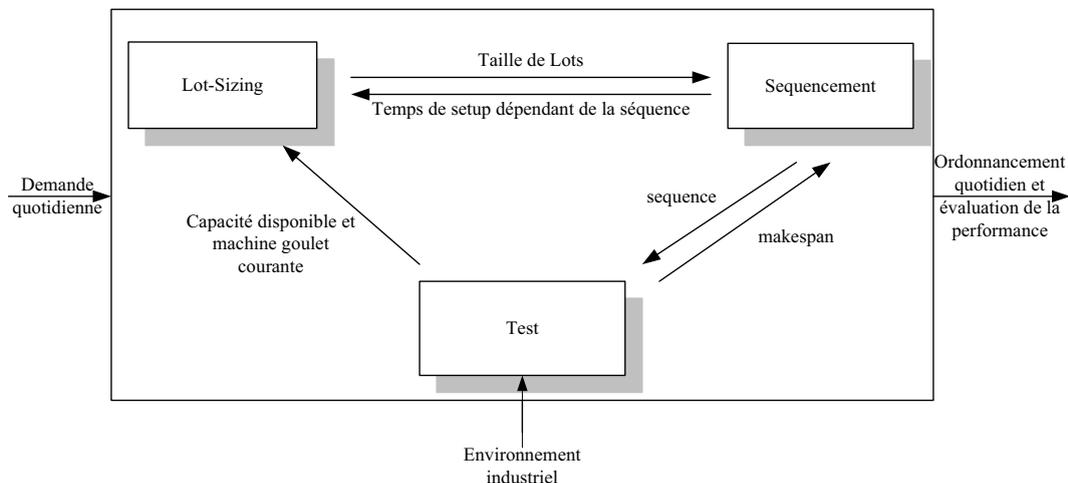


Figure 5.2 – Approche proposée
(Sikora et al. 1996)

Le fait que cette approche soit dédiée à un atelier de type Flow-Shop n'est pas anodin. En effet, le séquençement de la production, la détermination de la machine goulet et surtout le calcul de capacité restante pour produire s'en trouvent tout à fait facilités : le fait que les produits aient la même gamme opératoire permet la détermination de la machine goulet (qui est considérée par les auteurs comme étant celle disposant du moins de capacité disponible). Du coup, cette approche paraît très difficilement adaptable à un autre type d'atelier ce qui réduit d'autant son applicabilité dans un contexte industriel.

(Byrne et Bakir 1999) et (Byrne et Hossain 2005) proposent une méthodologie s'appuyant sur le couplage entre un modèle mathématique de planification et un modèle de simulation sensé décrire au mieux le fonctionnement de l'atelier. Le principe de fonctionnement est le suivant : le modèle de

planification prend en paramètre la demande et la capacité du système. Il élabore un planning optimal (en termes de coût). Ce dernier est passé en paramètre au modèle de simulation. Celui-ci vérifie si le planning proposé est réalisable. Dans le cas contraire, les capacités sont réévaluées en fonction des résultats obtenus par le modèle de simulation.

La figure 5.3 illustre cette approche.

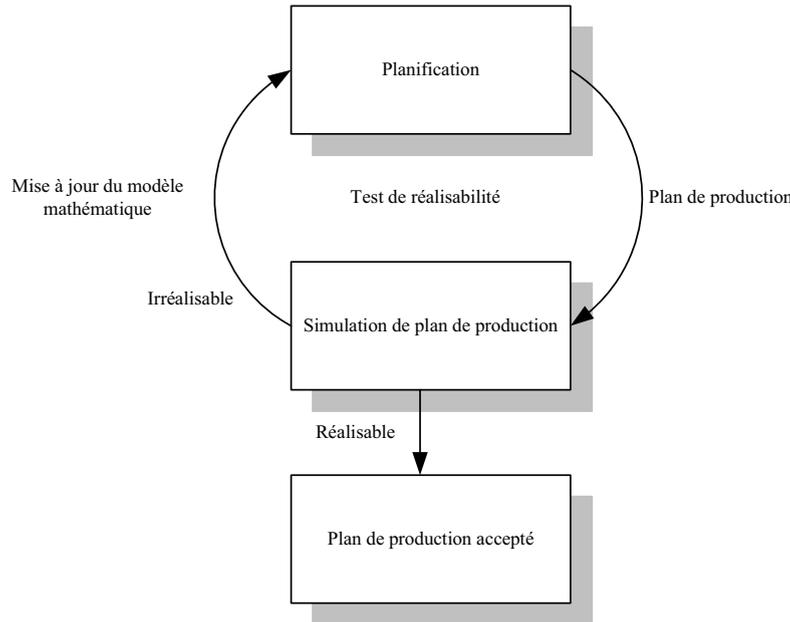


Figure 5.3 – Approche par couplage modélisation - simulation
(Byrne et Bakir 1999)

Cette approche a été reprise par (Kim et Kim 2001) qui utilisent un modèle de planification multi-ressources et utilisent les statistiques obtenues par le modèle de simulation afin de réajuster la capacité de chaque ressource indépendamment. Enfin (Gnoni et al. 2003) utilisent le même concept en l'élargissant à la planification d'une Supply Chain. Se basant sur un modèle de simulation pour évaluer la faisabilité opérationnelle et pour réestimer la capacité du système de production au niveau de la planification, cette méthodologie revêt un caractère beaucoup plus générique que l'approche défendue par (Sikora et al. 1996). De plus, il en découle une applicabilité bien plus grande dans un contexte industriel. Cependant, (Byrne et Bakir 1999) soulignent que cette approche ne garantit pas la convergence vers une solution mais remarquent que sur les instances qu'ils ont traitées, celle-ci a été obtenue assez rapidement.

Cet état de l'art a permis de faire émerger deux philosophies bien différentes. La première s'appuie sur un couplage de modèles mathématiques et propose l'élaboration d'une solution réalisable en intégrant des contraintes opérationnelles au niveau du modèle de planification : ainsi, la capacité n'est pas réestimée. Cependant, elle implique l'utilisation de cet ordonnancement au niveau opérationnel pour assurer la faisabilité du planning de production, ce qui constitue une hypothèse assez improbable. La seconde, illustrée par les travaux de (Byrne et Bakir 1999), s'appuie sur une description très fine du comportement de l'atelier grâce à un couplage entre un modèle d'optimisation et un modèle de simulation : elle fournit une solution réalisable en réestimant les capacités de production. L'ordonnancement utilisé par le modèle de simulation n'étant pas optimal, cela laisse plus de latitude au niveau opérationnel pour réaliser la production demandée. Cependant, la convergence de cette méthode n'est pas assurée.

La première méthode que nous proposons se situe à la frontière de deux précédentes puisqu'elle s'appuie sur un couplage de deux modèles mathématiques et une réestimation des capacités du système de production. De plus, la convergence de la méthode est assurée bien que l'ordonnancement utilisé pour

la réestimation ne soit pas optimal (laissant ainsi une plus grande latitude opérationnelle). La seconde méthode se rapproche plus de celle présentée par (Byrne et Bakir 1999) mais elle utilise plusieurs modèles de simulation basés sur diverses règles de gestion d'atelier permettant de décrire précisément le comportement du système de production, tout en préservant la propriété de convergence.

II Approche par couplage de modèles mathématiques

Dans cette partie, qui a fait l'objet d'une publication dans (Gourgand et al. 2007), nous expliquons le principe de la méthode envisagée.

La méthodologie que nous proposons est illustrée par la figure 5.4.

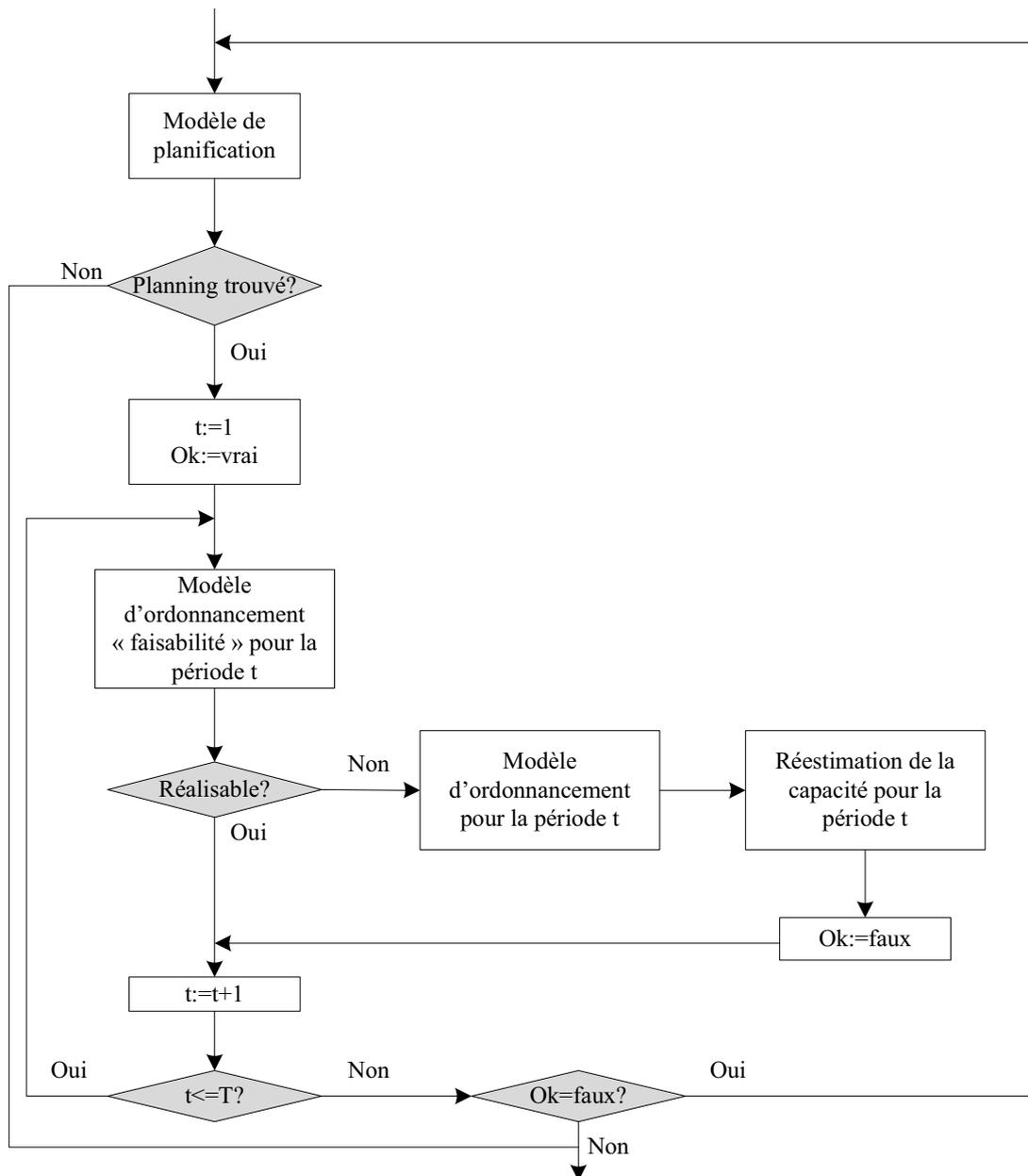


Figure 5.4 – Le principe de fonctionnement de notre méthodologie

Elle s'appuie sur une procédure itérative comprenant trois modèles mathématiques. Le premier d'entre eux est un modèle de planification tactique, il prend en paramètre la demande à planifier sur

tout l'horizon T ainsi que les capacités estimées du système de production. Le second est un modèle d'ordonnancement visant à déterminer la faisabilité de la solution fournie par le modèle tactique : pour une période donnée, il examine la possibilité de trouver au moins un ordonnancement de la production planifiée dont le makespan n'excède pas la fin de la période considérée. Ce dernier est appelé pour chaque période du plan de production. Le dernier modèle d'ordonnancement vise lui à obtenir un ordonnancement de la production pour une période donnée, tout en minimisant son makespan. Celui-ci sera utilisé si et seulement si le second modèle n'a pu trouver une solution faisable pour cette période et nous donnera le temps qu'il faudrait pour réaliser la production. La procédure utilisée se décompose en quatre grandes étapes :

1. A partir de la demande et des capacités estimées, le modèle de planification fournit un planning de production optimal en termes de coût. Ce planning donne pour chaque période de production les quantités d'articles à fabriquer. Celui-ci est envoyé au modèle d'ordonnancement « faisabilité ». Dans le cas où le module de planification ne parvient pas à trouver un plan de production satisfaisant les contraintes du modèle de « lot sizing », la procédure s'arrête : il n'existera donc pas de plan réalisable.
2. La seconde étape consiste à déterminer la faisabilité du plan tactique. Pour une période donnée, le modèle d'ordonnancement reçoit en entrée les quantités de chaque produit à fabriquer et examine la possibilité d'ordonnancer cette production sur cette même période. Cette étape s'effectue pour toutes les périodes de l'horizon. Si, pour toutes ces périodes, la réponse est positive, alors le plan proposé est réalisable : la procédure s'arrête. Sinon, la production planifiée pour les périodes pour lesquelles il n'existe pas un tel ordonnancement est dirigée vers le troisième modèle.
3. Le but de la troisième étape est de fournir, pour les périodes pour lesquelles l'étape précédente n'a pu apporter une réponse positive, une estimation du temps qu'il faudrait pour réaliser la production planifiée. C'est pour cela que celle-ci est composée d'un modèle visant à ordonnancer la production planifiée et dont l'objectif est de minimiser le makespan.
4. La dernière étape vise à réévaluer les capacités estimées du modèle de planification grâce aux résultats fournis par les étapes 2 et 3. Ceci se fait au travers de la formule suivante :

$$C_t^{(n+1)} = \left\lfloor \alpha_t C_t^{(n)} \right\rfloor \quad \forall t \in \llbracket 1, T \rrbracket$$

où $C_t^{(n)}$ représente la capacité du système à la période t à la n -ième itération du processus, T la longueur de l'horizon de planification et α_t le coefficient correctif. Celui-ci est calculé de la manière suivante : pour la période t ,

$$\alpha_t = \begin{cases} 1 & \text{si on a trouvé un ordonnancement réalisable à l'étape 2} \\ \frac{\text{Taille période}}{\text{Makespan obtenu à l'étape 3}} & \text{sinon} \end{cases}$$

Un plan fourni par le module de planification sera dit réalisable s'il est faisable au niveau opérationnelle, le cas extrême étant lorsque celui-ci est vide (c'est-à-dire dans le cas où rien n'est fabriqué).

La seconde étape peut paraître de prime abord inutile : en effet, la 3^e étape pourrait être considérée comme suffisante (en effet, on est sûr qu'au moins un ordonnancement permettant la fabrication sur cette période existe, si la solution optimale le permet), cependant, dans beaucoup de cas, les temps de calcul pour la seconde étape sont bien moindres que pour la troisième.

L'utilisation de la fonction « partie entière » dans la réévaluation de la capacité du système permet d'obtenir la convergence de la procédure comme le montre la proposition suivante :

Proposition 5.1 *Sous réserve d'existence d'un plan de production défini au niveau du module de planification tactique, la méthode converge vers un plan réalisable en un nombre fini d'itérations.*

Pour démontrer cette proposition, nous nous plaçons dans le cas où le module de planification tactique permet de déterminer à chaque itération un plan de production. Dans le cas contraire, la procédure pourrait s'arrêter d'elle-même sans plan réalisable.

Nous allons utiliser les deux lemmes suivants :

Lemme 5.1 La suite $(C_t^{(n)})_{n \in \mathbb{N}^*}$ est une suite décroissante de nombres entiers positifs, pour tout $t \in \llbracket 1, T \rrbracket$

Preuve :

Supposons que nous sommes à l'itération n , à la période t et que nous désirions réestimer la capacité du système pour la période suivante, nous avons donc :

$$C_t^{(n+1)} = \lfloor \alpha_t C_t^{(n)} \rfloor$$

Nous pouvons distinguer deux cas :

1. $C_t^{(n)} = 0$: dans cette configuration, $C_t^{(n+1)} = C_t^{(n)}$
2. $C_t^{(n)} > 0$: La valeur de $C_t^{(n+1)}$ va dépendre de la valeur que prendra α_t :
 - a. On a trouvé un ordonnancement réalisable à l'étape 2 pour cette période, dans ce cas $\alpha_t = 1$ donc $C_t^{(n+1)} = C_t^{(n)}$.
 - b. On n'a pas trouvé d'ordonnancement réalisable à l'étape 2, l'étape 3 nous a donc fourni un ordonnancement dont le makespan excède la longueur de la période t . Par suite, $0 < \alpha_t < 1$ ce qui entraîne que

$$\begin{aligned} 0 &\leq \alpha_t C_t^{(n)} < C_t^{(n)} && \text{donc que} \\ 0 &\leq \lfloor \alpha_t C_t^{(n)} \rfloor < C_t^{(n)} && \text{soit} \\ C_t^{(n+1)} &\leq C_t^{(n)} - 1 \end{aligned}$$

La suite $(C_t^{(n)})_{n \in \mathbb{N}^*}$ est donc bien une suite décroissante de nombres entiers positifs.

Dans la suite, nous considérons la suite $(u_n)_{n \in \mathbb{N}^*}$ de terme général

$$u_n = \sum_{t=1}^T C_t^{(n)}$$

Lemme 5.2 La suite $(u_n)_{n \in \mathbb{N}^*}$ est une suite décroissante de nombres entiers positifs.

Preuve :

Calculons $u_n - u_{n+1}$:

$$\begin{aligned} u_n - u_{n+1} &= \sum_{t=1}^T C_t^{(n)} - \sum_{t=1}^T C_t^{(n+1)} \\ &= \sum_{t=1}^T \underbrace{(C_t^{(n)} - C_t^{(n+1)})}_{\geq 0 \text{ via le lemme 5.1}} \end{aligned}$$

La suite $(u_n)_{n \in \mathbb{N}^*}$ est donc bien une suite décroissante de nombres entiers positifs.

Preuve de la proposition :

Examinons le comportement de la suite $(u_n)_{n \in \mathbb{N}^*}$ qui, d'après le lemme 5.2, est décroissante. Deux cas sont à considérer :

Cas 1 : la suite $(u_n)_{n \in \mathbb{N}^*}$ n'est pas strictement décroissante (dans ce cas au moins deux éléments consécutifs de la suite sont égaux),

Cas 2 : la suite $(u_n)_{n \in \mathbb{N}^*}$ est strictement décroissante.

Examinons successivement les deux éventualités possibles :

Cas 1 : La suite n'est pas strictement décroissante.

Il existe donc $k \in \mathbb{N}$ tel que $u_k = u_{k+1}$, donc $k \in \mathbb{N}$ tel que $u_k - u_{k+1} = 0$.

Or $u_k - u_{k+1} = \sum_{t=1}^T \underbrace{(C_t^{(k)} - C_t^{(k+1)})}_{\geq 0 \text{ via le lemme 5.1}}$ donc $\forall t \in \llbracket 1, T \rrbracket C_t^{(k)} = C_t^{(k+1)}$.

Nous avons donc $\lfloor \alpha_t C_t^{(k)} \rfloor = C_t^{(k)}$. Nous avons donc deux possibilités :

Possibilité n°1 : $\forall t \in \llbracket 1, T \rrbracket C_t^{(k)} = 0$. Le planning de production est vide, il est donc bien réalisable au niveau ordonnancement : la procédure s'arrête.

Possibilité n°2 : $\exists t \in \llbracket 1, T \rrbracket C_t^{(k)} > 0$. Dans ce cas $\alpha_t = 1$, donc il existe bien un ordonnancement réalisable à l'étape 2 pour cette période. De plus, pour les périodes t' où $C_{t'}^{(k)} = 0$, il n'y a pas de fabrication donc celles-ci sont bien réalisables au niveau ordonnancement. Ainsi, nous avons déterminé un plan réalisable : la procédure s'arrête.

Cas 2 : La suite est strictement décroissante.

On a donc $\forall k \in \mathbb{N}^* u_{k+1} < u_k$ et donc, plus précisément, $\forall k \in \mathbb{N}^* u_{k+1} \leq u_k - 1$.

Nous pouvons donc écrire les relations suivantes :

$$\begin{array}{rcl} u_{k+1} & \leq & u_k - 1 \\ u_k & \leq & u_{k-1} - 1 \\ \vdots & \vdots & \vdots \\ u_2 & \leq & u_1 - 1 \\ \hline u_{k+1} & \leq & u_1 - k \end{array}$$

u_1 étant la somme des capacités estimées initiales. Nous pouvons donc voir que dès que k sera supérieur à u_1 , alors u_{k+1} deviendra strictement négatif ce qui est impossible. Ainsi la suite $(u_n)_{n \in \mathbb{N}^*}$ ne peut être strictement décroissante ce qui nous assure la convergence de la méthode en au plus u_1 itérations (voir Cas 1).

Cependant, il est facile de voir que nous allons nous heurter à deux difficultés de deux ordres différents :

- la première est liée à la résolution de deux modèles : la complexité des modèles de lot-sizing et d'ordonnancement utilisés dans le monde industriel jointe à la très grande taille des instances à traiter montrent les limites de l'utilisation d'un solveur ou de méthodes exactes pour l'application d'un tel principe. Un moyen d'y pallier est l'utilisation de méthodes approchées.
- la seconde est liée à la mise en oeuvre d'un tel procédé dans un contexte industriel. En effet, dans ce cas-là, les ordonnancements de la production se font plus par application de règles de gestion basiques que par application d'une solution déterminée par une méthode d'optimisation. Cela signifie que l'utilisation d'un modèle mathématique ou d'une méthode d'optimisation au niveau de l'ordonnancement de la production n'est peut être pas le meilleur moyen de représenter le fonctionnement de l'atelier et donc d'assurer la réalisabilité du planning de production obtenu.

Pour ces deux raisons, nous proposons une modification de notre approche permettant de prendre en compte ces deux points tout en conservant les propriétés de convergence démontrées précédemment. Le modèle de planification pourra être remplacé par une méthode approchée : nous parlerons désormais de « module de planification » et non plus de « modèle de planification » et nous remplacerons les modèles d'ordonnancement par une série de modèles de simulation décrivant au mieux les règles de gestion en vigueur dans le système de production.

III Approche par couplage d'une méthode approchée et d'un modèle de simulation

Dans cette partie, nous proposons une adaptation de l'approche précédente permettant une meilleure prise en compte des contraintes industrielles au niveau de l'ordonnancement. De la même façon que

(Byrne et Bakir 1999), nous nous appuyons sur une approche par simulation du système de production. Cependant, nous proposons de modéliser les différentes règles de gestion de l'atelier au travers de plusieurs modèles de simulation et de nous en servir pour évaluer le temps nécessaire à la fabrication des produits prévus dans le planning de production pour chaque période : nous baserons donc la réestimation de la capacité sur un paramètre dépendant des résultats obtenus par l'ensemble de ces simulations. On peut facilement remarquer que la réalisabilité d'un plan va donc être fonction de la règle de gestion qui est appliquée au niveau ordonnancement. Cela induit donc que les décideurs en charge de la planification définissent un « taux d'acceptation d'une solution » : ils doivent donc définir la réalisabilité d'un plan comme étant sa propension à pouvoir être ordonnancé pour un certain pourcentage de règles de gestion sans excéder la taille de la période (par exemple, ils peuvent décider qu'un plan est réalisable s'il est réalisable pour 90% des règles de gestion). Désormais, nous parlerons de « taux » pour « taux d'acceptation d'une solution ».

Par la suite, nous noterons par N le nombre de simulations (donc de règles de gestion) considéré.

L'approche que nous proposons est illustrée par la figure 5.5.

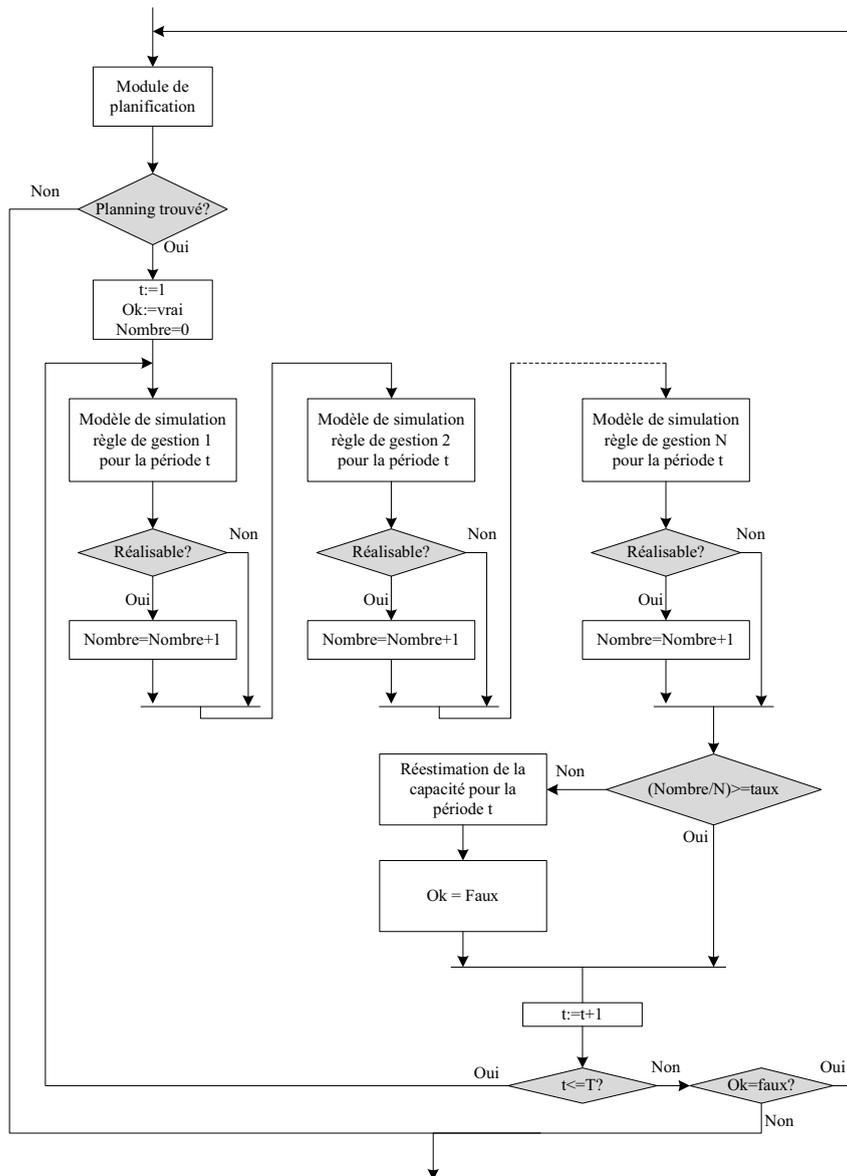


Figure 5.5 – Le principe de fonctionnement de notre méthodologie

Elle s'appuie donc sur une procédure itérative comportant 3 étapes :

1. Pour l'itération n , à partir de la demande et des capacités estimées $C_t^{(n)}$, le modèle de planification fournit un planning de production optimal en termes de coût. Ce planning donne pour chaque période de production les quantités d'articles à fabriquer. Celui-ci est envoyé au module d'ordonnancement « faisabilité ». Dans le cas où le module de planification ne parvient pas à trouver un plan de production satisfaisant les contraintes du modèle de « lot sizing », la procédure s'arrête : il n'existera donc pas de plan réalisable.
2. La seconde étape consiste à déterminer la faisabilité du plan tactique. Pour une période donnée, chaque modèle de simulation reçoit en entrée les quantités de chaque produit à fabriquer et évalue le temps nécessaire à la fabrication de cette production en suivant la règle de gestion qui lui est dédiée. Si, pour une simulation, le temps de fabrication obtenu (le makespan) est inférieur au nombre d'heures d'ouverture de l'atelier pour cette période alors le plan est considéré comme réalisable pour cette règle de gestion. Si, à la fin de toutes les simulations de cette période, le ratio entre le nombre de fois où le plan a été réalisable et le nombre de simulation est supérieur ou égal au taux défini par les décideurs, alors le plan est considéré comme réalisable pour cette période. Sinon, il va falloir réestimer la capacité de cette période. Cette étape s'effectue pour toutes les périodes de l'horizon. Si, pour toutes ces périodes, la réponse est positive, alors le plan proposé est réalisable : la procédure s'arrête. Sinon, on passe à la troisième étape qui consiste à réestimer les capacités pour les périodes où le plan n'est pas réalisable.
3. La troisième étape consiste donc à réestimer les capacités pour les périodes dont le plan n'est pas réalisable. A l'instar de la méthodologie basée sur un couplage de modèle mathématique, celui-ci va se faire par le biais du multiplicateur α_t de la façon suivante :
 - si pour la période t le plan est réalisable, alors $\alpha_t = 1$ ce qui revient à ne pas réestimer la capacité,
 - si pour la période t le plan n'est pas réalisable, alors
 - nous calculons $numero = \min(N, \lfloor \text{taux} \times N \rfloor + 1)$: cela nous donne le nombre de modèles de simulation dont la réponse aurait dû être positive pour que le plan soit réalisable,
 - nous classons par ordre croissant les makespans obtenus pour cette période et nous considérons celui positionné à la place « numéro »,
 - nous calculons $\alpha_t = \frac{\text{Taille période}}{\text{Makespan obtenu par le modèle de simulation numéro}}$
 Pour l'itération $n + 1$, la capacité de la période est donc réestimée par la formule :

$$C_t^{(n+1)} = \left\lfloor \alpha_t C_t^{(n)} \right\rfloor$$

Comme nous pouvons le remarquer, la façon dont est réestimée la capacité permet de conserver les propriétés vues dans la partie précédente : on a donc bien une convergence en un nombre fini d'itérations.

Nous venons d'exposer une adaptation de notre approche prenant en compte une description très fine du comportement de l'atelier, via des modèles de simulation. Il est facile de voir que cette dernière est capable de prendre en compte des événements aléatoires inhérents au fonctionnement d'un système de production (pannes, etc.) : en effet, pour chaque période il suffit de faire plusieurs répliques de chaque modèle de simulation. Le critère de réussite pour une règle de gestion ne sera plus le makespan mais le makespan moyen de toutes les répliques du modèle simulation lui correspondant pour cette période. Le reste de l'approche restera inchangée.

IV Illustration de l'approche utilisant la simulation sur la planification d'un atelier de type Flow-Shop hybride

Pour illustrer l'approche par couplage de méthode approchée et de modèles de simulation, nous proposons d'élaborer le PDP à la maille semaine d'une usine fabriquant 6 produits finis. La demande client hebdomadaire pour l'horizon de planification est donnée par le tableau 5.1

	S. 1	S. 2	S. 3	S. 4	S. 5	S. 6	S. 7	S. 8	S. 9	S. 10
produit1	50	10	0	100	50	0	0	30	0	100
produit2	10	0	70	0	20	20	30	0	50	0
produit3	0	100	0	10	50	0	50	20	10	0
produit4	10	5	0	30	10	50	5	10	50	100
produit5	30	0	30	30	10	10	5	30	10	60
produit6	0	20	10	0	10	10	5	40	30	1

Tableau 5.1 – La demande client à satisfaire

L'usine considérée est formée d'un atelier de type Flow-Shop Hybride à trois étages représenté par la figure 5.6

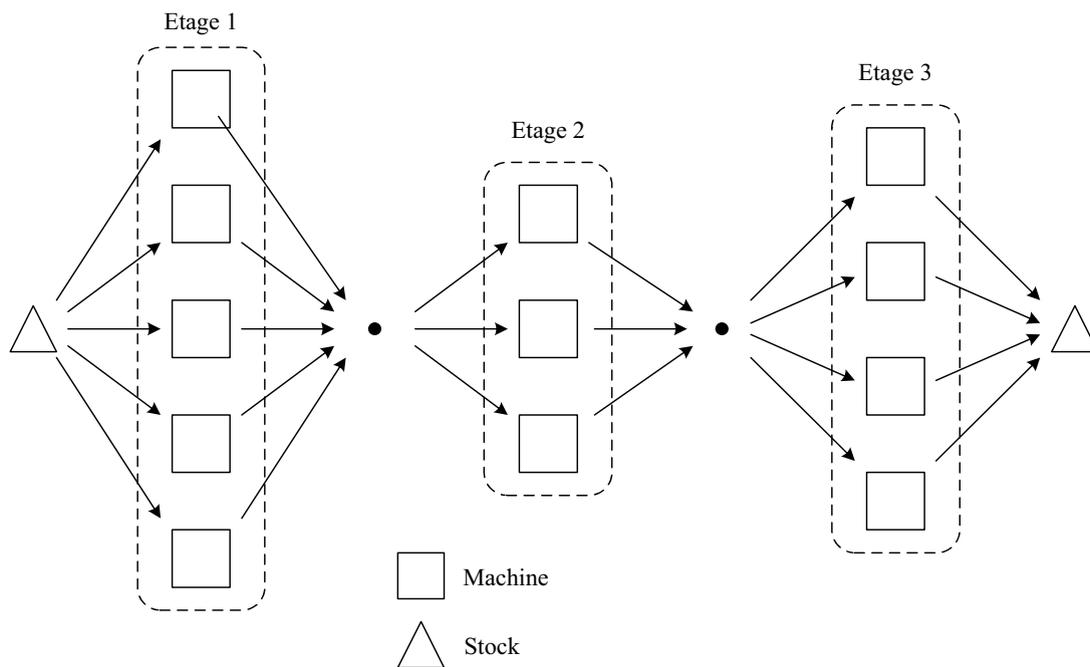


Figure 5.6 – L'atelier simulé

Le premier étage est formé de 5 machines identiques, le second de 3 et le troisième de 4.

Entre chaque étage, nous n'avons pas de stock intermédiaire : les seuls stocks présents sont ceux en début et fin de l'atelier.

Les temps opératoires (en heure) de chaque produit sur chaque étage sont donnés par le tableau 5.2.

	Produit1	Produit 2	Produit 3	Produit 4	Produit 5	Produit 6
Etage 1	0,1	0,15	0,05	0,01	0,5	0,02
Etage 2	0,2	0,3	0,5	0,6	0,8	0,1
Etage 3	0,1	0,2	0,5	0,4	0,3	0,7
Total	0,4	0,65	1,05	1,01	1,6	0,82

Tableau 5.2 – Temps de gamme pour les produits finis considérés

Notre but étant d'élaborer le PDP de l'usine, nous nous appuyons sur le modèle du CLSP pour élaborer ce planning. Nous faisons donc l'hypothèse de considérer l'usine comme une ressource unique ayant une capacité hebdomadaire (capacité agrégée sensée représenter la capacité de production de l'atelier) : nous supposons que chaque semaine, l'usine est ouverte 40h00 et donc que la capacité de production de chaque machine est de 40h00 hebdomadaire. Nous prenons arbitrairement 300h00 comme capacité agrégée hebdomadaire de l'usine (ce qui correspond à un peu plus de la somme des capacités de la moitié des machines de l'atelier) et pour capacité consommée pour la fabrication d'un produit la somme des temps de fabrication du produit pour chaque étage (c.f tableau 5.2)

Les coûts engendrés par la production (lancement et stockage) sont données par le tableau 5.3

	Produit 1	Produit 2	Produit 3	Produit 4	Produit 5	Produit 6
lancement	500	400	0	300	250	200
stockage	1	1	1	1	1	1

Tableau 5.3 – Coûts engendrés par la fabrication des produits

Comme la planification de la production de notre usine s'appuie sur le modèle du CLSP, le module de planification de notre approche sera constituée de la métaheuristique bi-objectif définie dans le chapitre III.

En ce qui concerne la modélisation du comportement de notre atelier, nous supposons qu'il peut être soumis à 12 règles de gestion différentes, toutes basées sur un ordonnancement par lot. Ainsi on retrouvera les règles d'ordonnancement par lot suivant le produit :

- règle 1 : ayant le numéro le plus petit au produit ayant le numéro le plus grand,
- règle 2 : ayant le numéro le plus grand au produit ayant le numéro le plus petit,
- règle 3 : dont la quantité à produire est la plus petite au produit dont la quantité à produire est la plus grande,
- règle 4 : dont la quantité à produire est la plus grande au produit dont la quantité à produire est la plus petite,
- règle 5 : ayant le temps opératoire le plus court au produit ayant le temps opératoire le plus long sur l'étage 1,
- règle 6 : ayant le temps opératoire le plus long au produit ayant le temps opératoire le plus court sur l'étage 1,
- règle 7 : ayant le temps opératoire le plus court au produit ayant le temps opératoire le plus long sur l'étage 2,
- règle 8 : ayant le temps opératoire le plus long au produit ayant le temps opératoire le plus court sur l'étage 2,
- règle 9 : ayant le temps opératoire le plus court au produit ayant le temps opératoire le plus long sur l'étage 3,
- règle 10 : ayant le temps opératoire le plus long au produit ayant le temps opératoire le plus court sur l'étage 3,
- règle 11 : ayant le temps opératoire globale le plus court au produit ayant le temps opératoire globale le plus long (temps opératoire globale = somme des temps opératoire de chaque étage),
- règle 12 : ayant le temps opératoire globale le plus long au produit ayant le temps opératoire globale le plus court.

Nous définissons ainsi 12 modèles de simulation qui entreront en compte dans la seconde étape de la démarche proposée. Enfin, il nous reste à fixer le taux d'acceptation de la solution. Nous décidons de

le fixer à 80% : cela signifie que le planning sera réalisable si, pour chaque période, 80% des makespans donnés par les modèles de simulation est inférieur à 40h00.

les tableaux suivants donnent les résultats obtenus à chaque itération.

Les capacités initiales ont été fixées à 300 ($C_t^{(1)} = 300, \forall t \in \llbracket 1, 10 \rrbracket$). Le planning de production obtenu est donnée par le tableau 5.4.

Itération n°1	P. 1	P. 2	P. 3	P. 4	P. 5	P. 6	P. 7	P. 8	P. 9	P. 10
Capacité	300	300	300	300	300	300	300	300	300	300
produit1	210	0	0	0	0	0	0	130	0	0
produit2	80	0	0	0	120	0	0	0	0	0
produit3	0	100	0	10	50	0	50	20	10	0
produit4	110	0	0	0	0	0	0	160	0	0
produit5	30	0	115	0	0	0	0	0	70	0
produit6	0	50	0	0	0	0	76	0	0	0

Tableau 5.4 – Planning obtenu à la première itération

Le planning de production est passé période par période aux modèles de simulation. Les résultats obtenus sont consignés dans le tableau 5.5.

Itération n°1	P. 1	P. 2	P. 3	P. 4	P. 5	P. 6	P. 7	P. 8	P. 9	P. 10
Makespan max	40	40	40	40	40	40	40	40	40	40
Règle n° 1	52,7	25,95	32	2,55	21,15	0	22,35	44,7	21,05	0
Règle n° 2	52,7	25,72	32	2,55	20,75	0	21,92	44,21	21,4	0
Règle n° 3	52,7	25,72	32	2,55	20,75	0	22,35	44,75	21,05	0
Règle n° 4	52,5	25,95	32	2,55	21,15	0	21,92	44,71	21,4	0
Règle n° 5	52,7	25,95	32	2,55	21,15	0	22,35	44,7	21,05	0
Règle n° 6	52,7	25,95	32	2,55	21,15	0	22,35	44,7	21,05	0
Règle n° 7	52,7	25,95	32	2,55	21,15	0	22,35	44,7	21,05	0
Règle n° 8	52,7	25,95	32	2,55	21,15	0	22,35	44,7	21,05	0
Règle n° 9	52,7	25,95	32	2,55	21,15	0	22,35	44,7	21,05	0
Règle n° 10	52,7	25,95	32	2,55	21,15	0	22,35	44,7	21,05	0
Règle n° 11	52,7	25,72	32	2,55	21,15	0	21,92	44,7	21,05	0
Règle n° 12	52,31	25,95	32	2,55	20,75	0	22,35	44,25	21,05	0

Tableau 5.5 – Makespans obtenus à la première itération

Comme on peut le voir, les makespans obtenus sont assez proches : cela est dû aux instances considérées. En effet, les temps de gamme sont assez proches surtout au niveau de l'étage 2 qui est l'étage goulet et, de plus, le flow shop hybride considéré est sans stock.

Les makespans obtenus par les différentes simulations excèdent 40h00 pour les périodes 1 et 8, ce

qui nécessite donc une réestimation de la capacité pour ces périodes. On calcule donc

$$\alpha_1 = \frac{40}{52,7} \approx 0,759 \text{ ce qui nous donne } C_1^{(2)} = [0,759 \times 300] = 227$$

$$\alpha_8 = \frac{40}{44,7} \approx 0,895 \text{ ce qui nous donne } C_8^{(2)} = [0,895 \times 300] = 268$$

Les capacités sont donc mises à jour et la seconde itération peut donc débuter. Le planning de production obtenu après la seconde itération est donné par le tableau 5.6.

Itération n°2	P. 1	P. 2	P. 3	P. 4	P. 5	P. 6	P. 7	P. 8	P. 9	P. 10
Capacité	227	300	300	300	300	300	300	268	300	300
produit1	210	0	0	0	0	0	0	130	0	0
produit2	120	0	0	0	0	0	80	0	0	0
produit3	0	100	0	10	50	0	50	20	10	0
produit4	10	110	0	0	0	0	0	0	150	0
produit5	30	0	80	0	0	0	105	0	0	0
produit6	0	55	0	0	0	0	0	71	0	0

Tableau 5.6 – Planning obtenu à la seconde itération

Le planning de production est de nouveau passé période par période aux modèles de simulation. Les résultats obtenus sont consignés dans le tableau 5.7.

Itération n°2	P. 1	P. 2	P. 3	P. 4	P. 5	P. 6	P. 7	P. 8	P. 9	P. 10
Makespan max	40	40	40	40	40	40	40	40	40	40
Règle n° 1	36,5	48,85	22,4	2,55	9,05	0	45,05	25,1	32,45	0
Règle n° 2	36,7	48,52	22,4	2,55	9,05	0	45,2	24,32	32,51	0
Règle n° 3	36,21	48,52	22,4	2,55	9,05	0	44,85	24,65	32,45	0
Règle n° 4	36,9	49,01	22,4	2,55	9,05	0	45,6	24,7	32,51	0
Règle n° 5	36,5	48,85	22,4	2,55	9,05	0	45,05	25,1	32,45	0
Règle n° 6	36,5	48,85	22,4	2,55	9,05	0	45,05	25,1	32,45	0
Règle n° 7	36,5	48,85	22,4	2,55	9,05	0	45,05	25,1	32,45	0
Règle n° 8	36,5	48,85	22,4	2,55	9,05	0	45,05	25,1	32,45	0
Règle n° 9	36,5	48,85	22,4	2,55	9,05	0	45,05	25,1	32,45	0
Règle n° 10	36,5	48,85	22,4	2,55	9,05	0	45,05	25,1	32,45	0
Règle n° 11	36,5	48,52	22,4	2,55	9,05	0	45,05	24,7	32,45	0
Règle n° 12	36,21	48,85	22,4	2,55	9,05	0	44,65	24,75	32,45	0

Tableau 5.7 – Makespans obtenus à la seconde itération

Cette fois-ci, les makespans obtenus par les différentes simulations excèdent 40h00 pour les périodes 2 et 7, ce qui nécessite donc une réestimation de la capacité pour ces deux périodes. On calcule donc

$$\alpha_2 = \frac{40}{48,85} \approx 0,819 \text{ ce qui nous donne } C_2^{(3)} = [0,819 \times 300] = 245$$

$$\alpha_7 = \frac{40}{45,05} \approx 0,888 \text{ ce qui nous donne } C_7^{(3)} = \lfloor 0,888 \times 300 \rfloor = 266$$

Les capacités sont donc mises à jour et la troisième itération peut donc débuter. Le planning de production obtenu après cette itération est donné par le tableau 5.8.

Itération n°3	P. 1	P. 2	P. 3	P. 4	P. 5	P. 6	P. 7	P. 8	P. 9	P. 10
Capacité	227	245	300	300	300	300	266	268	300	300
produit1	60	0	0	180	0	0	0	0	0	100
produit2	120	0	0	0	0	0	80	0	0	0
produit3	0	100	0	10	50	0	50	20	10	0
produit4	10	100	0	0	0	0	0	160	0	0
produit5	30	0	125	0	0	0	0	0	0	60
produit6	0	30	0	0	25	0	0	71	0	0

Tableau 5.8 – Planning obtenu à la troisième itération

Le planning de production est de nouveau passé période par période aux modèles de simulation. Les résultats obtenus sont consignés dans le tableau 5.9.

Itération n°3	P. 1	P. 2	P. 3	P. 4	P. 5	P. 6	P. 7	P. 8	P. 9	P. 10
Makespan max	40	40	40	40	40	40	40	40	40	40
Règle n° 1	26,5	42,35	34,4	14,4	13,25	0	17,25	48,35	2,55	23,3
Règle n° 2	26,7	42,32	34,4	13,95	12,82	0	16,75	47,92	2,55	23,4
Règle n° 3	26,21	42,22	34,4	13,95	12,82	0	16,75	48,15	2,55	23,4
Règle n° 4	26,95	42,35	34,4	14,4	13,25	0	17,25	48,11	2,55	23,3
Règle n° 5	26,5	42,35	34,4	14,4	13,25	0	17,25	48,35	2,55	23,3
Règle n° 6	26,5	42,35	34,4	14,4	13,25	0	17,25	48,35	2,55	23,3
Règle n° 7	26,5	42,35	34,4	14,4	13,25	0	17,25	48,35	2,55	23,3
Règle n° 8	26,5	42,35	34,4	14,4	13,25	0	17,25	48,35	2,55	23,3
Règle n° 9	26,5	42,35	34,4	14,4	13,25	0	17,25	48,35	2,55	23,3
Règle n° 10	26,5	42,35	34,4	14,4	13,25	0	17,25	48,35	2,55	23,3
Règle n° 11	26,5	42,22	34,4	14,4	12,82	0	17,25	48,02	2,55	23,3
Règle n° 12	26,21	42,35	34,4	13,95	13,25	0	16,75	48,35	2,55	23,4

Tableau 5.9 – Makespans obtenus à la troisième itération

De nouveau, les makespans obtenus par les différentes simulations excèdent 40h00 pour les périodes 2 et 8, on va donc encore une fois réestimer la capacité pour ces deux périodes. On calcule donc :

$$\alpha_2 = \frac{40}{42,35} \approx 0,945 \text{ ce qui nous donne } C_2^{(4)} = \lfloor 0,945 \times 245 \rfloor = 231$$

$$\alpha_8 = \frac{40}{48,35} \approx 0,827 \text{ ce qui nous donne } C_8^{(4)} = \lfloor 0,827 \times 268 \rfloor = 221$$

Les capacités sont donc mises à jour et la quatrième itération peut donc débuter. Le planning de production obtenu après cette itération est donné par le tableau 5.10.

Itération n°4	P. 1	P. 2	P. 3	P. 4	P. 5	P. 6	P. 7	P. 8	P. 9	P. 10
Capacité	227	231	300	300	300	300	266	221	300	300
produit1	60	0	0	180	0	0	0	0	0	100
produit2	10	0	190	0	0	0	0	0	0	0
produit3	0	100	0	10	50	0	50	20	10	0
produit4	55	0	0	0	0	65	0	0	150	0
produit5	45	0	110	0	0	0	0	0	0	60
produit6	0	55	0	0	0	0	0	71	0	0

Tableau 5.10 – Planning obtenu à la quatrième itération

Le planning de production est de nouveau passé période par période aux modèles de simulation. Les résultats obtenus sont consignés dans le tableau 5.11.

Itération n°4	P. 1	P. 2	P. 3	P. 4	P. 5	P. 6	P. 7	P. 8	P. 9	P. 10
Makespan max	40	40	40	40	40	40	40	40	40	40
Règle n° 1	28,4	26,85	49,15	14,4	9,05	13,61	9,05	16,45	32,45	23,3
Règle n° 2	28,7	26,72	49,2	13,95	9,05	13,61	9,05	16,22	32,51	23,4
Règle n° 3	28,35	26,72	49,2	13,95	9,05	13,61	9,05	16,45	32,45	23,4
Règle n° 4	28,4	26,85	49,15	14,4	9,05	13,61	9,05	16,22	32,51	23,3
Règle n° 5	28,4	26,85	49,15	14,4	9,05	13,61	9,05	16,45	32,45	23,3
Règle n° 6	28,4	26,85	49,15	14,4	9,05	13,61	9,05	16,45	32,45	23,3
Règle n° 7	28,4	26,85	49,15	14,4	9,05	13,61	9,05	16,45	32,45	23,3
Règle n° 8	28,4	26,85	49,15	14,4	9,05	13,61	9,05	16,45	32,45	23,3
Règle n° 9	28,4	26,85	49,15	14,4	9,05	13,61	9,05	16,45	32,45	23,3
Règle n° 10	28,4	26,85	49,15	14,4	9,05	13,61	9,05	16,45	32,45	23,3
Règle n° 11	28,4	26,72	49,15	14,4	9,05	13,61	9,05	16,22	32,45	23,3
Règle n° 12	28,41	26,85	49,2	13,95	9,05	13,61	9,05	16,45	32,45	23,4

Tableau 5.11 – Makespans obtenus à la troisième itération

Cette fois-ci, seuls les makespans obtenus pour le troisième période excèdent les 40h00 : on aura donc seulement à ajuster la capacité pour cette période-là. on aura donc :

$$\alpha_3 = \frac{40}{49,15} \approx 0,814 \text{ ce qui nous donne } C_3^{(5)} = [0,813 \times 300] = 243$$

Les capacités sont donc mises à jour et la cinquième itération peut donc débuter. Le planning de production obtenu après cette itération est donné par le tableau 5.12.

Itération n°5	P. 1	P. 2	P. 3	P. 4	P. 5	P. 6	P. 7	P. 8	P. 9	P. 10
Capacité	<i>227</i>	<i>231</i>	<i>243</i>	300	300	300	<i>266</i>	<i>221</i>	300	300
produit1	60	0	0	180	0	0	0	0	0	100
produit2	10	0	190	0	0	0	0	0	0	0
produit3	0	100	0	10	50	0	50	20	10	0
produit4	15	0	0	105	0	0	0	0	150	0
produit5	100	0	0	0	0	115	0	0	0	0
produit6	0	50	0	0	0	0	76	0	0	0

Tableau 5.12 – Planning obtenu à la dernière itération

Le planning de production est de nouveau passé période par période aux modèles de simulation. Les résultats obtenus sont consignés dans le tableau 5.13.

Itération n°5	P. 1	P. 2	P. 3	P. 4	P. 5	P. 6	P. 7	P. 8	P. 9	P. 10
Makespan max	40	40	40	40	40	40	40	40	40	40
Règle n° 1	35,2	25,95	19,55	35,3	9,05	32	22,35	4,05	32,45	7
Règle n° 2	35,4	25,72	19,55	34,91	9,05	32	21,92	4,05	32,51	7
Règle n° 3	35,75	25,72	19,55	34,95	9,05	32	22,35	4,05	32,45	7
Règle n° 4	35,5	25,95	19,55	35,4	9,05	32	21,92	4,05	32,51	7
Règle n° 5	35,2	25,95	19,55	35,3	9,05	32	22,35	4,05	32,45	7
Règle n° 6	35,2	25,95	19,55	35,3	9,05	32	22,35	4,05	32,45	7
Règle n° 7	35,2	25,95	19,55	35,3	9,05	32	22,35	4,05	32,45	7
Règle n° 8	35,2	25,95	19,55	35,3	9,05	32	22,35	4,05	32,45	7
Règle n° 9	35,2	25,95	19,55	35,3	9,05	32	22,35	4,05	32,45	7
Règle n° 10	35,2	25,95	19,55	35,3	9,05	32	22,35	4,05	32,45	7
Règle n° 11	35,2	25,72	19,55	35,3	9,05	32	21,92	4,05	32,45	7
Règle n° 12	35,01	25,95	19,55	34,95	9,05	32	22,35	4,05	32,45	7

Tableau 5.13 – Makespans obtenus à la dernière itération

Cette fois-ci, tous les makespans sont inférieurs à 40h00, et ceci pour toutes les périodes : le plan de production déterminé est donc considéré comme réalisable. La procédure s'arrête.

La figure 5.7 représente l'évolution de l'estimation des capacités du système de production durant la démarche proposée.

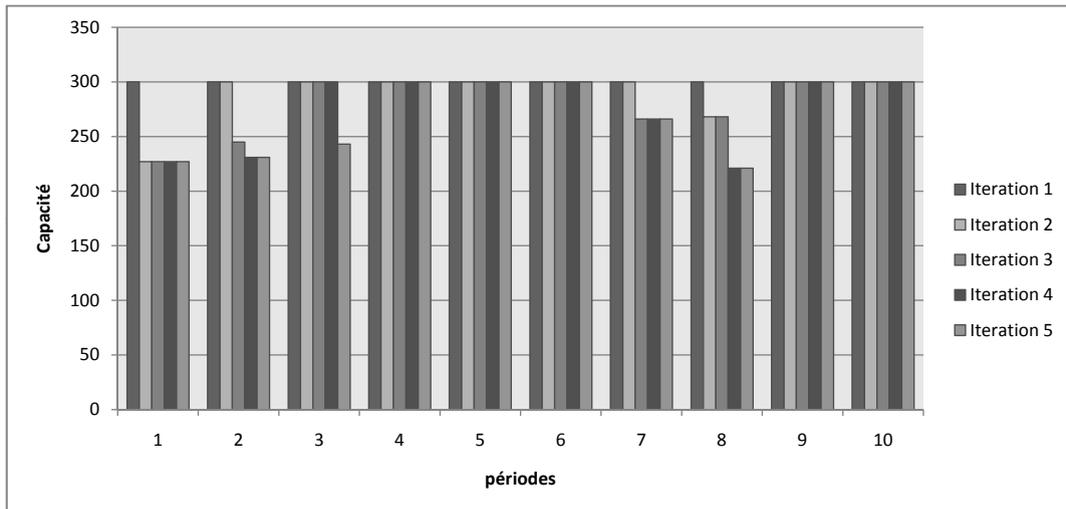


Figure 5.7 – Evolution des capacités suivant les itérations

Conclusion

Dans ce chapitre, nous avons présenté deux approches permettant d'assurer la synchronisation entre les planifications tactique et opérationnelle.

La première d'entre elles utilise un couplage entre deux modèles mathématiques : le premier est un modèle de planification tactique et le second un modèle d'ordonnancement, décliné en deux versions (l'un pour la faisabilité, l'autre pour la minimisation du makespan) et qui sert à réestimer les capacités de production si nécessaire. Nous avons prouvé sa convergence. Cependant, malgré son efficacité, nous avons montré ses limites liées à la complexité des modèles d'optimisation à résoudre ainsi qu'à la finesse de modélisation du fonctionnement d'un atelier de production.

La seconde approche que nous avons proposée vise à outrepasser les limites de la première : en effet, l'utilisation de modèle mathématique est abandonnée au profit de méthodes d'optimisation approchées (pour vaincre la complexité algorithmique lié à l'utilisation des modèles mathématiques) et de plusieurs modèles de simulation (pour décrire au plus prêt le comportement de l'atelier). Les propriétés de convergence ont été maintenues. Nous avons montré qu'il était possible de prendre en compte les événement aléatoires pouvant subvenir dans l'atelier (panne, etc.).

Bien que nous ayons pu vérifier le bon fonctionnement de ces deux approches au travers de cas d'étude, des tests sur des cas industriels devront être menés pour leur apporter une réelle validation.

Chapitre 6

Gestion des stocks à demande différenciée : application à un contexte industriel

Sommaire

Introduction	222
I Le contexte de l'étude	223
1 Les principales politiques de gestion des stocks	223
2 La gestion des stocks à demande différenciée	224
3 Les travaux du LIMOS	224
II Le projet	226
1 Etude de faisabilité et de condition du test	226
1.1 Interactions entre les outils de prise de commande et de gestion des stocks	227
1.2 Le processus de prise de commande	228
1.3 Intégration d'une politique de seuil au sein de la prise de commande	231
2 Conception de l'application informatique permettant le calcul des seuils	235
2.1 Le modèle de données	235
2.2 Le schéma fonctionnelle de l'application	237
2.3 L'algorithme du DRP	238
2.4 L'algorithme de gestion de stock	239
2.5 L'algorithme de calcul des seuils (Tempelmeier 2006)	240
Conclusion	240

Introduction

La problématique majeure de la planification tactique est de trouver un compromis entre les coûts logistiques (lancement de campagne de production, production, stock, transport etc.) et le taux de satisfaction de la demande. Dans les chapitres précédents, nous avons implicitement supposé que la demande était connue de façon certaine lors du processus de planification. Cependant, dans l'immense majorité des cas, il n'en est rien : la planification de la production s'appuie sur des prévisions de vente qui sont par nature tout à fait incertaines. Outre l'adéquation des plans de production à la demande réelle (et donc la pertinence d'un processus de planification), la question qui peut alors se poser est de savoir quelles stratégies adopter afin de pouvoir répondre aux fluctuations non prévues de cette demande et maintenir la qualité de la solution obtenue dans le processus de planification, en termes de coûts et de satisfaction de la demande.

Pour répondre à cette problématique, les entreprises mettent en place des « stocks de sécurité » dont le but est de pouvoir pallier cette incertitude sur la demande. En reprenant les processus logistiques définis par (Stadler 2005), c'est au niveau de la distribution des produits que va prendre place cette stratégie, c'est-à-dire au niveau du circuit de distribution : les entrepôts et les magasins représentés par la figure (6.1). Cependant, le dimensionnement de tels stocks est une entreprise délicate car, bien qu'ils permettent d'assurer en moyenne un niveau de satisfaction de la demande minimum, les coûts qu'ils engendrent ne sont pas négligeables. De plus, ces stocks ne sont en général pas dédiés à une certaine classe de client : cela implique que leur dimensionnement s'effectue en fonction des prévisions de tous les clients et que le taux de satisfaction obtenu est identique pour tous. L'unicité de ce taux de satisfaction oblige l'entreprise à dimensionner le stock de sécurité afin d'assurer à tous les clients le taux de satisfaction le plus élevé exigé, provoquant d'une part une inflation du niveau de stock et d'autre part une surqualité pour les clients ne demandant pas un taux de satisfaction aussi élevé.

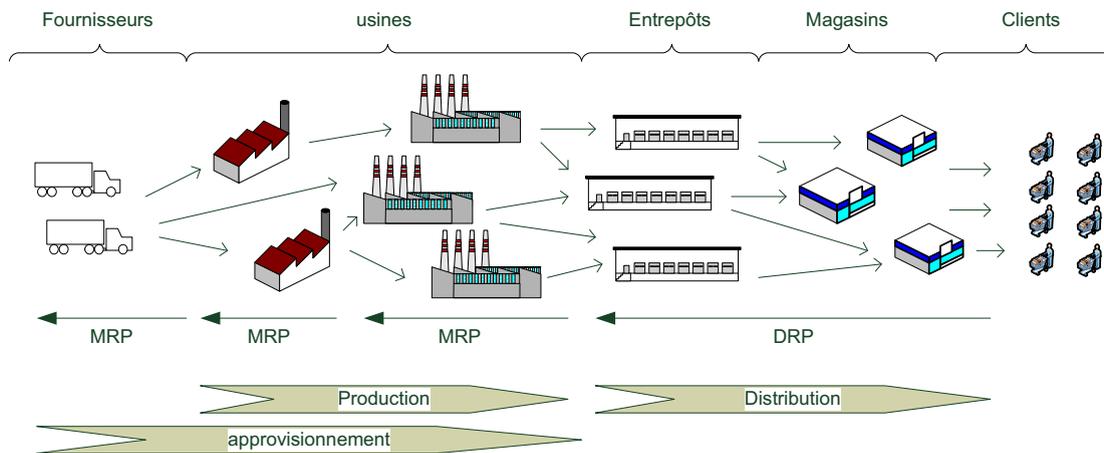


Figure 6.1 – l'approvisionnement, la production et la distribution dans une chaîne logistique

Pour contrer ces deux effets, une entreprise clermontoise a demandé au laboratoire du LIMOS d'étudier la possibilité de différencier les demandes clients pour tenter de dissocier les taux de satisfaction de chaque client et donc de dimensionner plus justement leurs stocks de sécurité. A l'issue de l'étude menée par (Comelli 2008), l'entreprise nous a demandé d'élaborer un outil logiciel permettant la mise en œuvre d'une telle politique afin d'élaborer un test de cette dernière dans un contexte industriel. Dans ce chapitre, nous commençons par présenter les différentes politiques de gestion des stocks rencontrées dans la littérature et comment s'y est intégrée la problématique de la demande différenciée. Nous exposons les résultats obtenus par (Comelli 2008). Dans une seconde partie, nous montrons quels scénarii ont été élaborés pour mettre en œuvre une politique de gestion des stocks à demande différenciée dans le contexte industriel de l'entreprise et, pour finir, nous donnons les algorithmes implémentés au sein de l'outil logiciel proposée.

I Le contexte de l'étude

Dans cette section, nous présentons les principales politiques de gestion des stocks, ensuite nous donnons les principales façons dont la demande est prise en compte (différenciée ou non). Ensuite, nous présentons les principales politiques de gestion des stocks permettant de prendre en compte la demande différenciée et nous finissons en exposant les travaux réalisés par (Comelli 2008)

1 Les principales politiques de gestion des stocks

(Giard 2003) présente et analyse les principales politiques de stock en distinguant deux modes de gestion possibles :

- le mode de gestion à révision périodique : la décision de lancer une commande est effectuée périodiquement (à intervalles de temps réguliers) en fonction du niveau des stocks. La révision périodique permet l'optimisation de commandes groupées et convient parfaitement aux cas où la demande est saisonnière.
- le mode de gestion à révision continue : les stocks sont gérés en temps réel ou à chaque mouvement de stock (entrée ou sortie), la décision du lancement de commande est alors effectuée lorsque le niveau des stocks est en dessous d'un certain seuil. Ce mode de gestion est en général moins coûteux que celui à révision périodique si on considère des coûts de commande identiques.

Pour chacun des modes de gestion, il identifie trois façons de calculer les quantités à commander :

- en quantités fixes (dont le calcul est basé le plus souvent sur la quantité économique donnée par les formules de Wilson),
- permettant de remonter à un certain niveau de stock,
- variables et quelconques.

Les principales politiques de gestion de stock rencontrées selon le mode de gestion considéré sont ensuite listées :

1. Révision périodique (réapprovisionnement à date fixe toutes les R unités de temps) :
 - la *politique* (R, Q) : le réapprovisionnement se fait à quantité fixe (Q) ,
 - la *politique* (R, S) : la quantité commandée varie afin d'atteindre un niveau de stock S . Cette politique de stock porte le nom de « Periodic Review Inventory Policy (R, S) » ,
 - la *politique* (R, s, S) : une quantité est commandée si et seulement si le niveau est en dessous d'un certain seuil s et varie afin d'atteindre un niveau de stock S .
2. Révision continue (les réapprovisionnements ont lieu à date variable et sont fonction d'un certain niveau de stock) :
 - la *politique* (s, Q) : une quantité fixe est commandée dès que le niveau est en dessous d'un niveau s ,
 - la *politique* (s, S) : une quantité est commandée dès que le niveau est en dessous d'un certain niveau s et varie afin d'atteindre un niveau de stock S . Cette politique porte le nom de « Order up to level (s, S) » ,
 - la *politique* $(S - 1, S)$: un ordre de réapprovisionnement est envoyé à chaque fois qu'une unité de stock est consommée.

La quasi totalité de la littérature portant sur la gestion des stocks vise à proposer des politiques de gestion optimales en prenant en compte diverses hypothèses telles que la stochasticité de la demande, le lead time de production ou encore les délais de livraison etc.

Bien qu'ils fournissent des solutions satisfaisantes, ces modèles s'appuient sur la consommation effective des stocks, c'est-à-dire qu'ils ne prennent pas en compte les prévisions de demande : ils sont dits « réactifs » contrairement aux systèmes « proactifs » qui, eux, s'appuient sur les prévisions de demande, donc de consommation et qui vont permettre d'anticiper les demandes de réapprovisionnement. Parmi les systèmes proactifs, on retrouve le D.R.P. (Distribution Resource Planning) (Martin 1990). Comme

nous l'avons vu dans le chapitre I, l'avantage de celui-ci est de prévoir la consommation de stock, et donc d'anticiper les demandes de réapprovisionnement.

2 La gestion des stocks à demande différenciée

La gestion des stocks à demande différenciée vise à assurer à n clients différents n niveaux de service différents. (Tempelmeier 2006) remarque le peu de recherches menées sur ce concept de différenciation et énumère les trois modes de gestion de stocks rencontrés :

- Le « Round up Stock Policy » : chaque stock est mutualisé et permet de servir n'importe quel client sans restriction. Le stock de sécurité est dimensionné en fonction du taux de satisfaction client le plus élevé. C'est la politique la plus couramment employée.
- Le « Separate Stock Policy » : à chaque client est affecté un stock. Son stock de sécurité est donc calculé en fonction du taux de satisfaction du client auquel il est associé. La contrepartie d'une telle politique est de provoquer une inflation des stocks due à la multiplication des stocks de sécurité.
- Le « Rationing Stock Policy » : à l'instar du « Round up Stock Policy », chaque stock est mutualisé et permet de servir n'importe quel client. Cependant, des restrictions sont mises quant à la façon de satisfaire la demande de ce dernier. En effet, à chaque client k est associé un niveau de stock S_k (appelé seuil) en dessous duquel il n'est plus servi (on le « rationne »). Par conséquent, dans cette politique peuvent cohabiter niveau de stock positif et présence de backorder. Le niveau du seuil de chaque client influe directement sur son taux de satisfaction.

Seuls les deux derniers modes de gestion répondent à la problématique des stocks à demande différenciée. Cependant, le « Separate Stock Policy » ne revêt pas le caractère innovant du « Rationing Stock Policy » : en effet, une fois un stock attribué à chaque client, il suffit d'appliquer les techniques du « Round up stock policy » à chacun d'eux. La recherche s'est donc orientée vers l'application du « Rationing Stock Policy » dans le cadre des politiques de gestion définies précédemment : le lecteur pourra trouver un état de l'art sur cette problématique dans (Koçaga et Sen 2007) et (Comelli 2008). Cependant, (Comelli 2008) remarque que la littérature s'est plus axée sur la preuve de l'optimalité de ce mode de gestion que sur la détermination des paramètres utiles à son application. L'objet de ce chapitre est de montrer comment implanter cette politique de gestion des stocks sur un cas industriel et de fournir des algorithmes permettant de déterminer les paramètres nécessaires à sa mise en œuvre.

3 Les travaux du LIMOS

Dans un but d'économie, l'objectif de l'entreprise clermontoise est d'abaisser son niveau de stock à l'orée 2012, tout en maintenant des niveaux de service tout à fait concurrentiels. Pour atteindre ce but, un des axes dégagés est de redéfinir son mode de gestion des stocks, sa politique actuelle s'apparentant à du « Round up Stock Policy ».

L'étude confiée à (Comelli 2008) consistait initialement à évaluer les gains potentiels dégagés par l'application d'une politique de « Separate Stock Policy » dans le contexte industriel de cette entreprise. Cependant, ce dernier a rapidement fait émergé l'existence de la politique à seuil de rationnement et ses résultats prometteurs. Ainsi, l'objectif de son étude a été redéfini puisqu'il s'agissait alors de comparer ces 3 modes de gestion. Après une recherche approfondie des divers travaux relatifs à cette problématique, il s'est rendu compte d'une part que peu de recherches s'étaient orientés vers la comparaison de ces politiques et d'autre part que l'impact de celles-ci n'avaient jamais été mesuré sur un système proactif de type D.R.P., utilisé par l'entreprise initiatrice de ces travaux. S'appuyant sur les travaux de (Deshpande et al. 2003), il a mené une analyse très fine et une comparaison détaillée des trois politiques de gestion dans un contexte D.R.P., (Deshpande et al. 2003) ayant effectué des travaux similaires pour un système de gestion de stock (s, Q) .

Pour mener son analyse, (Comelli 2008) considère le réseau de distribution suivant :

- 2 usines,
- 1 produit,
- 1 magasin physique,
- 2 types de clients : l'un privilégié, l'autre non.

La figure 6.2 représente le fonctionnement de ce réseau.

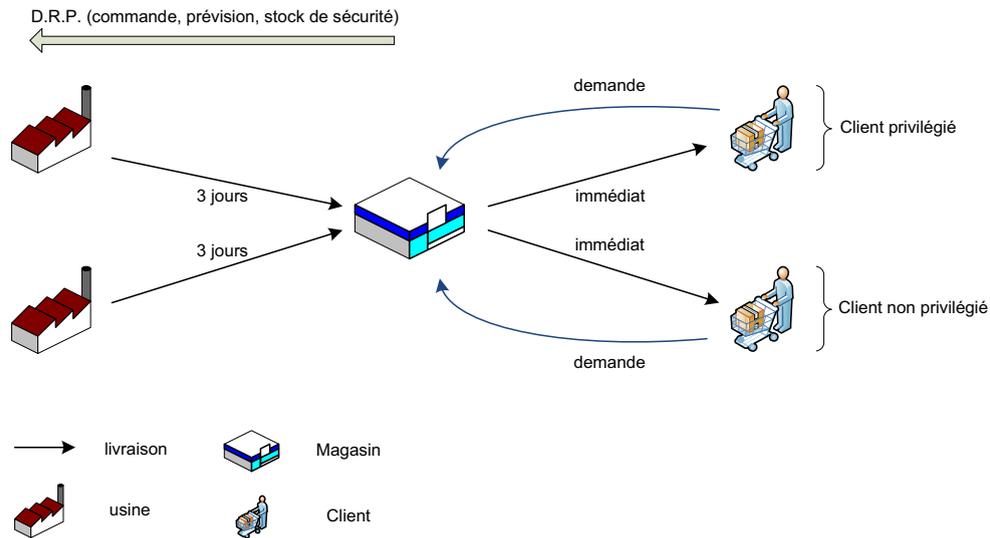


Figure 6.2 – Cas d'étude

Les hypothèses de fonctionnement sont les suivantes :

- lorsqu'un client passe une commande, il souhaite être livré le jour même,
- la commande du client est transmise au magasin qui la satisfait selon son niveau de stock et sa politique de gestion : la demande non satisfaite passe en backorder,
- quotidiennement, le calcul du DRP est effectué : les ordres d'approvisionnement sont calculés en fonction du niveau de backorder du magasin, de son stock de sécurité, des prévisions de vente et des commandes fermes non encore prises en compte en considérant des règles de lotissement d'une durée de plusieurs jours (le « custorder dur »),
- les ordres d'approvisionnement sont transmis au système de production selon une table d'affectation qui répartit les quantités demandées entre les diverses usines,
- les usines répondent à la demande émise par le magasin en tenant compte d'un lead time de trois jours correspondant au délai de fabrication et de massification (constitution des stocks de transport),
- les quantités livrées par les usines au magasin sont mises en stock ou permettent de solder les backorders selon des règles de gestion pré-établies (soldes des backorders les plus anciens, des clients prioritaires en premier etc.).

Pour mener à bien son étude, (Comelli 2008) simule le fonctionnement du système en utilisant le logiciel Siman Arena en y intégrant les algorithmes régissant le calcul du D.R.P. Grâce à cette technique, il compare les trois politiques considérées en faisant varier plusieurs paramètres tels que le taux de commandes prioritaires et l'erreur liée aux prévisions afin de tester la sensibilité des résultats obtenus. Pour déterminer le seuil de rationnement correspondant au taux de satisfaction désiré, plusieurs valeurs sont déterminées selon le principe donné par (Tempelmeier 2006). La figure 6.3 illustre les résultats obtenus.

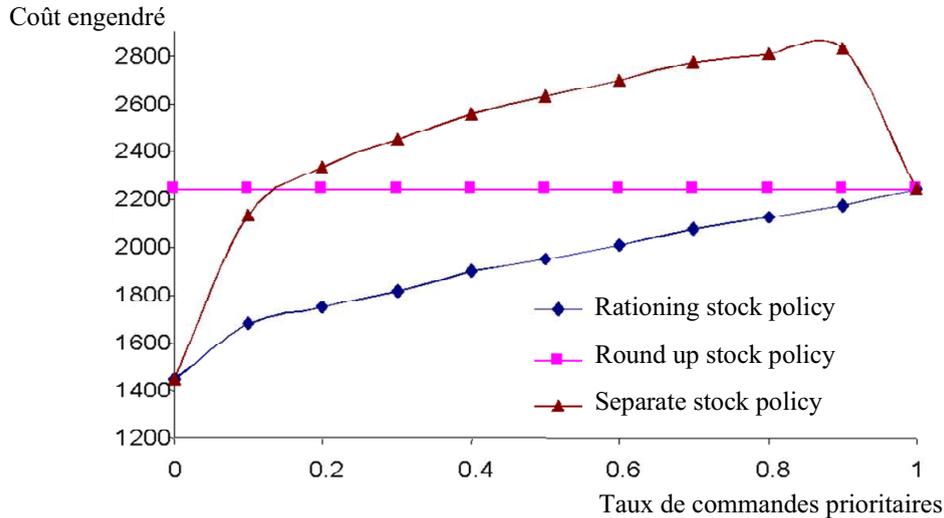


Figure 6.3 – Résultats obtenus

(Comelli 2008) remarque que ses résultats sont analogues à ceux de (Deshpande et al. 2003) : la politique « Separate Stock Policy » surpasse la politique du « Round up Stock Policy » lorsque le taux de demande émise par le client prioritaire est inférieur à 15 %. Cependant, la politique du seuil de rationnement est la moins coûteuse quel que soit ce taux.

Devant les résultats concluants de cette étude, l'entreprise a décidé de tester la politique du seuil de rationnement en situation réelle : un projet a donc été lancé pour mettre en œuvre un pilote au sein de l'entreprise.

II Le projet

Le but du projet est de vérifier la viabilité d'un mode de gestion à seuil de rationnement grâce à un test en grandeur nature. Plus précisément, son objectif est :

- de vérifier que l'utilisation d'une telle politique est suffisamment simple pour pouvoir être gérée opérationnellement,
- d'identifier les limites d'un tel concept,
- de définir comment calculer les seuils de rationnement et comment les exploiter opérationnellement.

La mise en place d'une telle politique des stocks n'est pas chose aisée car les répercussions qu'elle engendre s'étendent du système d'information de l'entreprise jusqu'à la relation que cette dernière entretient avec ses clients finaux. C'est pour cela que le projet a été découpé en trois étapes :

étape 1 : étude de faisabilité et de condition du test,

étape 2 : conception de l'application informatique permettant le calcul des seuils,

étape 3 : développement et mise en œuvre du test en grandeur nature et analyse des résultats obtenus.

A l'heure de la rédaction de ce manuscrit, seules les étapes 1 et 2 ont été abordées, le test grandeur nature devrait avoir lieu début 2009.

1 Etude de faisabilité et de condition du test

Cette étape essentielle permet d'appréhender la complexité effective de la mise en œuvre d'un tel test et d'en préciser le cadre expérimental.

Pour l'entreprise, trois points cruciaux sont à prendre en compte :

- le fonctionnement du service commercial et notamment de la prise de commande doit subir le moins d'impact possible afin de ne pas dégrader ses performances,
- les solutions envisagées informatiquement ne doivent pas engendrer de coûts prohibitifs, c'est-à-dire d'évolutions majeures des outils informatiques utilisés pour la prise de commande et la gestion des stocks,
- le test doit être significatif : le choix du ou des magasins où sera implantée la politique de seuil de rationnement, du ou des clients prioritaires doivent avoir une signification en terme business. Afin de simplifier la mise en place du test grandeur nature, il nous est demandé de ne considérer que deux groupes de clients : l'un est composé des clients prioritaires et le second des clients ne l'étant pas. Les taux de satisfaction du groupe des clients prioritaires (resp. non prioritaire) est identique pour chaque client le composant. Cette nouvelle hypothèse de travail permet de nous ramener au cas où nous n'avons que deux clients : l'un prioritaire, l'autre non. Dans la suite nous appellerons A le client prioritaire et B le second. Il est à noter que, dans ce cas, le seuil de rationnement du client A est égale à zéro.

Au travers de ces trois points apparaît clairement la diversité et la transversalité des problèmes à résoudre. Cela implique l'intégration au sein de l'équipe projet d'un nombre important d'acteurs venant d'horizons différents et dont les intérêts, sans être antagonistes, peuvent parfois être divergents. Ainsi, l'équipe projet est composé :

- de l'initiateur du projet en la personne du directeur Supply Chain,
- d'un chef de projet faisant partie de l'équipe Supply Chain,
- d'experts business,
- d'experts du système d'information,
- de représentants de la partie commerciale axée vers la prise de commande,
- des intervenants du laboratoire LIMOS.

Grâce à son expertise lié à son domaine d'activités, chaque participant veille à ce que tous les aspects liés au test d'une telle politique soient bien pris en compte.

1.1 Interactions entre les outils de prise de commande et de gestion des stocks

Pour bien comprendre où va se situer la problématique d'intégration du seuil de rationnement, nous commençons par expliquer brièvement comment sont modélisés les stocks dans l'outil de prise de commande et l'outil de gestion des stocks et quels sont les interactions entre ces deux outils. La figure 6.4 illustre cette modélisation.

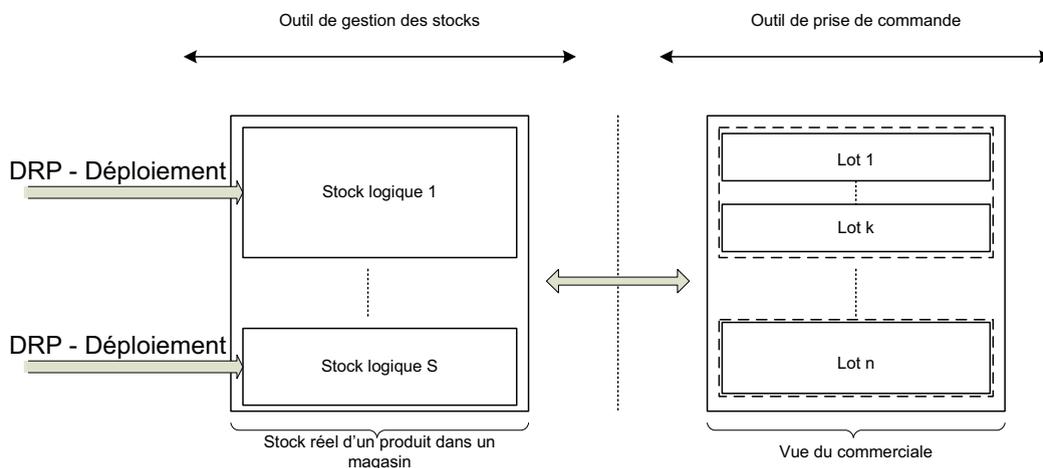


Figure 6.4 – Représentation des stocks dans les outils de gestion des stocks et de prise de commande

L'outil de gestion des stocks permet de gérer les réapprovisionnements du stock physique d'un magasin. Le stock physique de chaque article du magasin peut être scindé en plusieurs stocks logiques : dans ce cas, chaque stock logique aura sa propre politique de réapprovisionnement. Plus précisément,

chaque stock logique aura son propre stock de sécurité et lancera des ordres d'approvisionnement via le DRP indépendamment des autres stocks logiques.

Du côté de la prise de commande, les stocks logiques apparaissent comme autant de stocks indépendants. Chacun d'entre eux peut être scindés en plusieurs parties, appelés «lots» qui peuvent être attribués à un ou plusieurs clients particuliers.

Ce découplage entre l'outil de gestion des stocks et l'outil de prise de commande permet une gestion des stocks particulièrement souple puisqu'il permet, entre autres, de transférer des quantités d'un lot particulier à un autre au sein d'un même stock logique sans avoir à enclencher de demande de réapprovisionnement. On voit donc que nous avons deux façons d'intégrer la notion de seuil de rationnement : soit au niveau de l'outil de gestion des stocks, soit au niveau de l'outil de prise de commande.

Pour simplifier l'exposé, nous utilisons la convention suivante : un stock physique contient au moins un stock logique (qui peut être le stock physique lui-même) et du point de vue de la prise de commande, à un stock logique correspond toujours au moins un lot (qui peut être le stock logique lui-même).

Le seuil de rationnement étant un concept rattaché à la gestion des clients, c'est surtout au niveau de l'outil de prise de commande qu'il doit être intégré. De plus, une politique de priorisation par seuil ne peut s'effectuer qu'au sein d'un même lot puisque le rationnement s'effectue en fonction des quantités disponibles dans celui-ci : nous supposons dans un premier temps que pour un article donné, le magasin ne possède qu'un stock logique et que du côté de la prise de commande, il n'y a qu'un seul lot affecté à ce stock logique.

1.2 Le processus de prise de commande

Pour bien comprendre les implications de la politique par seuil, nous décrivons le processus de prise de commande. Cependant, en préliminaire à cette description, il faut savoir que chaque client est affecté à un lot d'un magasin particulier : cette affectation dépend de sa localisation et du type d'article qu'il souhaite.

Le processus se déroule selon le script suivant :

Phase 1 : Afin de passer une commande, le client téléphone au service de prise de commande qui est constitué d'une plateforme téléphonique découpée en zones selon le type de client. Ce dernier est donc automatiquement aiguillé vers la zone qui lui est attribuée.

Phase 2 : Le commercial qui réceptionne l'appel du client enregistre sa commande et, grâce à un logiciel de prise de commande, consulte l'état du lot qui lui est affecté.

Phase 3 : Si la quantité disponible dans le lot suffit à couvrir la demande du client, la commande est validée sinon le commercial consulte les quantités en transit jusqu'au stock logique correspondant (les « cours de route ») et les quantités en cours de fabrication (les « encours ») destinées à ce même stock logique, afin de donner une date prévisionnelle de livraison au client pour les quantités manquantes. Au cas où celles-ci sont couvertes par le cours de route ou l'encours et où le client est satisfait des dates prévisionnelles de livraison, la commande est validée : les quantités manquantes passent en backorder et le commercial a la possibilité de les réserver sur l'encours et le cours de route.

Phase 4 : Si les encours ou les cours de route ne suffisent pas à couvrir les quantités manquantes, le client en est informé et peut ou non maintenir sa commande. S'il décide de la maintenir, ces dernières sont passées en backorder mais aucune date de livraison ne lui est donnée.

Bien qu'un peu schématique, le déroulement de ce script permet de voir où va intervenir la notion de seuil dans le processus de prise de commande. C'est au niveau de la phase 3 que vont se situer les difficultés de mise en œuvre de la politique de rationnement et ceci pour deux raisons :

- la première réside dans le fait que le commercial ne doit considérer que la quantité présente dans le lot au dessus du seuil de rationnement du client pour satisfaire la demande de celui-ci,
- la seconde est que dans une politique de seuil de rationnement, les quantités en cours de route

et en encours ne sont pas prises en compte pour satisfaire la demande client.

Bien que d'apparence anodine, les conséquences de ces deux points sont importantes. En effet, ne pas considérer les encours et le cours de route entraîne une baisse significative du niveau d'informations délivrées au client ce qui n'est pas acceptable du point de vue de l'entreprise : en effet, pouvoir réserver du cours de route et de l'encours permet, dans la plupart des cas, de donner une date précise de livraison au client. Cependant, les prendre en compte peut entraîner une réservation de la quantité pour un client non prioritaire au détriment d'un client prioritaire.

Prenons pour exemple deux clients A et B. Le client A est prioritaire et a un seuil de rationnement à 0 et le client B est non prioritaire et a un seuil de rationnement de 50.

La figure 6.5 illustre le cas où l'encours et le cours de route sont pris en compte au niveau de la prise de commande.

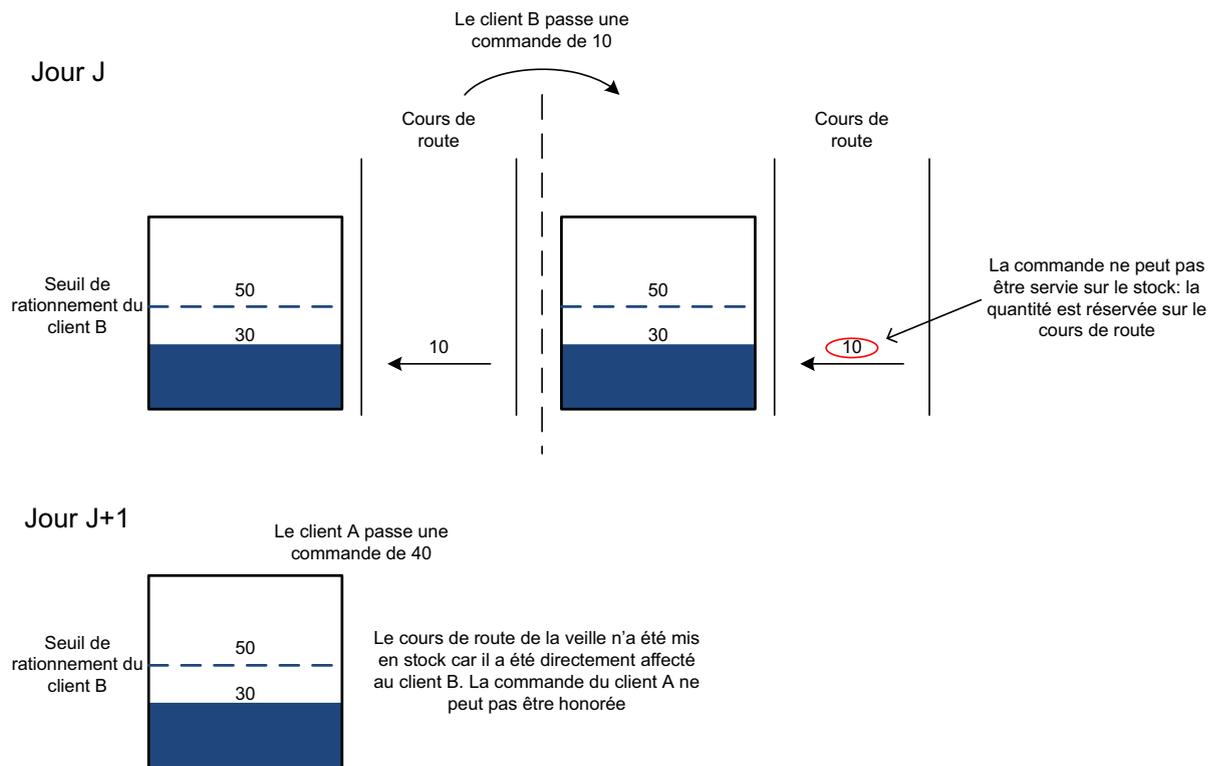


Figure 6.5 – Cas d'une politique de rationnement par seuil avec prise en compte du cours de route

Nous supposons qu'au jour J le niveau du lot est à 30. Le client B passe une commande de 10. Le niveau du lot étant inférieur à son seuil de rationnement, le commercial ne peut lui servir sa commande sur le lot : il consulte donc le cours de route et constate qu'une quantité de 10 arrivera le lendemain. Il réserve donc cette quantité pour le client B. Au jour J+1, le client B consomme donc le cours de route qui lui était réservé : le niveau du lot reste donc inchangé. Le client A passe une commande de 40 qui ne peut donc pas être intégralement satisfaite.

La figure 6.6 illustre le cas où l'encours et le cours de route ne sont pas pris en compte au niveau de la prise de commande.

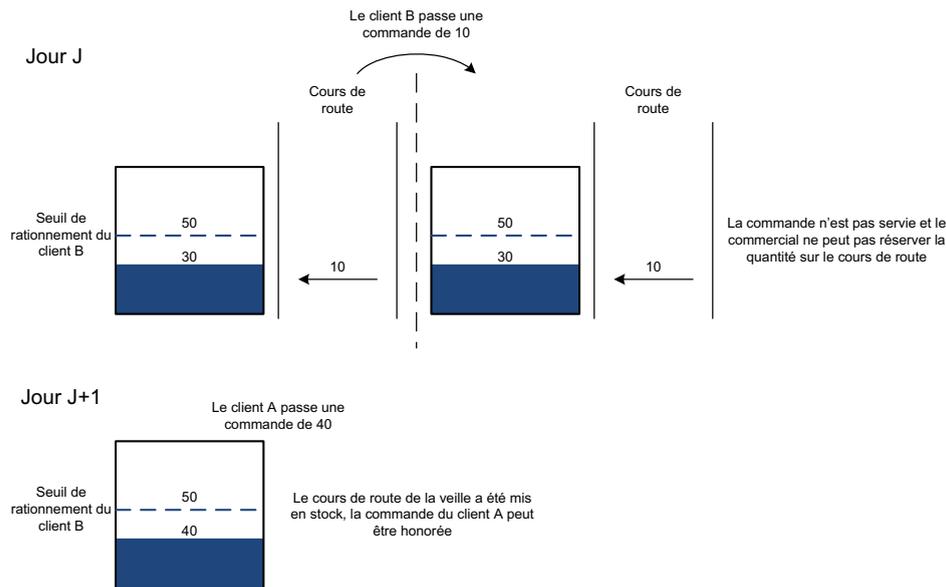


Figure 6.6 – Cas d'une politique de rationnement par seuil sans prise en compte du cours de route

Dans ce cas là, comme la quantité en cours de route n'a pas pu être réservée par le commercial, le lot remonte à un niveau de 40 au jour J+1, ce qui permet de répondre à la commande du client A.

On voit donc l'impact sur le taux de satisfaction du client A de la prise en compte des encours et du cours de route et donc le biais introduit dans la politique de seuil de rationnement.

Cette possibilité de réservation de quantités sur les encours et les cours de route empêche donc une stricte application d'une politique de rationnement. La solution retenue devra donc tenir compte des impératifs commerciaux de l'entreprise en termes de communication et de visibilité pour le client final. Il apparaît déjà que la politique de seuil de rationnement devra être aménagée pour tenir compte de ces impératifs.

Cependant, la phase 3 du script pose un autre problème : comment s'assurer que le commercial ne délivre au client que la quantité à laquelle il a droit. Reformuler différemment, comment renseigner le commercial sur le seuil de rationnement auquel son client est astreint ?

Deux solutions sont envisageables :

- Soit le commercial possède à ses côtés la liste des clients et leur seuil de rationnement et s'y réfère systématiquement à chaque prise de commande. Cette solution apparaît rapidement comme irréaliste pour plusieurs raisons :
 - . comme le font remarquer les représentants du service commercial, l'utilisation d'une telle liste n'est pas pratique en fonctionnement courant : outre les erreurs pouvant être générées par leur utilisation, le temps perdu pour rechercher les bonnes informations serait vraiment pénalisant pour les commerciaux.
 - . les seuils de rationnement pouvant varier au fil du temps, il faut veiller à l'actualisation de cette liste au niveau de chaque commercial
- Soit le système informatique lui fournit le seuil en fonction du client, voire lui indique la quantité maximum qu'il peut allouer à sa commande. Celle-ci implique le renseignement du seuil de rationnement directement au niveau de l'outil de prise de commande pour chaque client et chaque article. Celui-ci peut se faire de deux façons :
 - . soit en automatique : l'application informatique de calcul de seuil est directement interfacée avec le système d'information de l'entreprise et les logiciels de prise de commande et de gestion des stocks. Cependant, les experts du système d'information pointent les coûts très élevés en termes de développement d'une telle solution ce qui n'est pas envisageable dans le cadre d'un simple test. De plus, les outils de gestion de stocks et de prise de commande étant promis à une évolution à court terme, l'opportunité d'un tel développement n'est pas évidente.
 - . soit en manuel : dans ce cas là, les seuils sont renseignés par un opérateur à une fréquence restant à déterminer. L'équipe projet propose donc la création d'abaques de seuils pour faciliter

ce travail.

Au bilan, l'examen du processus de prise de commande permet de tirer les enseignements suivants :

- la politique de rationnement par seuil ne peut pas être implantée sans aménagement,
- l'application de calcul de seuil devra fournir des abaques de seuils qui seront exploitées par la personne chargée de renseigner le système d'information durant le test.

1.3 Intégration d'une politique de seuil au sein de la prise de commande

Pour mettre en œuvre une politique par seuil de rationnement, il faut pouvoir déterminer la meilleure manière d'intégrer la notion de seuil au sein de l'outil informatique dédié à la gestion des prises de commande.

Malgré sa performance, l'outil informatique utilisé n'intègre pas le concept de seuil de rationnement. La seule possibilité de gérer la priorisation client en évitant le recours à un développement informatique lourd et coûteux, est d'utiliser la politique du « Round up Stock Policy » ou du « Separate Stock Policy » (c'est à dire d'utiliser les possibilités qu'offrent la notion de lot) : c'est donc bien le concept même de seuil qui n'est pas implémenté dans ce logiciel. L'évolution de ce logiciel pour prendre en compte celui-ci étant inconcevable en termes de coût, le challenge à relever est donc de modéliser au mieux le comportement d'une politique par seuil en utilisant les possibilités de cet outil.

Nous rappelons que dans un premier temps, pour un article donné, le magasin ne possède qu'un stock logique et que du côté de la prise de commande, il n'y a qu'un seul lot affecté à celui-ci. La figure 6.7 illustre cette hypothèse de départ.

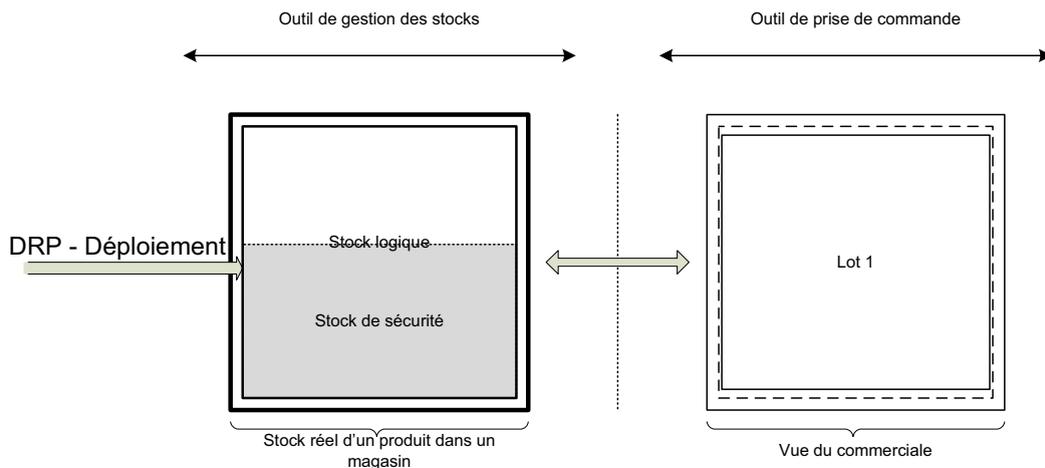


Figure 6.7 – Situation initiale

Au cours de réunions de travail, l'équipe projet a identifié cinq scénarii de fonctionnement possible pour approcher au mieux la politique de rationnement.

1.3.1 Scénario 1 : « la comparaison en temps réel »

Dans ce scénario, les seuils sont directement intégrés dans le logiciel de prise de commande. A chaque commande d'un client, le niveau du lot est comparé à son seuil de rationnement et renvoie la quantité maximale allouable à la commande. Ce scénario correspond à une vraie politique par seuil mais nécessite, selon les experts du système d'information, un évolution majeure (et coûteuse) du logiciel de gestion des prises de commande. Ce scénario n'est donc pas envisageable.

1.3.2 Scénario 2 : « le message d'alerte »

A l'instar du scénario précédent, les seuils sont également intégrés dans le logiciel de prise de commande et à chaque commande d'un client, le niveau du lot est comparé au seuil de rationnement qui lui est associé. Si la commande client excède la quantité de produit allouable au client, alors un

message alerte le commercial afin qu'il gère la commande en conséquence. L'idée de ce scénario est donc de minimiser le développement informatique liée à l'intégration d'une vraie politique de seuil de rationnement. Cependant, les experts du système d'information font remarquer que c'est la comparaison en temps réel qui est délicate à intégrer et que, même pour ce scénario, les développements à envisager seront coûteux. Ce scénario n'est donc pas envisageable.

Les deux premiers scénarii ont fait émerger l'impossibilité de comparer le niveau du lot aux seuils de rationnement de chaque client. Pour contourner cette difficulté, l'idée a été d'essayer d'approcher la politique de rationnement par seuil en la modélisant grâce à une politique basée sur du « Separate Stock Policy » : en effet, le logiciel de prise de commande permet la création de lot dédié à un client.

Prenons l'exemple des deux clients A et B. L'idée est de partitionner le stock logique en deux lots, l'un dédié au client A de la taille du seuil de rationnement du client B et l'autre accessible aux deux clients. Ainsi, on ne pourra pas servir de commande du client B sur le lot dédié au client A, le cantonnant ainsi au lot commun à tous. La figure 6.8 illustre cette façon de procéder.

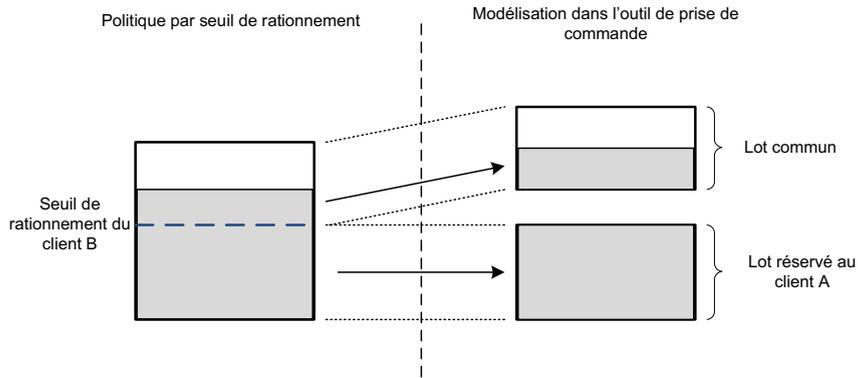


Figure 6.8 – Modélisation du seuil de rationnement dans l'outil de gestion des stocks

Bien évidemment, on voit au travers de cette figure que toute la difficulté réside dans la façon dont seront réparties les quantités en stock et comment seront affectées les commandes du client A. Les trois scénarii suivants utilisent donc cette façon de modéliser les stocks et définissent chacun une façon de répartir les quantités stockés et les commandes du client prioritaire.

La figure 6.9 illustre la nouvelle hypothèse de travail envisagée pour intégrer la notion de seuil de rationnement dans l'environnement de prise de commande.

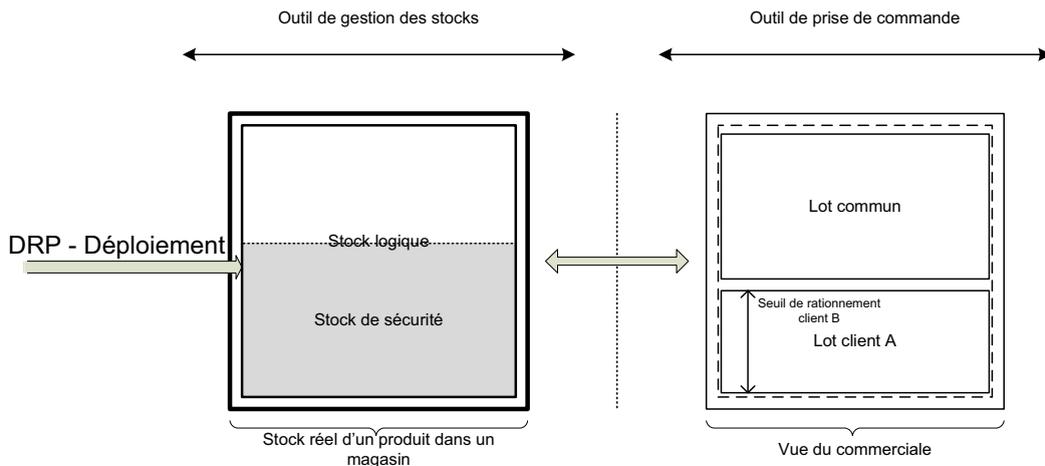


Figure 6.9 – Utilisation des lots dans l'outil de prise de commande

1.3.3 Scénario 3 : « Lot prioritaire en premier »

Dans ce scénario, le client prioritaire commence par être servi sur son lot dédié. Une fois que celui-ci est épuisé, alors ses commandes sont servies sur le stock commun. La figure 6.10 compare ce scénario à une réelle politique par seuil.

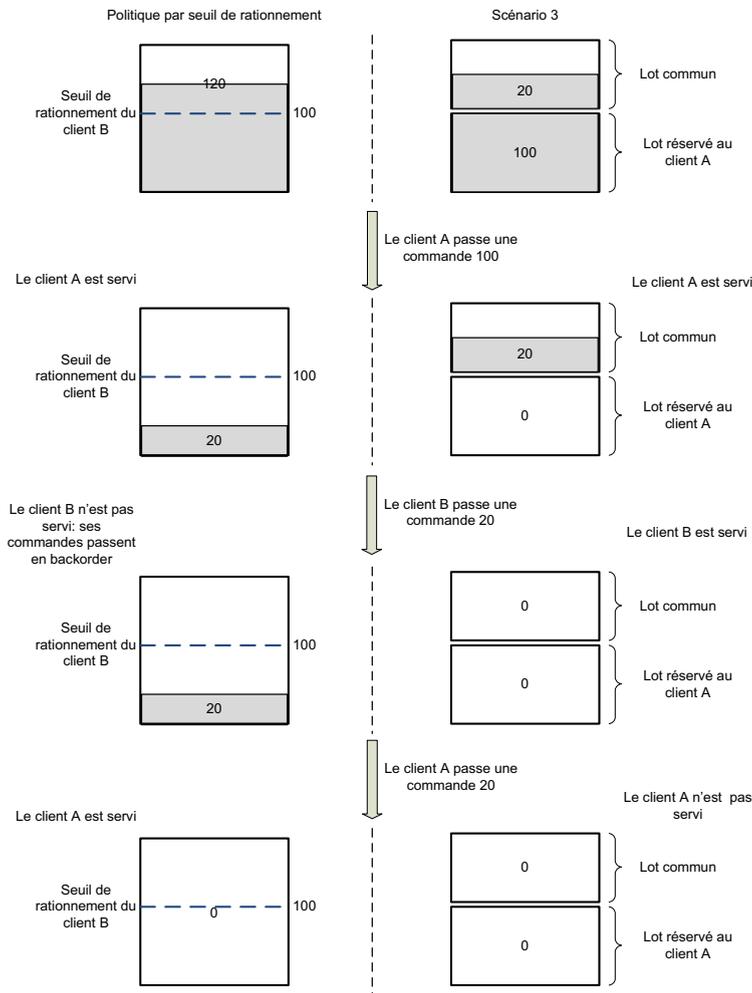


Figure 6.10 – Comparaison du scénario 3 avec une politique de rationnement par seuil

Grâce à cette illustration, on voit l'impact de cette configuration : en effet, dans une véritable politique par seuil, le client A aurait été servi lors de sa dernière commande. On a donc bien une conséquence sur son taux de satisfaction.

1.3.4 Scénario 4 : « Lot prioritaire en dernier »

Dans ce scénario, on prend le contrepied du précédent : le client prioritaire est servi sur le lot commun jusqu'à épuisement de celui-ci puis est servi sur son lot réservé. En reprenant l'exemple précédent, on constate que la politique par rationnement et le scénario 4 coïncide. Il est clair que cette configuration se rapproche bien plus de la politique de rationnement que le scénario 3 par son fonctionnement. Fort de ce constat, l'équipe projet décide de se pencher sur le problème de réapprovisionnement de ces deux lots : lors d'une arrivée de quantité en stock, comment répartir cette dernière entre les deux lots ?

Les experts du système d'information indiquent que réapprovisionner automatiquement le lot prioritaire à hauteur du seuil de rationnement du client non prioritaire n'est pas aisée. Il est donc préconisé que cette affectation soit effectuée manuellement. Deux solutions sont donc envisagées :

- soit cette affectation est faite dès qu'une quantité arrive en stock : dans ce cas, les arrivées doivent être surveillées en continu. Cela implique qu'une personne dédie une grande partie de son temps à cette tâche,

- soit cette affectation est faite quotidiennement : en début de journée, le commercial examine l'état du stock logique et réapprovisionne le lot prioritaire à hauteur du seuil.

Le groupe projet constate que la première solution est très lourde à mettre en œuvre et ne peut être généralisée ultérieurement si le test s'avérait concluant. Quoiqu'il en soit, les deux solutions envisagées ne permettent pas de régler le problème des encours et des cours de route que le commercial peut réserver pour un client non prioritaire. Mais, il apparaît clairement que c'est en se basant sur le scénario 4 que la politique par seuil de rationnement sera la mieux approchée.

Puisque c'est au niveau du réapprovisionnement des lots que se trouvent les difficultés de mise en œuvre du scénario 4, le groupe projet le fait évoluer afin de tenter d'automatiser celui-ci, ce qui donne lieu au scénario 5.

1.3.5 Scénario 5 : « Stock et Lot Prioritaires »

Un réapprovisionnement automatique ne pouvant s'effectuer au niveau du logiciel de prise de commande, l'idée a donc été de se tourner vers le logiciel de gestion des stocks. Comme nous l'avons expliqué précédemment, dans ce dernier chaque stock logique a sa propre politique de réapprovisionnement via l'outil DRP : dit autrement, chaque stock logique possède son propre mécanisme de réapprovisionnement automatique. C'est donc en utilisant cette possibilité que le scénario 4 a évolué : le stock logique est scindé en deux et à chacun des deux stocks logiques résultants est associé bijectivement l'un des deux lots. Par abus de langage, nous appellerons « Stock Logique Prioritaire » (resp. « Stock Logique Commun») celui associé au lot prioritaire (resp. commun).

La figure 6.11 illustre ce procédé.

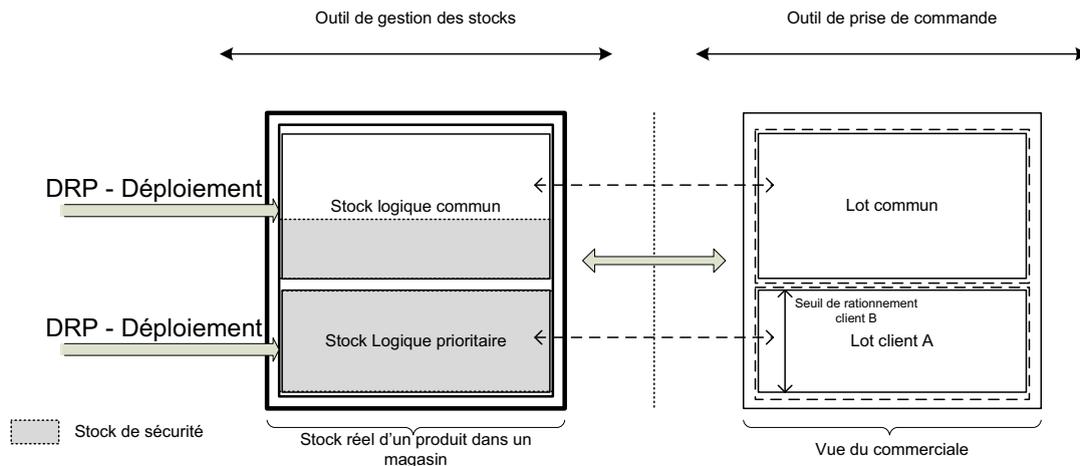


Figure 6.11 – Utilisation des stocks logiques dans l'outil de gestion des stocks

Le stock logique prioritaire est dimensionné à hauteur du seuil de rationnement du client B. Le stock de sécurité du stock logique initial est également partitionné en deux : il est affecté au stock logique prioritaire jusqu'à concurrence du seuil de rationnement et le reste est affecté au stock logique commun. (Tempelmeier 2006) remarque que pour qu'une politique de rationnement soit réellement efficace, il faut que le niveau de stock de sécurité soit supérieur au seuil de rationnement du client non prioritaire. Ainsi, le stock logique non prioritaire aura bien un stock de sécurité non nul.

Cette répartition du stock de sécurité n'est pas neutre : en effet, cela implique que dès qu'il y aura consommation d'une quantité du lot prioritaire, un ordre de réapprovisionnement sera déclenché afin de le remettre à son niveau initial. On voit bien qu'on a réussi par ce biais à automatiser le réapprovisionnement des deux lots. En corollaire de cette nouvelle configuration, nous avons apporté une solution au problème des encours et du cours de route. En effet, en créant deux stocks logiques nous avons créé un second flux d'approvisionnement dédié au stock logique prioritaire : sur celui-ci, le commercial ne pourra pas réserver de quantités sur les encours ou les cours de route pour un client non prioritaire car il n'aura pas la vue de ce flux au niveau du lot commun. Cependant, ce dernier conservera la main sur le flux de réapprovisionnement du stock logique non prioritaire, permettant

ainsi de donner une certaine visibilité sur les dates de livraison au client non prioritaire.

Actant ce résultat, le groupe projet a décidé la mise en œuvre du scénario 5 dans le cas du test grandeur nature et confirme la faisabilité d'un tel test, les seuls points restant encore à décider étant le choix du client prioritaire et le choix des articles sur lesquels portera le test.

2 Conception de l'application informatique permettant le calcul des seuils

Le but de l'application informatique est de calculer les seuils de rationnement pour chaque client. Conformément aux préconisations faites, nous nous sommes limité à deux clients pour développer celle-ci : si le test grandeur nature s'avère concluant et si l'entreprise décide de généraliser l'emploi de seuils de rationnement alors une nouvelle application sera développée, et la question de l'interfaçage avec le système d'information et de l'évolution des outils de prise de commandes et de gestion de stock sera soulevée.

Sous l'impulsion des experts business, il est décidé de reprendre la configuration du cas d'étude de (Comelli 2008). Cependant, une seule usine sera prise en compte : en effet, les leads time de production étant supposés constants et égaux pour chaque usine, on peut alors agréger les usines. La figure 6.12 représente donc la configuration sur laquelle s'appuie l'application informatique.

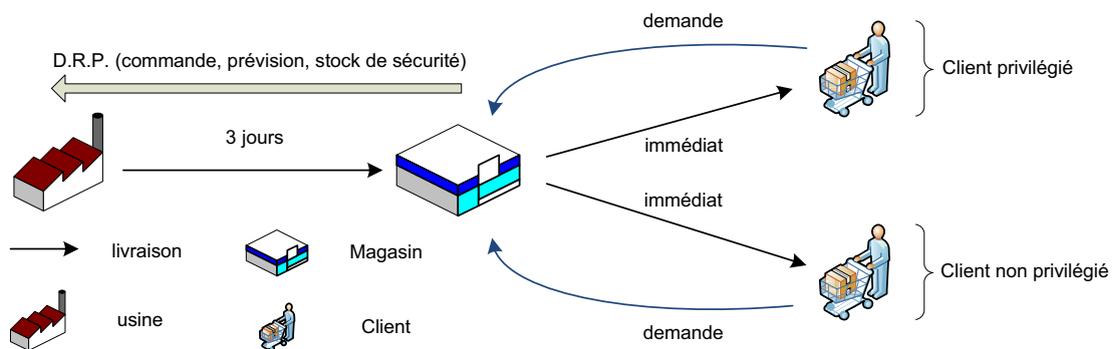


Figure 6.12 – Configuration du test

L'application à développer reprend le même principe que celle de (Comelli et al. 2008), cependant le modèle utilisée par ce dernier est remplacé par un algorithme reprenant le calcul du DRP. En effet, les hypothèses prises en compte (une seule usine, lead time constant) rendent l'utilisation d'un modèle de simulation inutile et de plus, le test portant sur plusieurs articles, l'abandon du modèle de simulation au profit d'un outil purement algorithmique permettra une réduction non négligeable des temps de calcul (et rendra inutile l'achat d'une licence Siman Arena).

2.1 Le modèle de données

la figure 6.13 donne le modèle de données de l'application sous forme d'un diagramme UML.

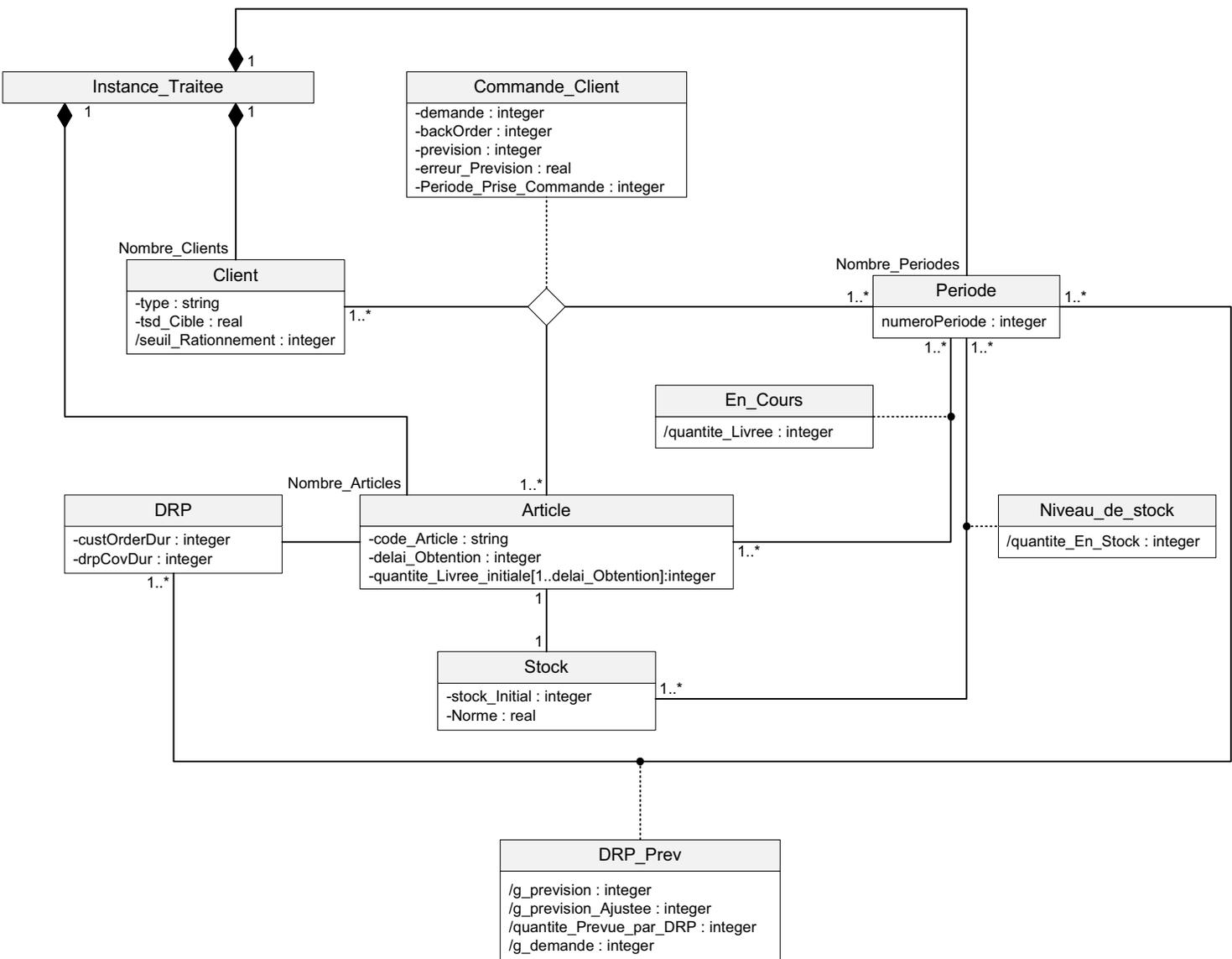


Figure 6.13 – Le modèle de données

Une instance est composée de clients, d'articles et d'un horizon discrétisé en un certain nombre de périodes.

Un client est caractérisé par son type (prioritaire ou non prioritaire), son taux de satisfaction voulu

ainsi que son seuil de rationnement. Quant à l'article, on retrouve son identifiant (code article), son délai d'obtention et les quantités livrées en début d'horizon (afin d'initialiser le calcul du DRP). Les périodes sont seulement caractérisées par leur numéro.

Un client peut passer une commande pour chaque article, à livrer pour une période donnée. La classe « Commande_client » modélise celle-ci. On y retrouve la demande du client (qui peut être soit donnée par un historique, soit générée par la prévision et l'erreur de prévision), le backorder éventuel sur cette commande et la période à laquelle a été émise la commande (cela permet de respecter la dynamique du DRP en ne faisant apparaître les commandes clients qu'à la période où elles ont été effectivement passées).

A un article est associé un unique stock logique et un DRP unique. Un stock est caractérisé par son niveau en début d'horizon (le stock initial) ainsi que son niveau de stock de sécurité (la norme). Le niveau du stock est calculé pour chaque période. Pour le DRP, on retrouve ses règles de lotissement (CustOrderDur qui correspond à la fenêtre temps durant laquelle les commandes clients consomment les prévisions, et le DrpCovDur qui indique le nombre de périodes sur lesquelles les commandes sont regroupées). Pour fonctionner, le DRP utilise pour chaque période les prévisions de commandes, les prévisions ajustées (c'est-à-dire tenant compte des commandes fermes déjà reçues), la demande connue (les commandes fermes) et il en déduit les quantités à commander à l'usine.

2.2 Le schéma fonctionnelle de l'application

La figure 6.14 illustre le fonctionnement l'application.

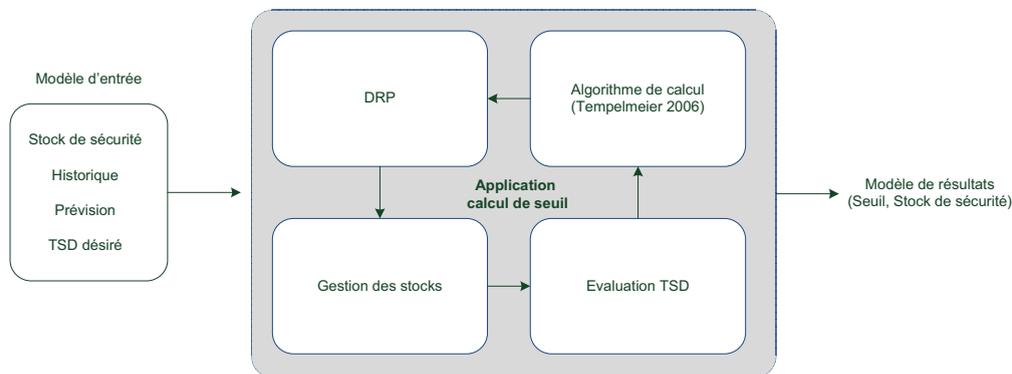


Figure 6.14 – Schéma de fonctionnement de l'application

Plus précisément, l'application fonctionne selon le principe décrit dans la figure 6.15.

L'application commence par initialiser le seuil de rationnement du client B à la moitié du stock de sécurité. Ensuite, l'algorithme simulant le DRP est lancé : les commandes des clients sont générées soit selon un historique des ventes soit aléatoirement en utilisant les prévisions de vente et les erreurs liées à ces mêmes prévisions. L'algorithme de gestion de stock calcule les niveaux de stocks, de demande servie et de backorder pour chaque client, en tenant compte de son seuil de rationnement. A l'issue de cet algorithme, les taux de satisfaction sont calculés. Au bout d'un certain nombre de répliques, les taux de satisfaction moyens sont calculés et s'ils ne correspondent pas à ceux attendus, l'algorithme de (Tempelmeier 2006) est utilisé pour calculer un nouveau seuil de rationnement pour le client B ainsi qu'un nouveau stock de sécurité.

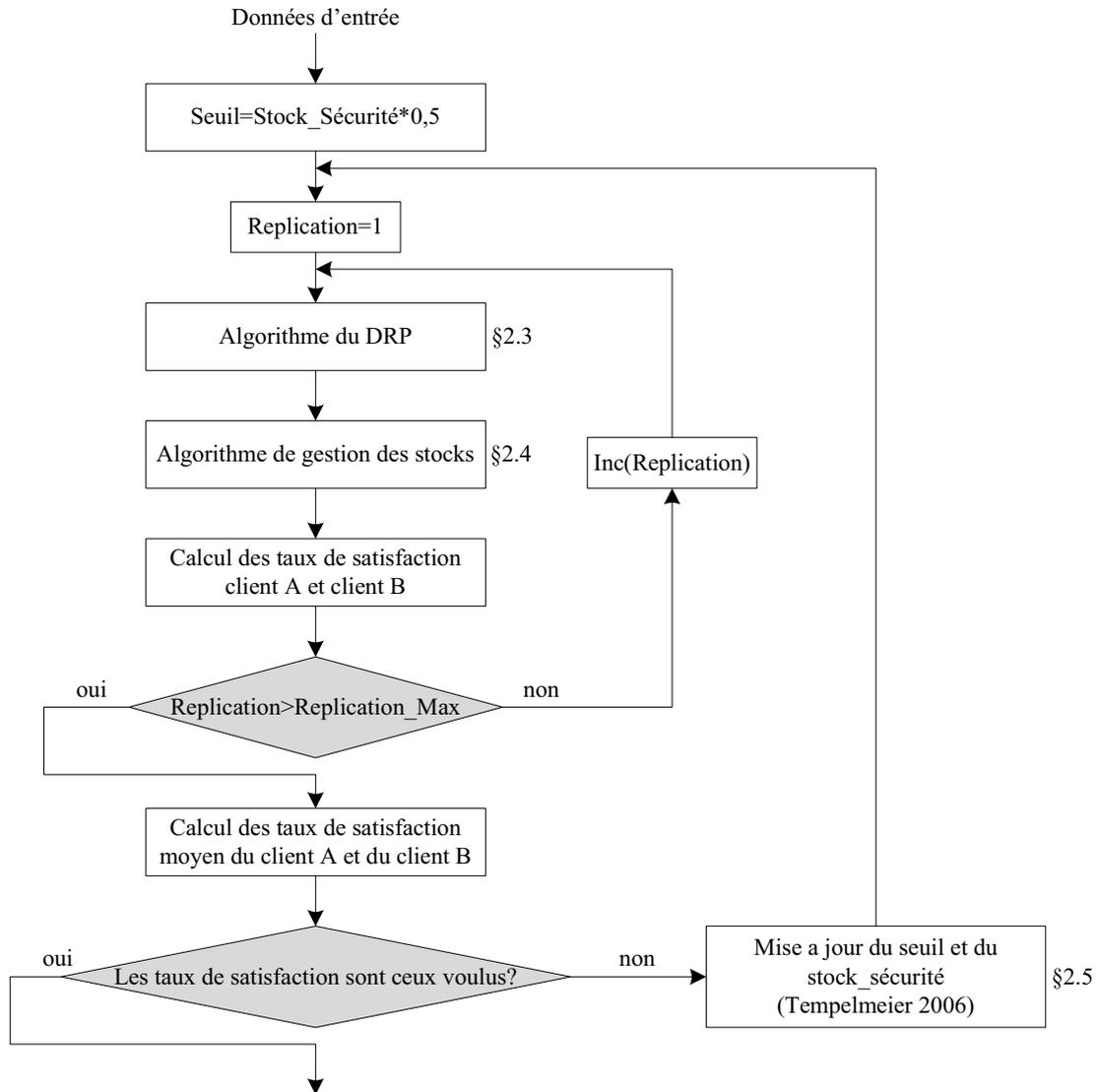


Figure 6.15 – Enchaînement des algorithmes

2.3 L'algorithme du DRP

Le calcul de la commande générée par le DRP pour un article à la période t est effectué en fonction :

- des prévisions de vente de cet article à la période t ,
- la commande des clients à la période t ,
- le $CustOrderDur$ qui correspond à la fenêtre de temps durant laquelle les commandes clients consomment les prévisions,
- le $DRPCovDur$ qui correspond au nombre de périodes sur lesquelles les commandes sont regroupées : c'est une simple règle de lotissement,
- le stock de sécurité désiré à la période t ,
- le niveau de stock et les cours de route à la période t ,
- le niveau de backorders à la période t ,
- le lead time pour l'article considéré.

Le calcul du DRP s'effectue de façon classique en commençant par réajuster les prévisions de vente en fonction des commandes fermes des clients (les prévisions sont consommées par ces commandes

et les prévisions restantes sont réparties de façon équitable sur la fenêtre de temps $CustOrderDur$). En fonction des niveaux de stocks de sécurité désirés, du niveau de backorder, des cours de route et de la règle de lotissement (le $DRPCovDur$), le DRP détermine sa demande réelle et l'ordre de réapprovisionnement est lancé en tenant compte du délai d'obtention de l'article.

Les détails de configuration du DRP ne seront pas donnés pour une question de confidentialité.

2.4 L'algorithme de gestion de stock

L'algorithme de gestion des stocks est représenté par la figure 6.16. On voit que le principe de gestion est basé sur une politique FIFO en privilégiant, à chaque période t où il y a du backorder, celui du client prioritaire.

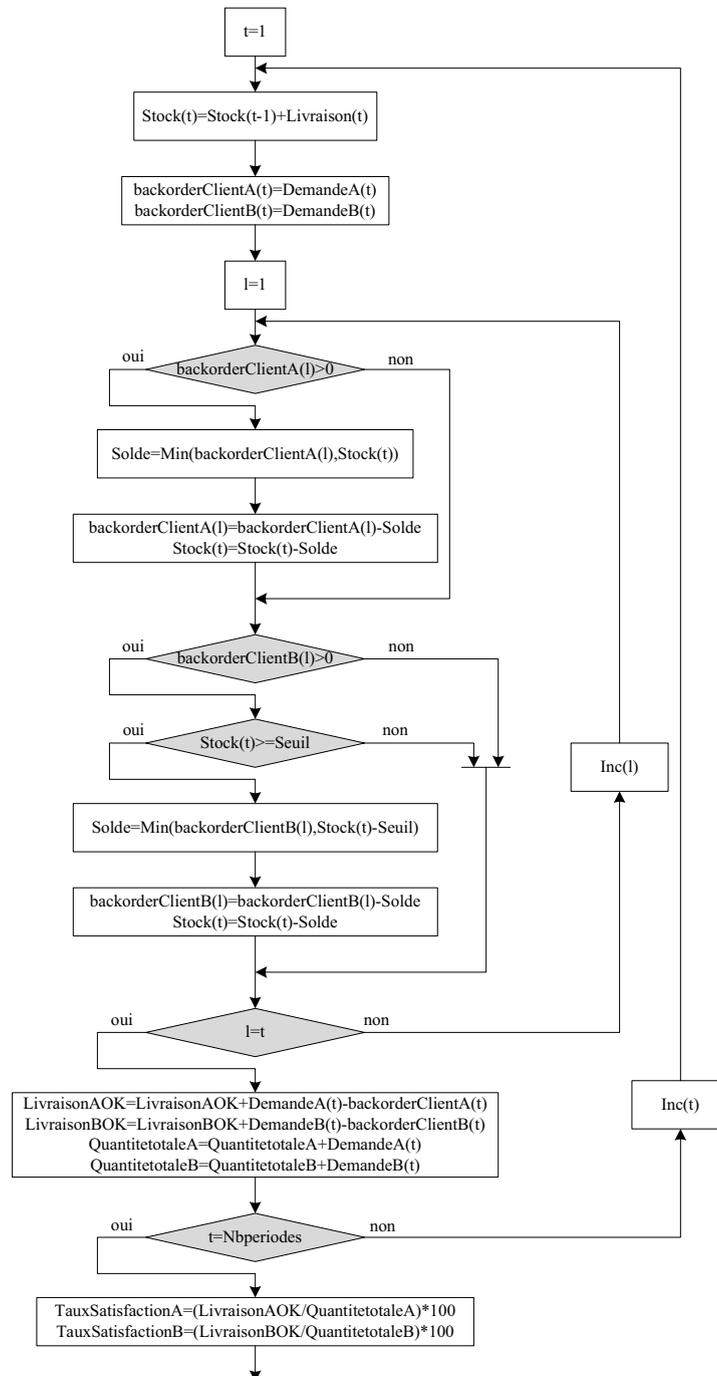


Figure 6.16 – L'algorithme de gestion des stocks

2.5 L'algorithme de calcul des seuils (Tempelmeier 2006)

Dans cette partie, nous considérons deux clients A et B dont le premier est prioritaire et nous utilisons les notations suivantes :

- $TSDcible_A$ (resp. $TSDcible_B$) est le taux de satisfaction désiré pour le client A (resp. B),
- TSD_A (resp. TSD_B) est le taux de satisfaction courant pour le client A (resp. B),
- SS est le stock de sécurité,
- SR est le seuil de rationnement du client B.

(Tempelmeier 2006) base son algorithme sur les constats suivants :

- Faire décroître SS en laissant SR constant diminue simultanément TSD_A et TSD_B . Cependant, c'est le second qui diminue le plus rapidement,
- Augmenter SR en laissant SS constant fait croître TSD_A en diminuant TSD_B ,
- Augmenter SR et SS de la même quantité augmente TSD_A tout en laissant TSD_B stable.

Ainsi, l'algorithme donné par (Tempelmeier 2006) se décompose en trois étapes de la façon suivante :

- Etape 1 : Commencer avec un stock de sécurité initial SS_0 et un seuil initial SR_0 .
Calculer $TSD_A^0 = TSD_A$ et $TSD_B^0 = TSD_B$.
- Etape 2 : Si $TSD_B^0 > TSDcible_B$ alors augmenter SR jusqu'à ce que $TSD_B \leq TSDcible_B$
Sinon diminuer SR jusqu'à ce que $TSD_B \geq TSDcible_B$.
Si $SR = 0$ alors augmenter SS jusqu'à ce que $TSD_B \geq TSDcible_B$.
- Etape 3 : Augmenter SR et SS de la même quantité jusqu'à ce que $TSD_A > TSDcible_A$.

Conclusion

Dans ce chapitre, nous avons présenté une politique de gestion de stock à demande différenciée basée sur le concept de seuil de rationnement. Suite à l'étude menée par (Comelli 2008), l'entreprise nous a confié l'élaboration d'une application permettant de calculer ces seuils afin de mettre sur pied un test «grandeur nature» de cette politique. Nous avons donc décrit les principales étapes qui ont permis de valider la faisabilité d'un tel test : il ne reste plus qu'à spécifier les clients à prioriser ainsi que le magasin où celui-ci sera mené. Nous avons également donné les détails des algorithmes utilisés dans l'application informatique de calcul de seuil. Les étapes 1 et 2 du projet s'achevant, le test grandeur nature devrait pouvoir débiter rapidement (son lancement est prévu début 2009).

On peut remarquer que cette politique de gestion de stock est une méthode particulièrement élégante d'assurer un taux de satisfaction moyen minimum pour un client prioritaire bien que la demande de celui-ci ne soit pas connue de façon certaine. Ainsi, il est possible grâce à cette technique d'atténuer les effets des aléas de la demande sur une planification tactique.

L'algorithme permettant d'ajuster le seuil pour obtenir les taux de satisfaction désirés a été donné par (Tempelmeier 2006). Cependant, il a été conçu dans le cas de deux clients. Pour pouvoir passer d'un test grandeur nature à une industrialisation complète de ce procédé, une recherche devra être menée afin de construire un algorithme permettant de calculer les seuils de rationnements pour différents clients, selon leurs niveaux de priorité.

Conclusion

Dans ce manuscrit, nous nous sommes intéressés à la problématique de la planification tactique, partie intégrante du management de la chaîne logistique. L'objectif de ce manuscrit est d'apporter une contribution au niveau de la modélisation et de la résolution des problèmes liés à cette problématique.

Dans le chapitre I, nous avons défini la problématique de la planification tactique et nous l'avons située au niveau des différentes approches rencontrées pour le management de la chaîne logistique. Nous avons mis en avant les différents problèmes rencontrés dans la mise en œuvre de ces dernières (problème de synchronisation horizontale, verticale, aléas sur la demande, etc.) puis nous avons donné un état de l'art des modèles mathématiques de référence dédiés à l'élaboration de plan de production. Parmi ceux-ci, nous avons distingué ceux répondant à notre problématique et nous avons remarqué l'absence de modèle de référence pour l'élaboration du master planning (planification multi-site).

Partant de ce constat, le but du chapitre II a été de fournir un modèle de connaissance pour la planification tactique. Ce dernier permet d'obtenir un modèle de lot sizing générique englobant les différentes problématiques d'élaboration de plans de production (PDP, PIC et master planning). En instanciant ce modèle, nous avons retrouvé les modèles mathématiques de référence pour la planification tactique. Lors de nos travaux, nous nous sommes souvent heurtés aux difficultés liées au manque d'uniformisation des instances de la littérature. Partant du modèle générique proposé, une perspective intéressante serait de définir une norme permettant d'homogénéiser l'écriture des instances. Une telle normalisation permettrait de proposer une bibliothèque facile d'utilisation pour la communauté de planification.

Le chapitre III se focalise sur l'élaboration de plans directeurs de production dans un contexte mono-site. Le modèle de lot sizing de référence pour cette problématique est le CLSP. Nous avons donc commencé par présenter ce modèle et ses extensions principales. Ensuite, un état de l'art des méthodes de résolution a été donné. Devant la multitude des instances utilisées, celui-ci ne permettait pas de conclure réellement sur l'efficacité des méthodes proposées. Cependant, il a permis de dégager celles qui sont le plus utilisées et de les proposer comme instances de référence. L'adoption de ces dernières par la communauté permettrait de faciliter les comparaisons entre les diverses méthodes de résolution de la littérature. Nous avons ensuite proposé deux méthodes d'optimisation pour le CLSP. La première s'appuie sur le chaînage de deux métaheuristiques tandis que la seconde utilise une fonction bi-objectif permettant de s'affranchir des difficultés liées à l'accessibilité des systèmes de voisinage proposés. Ces méthodes ont été testées sur les instances de la littérature et la métaheuristique bi-objectif s'avère particulièrement efficace. Sur les instances de (Eppen et Martin 1987), elle parvient même à améliorer certains des meilleurs résultats connus. Nous avons donc adapté cette dernière au cas du CLSP avec setup time et nous l'avons testée sur les instances de (Trigeiro et al. 1989) : les résultats que nous avons obtenus sont plus qu'encourageants et laissent présager de ceux qui pourraient être obtenus si les systèmes de voisinage utilisés étaient dédiés à ce problème.

Le chapitre IV s'intéresse à la planification multi-niveau et multi-site au travers d'un cas d'étude basée sur une chaîne logistique de type « Flow-Shop Hybride ». En utilisant le modèle mathématique générique que nous avons instancié sur la topologie particulière de cette chaîne et en intégrant

quelques contraintes spécifiques, nous avons obtenu un modèle mathématique générique pour la planification d'une chaîne logistique de type « Flow-Shop Hybride ». Nous avons ensuite proposé une méthode de résolution pour ce problème qui s'appuie sur une décomposition en deux sous-problèmes : un ayant trait à la planification multi-niveau à nomenclature série et à capacité finie et l'autre étant un problème d'affectation. Le premier sous-problème s'apparentant à un MLCLSP, nous avons donné un état de l'art des méthodes de résolution pour ce modèle et sa version à capacité infinie et, à l'instar du CLSP, nous avons tenté de définir des instances qui pourraient être considérées comme référentes pour ce problème. Nous avons ensuite proposé une méthode d'optimisation basée sur l'utilisation d'une métaheuristique pour le problème de planification et d'une heuristique pour le problème d'affectation. La métaheuristique utilise un système de voisinage basé sur 4 mouvements possibles. Cependant, comme chacun d'eux est susceptible de générer une solution non admissible, des algorithmes de réparation ont été conçus pour en restaurer la faisabilité. Enfin, une piste de recherche a été donnée afin d'améliorer et de généraliser notre approche.

Le chapitre V propose deux approches pour assurer la synchronisation entre les niveaux tactique et opérationnel en réestimant les capacités du système de production. La première d'entre elles s'appuie sur un couplage de deux modèles mathématiques : l'un est un modèle de planification tactique et l'autre un modèle d'ordonnancement. Ce dernier permet de vérifier la faisabilité du plan généré au niveau tactique ou de réestimer, le cas échéant, les capacités utilisées par le modèle tactique. Nous avons prouvé la convergence de cette méthode. Cependant, nous avons rapidement remarqué que celle-ci était confrontée d'une part aux complexités algorithmiques des problèmes de planification et d'ordonnancement sous-jacents et d'autre part aux limites liées à la modélisation mathématique du fonctionnement d'un atelier. C'est pour cela que la seconde approche proposée est basée sur le couplage entre une méthode approchée d'optimisation permettant de traiter des instances de taille industrielle, et de plusieurs modèles de simulation représentant chacun une règle de gestion de l'atelier afin de modéliser beaucoup plus finement son comportement. Les résultats sur la convergence sont préservés. Nous avons illustré cette dernière approche sur un problème d'atelier de type Flow-shop Hybride.

Le chapitre VI est issue d'une collaboration industrielle entre le laboratoire du LIMOS et une entreprise clermontoise. Dans celui-ci, nous avons montré l'influence d'une politique de gestion de stock à demande différenciée sur le taux de satisfaction client. Nous avons présenté ensuite les travaux menés pour élaborer un test grandeur nature de cette politique au sein de cette entreprise puis nous avons donné l'architecture fonctionnelle et les algorithmes de l'application que nous avons conçus pour déterminer les paramètres utiles à la mise en œuvre d'une telle politique.

Dans ce manuscrit, nous avons proposé un modèle générique de connaissance pour le domaine de la planification tactique à partir duquel on a pu obtenir un modèle mathématique générique permettant de répondre aux diverses problématiques telles que l'élaboration de PDP, PIC et Master Planning. Nous avons également proposé des méthodes de résolution (des modèles mathématiques, des heuristiques, des métaheuristiques ainsi que des couplages entre ces diverses méthodes, ainsi qu'un couplage entre une méthode d'optimisation et un modèle de simulation). En nous référant à la figure 6.17, nous voyons que nous avons exploité complètement le processus de modélisation ASCI et couvert la quasi totalité des modèles d'action définis dans cette méthodologie.

Tout au long de nos travaux, nous avons remarqué l'absence de cadre de travail formalisé (modèles et instances de référence, notation des problèmes rencontrés etc.) comme nous pouvons en rencontrer dans d'autres communautés telles que celle d'ordonnancement. L'une des directions de recherche de cette thèse a été de tenter d'unifier les modèles rencontrés et d'identifier, pour chaque problème de référence, les instances qui pourraient être considérées comme référentes. Bien qu'il existe déjà un site internet dédié aux problèmes de Lot-Sizing (Le LotsizeLib mis en ligne par Wolsey et disponible à l'adresse « <http://www.core.ucl.ac.be/wolsey/lotsizel.htm> ») qui fournit quelques instances et résultats de la

littérature, il serait sans doute utile de mettre en ligne un site où les chercheurs pourraient retrouver les instances de la littérature (et notamment les référentes) avec une écriture normalisée, les résultats obtenus par les différentes méthodes d'optimisation mises en œuvre sur ces dernières. Ceci permettrait une comparaison plus aisée des résultats obtenus par une nouvelle méthode d'optimisation. Le modèle de connaissance générique proposé dans ce manuscrit semble une bonne base de travail pour développer un tel site.

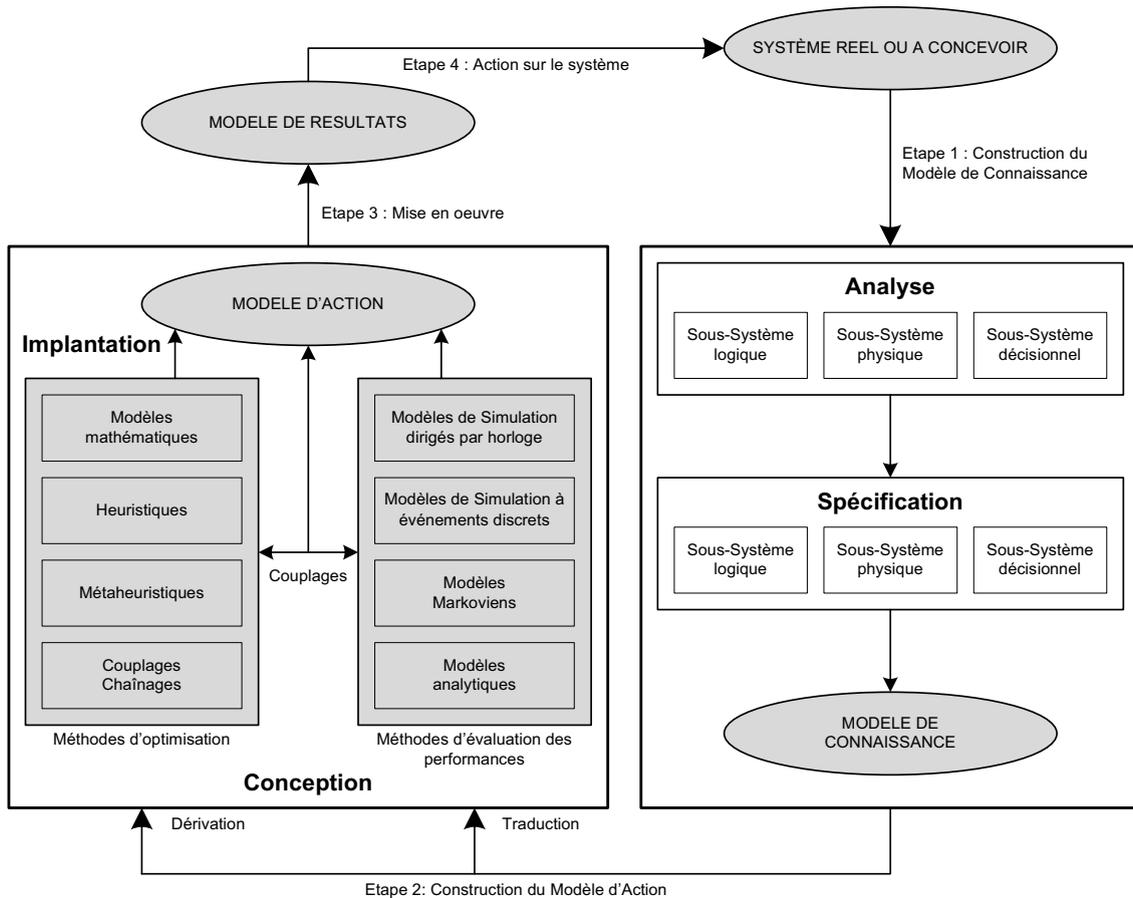


Figure 6.17 – Processus de modélisation ASCI

Yes!

Simon

Références bibliographiques

- Absi, N. et Kedad-Sidhoum, S. (2006). The multi-item capacitated lot-sizing problem with setup-times and shortage costs, *Research paper*, Laboratoire Lip6.
- Afentakis, P. et Gavish, B. (1986). Optimal lot-sizing for complex product structures, *Operations Research* **34** : 237–249.
- Aggarwal, A. et Park, J. (1993). Improved algorithms for economic lot-size problems, *Operations research* **41**(3) : 549–571.
- Akartunali, K. et Miller, A. (2008). A heuristic approach for big bucket multi-level production planning problems, *European Journal of Operational research*, à paraître .
- Arkin, E., Joneja, D. et Roundy, R. (1989). Computational complexity of uncapacitated multi-echelon production planning problems, *Operations Research Letters* **8** : 61–66.
- Baglin, G., Bruel, O., Garreau, A., Grief, M., Kerbache, L. et Van Delft, C. (2005). *Management Industriel et Logistique : Conception et pilotage de la Supply Chain*, 4th ed., Economica.
- Bahl, H., Ritzman, L. et Gupta, J. (1987). Determining lot sizes and resource requirements : a review, *Operations Research* **35**(3) : 329–345.
- Barany, I., Van Roy, T. et Wolsey, L. (1984). Strong formulations for multi-items capacitated lotsizing, *Management Science*, **30** : 1255–1261.
- Barbarosoglu, G. et Ozdamar, L. (2000). Analysis of solution space-dependent performance of simulated annealing : the case of the multi-level capacitated lot sizing problem, *Computers and Operations Research* **27** : 895–903.
- Belvaux, G. et Wolsey, L. (2000). Lot-sizing problems : Modeling issues and a specialized branch-and-cut system bc-prod, *Management Science* **46**(5) : 724–738.
- Berretta, R. et Rodrigues, L. (2007). A memetic algorithm for a multistage capacitated lot-sizing problem, *International Journal of Production Economics* **87** : 67–81.
- Billington, P., McClain, J. et Thomas, L. (1983). Mathematical programming approaches to capacity-constrained mrp system : review, formulation and problem reduction, *Management Science* **29**(10) : 1126–1141.
- Billington, P., McClain, J. et Thomas, L. (1986). Heuristics for multilevel lot-sizing with a bottleneck, *Management Science* **32**(8) : 989–1006.
- Bitran, G. et Matsuo, H. (1986). The multi-item capacitated lot-size problem : Error bounds of manne’s formulation, *Management Science* **32**(3) : 350–359.
- Bitran, G. et Yanasse, H. (1982). Computational complexity of the capacitated lot size problem, *Management Science* **46**(5) : 724–738.
- Blackburn, J. et Millen, R. (1982). Improved heuristics for multi-stage requirements planning systems, *Management Science* **28**(1) : 44–56.
- Boctor, F. (1987). The g-group heuristic for single machine lot scheduling, *International Journal of Production Research* **25** : 363–379.
- Bomberger, E. (1966). A dynamic programming approach to a lot size scheduling problem, *Management Science* **12** : 778–784.

- Bookbinder, J. et Koch, L. (1990). Production planning for mixed assembly/arborescent systems, *Journal of Operations Management* **9** : 7–23.
- Bourland, K. et Yano, C. (1997). A comparison of solution approaches for the fixed-sequence economic lot scheduling problem, *IEE Transactions* **29** : 103–108.
- Brahimi, N. (2004). *Planification de la production : modèles et algorithmes pour les problèmes de dimensionnement de lots*, PhD thesis, Université de Nantes.
- Brahimi, N., Dauzère Pérés, S., Najid, N. et Nordli, A. (2006). Single item lot sizing problems, *European Journal of Operational Research* **168**(1) : 1–16.
- Bruggemann, W. et Jahnke, H. (2000). The discrete lot-sizing and scheduling problem : Complexity and modification for batch availability, *European Journal of Operational Research* **124**(3) : 511–528.
- Buxey, G. (1989). Production scheduling : Practice and theory, *European Journal of Operational Research* **39** : 17–31.
- Byrne, M. et Bakir, M. (1999). Production planning using a hybrid simulation - analytical approach, *International Journal of Production Economic* **59**(3) : 305–311.
- Byrne, M. et Hossain, M. (2005). Production planning : An improved hybrid approach, *International Journal of Production Economic* **94** : 225–229.
- Carlier, J. et Chrétienne, P. (1988). Timed petri nets schedules, *Advances in Petri Nets : Lecture notes in Computer Science* **340** : 62–84.
- Chassang, G. (1983). *Gérer la production avec l'ordinateur*, Dunod.
- Chen, H. et Chu, C. (2003). A lagrangian relaxation approach for supply chain planning with order/setup costs and capacity constraints, *Journal of System Science and Systems Engineering* **12**(1) : 98–110.
- Chen, W. et Thizy, J. (1990). Analysis of relaxation for the multi-item capacities lot-sizing problem, *Annal of Operations Research* **26** : 29–72.
- Clark, A. et Armentano, V. (1995). The application of valid inequalities to the multi-stage lot-sizing problem, *Operations Research* **22**(7) : 669–680.
- Clark, A. et Scarf, H. (1960). Optimal policies for a multi-echelon inventory problem, *Management Science* **6** : 475–490.
- Comelli, M. (2008). *Modélisation, optimisation et simulation pour la planification tactique des chaînes logistiques*, PhD thesis, Université Blaise Pascal, Clermont-Ferrand.
- Comelli, M., Gourgand, M. et Lemoine, D. (2008). A review of tactical planning models, *Journal of Systems Science and Systems Engineering* **18 - N°2** : 204–229.
- Cossard, N. (2004). *Un environnement logiciel de modélisation et d'optimisation pour la planification de la production dans la chaîne logistique*, PhD thesis, Université Blaise Pascal, Clermont-Ferrand II.
- Courtois, A., Pillet, M. et Martin-Bonnefous, C. (2003). *Gestion de production, 4^{ème} édition*, Editions d'Organisation.
- Crama, Y., Pochet, Y. et Wera, Y. (2001). A discussion of production planning approaches in the process industry, *Working paper*, Université de Liège.
- Crandall, R. (1998). Production planning in a variable demand environment, *Production and Inventory Management Journal, 4th Quarter, APICS*, pp. 41–48.
- cxp (2006). conseil choix logiciel et progiciel. <http://www.cxp.fr>.
- Dauzère Pérés, S. et Lasserre, J. (1994). Integration of lotsizing and scheduling decisions in a job-shop, *European Journal of Operational Research* **75**(2) : 413–426.
- Dauzère Pérés, S. et Lasserre, J. (2002). On the importance of sequencing decisions in production planning and scheduling, *International Transactions in Operational Research* **9**(6) : 779–793.

- Degraeve, Z. et Jans, R. (2003). *a new Dantzig-Wolfe reformulation and branch-and-price algorithm for the capacitated lot sizing model with set-up times*, ERIM Report Series in Management ERS-2003-010-LIS, Erasmus University Rotterdam.
- Dellaert, N. et Jeunet, J. (2000). Solving large unconstrained multilevel lot-sizing problems using a hybrid genetic algorithm, *International Journal of Production Research* **38**(5) : 1083–1099.
- Dellaert, N. et Jeunet, J. (2002). Randomized multi-level lot-sizing heuristics for general product structures, *European Journal of Operational Research* **148**(1) : 211–228.
- Dellaert, N., Jeunet, J. et Jonard, N. (2000). A genetic algorithm to solve the general multi-level lot-sizing problem with time-varying costs, *International Journal of Production Economics* **68** : 241–257.
- Deshpande, V., Cohen, A. et Donohue, K. (2003). A threshold inventory rationing policy for service differentiates demand classes, *Management Science* **49** : 683–703.
- Diaby, M., Bahl, H., Karwan, M. et Zionts, S. (1992a). Capacitated lot-sizing and scheduling by lagrangean relaxation, *European Journal of Operational Research* **59** : 444–458.
- Diaby, M., Bahl, H., Karwan, M. et Zionts, S. (1992b). A lagrangean relaxation approach for very-large-scale capacitated lot-sizing, *Management Science* **38**(9) : 1329–1340.
- Dixon, P. et Silver, E. (1981). A heuristic solution procedure for the multi-item, single level, limited capacity, lotsizing problem, *Journal of operations management* **2** : 23–39.
- Dogramaci, A., Panayiotopoulos, J. et Adam, N. (1981). The dynamic lot-sizing problem for multiple items under limited capacity, *AIIE Transactions* **13**(4) : 294–303.
- Drexl, A. et Kimms, A. (1997). Lot sizing and scheduling - survey and extensions, *European Journal of Operational Research* **99** : 221–235.
- Du Merle, O., Goffin, J., Trouiller, C. et Vial, J. (1997). A lagrangian relaxation of the capacitated multi-item lot sizing problem solved with an interior point cutting plane algorithm, *Research paper*, McGill university, Montréal, Canada.
- Elmaghraby, S. (1978). The economic lot scheduling problem (elsp) : review and extension, *Management Science* **24** : 587–597.
- Eppen, G. et Martin, R. (1987). Solving multi-item lot-sizing problems using variable redefinition, *Operations Research* **35** : 832.
- Federgruen, A. et Tzur, M. (1991). Simple forward algorithm to solve general dynamic lot sizing models with n periods in $o(n \log n)$ or $o(n)$ time, *Management Science* **37**(8) : 909–925.
- Fleischmann, B. (2004). Advanced planning systems, *Mathematics in the Supply Chain, Oberwolfach report* **19** : 977–979.
- Fleischmann, B. et Meyr, H. (1997). The general lotsizing and scheduling problem, *OR Spectrum* **19**(1) : 11–21.
- Florian, M., Lenstra, J. et Rinnooy Kan, H. (1980). Deterministic production planning : Algorithms and complexity, *Management Science* **26** : 669–679.
- Féniès, P., Gourgand, M. et Tchernev, N. (2004). A framework for supply chain evaluation, *5e International Meeting for Research in Logistics, RIRL 04, Fortaleza, Brazil*.
- França, M., Armentano, V., Berretta, R. et Clark, A. (1997). A heuristic method for lot-sizing in multi-stage systems, *Operations Research* **24 - N°9** : 861–674.
- Ganeshan, R. et Harrison, T. (1995). An introduction to supply chain management, *Technical report*, Penn state university, department of management science and information system operations. Prentice Hall, New Jersey.
- Gelders, L., Maes, J. et Van Wassenhove, L. (1986). *A Branch and Bound Algorithm for the Multi-item Single Level Capacitated Dynamic Lotsizing Problem*, in S. Axsaster et al. (eds.), *Multi-stage Production Planning and Inventory Control, Lectures Notes in Economics and Mathematical Systems* 266, pp. 92–108.

- Genin, P. (2003). *Planification tactique robuste avec usage d'un A.P.S. Proposition d'un mode de gestion par plan de référence*, PhD thesis, Ecole supérieure des mines de Paris.
- Genin, P., Lamouri, S. et Thomas, A. (2001). Utilisation de la programmation linéaire dans l'optimisation du plan industriel et commercial : apports et limites, *3^e Conférence Francophone de MOdélisation et SIMulation (MOSIM'01) « Conception, Analyse et Gestion des Systèmes Industriels »*, Troyes, France.
- Gervais, M. (1995). *Stratégies de l'entreprise*, Economica.
- Giard, V. (2003). *Gestion de la production et des flux*, Economica, Coll.Gestion, 3^{eme} édition.
- Gilbert, K. et Madan, M. (1991). A heuristic for a class of production planning and scheduling problems, *IIE Transactions* **23** : 282–289.
- Gnoni, M. G., Iavagnilio, R., Mossa, G., G., M. et A., D. L. (2003). Production planning of a multi-site manufacturing system by hybrid modelling : A case study from the automotive industry, *International Journal of Production Economics* **85** : 251–262.
- Gopalakrishnan, M., Ding, K., Bourjolly, J. et Mohan, S. (2001). A tabu-search heuristic for the capacitated lot-sizing problem with set-up carryover, *Management Science* **47**(6) : 851–863.
- Gorham, T. (1968). Dynamic order quantities., *Production and Inventory Management*, **20** : 75–81.
- Gourgand, M., Grangeon, N., Lemoine, D. et Norre, S. (2006). Software tool for the master production schedule conception based on the capacitated lot sizing problem., *Doctoral Consortium of International Conference on Automated Planning and Scheduling (ICAPS)* pp. 68–70.
- Gourgand, M., Grangeon, N., Lemoine, D. et Norre, S. (2007). Couplage de modèles pour une planification tactique robuste. application à la planification dans un bloc opératoire., *5^{ème} conférence internationale de Conception et Production Intégrées (CPI'2007)*, Rabat.
- Gourgand, M. et Kellert, P. (1991). Conception d'un environnement de modélisation de systèmes de production, *Actes du 3^{ème} congrès international de Genie industriel*, Tours.
- Gourgand, M., Lemoine, D. et Norre, S. (2008). Metaheuristiques pour le capacitated lot sizing problem., *7^{ème} conférence francophone de modélisation et simulation (MOSIM08)*, Paris.
- Graves, S. (1981). In multi-stage production / inventory control systems : Theory and practice, *TIMS Studies in the Management Science* **16** : 95–110.
- Graves, S. (1999). *Manufacturing Planning and Control*, Massachusetts Institute of Technology.
- Grubbstrom, W. et Molinder, A. (1996). Safety production plans in mrp-systems using transform methodology, *Computer and Operations Research - Proceedings of the 8th International Working Seminar on Production Economics* **46-47** : 297–309.
- Haase, K. (1993). *Lot sizing and scheduling for production planning*, PhD thesis, University of Kiel.
- Haase, K. et Kohlmorgen, U. (1995). Parallel genetic algorithm for the capacitated lot-sizing problem, *Operations research proceedings* pp. 370–375.
- Haugen, K., Olstad, A. et Pettersen, B. (2006). The profit maximizing capacitated lot-size (pclsp) problem, *European Journal of Operational Research* **à paraître**.
- Hêche, J., Liebling, T. et De Werra, D. (2003). *Recherche opérationnelle pour ingénieurs*, Presses polytechniques et universitaires romandes.
- Heinrich, C. et Schneeweiss, C. (1986). Multi-stage lot-sizing for general production systems, in *S. Axsater, C. Schneeweiss, and E. Silver (Eds.), Lecture Notes in Economics and Mathematical Systems*, Springer Verlag, Heidelberg, Germany.
- Heizer et Render (1995). *Production and Operations management*, Prentice Hall.
- Held, M. et Karp, R. (1971). The traveling salesman problem and minimum spanning trees, *Operations Research* **18** : 1138–1162.
- Hennet, J. (1997). *Concepts et outils pour les systèmes de production*, Cepaduès - éditions.

- Hennet, J. (2001). Dimensionnement et ordonnancement cyclique de lots de fabrication, *Actes.3ème Conférence Francophone de MODélisation et SIMulation (MOSIM'01), Troyes (France)* **2** : 669–675.
- Hindi, K. (1995). Computationally efficient solution of multi-item capacitated lot sizing problems., *Computers and industrial Engineering* **28(4)** : 709–719.
- Hindi, K. (1996). Solving the clsp by a tabu search heuristic., *Journal of Operational Research Society* **47(1)** : 151–161.
- Hindi, K., Fleszar, K. et Charalambous, C. (2003). An effective heuristic for the clsp with set-up times, *Journal of Operational Research Society* **54** : 490–498.
- Hsu, W. (1983). On the general feasibility test of scheduling lot sizes for several products on one machine, *Management Science* **29** : 93–105.
- Huisman, D., Jans, R., Peeters, M. et Wagelmans, A. (2003). Combining column generation and lagrangian relaxation, *Technical report*.
- Hung, Y., C.C., S. et Chen, C. (1999). Evolutionary algorithms for production planning problems with setup décisions, *Journal of Operational Research Society* **50** : 857–866.
- Hung, Y. et Chien, K. (2000). A multi-class multi-level capacitated lot sizing model, *Journal of Operational Research Society* **51** : 1309–1318.
- Jans, R. et Degraeve, Z. (2007). Metaheuristics for dynamic lot sizing : A review and comparison of solution approaches, *European Journal of Operational Research* **177(3)** : 1855–1875.
- Jeunet, J. et Jonard, N. (2005). Single point stochastic search algorithms for the multi-level lot-sizing problem, *International Journal of Production Economics* **32** : 985–1006.
- Karimi, B., S.M.T., F. G. et J.M., W. (2005). A tabu-search heuristic for the clsp with backlogging and set-up carry-over, *Journal of Operational Research Society* **47(6)** : 851–863.
- Katok, E., Lewis, H. et Harrison, T. (1998). Lot sizing in general assembly systems with setup costs, setup times, and multiple constrained resources, *Management Science* **44(6)** : 859–877.
- Kim, B. et Kim, S. (2001). Extended model for a hybrid production planning approach, *International Journal of Production Economic* **73** : 165–173.
- Kim, H. et Hosni, Y. A. (1998). Manufacturing lot-sizing under mrp ii environment : an improved analytical model and a heuristic procedure, *Computers and Industrial Engineering* **35(3)** : 423–426.
- Kimms, A. (1999). A genetic algorithm for multi-level, multi-machine lot sizing and scheduling, *Computers and operations research* **26** : 829–848.
- Kirca, d. et Kökten, M. (1994). A new heuristic approach for the multi-item dynamic lot sizing problem, *European Journal of Operational Research* **75** : 332–341.
- Koçaga, Y. et Sen, A. (2007). Spare parts inventory management with delivery lead times and rationing, *IIE Transactions* **39(9)** : 879–898.
- Kuik, R. et Solomon, M. (1990). Multi-level lot-sizing problem : Evaluation of a simulated-annealing heuristic, *European Journal of Operational Research* **45(1)** : 25–37.
- Kuik, R., Solomon, M., Van Wassenhove, L. et Maes, J. (1993). Linear programming simulated annealing and tabu search heuristics for lotsizing in bottleneck assembly systems, *IEE Transactions* **25(1)** : 162–172.
- Lambrecht, M. et Vanderveken, H. (1979). Heuristic procedures for the single operation multi item loading problem, *AIIE Transactions* **11** : 319–326.
- Lamouri, S., Genin, P. et Thomas, A. (2005). Planification avancée : Aps, <http://www.techniques-ingenieur.fr>.
- Lamouri, S. et Thomas, A. (1999). Optimization of the process of development of the sop, *International Conference (CPI'99), Tanger, Maroc*, pp. 328–338.

- Lasdon, S. (2002). *Optimization theory for large system*, Dover publications.
- Lee, H. et Billington, C. (1993). Material management in decentralized supply chain, *Operation Research* **41**(5) : 835–847.
- Leung, J., Magnanti, T. et Vachani, R. (1989). Facets and algorithms for the capacitated lot sizing, *Mathematical programming* **45** : 331–359.
- Lin, J. et Chen, Y. (2007). A multi-site supply network planning problem considering variable time buckets- a tfl-lcd industry case, *International Journal Manufacturing Technology* **33** : 1031–1044.
- LotsizeLib (2006). Lot-sizing problems : A library of models and matrices. <http://www.core.ucl.ac.be/wolsey/lotsizel.htm>.
- Maes, J., McClain, J. et Van Wassenhove, L. (1991). Multilevel capacitated lotsizing complexity and lp-based heuristics, *European Journal of Operational Research* **53**(2) : 131–148.
- Maes, J. et Van Wassenhove, L. (1986a). Multi-item single-level capacitated dynamic lot-sizing heuristics : A computational comparison (part i : Static case, part ii : Rolling horizon), *IIE transaction* **18** : 114–129.
- Maes, J. et Van Wassenhove, L. (1986b). A simple heuristic for the multi-item single level capacitated lot sizing problem, *Letters of the Operational Research Society* **4** : 265–274.
- Maes, J. et Van Wassenhove, L. (1988). Multi-item single-level capacitated dynamic lot-sizing heuristics : A general review, *Journal of the Operational Research* **39**(11) : 991–1004.
- Manne, A. (1958). Programming of economic lot sizes, *Management Science* **4** : 115–135.
- Martin, A. (1990). *Drp Distribution Resource Planning*, Oliver Wight Publications.
- Mc Donald, C. et Karimi, I. (1997). Planning and scheduling of parallel semicontinuous processes. 1. production planning, *Industrial Engineering Chemical Research* **36** : 2691–2700.
- McClain, J., Thomas, L. et Weiss, E. (1987). Efficient solutions to linear programming model for the production scheduling with capacity constraints, *Working paper series 85-14*, Johnson Graduate School of Management.
- McLaren, B. (1977). *A study of Multiple level lot sizing techniques for Material Requierements Planning Systems*, PhD thesis, Purdue University.
- Meyr, H., Wagner, M. et Rohde, J. (2002). Structure of advanced planning systems, *Supply Chain Management and Advanced Planning - concepts, models, software and case studies*, Stadtler H., Kilger C., Springer-Verlag, Berlin.
- Meyr, H., Wagner, M. et Rohde, J. (2005). *Structure of Advanced Planning Systems*, 2nd ed., Springer.
- Michalewicz, Z. et Fogel, D. (2002). *How to solve it : Modern heuristics*, 3^{eme} édition, Springer-Verlag, Berlin.
- Miller, A., Nemhauser, G. et Savelsbergh, M. (2000). Solving multi-item capacitated lot-sizing problems with setup-times by branch-and-cut, *Technical report*.
- Nollet, J., Kélada, J. et Diorio, M. (1994). *La gestion des opérations et de la production : une approche systémique*, 2^{eme} édition, Gaëtan Morin éditeur.
- Norre, S. (2005). Heuristiques et métaheuristiques pour la résolution de problèmes d'optimisation combinatoire dans les systèmes de production, *Habilitation à diriger des recherches, Université Blaise Pascal-Clermont-Ferrand II*.
- Oliver, R. et Webber, M. (1992). Supply-chain management :logistics catches up with strategy (reprint from outlook 1982), *Logistics - the strategic issues*, Christopher, M. (Ed.), London.
- Orlicky, J. (1975). *Material Requirements Planning*, McGraw-Hill, London.
- Ozdamar, L. et Barbarosoglu, G. (2000). An integrated lagrangean relaxation-simulated annealing approach to the multi-level multi-item capacitated lot sizing problem, *International Journal of Production Economics* **68** : 319–331.

- Ozdamar, L., Birbil, S. et M.C., P. (2002). Technical note : New result for the capacitated lot sizing problem with overtime decisions and setup times, *Production Planning & control* **13**(1) : 2–10.
- Ozdamar, L. et Bozyel, M. (2000). The capacitated lot sizing problem with overtime decisions and setup times, *IIE Transactions* **32** : 1043–1057.
- Palmer, D. (1965). Sequencing jobs through a multistage process in the minimum total time - a quick method of obtaining a near optimum, *Operations research Quarterly* **16**(1) : 101–107.
- Pibernic, R. and Sucky, R. (2007). An approach to inter-domain master planning in supply chains, *International Journal Production Economics* **108** : 200–212.
- Pitakaso, R., Almeder, C., K.F., D. et Hartl, R. (2007). A max-min ant system for unconstrained multi-level lot-sizing problems, *Computer and Operations Research* **34** : 2533–2552.
- Pochet, Y. et Wolsey, L. (1991). Solving multi-item lot-sizing problems using strong cutting planes, *Management Science* **37** : 53–67.
- Porter, M. (1986). *L'avantage concurrentiel*, InterEditions.
- Raza, A. S. et Akgunduz, A. (2008). A comparative study of heuristic algorithms on economic lot scheduling problem, *Computers and Industrial Engineering* **55**(1) : 94–109.
- Rizk, N. et Martel, A. (2001). Supply chain flow planning methods : a review of the lot-sizing literature, *Working paper DT-2001-AM-1*, Université Laval (canada).
- Roll, Y. et Karni, R. (1991). Multi-item, multi-level lotsizing problem with an aggregate capacity constraint, *European Journal of Operational Research* **51** : 73–87.
- Rosling, K. (1986). *Optimal Lot-Sizing for Dynamic Assembly Systems*, S. Axsater, Ch. Schneeweiss and E. Silver edn, Springer-Verlag, Berlin, chapter Multi-Stage Production Planning and Inventory Control, pp. 119–131.
- Salomon, M. (1991). *Multi-stage Production Planning and Inventory Control*, Lectures Notes in Economics and Mathematical Systems 355, pp. 92–108.
- Salomon, M., Kuik, R. et Van Wassenhove, L. (1993). Statistical search methods for lot-sizing problems, *Annals Operations Research* **41** : 453–468.
- Samii, A. (2006). *Stratégie logistique, supply chain management*, 3^{ème} édition, Dunod.
- Schwarz, L. et Schrage, L. (1975). Optimal and system myopic policies for multi-echelon production/inventory assembly systems, *Management Science* **21**(11) : 1285–1294.
- Shapiro, J. F. (2001). *Modeling the Supply Chain*, Pacific Grove : Wadsworth Group, Duxbury Thomson Learning.
- Sikora, R., Chhajed, D. et Shaw, M. (1996). Integrating the lot-sizing and sequencing decisions for scheduling a capacitated flow line, *Computers & Industrial Engineering* **30**(4) : 659–679.
- Silver, E. et Meal, H. (1973). A heuristic for selecting lot size quantities for the case of a deterministic time-varying demand rate and discrete opportunities for replenishment, *Production and inventory management* **14** : 64–74.
- Süral, M., Denizel, M. et Van Wassenhove, L. (2008). Lagrangean relaxation based heuristics for lot sizing with setup times, *European Journal of Operational Research*, in press .
- Stadler, H. (1996). Mixed integer programming model formulation for dynamic multi-item multi-level capacitated lotsizing, *European Journal of Operational Research* **94** : 561–581.
- Stadler, H. (2002). Basics of supply chain management, *Supply Chain Management and Advanced Planning - concepts, models, software and case studies*, Stadler H., Kilger C., Springer-Verlag, Berlin.
- Stadler, H. (2005). Supply chain management and advanced planning - basics, overview and challenges, *European Journal of Operational Research* **163**(3) : 575–588.

- Stadler, H. et Kilger, C. (2004). *Supply Chain Management and Advanced Planning, concepts, models, software and case studies*, 3^{ème} édition, Springer.
- Tang, O. (2004). Simulated annealing in lot sizing problem, *International Journal of Production Economics* **88**(2) : 173–181.
- Tchernev, N. (1997). *La modélisation du processus logistique dans les systèmes flexibles de production*, PhD thesis, Université Blaise Pascal, Clermont-Ferrand.
- Tchernev, N. (2003). *Approvisionnement et logistique, Manuel de l'étudiant*, Institut Universitaire Professionnalisé « Management et gestion des entreprises », Université d'Auvergne.
- Tempelmeier, H. (2001). Supply chain planning with advanced planning systems, *Tinos* .
- Tempelmeier, H. (2006). Supply chain inventory optimization with two customer classes in discrete time, *European Journal of Operational Research* **174** : 600–621.
- Tempelmeier, H. et Derstroff, M. (1996). A lagrangean based heuristic for dynamic multilevel multiitem constrained lotsizing with setup times, *Management Science* **42**(5) : 738–756.
- Tempelmeier, H. et Helber, S. (1994). A heuristic for dynamic multi-item multi-level capacitated lotsizing for general product structures, *European Journal of Operational Research* **75** : 296–311.
- Thierry, C. (2003). Gestion de chaînes logistiques : Modèles et mise en oeuvre pour l'aide à la décision à moyen terme, *Habilitation à diriger des recherches, Université de Toulouse-le-Mirail*.
- Thierry, C., Chapeaublanc, N., Lepage, P. et Bel, G. (1994). Multi site planning : A centralized or a distributed approach?, *Conference INRIA, Sophia Antipolis, France* .
- Thizy, J. et Van Wassenhove, L. (1983). A subgradient algorithm for the multi-item capacitated lot-sizing problem, *Research Report EES-83-14*, Department of Civil Engineering, Princeton University.
- Thizy, J. et Van Wassenhove, L. (1985). Lagrangean relaxation for the multi-item capacitated lotsizing problem : a heuristic implementation, *IIE Transactions* **17** : 308–313.
- Thomas, A. et Lamouri, S. (2000). Flux poussés : Mrp et drp, *Techniques de l'Ingénieur* **AG 5 110**.
- Timpe, C. et Kallrath, J. (2000). Optimal planning in large multi-site production networks, *European Journal of Operational Research* **126** : 422–435.
- Trigeiro, W., Thomas, L. et Mc Clain, J. (1989). Capacitated lot sizing with setup times, *Management science* **35** : 353–366.
- Veral, E. et LaForge, R. (1985). The performance of a simple incremental lot-sizing rule in a multilevel inventory environment, *Decision Sciences* **16** : 57–72.
- Vollmann, T., Berry, D. et Whybark (1997). *Manufacturing planning and control systems*, 4th ed., New York et al.
- Voros, J. (2002). On the relaxation of multi-level dynamic lot-sizing models, *International Journal of Production Economics* **77**(1) : 53–61.
- Wagelmans, A., Van Hoesel, S. et Kolen, A. (1992). Economic lot sizing : an $O(n \log n)$ that runs in linear time in the wagner-whitin case, *Operations Research* **40**(1) : S145–S156.
- Wagner, H. et Whitin, T. (1958). Dynamic version of the economic lot size model, *Management science* **5** : 89–96.
- Wight, O. (1984). *Manufacturing Resource Planning : MRP II.*, Oliver Wigth éditeur,.
- Wolosewicz, C. (2008). *Approche intégrée en planification et ordonnancement de la production*, PhD thesis, Ecole National des Mines de Saint-Etienne.
- Wolosewicz, C., Dauzère Péres, S. et Aggoune, R. (2006). Modélisation et résolution d'un problème général de planification et d'ordonnancement, *6ème conférence francophone de modélisation et simulation (MOSIM06)*, Rabat.

- Womack, J., Jones, D. et Roos, D. (1992). *Le système qui va changer le monde : une analyse des industries automobiles mondiales dirigée par le MIT*, Dunod.
- Xie, J. et Dong, J. (2002). Heuristic genetic algorithms for general capacitated lot-sizing problem, *Computers and Mathematics with Applications* **44** : 263–276.
- Zangwill, W. (1966). A deterministic multiproduct, multifacility production and inventory model, *Operations Research* pp. 486–507.
- Zangwill, W. (1969). A backlogging model and a multi-echelon model of a dynamic economic lot size production system - a network approach, *Management Science* **15** : 506–527.
- Zhao, X., Luh, P. et Wang, J. (1999). The surrogate gradient algorithm for lagrangian relaxation method, *Journal of Optimization Theory and Applications* **100**(3) : 699–712.

Annexes

Annexe 1 - Comparaison des modèles de lot-sizing mono-niveau

1 Comparaison des trois modèles «small bucket»

Afin de percevoir les différences entre les modèles mono-niveau «small bucket» détaillés dans le chapitre I, ceux-ci ont été résolus sur une même instance que nous avons élaboré. Pour ce faire nous avons construit une instance comportant 8 produits et 2 macro-périodes. Pour tester les modèles «small bucket» nous avons «éclaté» chaque macro-période en 10 micro-périodes en ajustant les capacités, les coûts de stock pour chaque micro-période. L'horizon de planification a donc été subdivisé en 20 micro-périodes. On se propose de planifier la fabrication de 8 produits finis dont la demande est indiquée dans le tableau 7.1. On indique également dans ce tableau la capacité par période (C_t) en terme de produits du système de production.

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
d_{1t}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	70
d_{2t}	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	40
d_{3t}	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	50
d_{4t}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
d_{5t}	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
d_{6t}	0	0	0	0	0	0	0	0	0	70	0	0	0	0	0	0	0	0	0	0	40
d_{7t}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20
d_{8t}	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	20
C_t	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35

Tableau 7.1 – jeu d'essai : demandes et capacités «small bucket»

Le tableau 7.2 nous donne, pour chaque produit i , le stock initial (I_{i0}), la capacité nécessaire pour produire une unité de produit i (p_i), le coût de stockage par micro-période (h_i) ainsi que de lancement (s_i).

i	1	2	3	4	5	6	7	8
I_{i0}	0	0	0	0	0	0	0	0
p_i	1	1	1	1	1	1	1	1
h_i	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
s_i	100	200	200	300	400	250	500	300

Tableau 7.2 – jeu d'essai : Stock initial, Capacité requise, Coût de stockage, de lancement

On impose que la demande doit être satisfaite sans retard.

Nous avons représenté les solutions obtenues grâce à ces modèles par les diagrammes de Gantt suivants :

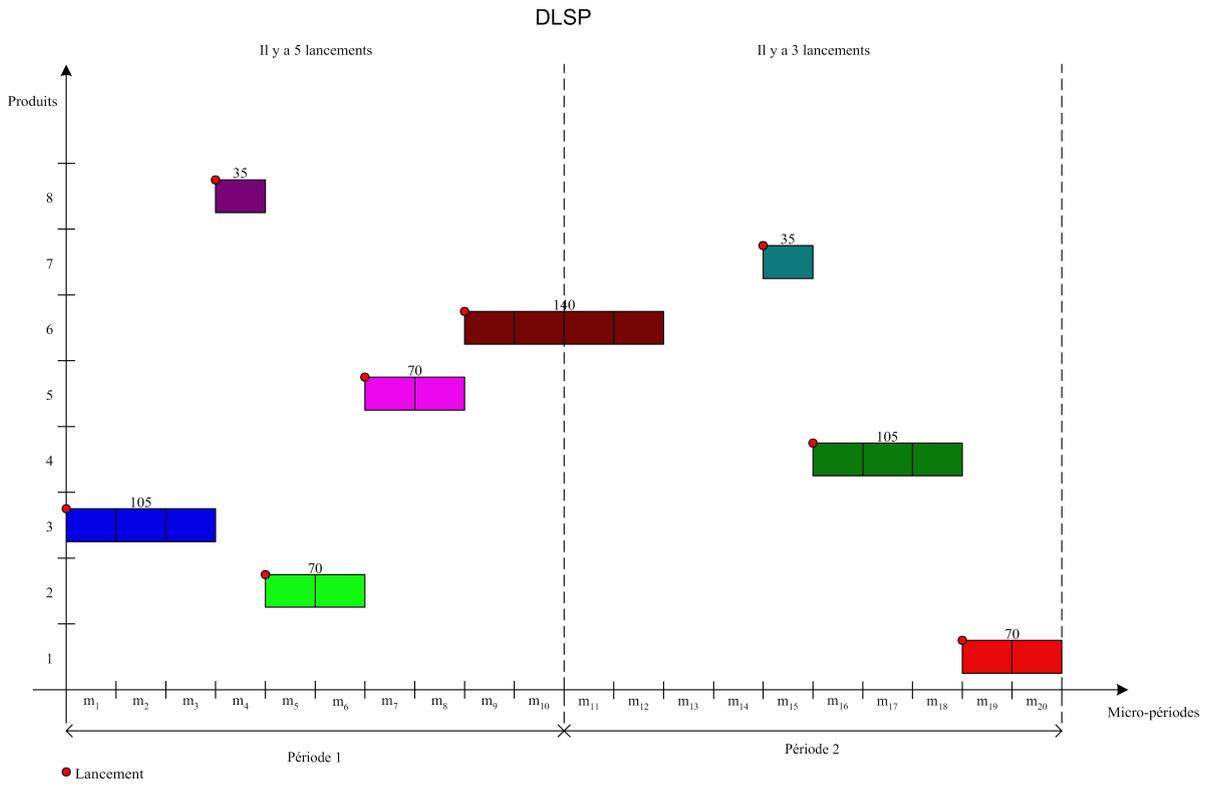


Figure 7.1 – Diagramme de Gantt obtenu pour le DLSP

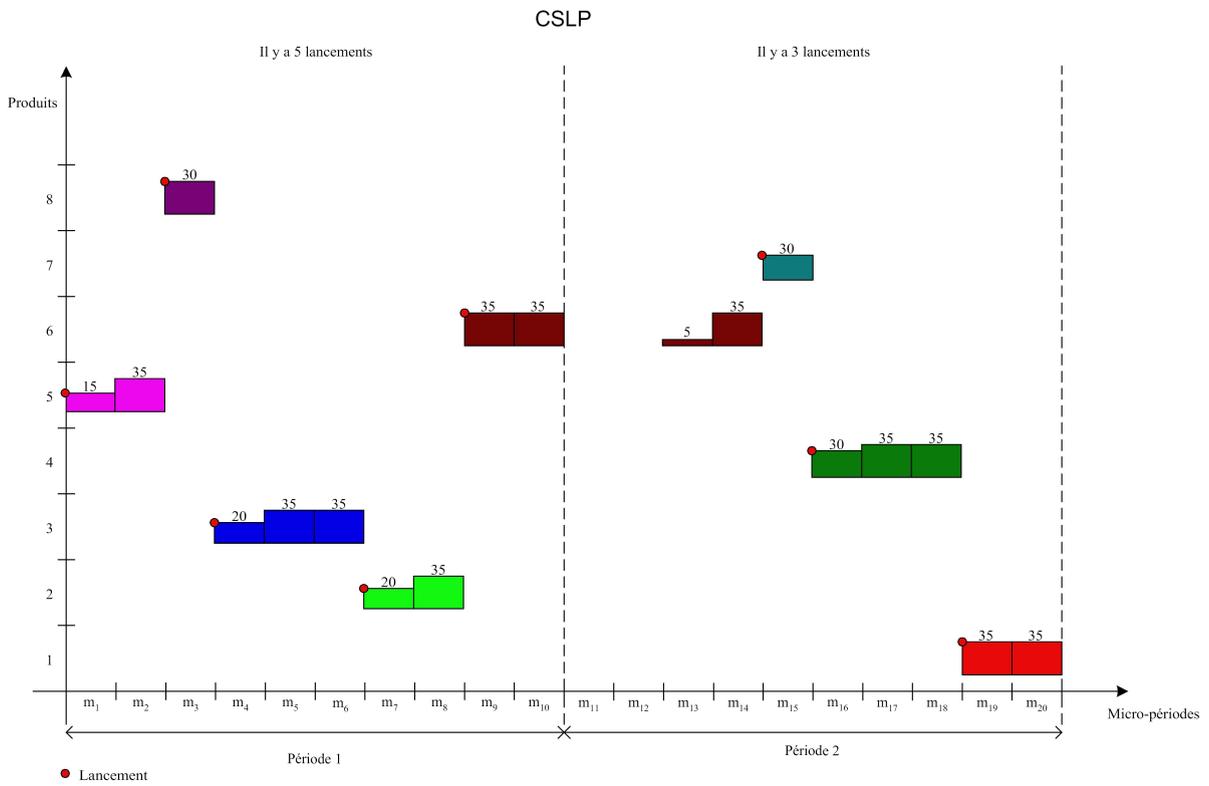


Figure 7.2 – Diagramme de Gantt obtenu pour le CSLP

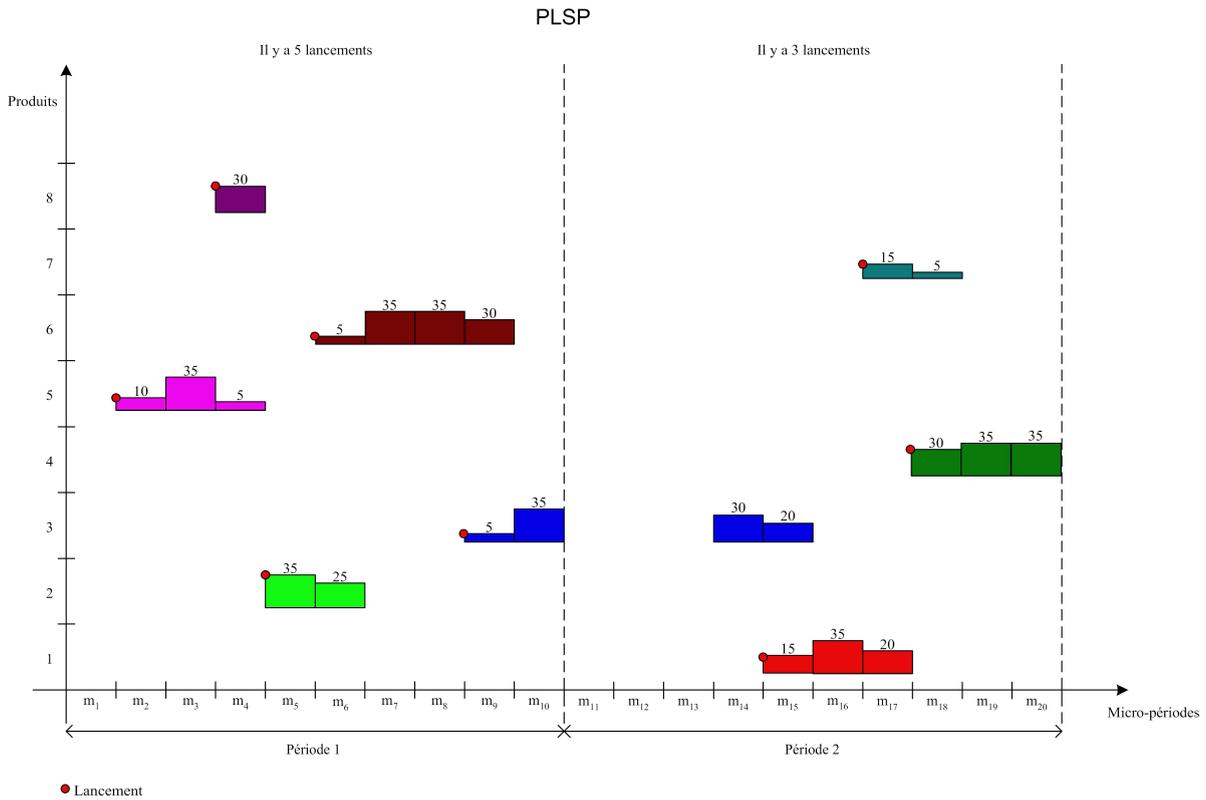


Figure 7.3 – Diagramme de Gantt obtenu pour le PLSP

Les trois modèles diffèrent essentiellement sur deux aspects :

- la production « tout ou rien » pour le DLSP, une production proportionnelle pour le CSLP et le PLSP,
- le calcul des coût de lancement : si on rencontre une période d'inactivité entre la fabrication du même produit, le DLSP comptera un coût de lancement contrairement aux deux autres.

Le tableau 7.3 donne pour la solution optimale obtenue par chaque modèle, la valeur de la fonction objectif ainsi que le temps mis pour l'obtenir au moyen du solveur cplex :

Modèle	Fonction objectif	Temps de calcul
DLSP	2689.5	8,77 s
CSLP	2551.5	1,31 s
PLSP	2526.5	587,25 s

Tableau 7.3 – Résultats obtenus au moyen de Cplex

On remarque également que la résolution du PLSP est beaucoup plus longue, ceci est dû au fait que celui-ci permet la production de deux produits par période, contrairement aux deux autres modèles « small bucket »

Pour chaque modèle, on obtient 8 lancements mais la répartition de la production est très différente : cela est dû à la politique de production inhérente à chaque modèle. On retrouve l'influence de celle-ci dans l'évolution des stocks présenté dans le graphique 7.4 :

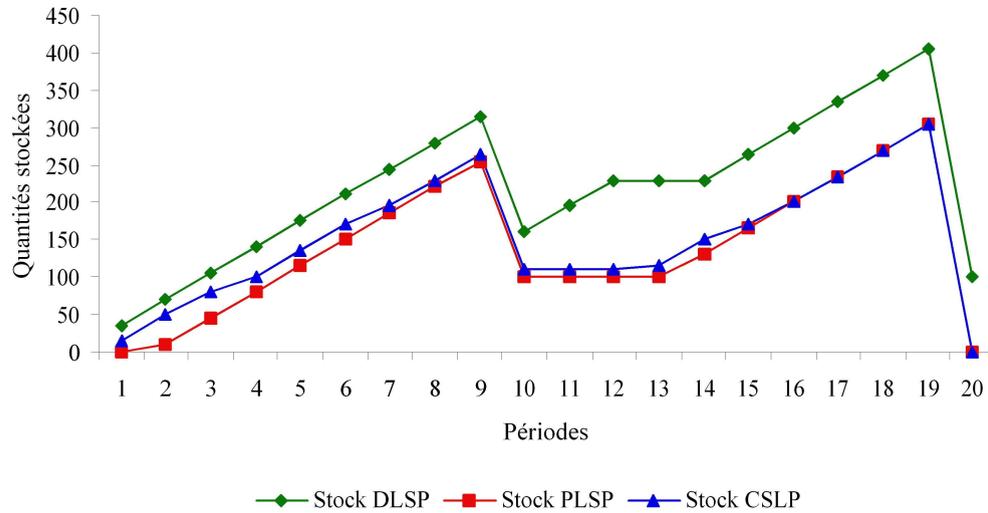


Figure 7.4 – Evolution des stocks cumulés pour les différents modèles

On remarque que l'évolution des stocks est plus favorable dans le cas du PLSP, ce qui n'est guère étonnant puisqu'on est capable de fabriquer deux produits durant la même période, ce qui a pour effet de diminuer la mise en stock. Evidemment, le DLSP utilisant une loi de « tout ou rien » obtient les plus mauvais résultats et on peut même voir qu'on aura trop produit à l'issue des 20 micro-périodes. Pour notre cas d'étude, le PLSP obtient donc la meilleure solution.

Néanmoins, ces seules comparaisons ne sont en aucun cas pertinentes quant aux choix d'un modèle pour traduire une réalité industrielle. En effet, le but n'est pas de choisir le plus performant mais celui le plus proche de la réalité physique de l'atelier de production. Ainsi, le but de cet exemple est d'illustrer les hypothèses de modélisation implicitement liées aux modèles. Ainsi le DLSP suppose qu'au cours d'une micro-période, la ressource est utilisée à 100% de sa capacité pour un et un seul type de produit, le CSLP modélise lui le fait qu'au cours d'une micro-période une machine peut être employée en dessous de sa capacité maximale, le PLSP modélisant le fait que sur une même micro-période deux produits peuvent être pour l'un commencé et pour l'autre terminé. Sachant cela, il reste à identifier quel modèle est le plus proche de la réalité et quel est le but de la modélisation et de l'optimisation.

2 Comparaison entre le CLSP et le GLSP

De même que pour les modèles « small bucket » le GLSP et le CLSP ont été comparés sur l'instance que nous avons fabriquée.

Cette comparaison permet de faire ressortir un aspect important du GLSP : ainsi, bien que le GLSP comme le CLSP compte un coût de lancement dès qu'un produit est fabriqué, il n'en compte pas entre deux macro-périodes dans le cas où au moins un produit est fabriqué au cours de ces macro-périodes. Il suppose ainsi que la machine n'a pas été démontée par exemple en fin de semaine.

L'horizon de planification a été subdivisé en 2 macro-périodes. On se propose de planifier la fabrication de 8 produits finis dont la demande est indiquée dans le tableau 7.4. On indique également dans ce tableau la capacité par période (C_t) en terme de produits du système de production.

Les détails des instances et des résultats obtenus sont donnés dans les tableaux 7.4, 7.5 et 7.6. Une solution de CLSP peut être représentée par un histogramme alors que celle du GLSP peut être représentée par un diagramme de Gantt.

t	1	2
d_{1t}	0	70
d_{2t}	20	40
d_{3t}	40	50
d_{4t}	0	100
d_{5t}	50	0
d_{6t}	70	40
d_{7t}	0	20
d_{8t}	10	20
C_t	350	350

Tableau 7.4 – jeu d’essai : demande et capacité « big bucket »

Le tableau suivant nous donne, pour chaque produit i , le stock initial (I_{i0}), la capacité nécessaire pour produire une unité de produit i (p_i), le coût de stockage par macro-période (h_i) ainsi que de préparation (s_i).

On impose que la demande doit être satisfaite sans retard.

i	1	2	3	4	5	6	7	8
I_{i0}	0	0	0	0	0	0	0	0
p_i	1	1	1	1	1	1	1	1
h_i	10	10	10	10	10	10	10	10
s_i	100	200	200	300	400	250	500	300

Tableau 7.5 – jeu d’essai : Stock initial, capacité requise, Coût de stockage, de lancement « big bucket »

Le tableau suivant donne les résultats de notre instance obtenus grâce au solveur Cplex.

Modèle	Fonction objectif	Temps de calcul
CLSP	3100	0,01 s
GLSP	2350	1,11 s

Tableau 7.6 – Résultats obtenus au moyen de Cplex

On remarque que pour notre exemple, le GLSP obtient de bien meilleurs résultats car il compte beaucoup moins de lancements ce qui influence beaucoup la répartition de la production.

Les solutions de ces modèles « small bucket » ont été représentées par les diagrammes de Gantt des figures 7.5 et 7.6

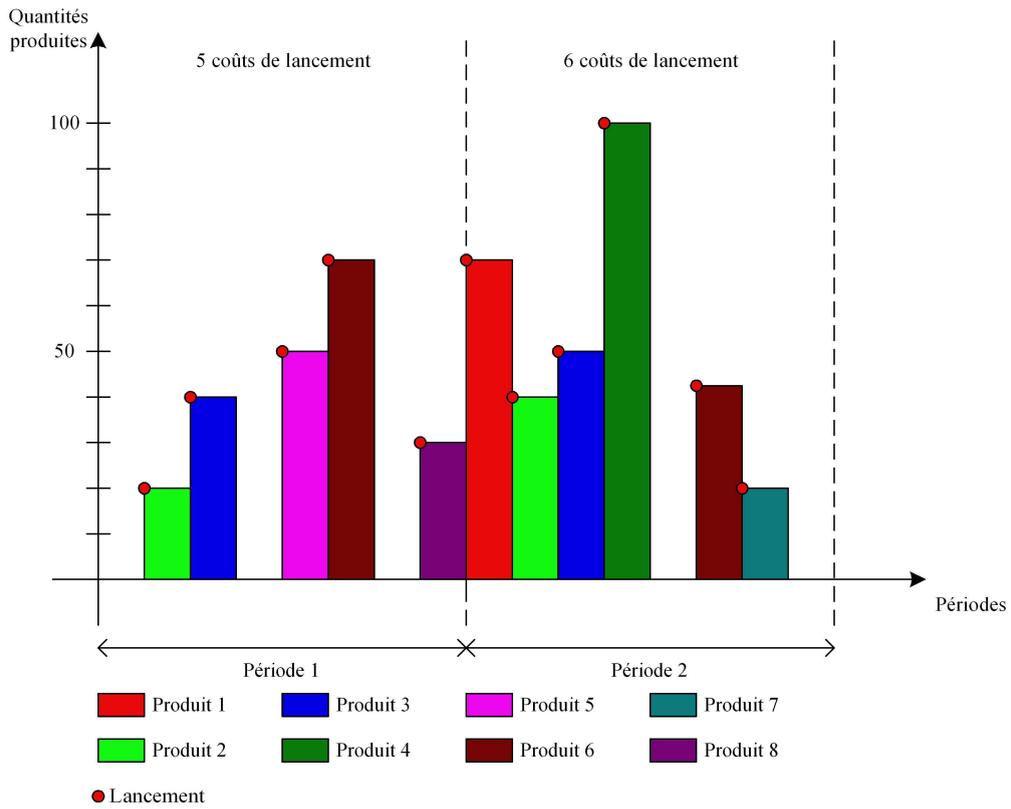


Figure 7.5 – CLSP

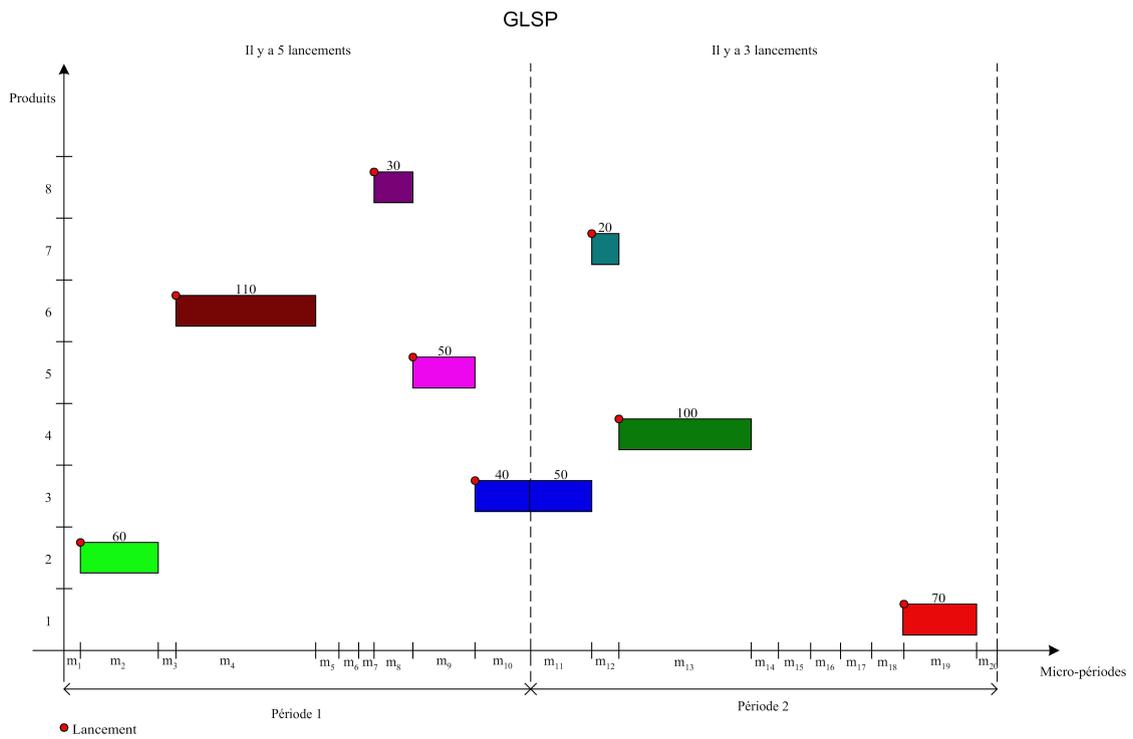


Figure 7.6 – GLSP

Pour conclure, le tableau 7.7 récapitule pour les modèles que nous avons rencontrés, leurs positionnements au regard des trois phases définies par (Drexel et Kimms 1997) et décrites dans le chapitre I :

phase	CLSP	DLSP	CSLP	PLSP	GLSP
I	×	×	×	×	×
II	×	×	×	×	×
III		×	×	×	×

Tableau 7.7 – Tableau récapitulatif pour les modèles dans la logique MRP

En effet, pour tous les modèles rencontrés, la planification se fait directement en capacité finie (phases I et II), mais seuls les modèles « small bucket » et le GLSP propose un séquençement de la production (phase III).

Annexe 2 - Les procédures de réparations

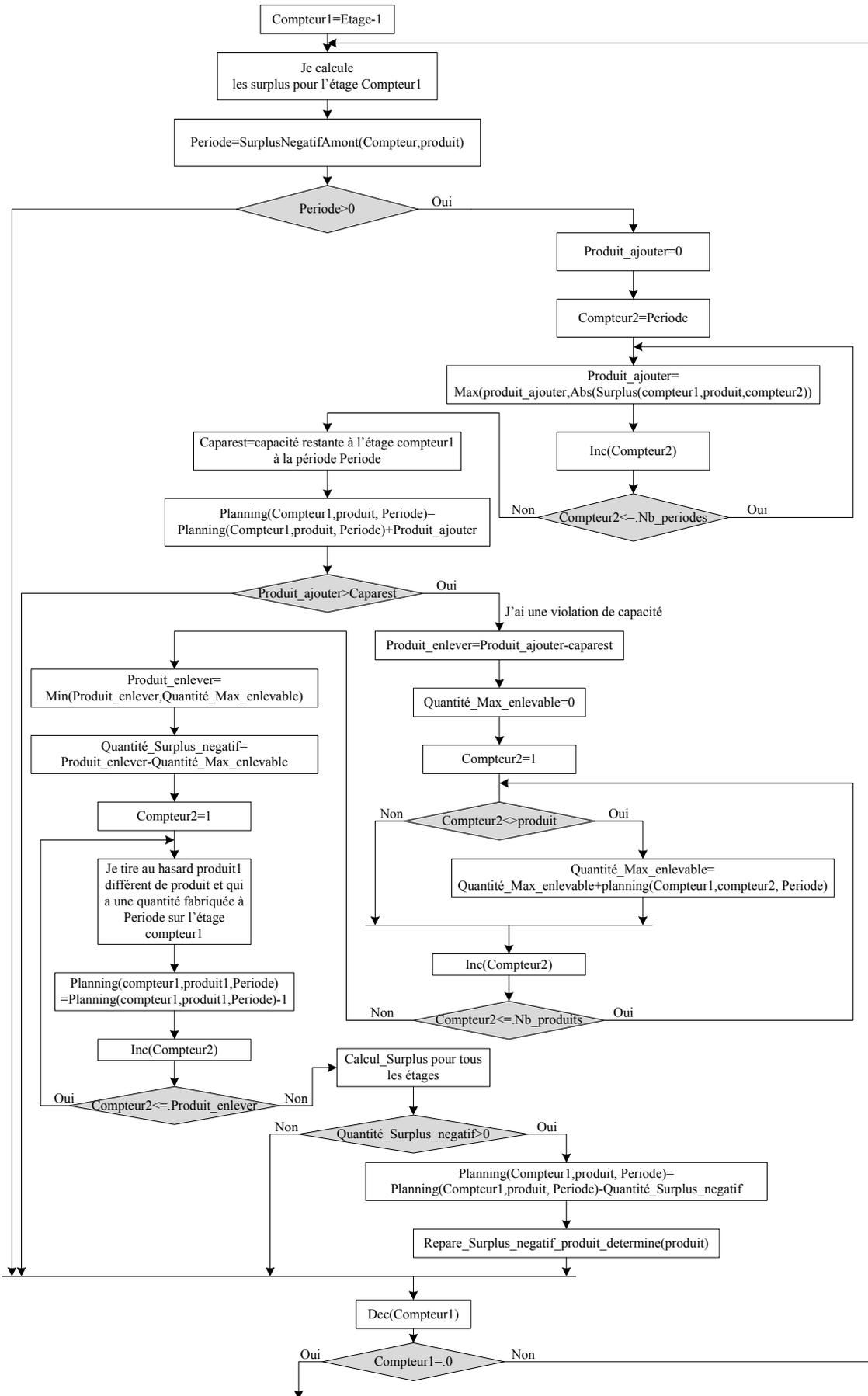


Figure 7.7 – Procédure 1

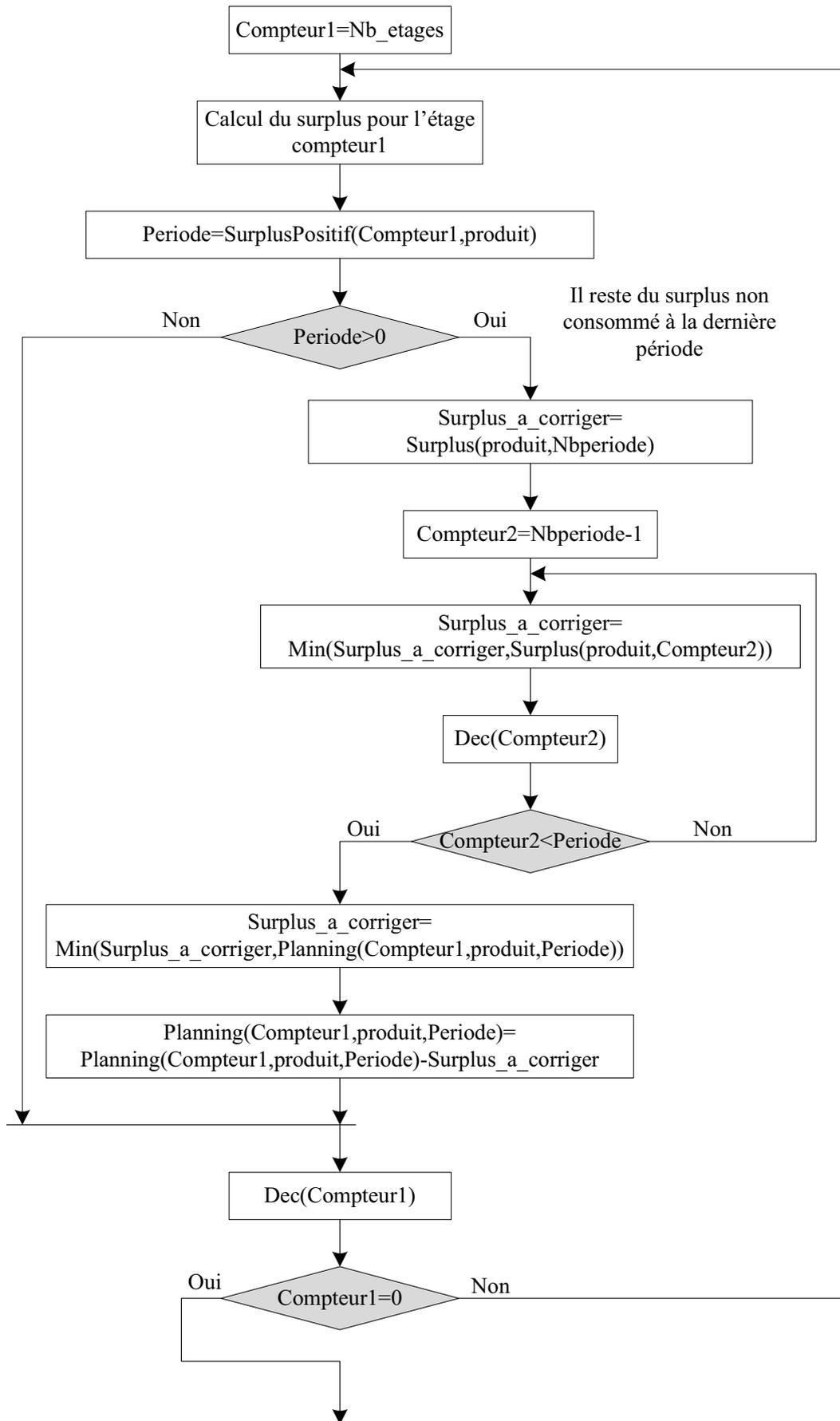


Figure 7.9 – Procédure 3

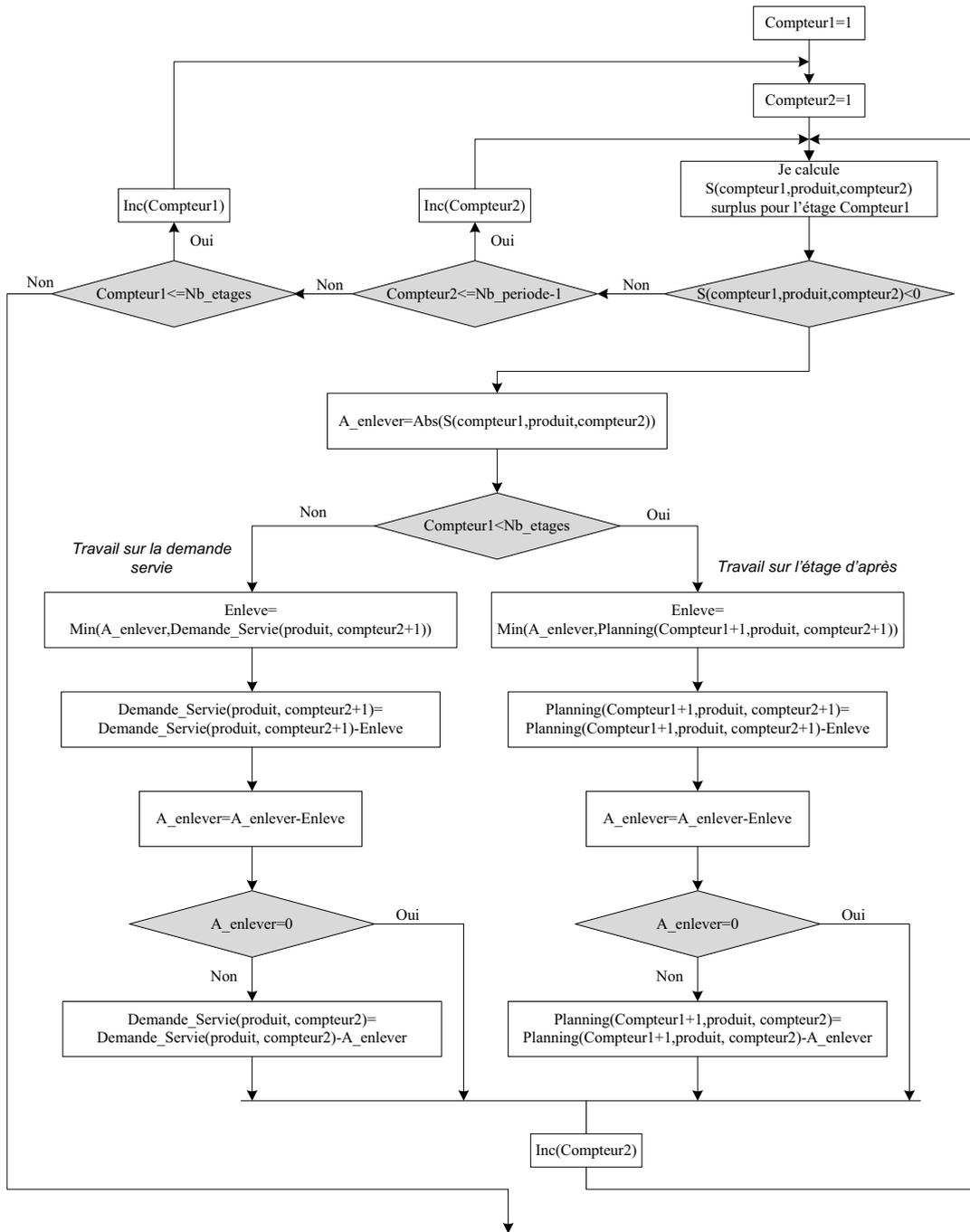


Figure 7.10 – Procédure 4

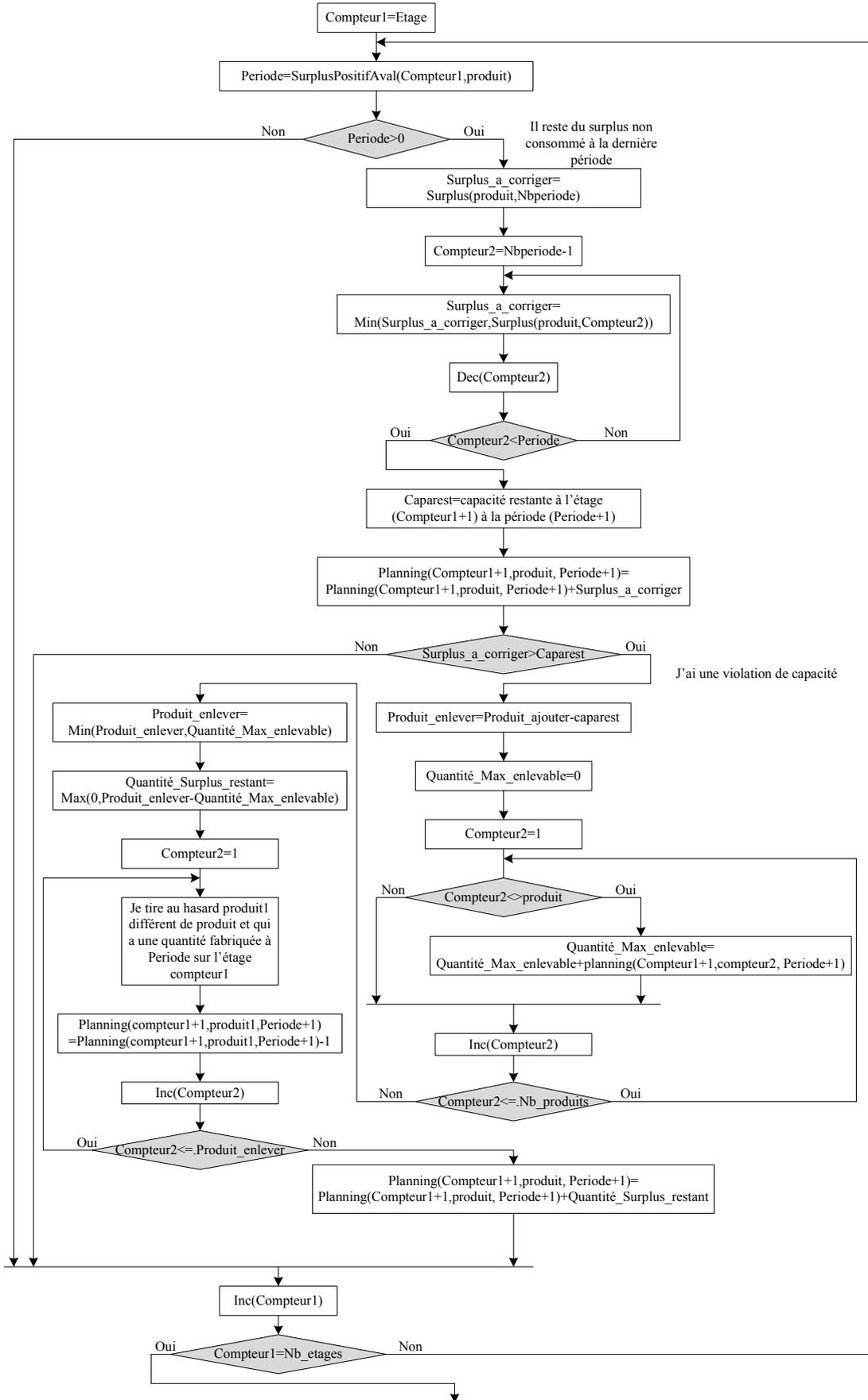


Figure 7.11 – Procédure 5

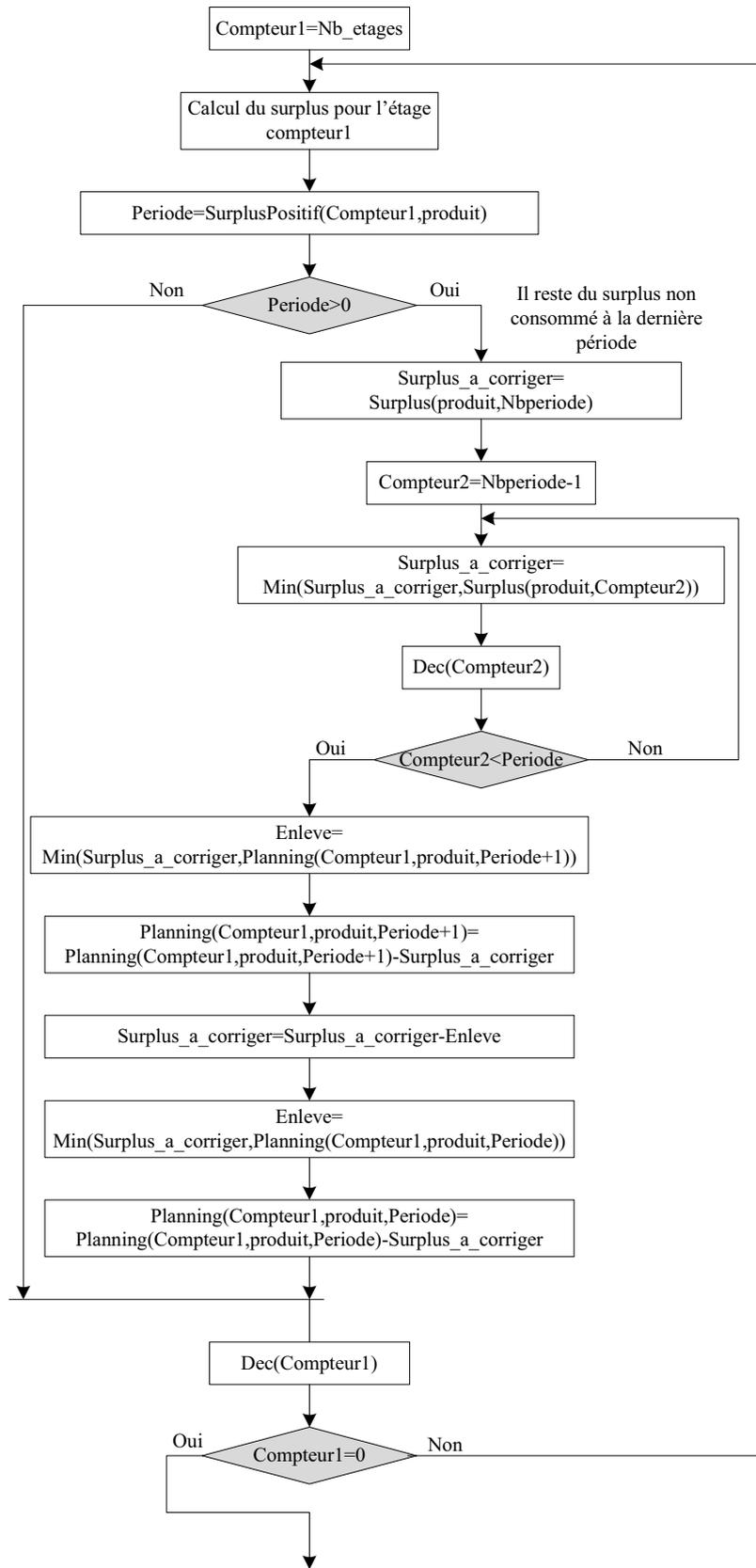


Figure 7.12 – Procédure 6

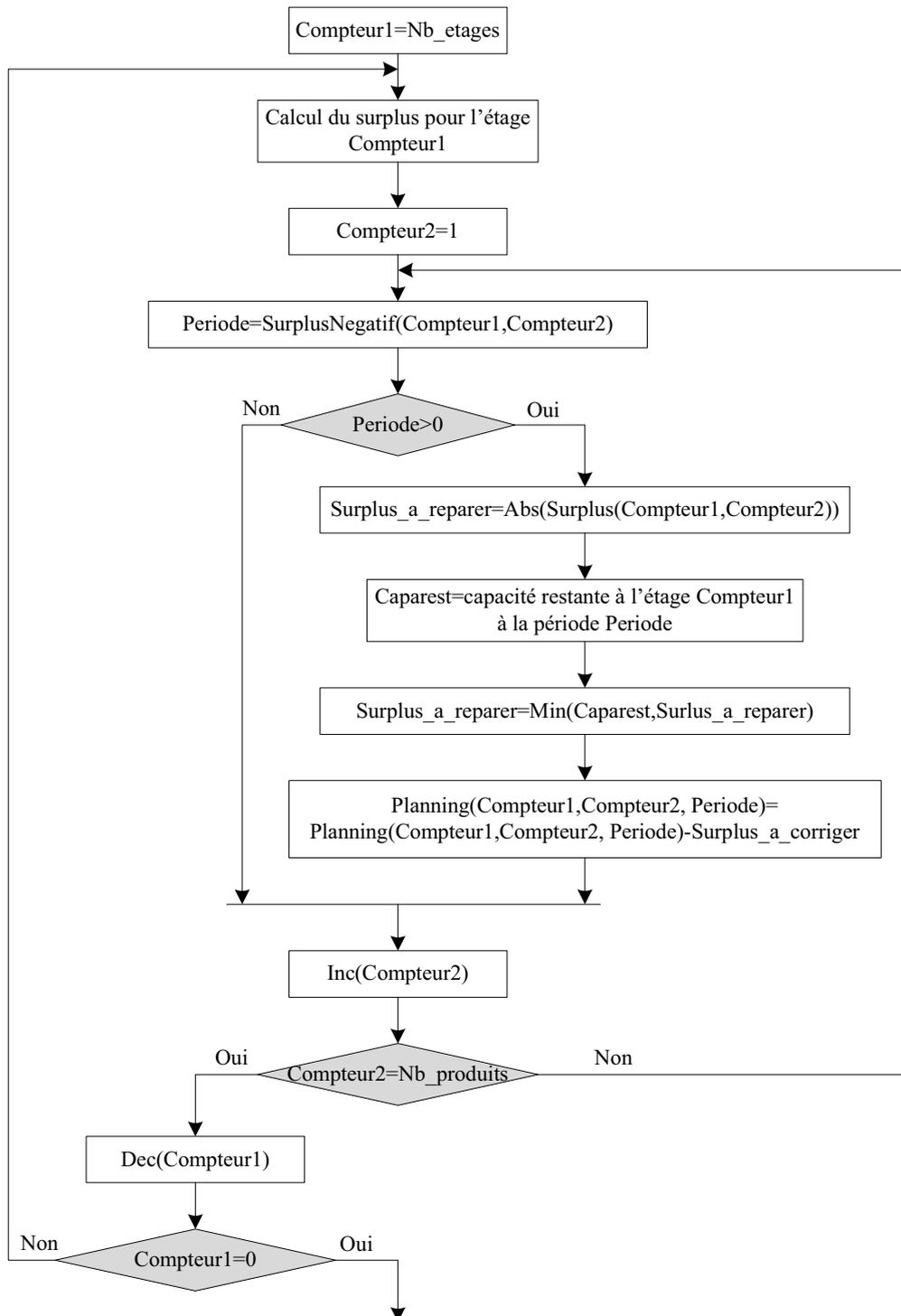


Figure 7.13 – Procédure 7

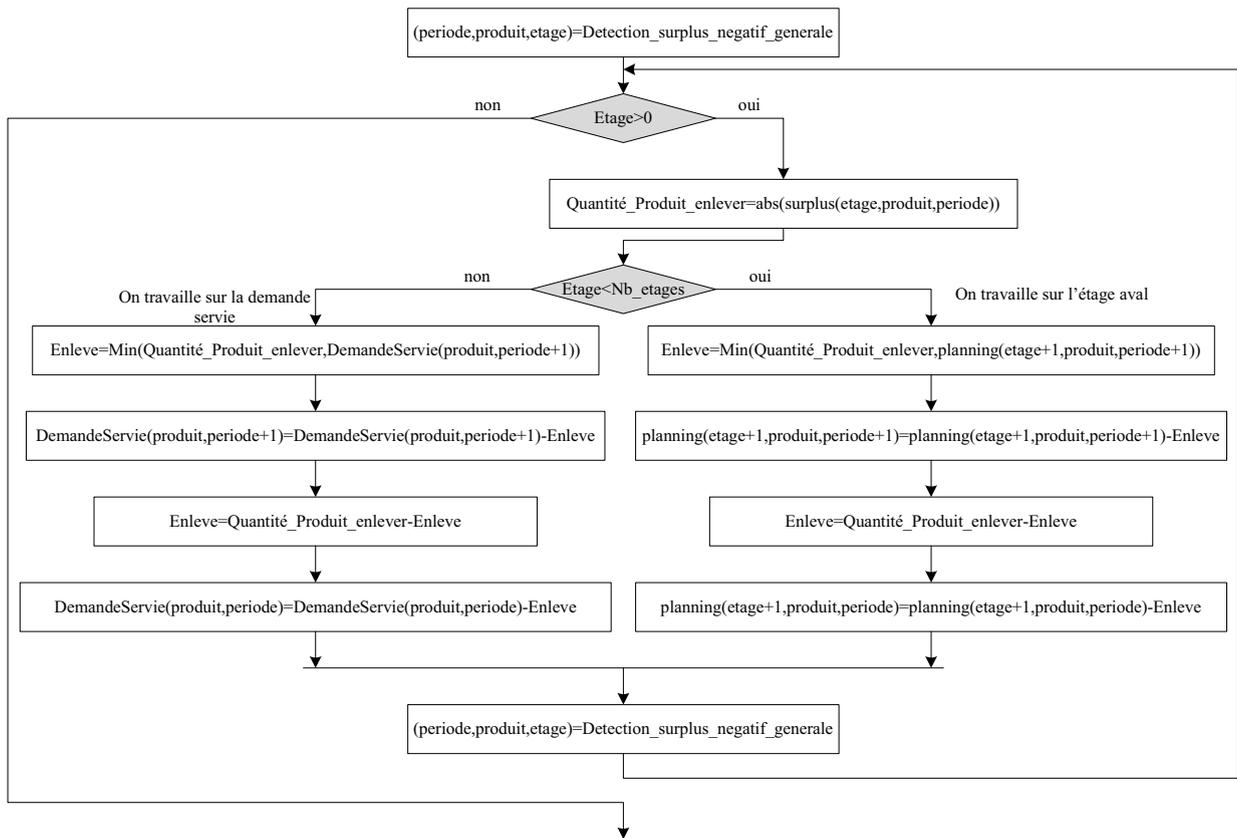


Figure 7.14 – Procédure 8

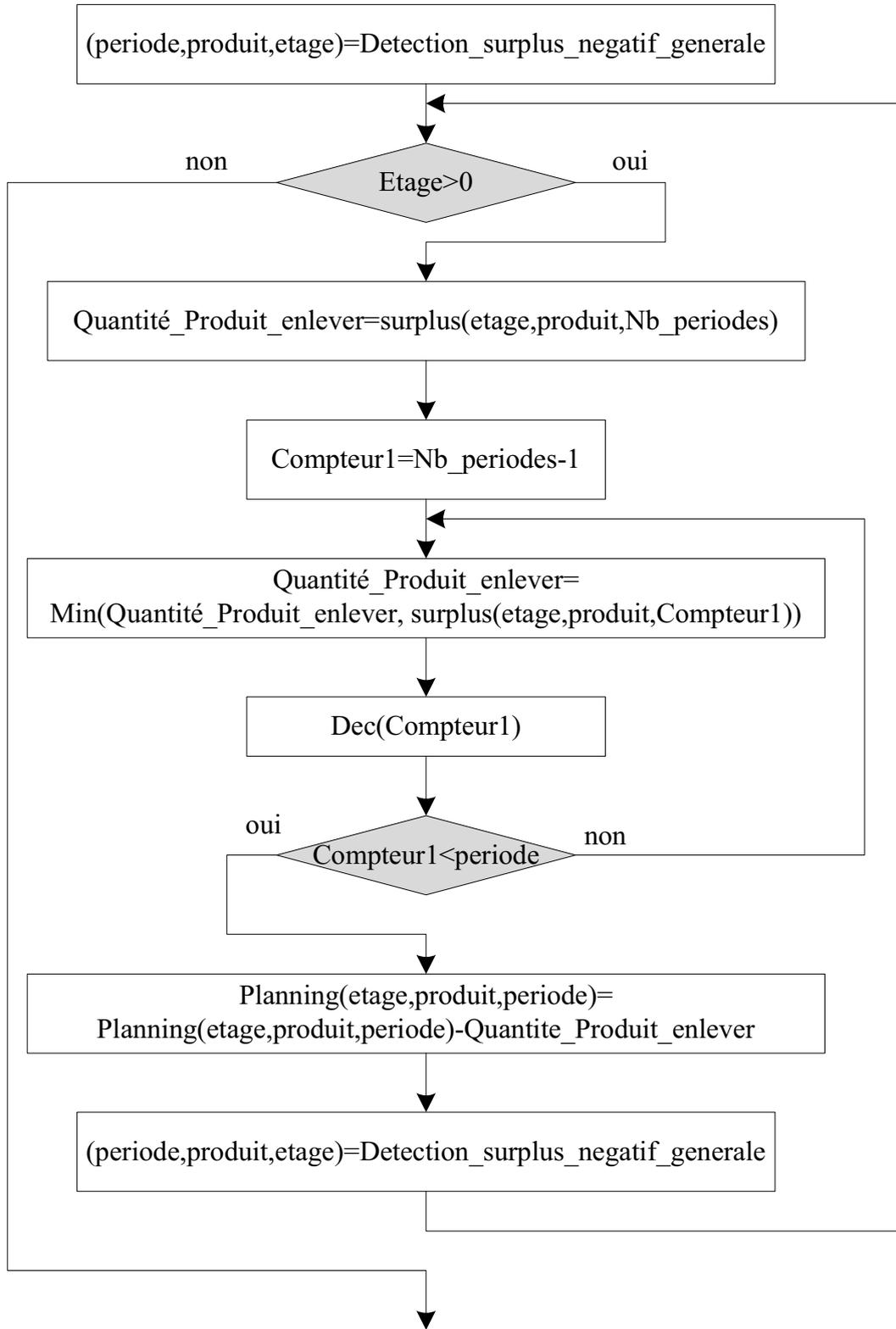


Figure 7.15 – Procédure 9

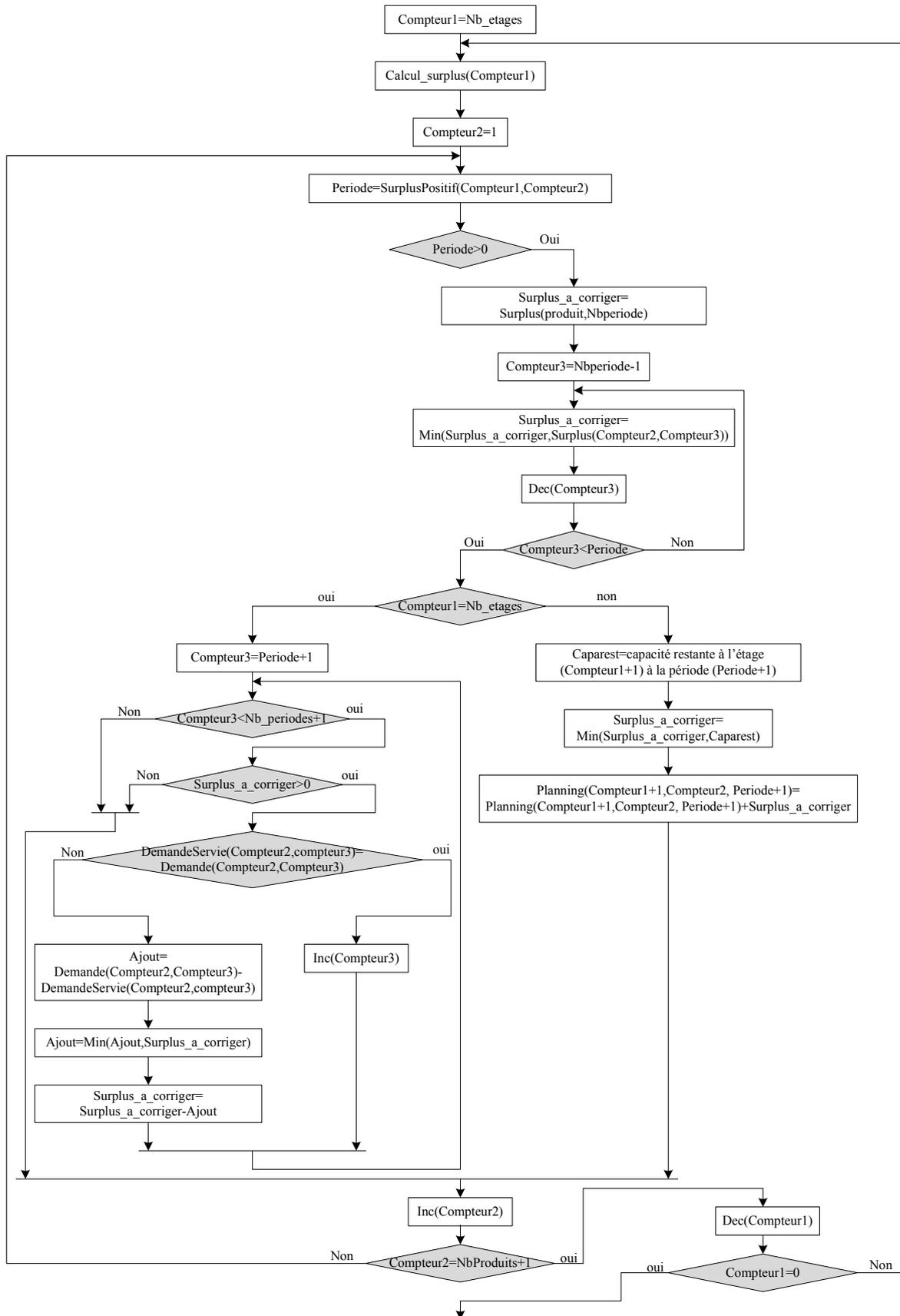


Figure 7.16 – Procédure 10

