



HAL
open science

Optimisation intégrée des décisions en planification et ordonnancement dans une chaîne logistique

Edwin David Gomez Urrutia

► **To cite this version:**

Edwin David Gomez Urrutia. Optimisation intégrée des décisions en planification et ordonnancement dans une chaîne logistique. Autre. Ecole Nationale Supérieure des Mines de Saint-Etienne, 2014. Français. NNT : 2014EMSE0744 . tel-01089338

HAL Id: tel-01089338

<https://theses.hal.science/tel-01089338>

Submitted on 1 Dec 2014

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.



NNT : 2014 EMSE 0744

THÈSE

présentée par

Edwin David GOMEZ URRUTIA

pour obtenir le grade de

Docteur de l'École Nationale Supérieure des Mines de Saint-Étienne

Spécialité : **Génie Industriel**

OPTIMISATION INTÉGRÉE DES DÉCISIONS EN PLANIFICATION ET ORDONNANCEMENT DANS UNE CHAÎNE LOGISTIQUE

soutenue à Gardanne, le 12 juin 2014

Membres du jury

Rapporteurs	Bernard Penz Safia Kedad-Sidhoum	Professeur, Institut Polytechnique de Grenoble Maître de Conférences, Université Pierre et Marie Curie
Examineurs	Jean-Bernard Lasserre Laurence Wolsey Christelle Jussien-Guéret Nabil Absi	Directeur de Recherche, LAAS-CNRS Professeur, Université Catholique de Louvain Professeur, Université d'Angers Chargé de Recherche, École des Mines de Saint-Étienne
Directeur de thèse	Stéphane Dauzère-Pérès	Professeur, École des Mines de Saint-Étienne
Encadrant	Riad Aggoune	R&D Manager, CRP Henri Tudor

Spécialités doctorales	Responsables :	Spécialités doctorales	Responsables
SCIENCES ET GENIE DES MATERIAUX MECANIQUE ET INGENIERIE GENIE DES PROCEDES SCIENCES DE LA TERRE SCIENCES ET GENIE DE L'ENVIRONNEMENT	K. Wolski Directeur de recherche S. Drapier, professeur F. Gruy, Maître de recherche B. Guy, Directeur de recherche D. Graillot, Directeur de recherche	MATHEMATIQUES APPLIQUEES INFORMATIQUE IMAGE, VISION, SIGNAL GENIE INDUSTRIEL MICROELECTRONIQUE	O. Roustant, Maître-assistant O. Boissier, Professeur J.C. Pinoli, Professeur A. Dolgui, Professeur S. Dauzere Peres, Professeur

EMSE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

ABSI	Nabil	CR		CMP
AVRIL	Stéphane	PR2	Mécanique et ingénierie	CIS
BALBO	Flavien	PR2		FAYOL
BASSEREAU	Jean-François	PR		SMS
BATTON-HUBERT	Mireille	PR2	Sciences et génie de l'environnement	FAYOL
BERGER DOUCE	Sandrine	PR2		FAYOL
BERNACHE-ASSOLLANT	Didier	PR0	Génie des Procédés	CIS
BIGOT	Jean Pierre	MR(DR2)	Génie des Procédés	SPIN
BILAL	Essaid	DR	Sciences de la Terre	SPIN
BOISSIER	Olivier	PR1	Informatique	FAYOL
BORBELY	Andras	MR(DR2)	Sciences et génie des matériaux	SMS
BOUCHER	Xavier	PR2	Génie Industriel	FAYOL
BRODHAG	Christian	DR	Sciences et génie de l'environnement	FAYOL
BRUCHON	Julien	MA(MDC)	Mécanique et ingénierie	SMS
BURLAT	Patrick	PR2	Génie Industriel	FAYOL
COURNIL	Michel	PR0	Génie des Procédés	DIR
DARRIEULAT	Michel	IGM	Sciences et génie des matériaux	SMS
DAUZERE-PERES	Stéphane	PR1	Génie Industriel	CMP
DEBAYLE	Johan	CR	Image Vision Signal	CIS
DELAFOSSÉ	David	PR1	Sciences et génie des matériaux	SMS
DESRAYAUD	Christophe	PR2	Mécanique et ingénierie	SMS
DOLGUI	Alexandre	PR0	Génie Industriel	FAYOL
DRAPIER	Sylvain	PR1	Mécanique et ingénierie	SMS
FEILLET	Dominique	PR2	Génie Industriel	CMP
FEVOTTE	Gilles	PR1	Génie des Procédés	SPIN
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GARCIA	Daniel	MR(DR2)	Génie des Procédés	SPIN
GERINGER	Jean	MA(MDC)	Sciences et génie des matériaux	CIS
GOEURIOT	Dominique	DR	Sciences et génie des matériaux	SMS
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR1	Génie des Procédés	SPIN
GUY	Bernard	DR	Sciences de la Terre	SPIN
HAN	Woo-Suck	CR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR1	Génie des Procédés	SPIN
KERMOUCHE	Guillaume	PR2	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFORÉST	Valérie	MR(DR2)	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	CR	Mécanique et ingénierie	FAYOL
LI	Jean-Michel		Microélectronique	CMP
MALLIARAS	Georges	PR1	Microélectronique	CMP
MOLIMARD	Jérôme	PR2	Mécanique et ingénierie	CIS
MONTHEILLET	Frank	DR	Sciences et génie des matériaux	SMS
MOUTTE	Jacques	CR	Génie des Procédés	SPIN
NEUBERT	Gilles			FAYOL
NIKOLOVSKI	Jean-Pierre			CMP
NORTIER	Patrice	PR1		SPIN
PIJOLAT	Christophe	PR0	Génie des Procédés	SPIN
PIJOLAT	Michèle	PR1	Génie des Procédés	SPIN
PINOLI	Jean Charles	PR0	Image Vision Signal	CIS
POURCHEZ	Jérémy	CR	Génie des Procédés	CIS
ROBISSON	Bruno			CMP
ROUSSY	Agnès	MA(MDC)		CMP
ROUSTANT	Olivier	MA(MDC)		FAYOL
ROUX	Christian	PR		CIS
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
TRIA	Assia	Ingénieur de recherche	Microélectronique	CMP
VALDIVIESO	François	MA(MDC)	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	MR(DR2)	Génie des Procédés	SPIN
WOLSKI	Krzysztof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR1	Génie industriel	CIS
YUGMA	Gallian	CR	Génie industriel	CMP
ENISE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)				
BERGHEAU	Jean-Michel	PU	Mécanique et Ingénierie	ENISE
BERTRAND	Philippe	MCF	Génie des procédés	ENISE
DUBUJET	Philippe	PU	Mécanique et Ingénierie	ENISE
FEULVARCH	Eric	MCF	Mécanique et Ingénierie	ENISE
FORTUNIER	Roland	PR	Sciences et Génie des matériaux	ENISE
GUSSAROV	Andrey	Enseignant contractuel	Génie des procédés	ENISE
HAMDI	Hédi	MCF	Mécanique et Ingénierie	ENISE
LYONNET	Patrick	PU	Mécanique et Ingénierie	ENISE
RECH	Joël	PU	Mécanique et Ingénierie	ENISE
SMUROV	Igor	PU	Mécanique et Ingénierie	ENISE
TOSCANO	Rosario	PU	Mécanique et Ingénierie	ENISE
ZAHOUANI	Hassan	PU	Mécanique et Ingénierie	ENISE

À ma mère et Ana Gisselle

Remerciements

Je tiens d'abord à remercier le Fonds National de la Recherche (FNR) du Luxembourg pour avoir soutenu cette thématique de recherche et avoir permis une collaboration entre le Centre de Recherche Public Henri Tudor du Luxembourg et l'École des Mines de Saint-Étienne, que je remercie également de m'avoir donné l'opportunité de m'épanouir dans le monde de la recherche.

J'adresse mes remerciements à Madame Safia Kedad-Sidhoum et Monsieur Bernard Penz pour avoir accepté d'être rapporteurs de ce travail. Je tiens également à remercier Monsieur Laurence Wolsey, Monsieur Jean-Bernard Lasserre, Madame Christelle Jussien-Guéret et Monsieur Nabil Absi, qui m'ont fait l'honneur de participer au jury de thèse en tant qu'examinateurs.

Mes plus vifs remerciements vont à Monsieur Stéphane Dauzère-Pères pour avoir encadré ce travail en tant que Directeur de thèse. Ses idées ont grandement enrichi les contributions de cette thèse. Son grand savoir m'a beaucoup apporté. J'adresse aussi mes plus sincères remerciements à Monsieur Riad Aggoune pour avoir co-encadré cette thèse. Ses conseils, sa disponibilité, ses encouragements et ses critiques constructives m'ont été d'une aide précieuse. J'apprécie énormément l'amitié qu'il m'a offerte. Je dois dire que la qualité de ce travail est en grande partie due au bon esprit de travail avec mes encadrants.

Je tiens à remercier mes collègues de l'unité KISS du CRP Henri Tudor, avec qui j'ai passé des moments très agréables. Je remercie particulièrement Yves Rangoni, qui m'a aidé sur le plan technique, en répondant à mes questions quotidiennes (et parfois embêtantes...) de LATEX et de français, qui a su construire une très bonne ambiance dans notre groupe travail et que je compte aujourd'hui parmi mes amis. Je tiens aussi à remercier très spécialement Younès Djaghloul, personnage mémorable du bureau pour sa grande sympathie. Son aide sur l'aspect informatique, ses conseils, ses encouragements et son amitié me sont d'une valeur inestimable.

Je souhaite aussi remercier tous les membres de l'équipe SFL du CMP Georges Charpak pour leur accueil et l'excellente ambiance qui régnait lors de chacune de mes visites à Gardanne. Le cadre de travail est magnifique et encore plus la qualité humaine des collaborateurs. Même si je n'ai pas pu y aller très souvent, j'en garderai toujours un très bon souvenir.

Enfin, je remercie ma famille, et très spécialement ma mère, qui malgré la distance m'a toujours apporté son soutien. Je remercie également Ana Gisselle pour son soutien et sa compagnie pendant ces années.

Table des matières

Introduction Générale	1
1 Gestion de la chaîne logistique	5
1.1 Introduction	6
1.2 Description de la chaîne logistique	7
1.2.1 Niveaux de décisions en la planification de la chaîne logistique	8
1.2.2 Activités de la chaîne logistique	9
1.3 Systèmes de production	10
1.3.1 Systèmes à flux continus	13
1.3.2 Systèmes à flux discrets	14
1.4 Planification de la production	16
1.4.1 Progiciels de Gestion Intégrée (ERP)	17
1.4.2 Planification des ressources de production (MRP-II)	19
1.4.3 Systèmes de planification avancée (APS)	24
1.5 Ordonnancement de la production	26
1.5.1 Configurations d'ateliers de production	27
1.5.2 Contraintes d'ordonnancement	29
1.5.3 Critères d'optimisation	31
1.6 Conclusion	34
2 État de l'art en planification et ordonnancement	37
2.1 Introduction	38
2.2 Généralités sur les problèmes d'optimisation	38
2.2.1 Complexité	39

2.2.2	Modélisation	41
2.2.3	Méthodes de résolution	42
2.3	Problèmes de dimensionnement de lots	51
2.3.1	Problèmes à courtes périodes	56
2.3.2	Problèmes à longues périodes	60
2.4	Problèmes d'ordonnancement	66
2.5	Intégration de la planification et de l'ordonnancement	68
2.5.1	Approches itératives	71
2.5.2	Approches intégrées	78
2.6	Conclusion	82
3	Approche intégrée pour des problèmes à un niveau	85
3.1	Introduction	86
3.2	Modélisation	87
3.2.1	Description du problème	87
3.2.2	Modèle intégré	88
3.2.3	Modèle avec chemins et séquence fixée	91
3.3	Méthode de résolution	96
3.3.1	Heuristique Lagrangienne pour la résolution du problème à séquence fixée	97
3.3.2	Méthode d'amélioration de la séquence	102
3.4	Analyses et améliorations	105
3.4.1	Résolution du problème à séquence fixée	105
3.4.2	Amélioration de la séquence	113
3.5	Résultats expérimentaux	120
3.5.1	Expérimentations avec un atelier de type job-shop 6x6	123
3.5.2	Expérimentations avec un atelier de type job-shop 10x10	125
3.5.3	Expérimentations avec un atelier de type job-shop 20x5	128
3.6	Conclusion	130
4	Approche intégrée pour des problèmes à plusieurs niveaux	133
4.1	Introduction	134
4.2	Évolution du modèle	134

4.2.1	Description du problème	134
4.2.2	Modèle intégré avec notation classique	140
4.2.3	Modèle intégré avec échelon stock	142
4.2.4	Modèle avec chemins, échelon stock et séquence fixée	144
4.3	Méthode de résolution	144
4.3.1	Heuristique Lagrangienne pour la résolution du problème avec séquence fixée	145
4.3.2	Méthode d'amélioration de la séquence	157
4.4	Résultats expérimentaux	158
4.4.1	Expérimentations avec des ateliers de type job-shop 6x6	158
4.4.2	Expérimentations avec des ateliers de type job-shop 10x10	176
4.5	Conclusion	190
Conclusion Générale et Perspectives		193
A Fonction objectif Lagrangienne du problème à plusieurs niveaux		197
B Expérimentations avec l'algorithme de lissage descendant		203
B.1	Nomenclature 6x6-2	203
B.2	Nomenclature 6x6-3	204
B.3	Nomenclature 6x6-4	205
B.4	Nomenclature 10x10-1	207
B.5	Nomenclature 10x10-2	208
B.6	Nomenclature 10x10-3	209
C Intégration de la planification de la distribution		211
D Prise en compte de l'aspect environnemental		229
Bibliographie		258

Liste des tableaux

3.1	Prédécesseurs et successeurs dans le graphe conjonctif	93
3.2	Comparaison de résultats avec différents voisinages	117
3.3	Compromis entre diversification et intensification	119
3.4	Problèmes étudiés	121
3.5	Paramètres généraux pour la génération d'instances	121
3.6	Résultats pour le job-shop 6x6	124
3.7	Écarts relatifs pour le job-shop 6x6 avec 60 secondes de temps de calcul	125
3.8	Écarts relatifs pour le job-shop 6x6 avec 300 secondes de temps de calcul	126
3.9	Résultats pour le job-shop 10x10	127
3.10	Écarts relatifs pour le job-shop 10x10	128
3.11	Résultats pour le job-shop 20x5	129
3.12	Écarts relatifs pour le job-shop 20x5	130
4.1	Paramètres généraux associés aux ateliers de type job-shop 6x6	159
4.2	Solutions obtenues avec IBM ILOG CPLEX pour la nomenclature 6x6-1	159
4.3	Résultats pour la nomenclature 6x6-1 avec l'algorithme glouton	160
4.4	Résultats pour la nomenclature 6x6-1 avec l'algorithme descendant	162
4.5	Résultats pour la nomenclature 6x6-1 avec l'algorithme ascendant	163
4.6	Solutions obtenues avec IBM ILOG CPLEX pour la nomenclature 6x6-2	165
4.7	Résultats pour la nomenclature 6x6-2 avec l'algorithme glouton	166
4.8	Résultats pour la nomenclature 6x6-2 avec l'algorithme ascendant	167
4.9	Solutions obtenues avec IBM ILOG CPLEX pour la nomenclature 6x6-3	168
4.10	Résultats pour la nomenclature 6x6-3 avec l'algorithme glouton	169

4.11	Résultats pour la nomenclature 6x6-3 avec l'algorithme ascendant	170
4.12	Solutions obtenues avec IBM ILOG CPLEX pour la nomenclature 6x6-4	171
4.13	Résultats pour la nomenclature 6x6-4 avec l'algorithme glouton	172
4.14	Résultats pour la nomenclature 6x6-4 avec l'algorithme ascendant	173
4.15	Paramètres généraux associés aux ateliers de type job-shop 10x10	177
4.16	Solutions obtenues avec IBM ILOG CPLEX pour la nomenclature 10x10-1	178
4.17	Résultats pour la nomenclature 10x10-1 avec l'algorithme glouton	179
4.18	Résultats pour la nomenclature 10x10-1 avec l'algorithme ascendant	180
4.19	Solutions obtenues avec IBM ILOG CPLEX pour la nomenclature 10x10-2	181
4.20	Résultats pour la nomenclature 10x10-2 avec l'algorithme glouton	182
4.21	Résultats pour la nomenclature 10x10-2 avec l'algorithme ascendant	183
4.22	Solutions obtenues avec IBM ILOG CPLEX pour la nomenclature 10x10-3	184
4.23	Résultats pour la nomenclature 10x10-3 avec l'algorithme glouton	185
4.24	Résultats pour la nomenclature 10x10-3 avec l'algorithme ascendant	186
B.1	Résultats pour la nomenclature 6x6-2 avec l'algorithme descendant	204
B.2	Résultats pour la nomenclature 6x6-3 avec l'algorithme descendant	205
B.3	Résultats pour la nomenclature 6x6-4 avec l'algorithme descendant	206
B.4	Résultats pour la nomenclature 10x10-1 avec l'algorithme descendant	207
B.5	Résultats pour la nomenclature 10x10-2 avec l'algorithme descendant	208
B.6	Résultats pour la nomenclature 10x10-3 avec l'algorithme descendant	210

Table des figures

1.1	La chaîne logistique [165]	7
1.2	Exemple de nomenclature à plusieurs niveaux	11
1.3	Types de systèmes de production [178]	13
1.4	Planification de la production et de la capacité avec MRP-II	20
1.5	Matrice de planification de la chaîne logistique [72]	24
2.1	Relation entre les principales classes de complexité	40
2.2	Exemple de structures de setups complexes	53
2.3	Classification des problèmes de dimensionnement de lots à un seul niveau [233]	54
2.4	Classification des problèmes de dimensionnement de lots à plusieurs niveaux [233]	54
2.5	Stratégies de résolution du problème intégré de planification et d'ordonnancement [155]	70
2.6	Approche itérative de Lasserre [144]	71
2.7	Approche avec horizon glissant [152]	75
2.8	Approche avec relaxation Lagrangienne augmentée [151]	78
2.9	Évolution de l'approche de Lasserre [144]	80
3.1	Exemple d'un graphe disjonctif et d'un graphe conjonctif associé pour un atelier de type job-shop avec 3 jobs et 3 machines	88
3.2	Graphe conjonctif pour un atelier de type job-shop avec 3 produits, 3 machines 3 périodes	93
3.3	Approche intégrée	97
3.4	Heuristique de lissage de la production	101
3.5	Méthode d'amélioration de la séquence	103

3.6	Influence du nombre maximum d'itérations de la relaxation Lagrangienne sur le résultat final	107
3.7	Heuristique de lissage gloutonne	109
3.8	Heuristique Lagrangienne de Wolosewicz <i>et al.</i> [235]	111
3.9	Heuristique Lagrangienne modifiée	113
3.10	Méthode d'amélioration de la séquence dans l'approche de Wolosewicz <i>et al.</i>	114
3.11	Résultats expérimentaux avec plusieurs tailles de liste taboue	118
4.1	Exemple de nomenclature	138
4.2	Heuristique Lagrangienne pour des problèmes de dimensionnement de lots à plusieurs niveaux	146
4.3	Stratégie de lissage SP1	149
4.4	Stratégie de lissage SP2	149
4.5	Stratégie de lissage SP3	150
4.6	Stratégie de lissage SP4	150
4.7	Stratégie de lissage SP5	150
4.8	Algorithme glouton pour satisfaire les contraintes de nomenclature	152
4.9	Algorithme descendant pour satisfaire les contraintes de nomenclature	155
4.10	Algorithme ascendant pour satisfaire les contraintes de nomenclature	156
4.11	Nomenclature 6x6-1	159
4.12	Nomenclature 6x6-2	165
4.13	Nomenclature 6x6-3	168
4.14	Nomenclature 6x6-4	171
4.15	Synthèse des résultats pour les nomenclatures 6x6 avec l'algorithme glouton	174
4.16	Synthèse des résultats pour les nomenclatures 6x6 avec l'algorithme descendant	175
4.17	Synthèse des résultats pour les nomenclatures 6x6 avec l'algorithme ascendant	176
4.18	Comparaison des algorithmes de validation des contraintes de nomenclature pour les nomenclatures 6x6	177
4.19	Nomenclature 10x10-1	178
4.20	Nomenclature 10x10-2	181
4.21	Nomenclature 10x10-3	184
4.22	Synthèse des résultats pour les nomenclatures 10x10 avec l'algorithme glouton	188

4.23 Synthèse des résultats pour les nomenclatures 10x10 avec l'algorithme descendant	188
4.24 Synthèse des résultats pour les nomenclatures 10x10 avec l'algorithme ascendant	189
4.25 Comparaison des algorithmes de validation des contraintes de nomenclature pour les nomenclatures 10x10	190
C.1 Exemple de réseau multi-sources et multi-sites	212
C.2 Exemple de nomenclature de produits pour un chaîne logistique multi-sources et multi-sites	212

Introduction Générale

De nos jours, le management de la chaîne logistique est une pratique fondamentale. La forte concurrence et la globalisation des marchés dans des secteurs industriels divers obligent les entreprises à adopter une stratégie de management globale, qui privilégie la recherche de décisions optimales pour l'ensemble de la chaîne logistique, plutôt que les intérêts individuels des différents acteurs. Mais ce besoin de management collaboratif n'est pas le seul constat. Une faiblesse importante en gestion de la chaîne logistique est la structure du processus décisionnel, qui suit régulièrement une logique hiérarchique organisée en trois niveaux : stratégique, tactique et opérationnel. Cette hiérarchisation contraint l'ensemble des décisions possibles à prendre, à moyen et à court terme, et conduit vers une planification par étapes séparées, certaines subordonnées à d'autres, empêchant souvent l'obtention de solutions cohérentes aux problèmes courants de la chaîne logistique tels que la planification de la production et de la distribution.

La gestion de la production constitue l'une des activités les plus importantes dans le management de tout système manufacturier, et globalement dans le management de la chaîne logistique. La réussite d'une activité économique dépend grandement d'une gestion efficace de la production, qui permet de satisfaire pleinement les besoins du marché tout en minimisant les coûts de gestion. Cette tâche n'est pas toujours simple à effectuer et ceci pour deux raisons principales : (i) la hiérarchisation des décisions ne garantit pas une cohérence entre les niveaux stratégique, tactique et opérationnel et, (ii) les problèmes sont si complexes qu'il est très difficile, avec les capacités de calcul actuellement disponibles, de gérer toutes les données et les contraintes associées au système de production. Par conséquent, il est souvent impossible de mettre en œuvre à court terme les décisions prises à moyen et à long terme. C'est pourquoi la gestion de la production représente un grand challenge qui n'est pas toujours efficacement mené par les progiciels de gestion intégrée (ERP) ni par les systèmes de planification avancée (APS).

Dans cette thèse, nous nous intéressons à l'intégration des niveaux de décision tactique et opérationnel, et plus particulièrement aux problèmes de planification et d'ordonnancement de la production. Ces deux problèmes ont été largement étudiés dans la littérature de manière séparée, mais très peu de travaux intégrant les deux problématiques d'une manière cohérente

et adaptée à des problèmes complexes ont été menés. Ainsi, les plans de production tactiques sont souvent irréalisables au niveau ordonnancement, c'est-à-dire que les ordres de fabrication ne peuvent pas être complétés dans les délais avec la capacité de production disponible du système. Pour répondre à ce problème, nous proposons une approche intégrée qui permet de prendre les décisions de planification et d'ordonnancement simultanément, et qui est capable de déterminer des solutions réalisables de bonne qualité, pour des problèmes de production complexes, en un temps de calcul raisonnable.

Cette thèse comporte 4 chapitres organisés de la manière suivante. Dans le Chapitre 1, nous présentons les principes et pratiques du management de la chaîne logistique, en mettant l'accent sur l'étape de production. Le but est de décrire le contexte autour du problème intégré de planification et d'ordonnancement de la production.

L'intégration de la planification et de l'ordonnancement fait l'objet d'étude du Chapitre 2, où nous présentons un état de l'art sur les travaux qui s'intéressent à la cohérence des décisions en planification et ordonnancement de la production. Nous mettons en évidence la difficulté associée à la mise en place des approches de résolution et présentons les problèmes couramment étudiés et les techniques d'optimisation les plus utilisées dans ce domaine. Un intérêt particulier est consacré aux approches intégrant les décisions de planification et d'ordonnancement qui permettent de résoudre des problèmes dans des ateliers de production complexes. Nous classifions ces méthodes en approches intégrées et approches itératives et distinguons les problèmes à un seul niveau des problèmes à plusieurs niveaux. Les premiers correspondent à des systèmes de production, où les flux de matières entre les produits sont indépendants. En revanche, les deuxièmes correspondent à des systèmes de production où des nomenclatures définissent la composition d'un produit à partir d'autres produits. Ce scénario est fréquemment rencontré dans l'industrie et peut être étendu au cas d'un problème avec plusieurs sites de production qui participent à la fabrication d'un produit fini commun.

Dans le Chapitre 3, nous présentons une approche intégrée pour résoudre le problème mono-niveau de planification et d'ordonnancement, dans des systèmes complexes. Cette approche s'inscrit dans la lignée des travaux commençant par l'approche de Lasserre [144] et finissant par l'approche de Wolosewicz *et al.* [233, 234, 235].

Dans le Chapitre 4, nous proposons une nouvelle approche intégrée pour la résolution efficace de problèmes avec nomenclatures à plusieurs niveaux. Comme dans le cas mono-niveau, nous proposons une modélisation du problème avant de détailler la méthode de résolution. Enfin, nous présentons les résultats de nombreuses expérimentations numériques qui permettent de montrer l'efficacité des approches intégrées.

INTRODUCTION GÉNÉRALE

Nous terminons ce manuscrit par une conclusion générale et des perspectives pour des travaux futurs.

Chapitre 1

Gestion de la chaîne logistique

Ce chapitre offre une vue globale sur les principales caractéristiques de la gestion de la chaîne logistique. Les différents niveaux de décisions (stratégique, tactique et opérationnel) et les différentes étapes dans la planification de la chaîne logistique sont décrits. Une classification des systèmes de production est présentée, et une description plus détaillée sur la planification et l'ordonnancement de la production, qui font l'objet d'étude de cette thèse, est mise en avant.

[1.1 Introduction](#)

[1.2 Description de la chaîne logistique](#)

[1.3 Systèmes de production](#)

[1.4 Planification de la production](#)

[1.5 Ordonnancement de la production](#)

[1.6 Conclusion](#)

1.1 Introduction

Dans ce chapitre, nous décrivons les principaux aspects de la gestion de la chaîne logistique, en mettant l'accent sur les étapes de planification et d'ordonnancement de la production. Dans la plupart des approches et des systèmes d'information pour la gestion de la chaîne logistique, les différentes étapes de planification sont étudiées de manière séparée. Le but de ce chapitre est d'expliquer, dans un premier temps, la structure décisionnelle de la chaîne logistique, à travers laquelle les différentes étapes sont liées, dans des échelles de temps différentes, par des flux d'informations et de matières.

La coordination des flux logistiques est primordiale pour assurer une gestion efficace de la chaîne d'approvisionnement globale. Cependant, il est souvent difficile de trouver un compromis entre tous les participants de la chaîne, de telle sorte que le bénéfice soit global et équitable, puisque chaque entité cherche à maximiser son propre bénéfice, sans tenir compte de la performance globale. Mais le challenge n'est pas seulement associé à la collaboration entre les différents participants de la chaîne logistique. Le problème dans la synchronisation de décisions est aussi présent au niveau interne. Pour améliorer la performance à moyen et long terme ou simplement, pour éviter la faillite, une entreprise doit tenir compte des contraintes internes et externes concernant les différentes étapes de planification de la chaîne logistique. Nous discutons plus en détail ces aspects dans la Section 1.2, où nous décrivons le rôle des participants, la structure décisionnelle, et les principales activités liant les flux de matières et d'informations.

Un facteur important à prendre en compte lorsque l'on analyse les possibles stratégies de gestion de la chaîne logistique est de connaître la nature du système. L'activité économique peut être transformatrice de produits ou de services, ou les deux. Nous nous focalisons dans ce travail sur les systèmes de production, et plus particulièrement sur les systèmes de production à flux discrets. Une classification des systèmes manufacturiers est présentée dans la Section 1.3.

De la même manière, parmi les activités impliquées dans la gestion de la chaîne logistique, nous nous intéressons particulièrement dans ce travail à la production. Donc, nous présentons en détail, dans les Sections 1.4 et 1.5, les étapes de management de la production, notamment la planification et l'ordonnancement. Nous expliquons les méthodes de gestion de la production couramment utilisées et leurs limitations. Finalement, dans la Section 1.6, nous présentons les conclusions de ce chapitre.

1.2 Description de la chaîne logistique

La chaîne logistique est un système intégrant un ensemble d'entreprises, de personnes, de machines, d'outils, de technologies, d'informations, etc., où le but est de satisfaire les demandes des clients finaux, à travers la transformation de matière, ayant pour point d'origine les fournisseurs de matières premières. Une chaîne logistique est composée principalement de fournisseurs, de fabricants, de distributeurs, de détaillants et de consommateurs.

Ces participants sont liés par des flux de matières et d'informations, comme illustré dans la Figure 1.1. Le flux d'informations démarre chez les clients avec les spécifications de la demande, qui est transmise aux détaillants (des commerçants de produits à l'unité, par petit volume). Étant donné que les détaillants ne sont pas des producteurs, ils ont besoin d'approvisionner périodiquement leurs stocks avec les produits demandés par les clients. Ce sont les distributeurs qui assurent cet approvisionnement, en suivant la demande transmise par les détaillants. De la même manière, les distributeurs envoient de l'information aux fabricants avec la demande de produits nécessaires pour approvisionner les détaillants. Finalement, compte tenu de la demande de produit fini réalisée par les distributeurs, les fabricants transmettent une demande de matière première et/ou en composants aux fournisseurs, afin de pouvoir fabriquer les produits.

Dans le sens inverse, le flux de matières démarre chez les fournisseurs. Ces matières sont transformées en produits finis par les fabricants et elles sont récupérées (souvent aussi stockées) par les distributeurs, qui les distribuent aux détaillants, pour être finalement achetées par les clients finaux. Le terme de *logistique interne* est souvent associé au management des flux entre les fournisseurs de matières premières ou composants et les fabricants; tandis que la gestion des flux à partir des fabricants jusqu'aux clients finaux est souvent appelée *logistique externe* [43].

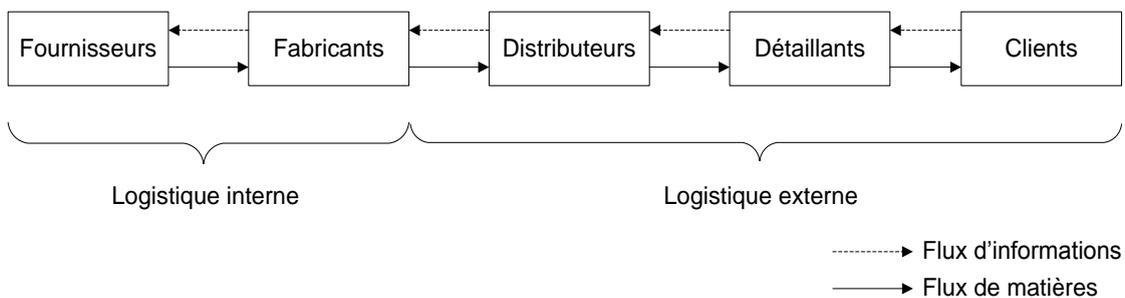


FIGURE 1.1 – La chaîne logistique [165]

Chaque groupe de participants du même maillon peut aussi constituer une chaîne logistique, faisant partie de la chaîne logistique globale. Ainsi par exemple, il peut y avoir plusieurs fabricants intégrant la même chaîne, où chaque usine produit des composants pour les autres usines et en même temps des produits pour les clients finaux. De manière similaire, les centres de distribution peuvent constituer un réseau, permettant la circulation et l'entreposage de produits entre eux, afin de réduire les coûts de transport et de stockage, ainsi que les délais de livraison.

En suivant le flux de matière, nous pouvons établir que chaque participant a des fournisseurs et des clients. Ainsi, les fournisseurs des distributeurs sont les fabricants, et ses clients sont les détaillants. En pratique, les entreprises ne suivent pas une stratégie de management globale de la chaîne d'approvisionnement, et considèrent seulement leurs fournisseurs et leurs clients directs pour gérer les opérations logistiques [57]. Ceci entraîne la perte d'information, circulant à partir des consommateurs finaux, et se traduit en une visibilité limitée du marché et des capacités de production, de stockage et de distribution des autres participants. Ceci peut faire varier les temps d'approvisionnement et peut engendrer des ruptures de stock (et potentiellement la perte de clients) ou un sur-stockage indésirable, incrémentant les coûts de gestion. Ce phénomène de perturbation des décisions en management de la chaîne logistique et connu sous le nom d'« *Effet coup de fouet* » [145].

Une stratégie de management global de la chaîne logistique implique la planification synchronisée des activités liant les flux d'information et de matière sur un horizon à long terme. Les activités de planification d'une chaîne logistique sont organisées en pratique en trois niveaux de décisions, à savoir : le niveau stratégique, le niveau tactique et le niveau opérationnel [209].

1.2.1 Niveaux de décisions en la planification de la chaîne logistique

Le **niveau stratégique** comprend les décisions qui ont un impact sur le long terme (plusieurs années). Ces décisions sont prises au plus haut niveau de l'entreprise et contraignent toutes les activités de planification aux niveaux tactique et opérationnel. Les décisions stratégiques incluent : la conception du réseau logistique, la conception du produit et les offres commerciales et la planification des ventes. La conception du réseau logistique consiste à :

- Déterminer le nombre d'usines, leur localisation et leur taille.
- Déterminer le nombre d'entrepôts, leur localisation et leur taille.
- Déterminer la nature du système de production et le nombre et type de ressources nécessaires (opérateurs, machines, outils, etc.)
- Identifier et sélectionner les fournisseurs, les distributeurs, les grossistes, les détaillants et les autres participants de la chaîne logistique.

Le **niveau tactique** est basé sur un horizon à moyen terme, de l'ordre de quelques mois, et consiste à planifier les activités d'approvisionnement, de production, de distribution et des ventes, de façon à atteindre les objectifs fixés dans le plan stratégique. Il s'agit de définir les dates d'approvisionnement, les quantités et périodes de production et moyens de distribution, pour assurer la satisfaction de la demande dans les délais, à un coût minimal. Les activités tactiques sont contraintes par la capacité des ressources.

Le **niveau opérationnel** comprend les activités à court terme (de l'ordre de quelques jours ou semaines), i.e. les activités d'exécution. Ceci correspond à l'achat de matière première et des composants, à l'affectation de tâches de production (opérations) aux machines, à la détermination de tournées de transports et l'entreposage et à l'élaboration d'un plan détaillé de ventes.

Une limite importante en management de la chaîne logistique est que les différents niveaux de décisions sont le plus souvent traités de manière séparée et hiérarchique, où les objectifs du niveau supérieur deviennent des contraintes pour le niveau immédiatement inférieur. Les contraintes du niveau inférieur sont habituellement mal considérées dans le niveau supérieur, ce qui fait que les solutions prises au niveau supérieur ne sont pas forcément réalisables. Nous en discutons plus en détail, en mettant l'accent sur l'étape de production, dans la Section 1.4.

Il existe aujourd'hui des logiciels de planification qui offrent un bon degré de cohérence dans la synchronisation des flux logistiques. Ce sont les Systèmes de Planification Avancée (APS), dont nous discutons dans la Section 1.4.3. Bien que les niveaux de décisions continuent à être traités de manière hiérarchique, le partage d'information entre eux permet dans la plupart des cas l'obtention de solutions réalisables [51]. La qualité des décisions peut, néanmoins, être considérablement améliorée, en intégrant les niveaux de décisions.

1.2.2 Activités de la chaîne logistique

De manière transversale aux niveaux de décisions, les différentes activités de transformation des flux de matières et d'informations sont menées. Ces étapes font le lien entre les participants de la chaîne logistique et sont principalement : l'approvisionnement, la production, la distribution et les ventes [128]. L'**approvisionnement** est l'étape qui consiste à approvisionner de les matières premières nécessaires pour fabriquer les produits, selon les spécifications indiquées dans le plan de production. Il faut que l'approvisionnement permette de respecter les quantités et les périodes de production décidées. L'approvisionnement est donc l'étape faisant le lien entre les fournisseurs et les sites de production. Lorsque l'on définit le plan d'approvisionnement, on doit tenir compte de la capacité de production des fournisseurs et des délais d'obtention et de

livraison. Dans une chaîne logistique comptant plusieurs fournisseurs, les facteurs différenciateurs au moment de choisir le fournisseur d'un produit i à un instant t sont le prix de vente et les délais de production et de livraison, sans oublier la qualité des matières premières (et/ou composants) qui peut varier d'un fournisseur à un autre.

La **production** est l'étape qui consiste en la fabrication de produit fini, à travers la transformation de la matière première issue de l'approvisionnement. Cette transformation peut être faite en une ou plusieurs opérations, et chaque opération implique la création de valeur ajoutée. Les produits finis doivent satisfaire les besoins des clients, à savoir : fonctionnalité, qualité et délais d'obtention. Les décisions à prendre dans cette étape sont : combien, quand et comment produire. Ces décisions ont pour but de satisfaire pleinement la demande client au moindre coût. Nous discutons ces facteurs qui ont une incidence sur les politiques de production dans la Section 1.3.

La **distribution** est l'activité permettant de mettre à disposition des consommateurs les produits élaborés par les fabricants. Elle est le lien entre les producteurs et les détaillants. Un distributeur doit récupérer, transporter, stocker et répartir, les produits fournis par les fabricants. À la fin, les produits doivent être à disposition des détaillants, en respectant les délais de livraison. L'objectif est de livrer tous les détaillants dans les délais, au moindre coût possible, en utilisant au mieux des ressources (véhicules) limitées. Le problème de distribution implique en particulier l'affectation des produits aux véhicules et la sélection des tournées, avec l'objectif de minimiser le nombre de véhicules et/ou de minimiser la distance parcourue.

Finalement, les **ventes** représentent l'étape où les produits sont acquis par les clients finaux. Ceci correspond donc au flux de matière entre les détaillants et les consommateurs. L'objectif est de satisfaire la demande au moindre coût. À ce stade, les coûts sont associés au management des stocks. Il s'agit donc de minimiser la somme des coûts de stockage et de rupture de stock. Dans certains secteurs économiques, comme l'alimentaire, il faut tenir compte de la péremption des produits et gérer les stocks de façon à réduire le nombre de produits avec date de péremption dépassée.

Pour plus de détails sur le management de la chaîne logistique, le lecteur peut consulter [210].

1.3 Systèmes de production

Un système de production est une organisation de personnes, machines, outils et d'autres composants, qui servent à transformer des matières premières en produits finis, destinés à la

consommation de clients finaux. La transformation d'un produit peut nécessiter une ou plusieurs opérations à réaliser sur une ou plusieurs machines ou stations de travail.

Si les produits finis sont fabriqués juste à partir de matières premières, mais ne requièrent pas l'assemblage ou l'intégration de composants, il s'agit d'un **système mono-niveau**. Si la fabrication des produits finis requiert l'inclusion d'autres produits (composants), le processus correspond à un **système multi-niveaux**. La différence entre ces deux types de systèmes est que dans le cas multi-niveaux, il existe une nomenclature ou *liste BOM* (Bill Of Materials) qui définit les besoins de production ou demandes internes, entre tous les produits (composants et produits finis) faisant partie du processus de fabrication ; tandis que dans les systèmes mono-niveau, chaque produit a une demande externe, mais il n'y a pas de demandes internes entre les produits. Le nombre de niveaux fait référence au plus grand nombre de liens de dépendance entre un composant basique (produit sans composants) et un produit fini, en passant par tous les composants intermédiaires. Un exemple de *nomenclature à plusieurs niveaux* est présenté dans la Figure 1.2. Dans les systèmes mono-niveau, on parle souvent de *nomenclature à un niveau*.

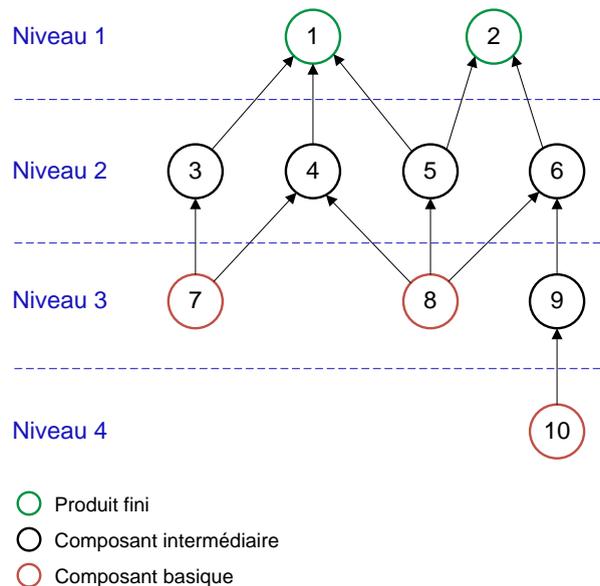


FIGURE 1.2 – Exemple de nomenclature à plusieurs niveaux

Dans la gestion d'un système de production, le but est de satisfaire la demande client au moindre coût possible. Pour ce faire, plusieurs politiques de production peuvent être implémentées, à savoir : la production à la demande, la production sur stock, l'assemblage à la demande et la conception à la demande [118].

La **production à la demande** consiste à ne déclencher la fabrication d'un article que lorsque la demande client est confirmée, ce qui permet de réduire les inventaires de produits semi-finis et de produits finis. Cette politique permet donc de diminuer les coûts de stockage. Pour plus de détails sur ce mode de production, le lecteur peut consulter [138].

La **production sur stock** est basée sur les prévisions de la demande, et consiste à démarrer le processus de fabrication avant d'avoir une demande confirmée. Ceci permet de réduire les délais de livraison (intervalle de temps entre la commande client et la livraison du produit) et les coûts de production.

L'**assemblage à la demande** consiste à construire un stock de produits semi-finis, en attendant l'arrivée de la demande. Une fois que la demande est confirmée, les produits sont assemblés et livrés aux consommateurs. C'est donc une stratégie qui combine les deux politiques précédentes : production sur stock pour obtenir des produits semi-finis et production à la demande pour réaliser l'assemblage final.

Dans la **conception à la demande**, une fois que la demande arrive, l'entreprise et le client conçoivent ensemble le produit, ce qui fait que chaque article fabriqué est unique. Une fois le produit conçu, la matière première est commandée, et une fois l'approvisionnement réalisé, la production et la distribution sont successivement enchaînées. À travers cette politique, il n'y a pas d'inventaires, mais il est difficile d'estimer les temps de production et les besoins en ressources. Par conséquent, il est compliqué de maîtriser la capacité du système et de respecter les délais d'obtention. Le système adoptant cette stratégie doit être flexible.

Dans un système de production, plusieurs politiques de production peuvent être implémentées. Cela dépend de la nature des produits, des contraintes du système de production, du comportement de la demande, des coûts de production et de stockage et des durées opératoires. Ainsi, dans un même atelier, certains produits peuvent suivre une politique de production sur stock, d'autres une production à la demande, et d'autres une combinaison de plusieurs politiques [118].

En gestion de stocks, il est commun d'utiliser les termes *flux poussé* et *flux tiré*, pour faire référence à la circulation de matières (en tant que matières premières, produits semi-finis et produits finis) dans les modes de production sur stock et de production à la demande, respectivement. Ainsi, dans un système à flux poussé, la matière est « poussée » tout au long du processus de fabrication étape par étape (une fois qu'une opération est terminée, l'opération suivante est déclenchée). D'autre part, dans un système à flux tiré, la demande client génère un besoin de produit fini, qui à la fois génère une demande de composants (chaque composant pouvant demander aussi d'autres composants), pour au final générer une demande de matière

première. Une fois que la demande de matière première arrive chez le fournisseur, le flux de matières est déclenché.

Un autre terme utilisé, lié au type de production *juste à temps* [118], est celui de *flux tendu*, qui consiste à générer le minimum de stock possible dans toutes les étapes de production. Il correspond souvent à un système suivant un flux tiré avec des lignes de production équilibrées.

Les systèmes de production sont classés en deux grands groupes : les systèmes à flux continus et les systèmes à flux discrets ou intermittents [171], comme détaillé sur la Figure 1.3.

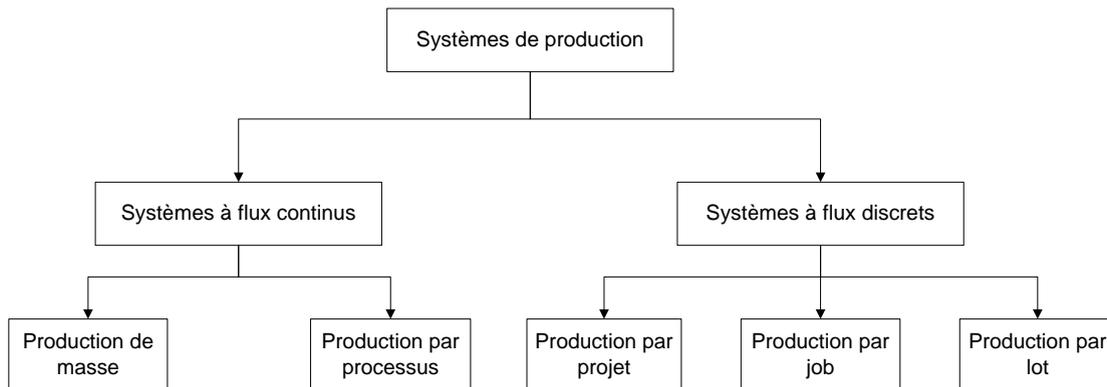


FIGURE 1.3 – Types de systèmes de production [178]

1.3.1 Systèmes à flux continus

Dans les systèmes de production à flux continus, les produits sont standardisés et circulent sans interruption tout au long de la ligne de production, le volume de production est normalement élevé, le taux de production est ajusté selon la prévision de la demande et le séquençement et l'ordonnancement sont standardisés (inchangéables). Les produits finis sont stockés dans des entrepôts, en attendant l'arrivée progressive de la demande. La politique de production suivie est celle de *production sur stock*.

Les systèmes à flux continus peuvent être classés en : *systèmes de production de masse* et *systèmes de production par processus*.

1.3.1.1 Production de masse

La production de masse est caractérisée par un gros volume de production, où des produits standard passent par différentes étapes ou opérations de manière continue. L'ordre de passage des produits sur les stations de travail est toujours le même et il n'y a quasiment pas

de temps d'attente, ce qui fait que l'inventaire de produits semi-finis est faible. Ce système est généralement composé des lignes d'assemblage et il est dans la plupart des cas complètement automatisé. Seules des opérations mineures sont effectuées par des opérateurs. Parmi les systèmes de production de masse, nous trouvons l'assemblage final d'automobiles, la production d'aliments et de boissons, d'essence, de médicaments et d'autres produits chimiques, etc.

1.3.1.2 Production par processus

La production par processus implique la transformation de composants en produits finis, à travers des réactions et des traitements irréversibles, qui suivent des recettes ou des formules spécifiques. En tenant compte que le système est dédié à la fabrication d'un produit en particulier, l'affectation des opérations aux ressources est fixe et le système reste très peu flexible. L'inventaire de produits semi-finis est presque inexistant (contrairement à celui de produits finis) et il y a très peu d'intervention humaine (les ressources sont majoritairement des machines). C'est particulièrement le cas de l'industrie pharmaceutique, de l'industrie du papier, de l'élaboration de boissons, de la fonderie, du raffinage du pétrole, de la production de gaz naturel, des centrales électriques, de la production de fibres synthétiques, etc.

1.3.2 Systèmes à flux discrets

Les systèmes de production à flux discrets sont caractérisés par un flux de matières interrompu à volume moyen ou faible, où les produits sont différenciables. L'affectation des produits aux ressources varie selon les spécifications des produits (demande client) et selon la disponibilité des ressources (temps de production différents). Ils peuvent être classés en : *production par projet*, *production par job* et *production par lot*.

1.3.2.1 Production par projet

Le mode de production par projet est caractérisé par un flux à bas volume et un système flexible. La demande est imprévisible et les spécifications demandées sont rarement répétitives, ce qui fait que les produits sont uniques. Normalement, dans le processus de fabrication, le produit reste dans un seul endroit, et ce sont les ressources qui se déplacent. Une fois que le produit est terminé, les ressources sont démontées, vu que la probabilité d'utiliser les mêmes ressources et dans la même disposition est presque nulle. La gestion de délais est une problématique majeure dans ces systèmes, surtout lorsque plusieurs projets sont abordés en même temps. La politique de production suivie est celle de *conception à la demande*.

Les systèmes de production par projet sont particulièrement présents dans le domaine de la construction. À titre d'exemple, nous pouvons mentionner la construction de bâtiments, de ponts et de navires.

1.3.2.2 Production par job

La production par job est caractérisée par un volume limité de production (à l'unité ou par lots de petite taille). Les produits sont fabriqués selon les spécifications du client, donc la production n'est pas standard mais le système est normalement flexible (l'organisation de l'atelier peut être adaptée pour traiter des commandes client différentes). Une seule commande est traitée à la fois, l'inventaire de produits semi-finis est important et les coûts de production sont élevés. Par contre, les ressources sont flexibles. La politique de production peut être *conception à la demande* ou *production à la demande*.

Ce type de production est présente dans les processus de confection, de réparation des équipements, de peinture et décoration, etc.

1.3.2.3 Production par lot

La production par lot est le processus où des produits similaires sont fabriqués par groupes, et pas unitairement. Les produits d'un lot suivent ensemble le processus de fabrication étape par étape. Ce mode de production permet de réduire les coûts unitaires et d'augmenter les niveaux de productivité. Cependant, les temps de lancement de la production entre lots peuvent être importants, et il faut gérer des niveaux non-négligeables d'inventaire de matières premières et de produits semi-finis. Plusieurs politiques de production peuvent être combinées, à savoir : la production sur stock, l'assemblage à la demande et la production à la demande. Le choix dépend de la variabilité de la demande. Si la demande est connue à l'avance, une production à la demande sera la stratégie à suivre. Si la demande est incertaine, il faudra construire du stock, soit de matières premières (et/ou des composants) avec une politique d'assemblage à la demande, soit de produits finis avec une politique de production sur stock.

La production par lot amène à un problème de dimensionnement de lots, qui consiste à choisir la taille des lots de produits dans chaque période de l'horizon de planification (plusieurs jours, semaines ou mois), pour satisfaire la demande client. Dans un tel problème, l'objectif est de minimiser la somme des coûts de fabrication, de stockage et de lancement de la production, entre autres. La fonction objectif peut être réduite à la somme des coûts de lancement de la production et de stockage si le coût de fabrication est considéré constant quand toute la demande est satisfaite, sans rupture de stock.

La production par lot est largement utilisée dans des secteurs économiques divers. À titre d'exemple, nous pouvons citer la production de pain, de médicaments, de peintures, d'appareils électroniques, de fourniture, etc.

Dans ce travail, nous nous intéressons aux systèmes à flux discrets, et plus spécifiquement, à la production par lot.

1.4 Planification de la production

Selon le schéma décisionnel de la planification de la chaîne logistique, le management de la production est organisé en pratique en trois niveaux de décision : stratégique, tactique et opérationnel. Le terme de *planification de la production* fait référence aux décisions prises au niveau tactique. L'objectif est de :

- déterminer les périodes de production (*quand produire ?*)
- déterminer les quantités de production (*combien produire ?*)

Ces décisions correspondent à gérer le flux de production et les inventaires. Elles sont soumises à plusieurs contraintes, comme : la capacité de production, la satisfaction de la demande, les périodes de maintenance, la capacité de stockage, les contraintes de calendrier, les quantités de production maximale et minimale, la nomenclature, etc.

Selon le type de système de production (continu ou discret), des stratégies et des modèles mathématiques différents peuvent être adoptés afin d'optimiser le plan de production, où le but est de minimiser la somme de tous les coûts liés à la production, à savoir : coûts de stockage et de rupture de stock, coûts de fabrication, coûts de lancement (setup) de la production, coûts de maintenance, coûts de main d'œuvre additionnelle, etc.

Au niveau opérationnel, l'objectif est d'ordonnancer la production, de façon à ce que le plan de production (délais et quantités) déterminé au niveau tactique soit respecté. L'allocation de ressources pour chaque ordre de fabrication doit être décidée, avec le but de non seulement garantir un plan de production réalisable, en respectant toutes les contraintes, mais aussi de fournir la meilleure solution possible. C'est-à-dire que les décisions opérationnelles doivent permettre de réaliser la meilleure allocation de ressources possible (séquencement) et de fixer les meilleures dates de début et de fin des ordres de fabrication (ordonnancement). Selon la stratégie adoptée dans la gestion de production, ou selon le comportement du système, des critères différents d'optimisation peuvent être considérés. Nous discutons plus en détail de l'ordonnancement de la production dans la Section 1.5.

Le défi le plus important en planification de production est la gestion de la capacité. La planification tactique idéale doit tenir complètement compte des contraintes de capacité du niveau opérationnel, c'est-à-dire, des contraintes détaillées concernant l'ordonnancement. Néanmoins, cette information est difficile à intégrer et peut requérir des capacités de calcul (du point de vue informatique) importantes. Dans les dernières décennies, des solutions logicielles ont été conçues pour faciliter la planification de la production, mais les stratégies adoptées ne mènent pas, dans beaucoup de cas, vers des plans de production réalisables au niveau opérationnel.

1.4.1 Progiciels de Gestion Intégrée (ERP)

Les progiciels de gestion intégrée (Enterprise Resource Planning ou ERP) [148] sont des outils informatiques qui permettent de prendre des décisions de différente nature, concernant les différents départements ou fonctions de l'entreprise. Les fonctions qui sont normalement considérées par les ERP sont :

- Gestion des ressources financières (Financial Resource Management ou FRM)
- Gestion de la chaîne logistique (Supply Chain Management ou SCM)
- Gestion des ressources humaines (Human Resource Management ou HRM)
- Gestion de la relation client (Customer Relationship Management ou CRM)
- Planification des ressources de production (Manufacturing Resource Planning ou MRP-II)

Le concept d'ERP a été créé en 1995, avec pour but de coordonner les différentes fonctions de l'entreprise avec la production comme activité principale. Il est le résultat de l'évolution des systèmes de gestion de production, qui ont commencé avec l'approche de *calcul des besoins nets* (ou planification des besoins en composants) [188] en 1965. Cette méthode, connue par le sigle MRP, permettait juste de calculer les quantités de production de tous les composants des produits finis, en suivant la structure de la nomenclature, sans tenir compte de la capacité. Vu la difficulté pour déterminer des plans de production réalisables au niveau opérationnel, la méthode de *planification des ressources de production* (MRP-II) [204] a été créée en 1979. MRP-II incorpore MRP en tant que module de planification à moyen terme, et considère la capacité dans le processus de planification, à travers différentes étapes avec différents niveaux d'agrégation de l'information.

C'est justement le fait d'agréger la capacité qui génère des problèmes pour calculer des plans de production cohérents. La phase critique est le lien entre la planification et l'ordonnancement de la production, deux des étapes les plus importantes en management de la chaîne logistique. En effet, les ERP ne garantissent pas, à notre connaissance, une cohérence entre les décisions tactiques et opérationnelles. Le problème principal, associé à l'approche de **Calcul des besoins**

nets (Material Requirements Planning ou MRP), est que la planification et l'ordonnancement sont réalisés de manière séparée et séquentielle, en suivant la structure hiérarchique des niveaux décisionnels. Dans ce contexte, les décisions tactiques (tailles de lots et périodes de production) sont considérées comme données d'entrée ou des contraintes pour le problème d'ordonnancement. Dans certains cas, les plans de production proposés au niveau tactique, sans tenir compte des contraintes d'ordonnancement, ne peuvent pas être respectés au niveau opérationnel (ils sont infaisables). De plus, dans l'approche MRP, la capacité est considérée infinie quand les tailles de lots sont déterminées, ce qui fait qu'un plan de production ne peut être mis en œuvre qu'à condition de réaliser plusieurs changements au niveau opérationnel. Ces modifications sont liées à l'utilisation d'heures de travail supplémentaires, de la sous-traitance, ou simplement au fait de décider de ne pas satisfaire certaines demandes ou de les satisfaire avec du retard. Dans tous les cas, ces décisions entraînent des coûts additionnels qui peuvent être très importants.

Cette incohérence dans la gestion de la capacité est bien connue depuis plusieurs décennies, et le problème a été pris en compte dans l'approche MRP-II. Cependant, la planification et l'ordonnancement se font toujours d'une manière séparée. Pour cette raison, les contraintes de capacité, détaillées au niveau opérationnel, deviennent agrégées au niveau tactique. L'objectif de MRP-II est de déterminer des solutions réalisables, sans chercher à optimiser le plan. Lorsqu'un plan de production réalisable n'est pas obtenu au niveau tactique (après la phase de MRP), des modifications sont réalisées sur le plan MRP ou sur le **plan directeur de production** (Master Production Schedule ou MPS) [118]. Souvent, la méthode MRP-II n'est pas capable de fournir des solutions réalisables. Une stratégie utilisée pour éviter des plans infaisables, consiste à surestimer la capacité. Cependant, le plan de production proposé n'est pas approprié dans la plupart des cas, car la production peut finir avant la date planifiée, générant ainsi des inventaires de produit semi-fini additionnels, ce qui représente des coûts supplémentaires. Les étapes de planification de la méthode MRP-II sont détaillées dans la Section 1.4.2.

Pour garantir la cohérence entre les décisions de planification et d'ordonnancement dans les systèmes de production complexes, des contraintes d'ordonnancement détaillées doivent être considérées dans les modèles mathématiques de dimensionnement de lots [144, 47]. Ouenniche *et al.* [174] et Dauzère-pères et Lasserre [48] ont étudié l'impact des décisions de séquençement sur la planification de la production. Une description sur les méthodes d'optimisation discrète pour l'intégration de la planification et de l'ordonnancement de la production peut être consultée dans [95].

Une approche récente pour gérer de manière efficace la capacité est le concept de **Système de Planification Avancée** (Advanced Planning and Scheduling ou APS), dont l'objectif est de déterminer des solutions optimales, en considérant non seulement les décisions concernant la

production, mais aussi l'approvisionnement, la distribution et les ventes [72]. Les systèmes APS sont intégrés dans les systèmes ERP, avec pour but d'optimiser la chaîne logistique globalement. Néanmoins, les différentes activités de la chaîne logistique sont affectées à des modules différents, et la résolution se fait donc de manière hiérarchique. L'avantage par rapport à MRP-II est la facilité offerte pour échanger les flux d'informations entre les différents modules, qui permet de favoriser l'obtention des solutions réalisables, sans toutefois le garantir. Une opportunité d'amélioration importante dans les APS est au niveau des formulations mathématiques, sur lesquelles la communauté scientifique focalise actuellement les travaux de recherche. Parfois, les modèles mathématiques utilisés pour planifier la chaîne logistique considèrent des contraintes qui ne permettent pas d'utiliser la capacité de façon optimale, ou ils ne sont pas adaptés à des problèmes spécifiques, et parfois la capacité de calcul des ordinateurs est insuffisante. En plus, les solveurs commerciaux ne sont pas souvent capables de résoudre des problèmes de grande taille combinant des variables continues et discrètes. Nous discutons plus en détail l'approche APS dans la Section 1.4.3.

1.4.2 Planification des ressources de production (MRP-II)

La première étape en planification de la chaîne logistique consiste à établir un **plan stratégique** pour le long terme (sur plusieurs années). Dans ce plan, les directives principales concernant l'évolution de l'activité économique de la chaîne sont considérées. Les décisions concernées ont un rapport avec la conception de la chaîne, le positionnement sur le marché et les politiques internes. À partir du plan stratégique, des objectifs sont fixés pour gérer la planification de la production et de la capacité.

L'apport de la méthode MRP-II est le fait d'intégrer la capacité dans le processus de décisions pour construire le plan de production, comme illustré sur la Figure 1.4.

MRP-II démarre avec la planification de la demande, qui consiste à définir les commandes clients fixes à satisfaire tout au long de l'horizon de planification, ainsi que les demandes prévues, selon des méthodes de prévision. À partir de cette planification, la prochaine étape consiste à élaborer le plan industriel et commercial (Sales and Operations Planning ou S&OP), en prenant en compte une vision agrégée de la capacité, où la consommation de ressources de production est estimée de manière globale. Ce plan doit satisfaire les objectifs du plan stratégique et la demande planifiée. L'étape suivante consiste à créer un plan directeur de production (MPS), en tenant compte de la capacité sommaire planifiée, qui correspond également à une capacité agrégée. Il sert à planifier la production des produits finis. Ensuite, le calcul des besoins nets (MRP) est réalisé à moyen terme, compte tenu d'une vue détaillée de la capacité, à travers une

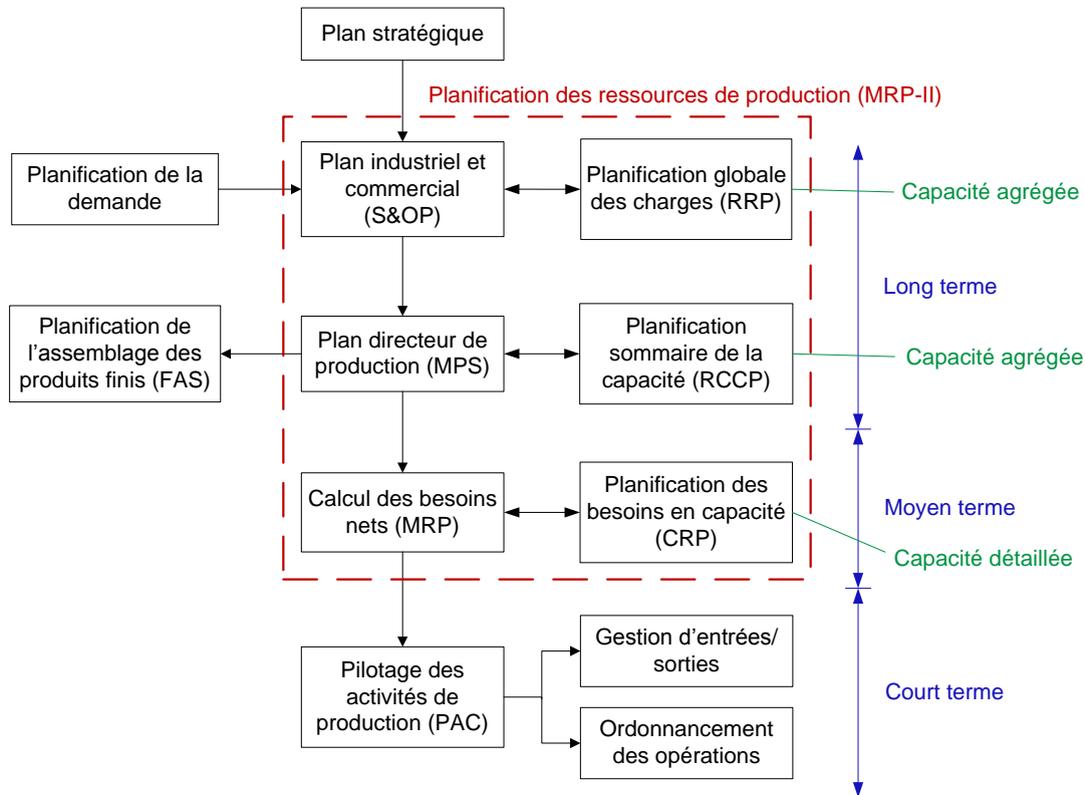


FIGURE 1.4 – Planification de la production et de la capacité avec MRP-II

planification des besoins en capacité (CRP). L'étape suivante, à court terme, est le pilotage des activités de production (PAC), où l'ordonnancement est effectué. Si aucun ordonnancement ne permet de respecter les délais (plan de production non réalisable), on retourne vers l'étape de MRP, pour proposer un nouveau plan, et ensuite on exécute à nouveau le module PAC. Ces deux étapes sont exécutées de façon itérative jusqu'à ce que le plan de production et d'approvisionnement soit réalisable au niveau ordonnancement.

Dans ce qui suit, nous décrivons brièvement les étapes de planification de la production et de la capacité à travers MRP-II. Pour des plus amples détails, le lecteur peut consulter [118] et [141].

1.4.2.1 Planification de la production

– Plan industriel et commercial

Son but est d'associer les décisions commerciales à la capacité de production globale, de façon à définir des objectifs de production par famille de produits, permettant de maximiser le profit. Les décisions commerciales consistent à planifier la demande sur le long terme et par périodes, en se basant sur des méthodes de prévisions. Les demandes deviennent des contraintes pour le S&OP, qui doit garantir leur satisfaction, tout en respectant la charge globale (capacité) estimée. Les données de sortie du S&OP correspondent à des quantités de production par période, par famille de produit.

– Plan directeur de production

Il a pour but de définir les objectifs de production de tous les produits finis (tailles de lots et périodes de production), ou d'un ensemble de configurations possibles de produits finis, dans le cas des environnements de production de type *assemblage à la demande*, tout au long d'un horizon de planification à long ou à moyen terme. Les objectifs fixés dans le S&OP sont considérés comme des contraintes, c'est-à-dire qu'il faut respecter les projections de ventes et les ordres de production prévus. Le MPS prend en considération une estimation agrégée de la capacité disponible, à travers une *planification sommaire de la capacité* (RCCP), et permet d'assurer le lien entre la demande du marché et la capacité d'approvisionnement des sites de production.

– Calcul des besoins nets

C'est une procédure qui consiste à planifier les quantités de production de tous les produits composants de la nomenclature, de manière à satisfaire les besoins des produits finis. Chaque produit est planifié de façon indépendante dans un ordre hiérarchique, en commençant par les produits les plus agrégés. L'objectif est de : (i) garantir la disponibilité de composants et de matière première pour réaliser la production, (ii) garantir la disponibilité de produits finis pour satisfaire la demande client, (iii) maintenir un niveau d'inventaire le plus petit possible et, (iv) planifier les ordres de fabrication, les dates de livraison et les dates de réapprovisionnement.

La procédure MRP peut être résumée à travers les deux étapes suivantes. À noter que l'étape 2 est réalisée systématiquement et hiérarchiquement jusqu'à planifier tous les produits de la nomenclature.

1. Les tailles de lots des produits les plus agrégés sont déterminées sans tenir compte de la capacité.
2. Les tailles de lots des prédécesseurs directs sont déterminées, en tenant compte des délais d'approvisionnement et en respectant les besoins internes (en matière) des successeurs directs.

La planification est donc effectuée niveau par niveau, en commençant par les produits finis. Les contraintes de nomenclatures sont respectées, mais il n'y a pas de garantie que le plan de production soit optimal, puisque les tailles de lots des produits les plus agrégés contraignent la planification des autres produits. Ceci est un inconvénient du point de vue de la qualité de la solution, mais l'inconvénient majeur est la non-prise en compte des contraintes de capacité, ce qui entraîne, la plupart du temps, la modification du plan de production au niveau opérationnel, au moment où les décisions d'ordonnancement sont prises. Ce manque d'information fait que MRP ne convient pas à des systèmes de production complexes, comme les job-shops et les flow-shops.

1.4.2.2 Planification de la capacité

– Planification globale des charges (Resource Requirements Planning ou RRP)

C'est une procédure qui consiste à mesurer la capacité globale du système de production, en termes de main d'œuvre, de machines et d'autres ressources, pour respecter le plan industriel et commercial (S&OP). Si d'après cette mesure de la capacité, le S&OP est réalisable, la prochaine étape est l'élaboration du plan directeur de production (MPS). Dans le cas contraire, si les ressources sont insuffisantes, deux alternatives sont possibles :

1. Augmenter la capacité de production, en développant l'infrastructure, en achetant des nouvelles machines, en augmentant le nombre de postes de travail, en contractant de la main d'œuvre externe, etc.
2. Modifier le S&OP. Cette mesure est adoptée lorsque le retour sur investissement de l'augmentation de la capacité est jugé inintéressant à long terme.

– **Planification sommaire de la capacité (Rough-Cut Capacity Planning ou RCCP)**

C'est une étape de validation des ressources nécessaires pour suivre le MPS. Un calcul de la capacité globale nécessaire pour fabriquer tous les produits finis, dans les périodes établies par le MPS, est réalisé, en tenant compte des ressources disponibles. S'il s'avère que le MPS est réalisable avec les ressources existantes, aucune mesure additionnelle n'est adoptée, donnant suite ainsi au calcul des besoins nets (MRP) à court terme. Par contre, si la capacité est insuffisante, les stratégies suivantes sont considérées :

1. Modifier le MPS, de façon à ce que la capacité soit satisfaite.
2. S'il n'est pas possible de modifier le MPS, en respectant les contraintes du S&OP, remonter d'un niveau dans la structure décisionnelle pour modifier le S&OP, et recalculer ensuite le MPS.

Il existe trois méthodes pour planifier la capacité à ce stade : la méthode des facteurs globaux, la méthode de macro-gammes et la méthode des profils des ressources. Le choix de la méthode dépend de la précision désirée (le niveau de détail ou de désagrégation de la capacité).

– **Planification des besoins en capacité (Capacity Requirements Planning ou CRP)**

Elle consiste à déterminer tous les besoins en main d'œuvre, en machines et en matière (y compris les produits composants), ainsi que les délais d'approvisionnement, de façon à satisfaire le plan de production issu du MRP (toutes les tailles de lots sont considérées). La capacité calculée est donc plus détaillée que celle définie dans les étapes de RRP et RCCP. S'il est impossible de respecter la capacité avec le plan de production courant, les mesures suivantes sont adoptées :

1. Modifier le plan de production des produits composants (au niveau du MRP).
2. Si la modification du MRP n'entraîne pas le respect de la capacité, remonter d'un niveau et modifier le MPS.

3. S'il est encore impossible de satisfaire la capacité, modifier le S&OP ou investir en une augmentation de la capacité.

La méthode CRP requiert comme données d'entrée : la nomenclature, les gammes de fabrication, les temps opératoires standard, les délais d'approvisionnement et les ordres de production du MRP (tailles de lots de tous les produits de la nomenclature sur toutes les périodes de l'horizon de planification). Le calcul de la capacité est beaucoup plus précis que dans les méthodes associées à RCCP, mais l'effort de calcul est plus important.

1.4.3 Systèmes de planification avancée (APS)

Les APS ont pour mission de déterminer la solution optimale concernant la planification globale de la chaîne logistique. Non seulement les décisions de production (planification et ordonnancement) sont considérées, mais aussi l'approvisionnement, la distribution et les ventes. Le problème global est décomposé en différents modules de planification, avec des interactions de flux verticaux (liant les niveaux stratégique, tactique et opérationnel) et horizontaux (liant les différentes activités de planification). Une vision globale des interactions des modules de planification de la chaîne logistique dans un APS est illustrée sur la Figure 1.5.

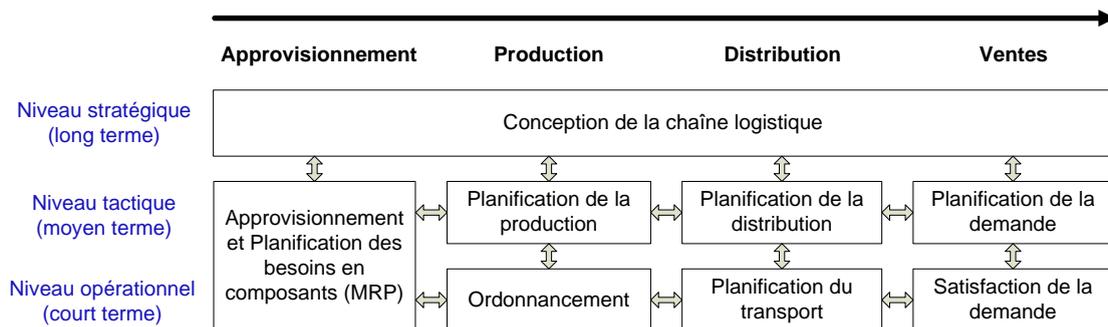


FIGURE 1.5 – Matrice de planification de la chaîne logistique [72]

Contrairement aux ERP classiques utilisant la méthode MRP-II, qui suit une logique de construction de solution manuelle, l'approche APS modélise chaque étape de planification (approvisionnement, production, distribution et satisfaction de la demande) comme un programme linéaire en nombres entiers ou mixte [185], et elle utilise différentes méthodes de *recherche opérationnelle* pour résoudre ces modèles mathématiques, le but étant d'obtenir une solution optimale [72]. Pour assurer la cohérence des décisions et éviter les problèmes de MRP-II, chaque

modèle mathématique inclut des contraintes liées aux problèmes adjacents (problèmes ayant une influence directe sur le flux de matière ou d'information). Ainsi, par exemple, le problème d'ordonnancement tient compte des contraintes issues de la planification de la production (taille de lots et délais) et des contraintes dérivées de la planification du transport (disponibilité des véhicules).

Le challenge le plus important à gérer dans un APS est la faisabilité du point de vue informatique (technologique). Les problèmes d'optimisation associés aux étapes de planification de la chaîne logistique, ayant souvent une grande complexité, peuvent conduire à un effort de calcul (mémoire physique et temps de calcul) très important. Pour réduire l'effort de calcul, les informations partagées entre les modules sont souvent agrégées, et certaines considérations mathématiques sont faites afin de privilégier certains objectifs (par exemple, subordonner les décisions d'approvisionnement aux décisions de planification), et ne pas avoir à gérer des objectifs multiples.

Le caractéristique commune des APS est le partage d'information des différents modules. Cependant, il n'y a pas de version standard définissant la coordination des flux entre les différentes activités. En effet, le mode de fonctionnement des APS peut varier grandement d'un logiciel/vendeur à l'autre. Ainsi, quelques compagnies proposent un seul type de logiciel, qui s'adapte avec plus ou moins de pertinence aux situations particulières de l'industrie ; tandis que d'autres conçoivent leurs systèmes en fonction du client, avec les contraintes et les besoins spécifiques de la chaîne logistique en question. Nous pourrions donc dire que ces dernières suivent un mode de production de type *conception à la demande*, comme nous avons vu dans la Section 1.3. L'inconvénient du premier type de concepteur d'APS est que le système peut être difficile à adapter dans certains cas, et la performance peut ne pas correspondre aux attentes. D'autre part, les délais de développement, dans le deuxième cas, peuvent être très importants, les coûts sont plus élevés et le logiciel éventuellement moins flexible (pas adapté si le système de production évolue régulièrement en fonction des besoins du marché). Une vue d'ensemble sur les principaux APS disponibles sur le marché international, et une comparaison des deux systèmes les plus utilisés (SAP APO et Oracle) sont présentées dans [51].

En ce qui concerne la planification de la production, certains APS adoptent la logique de MRP-II, en construisant un S&OP et un MPS avec différents niveaux de détail sur la capacité ; d'autres systèmes proposent une seule étape de planification, réalisée au niveau tactique, avec des contraintes de capacité relativement détaillées, mais qui n'intègrent pratiquement jamais les contraintes d'ordonnancement détaillées [77].

Les APS diffèrent aussi dans la façon dont les étapes de planification de la production et d'ordonnancement interagissent. D'une part, certains logiciels proposent un module unique pour

la planification et l'ordonnancement, où les décisions peuvent être prises simultanément. Cependant, les modèles mathématiques utilisés pour planifier la production contraignent souvent l'utilisation exacte de la capacité, en limitant le nombre de lancements de production (setups) par période, pour garantir la faisabilité du plan de production à court terme, sans avoir à inclure des contraintes d'ordonnancement détaillées, l'ordonnancement étant réalisé période par période. Dans des problèmes d'ordonnancement complexes (où plusieurs produits sont en compétition sur l'utilisation de plusieurs ressources), ces modèles peuvent s'avérer inefficaces. De plus, l'effort de calcul nécessaire pour résoudre un tel modèle est très élevé. D'autre part, d'autres APS proposent deux modules séparés, avec une méthode hiérarchique et un échange itératif d'informations permettant de faire évoluer, de façon permanente, les décisions de deux modules, jusqu'à obtenir un plan de production réalisable au niveau ordonnancement. Dans ce type d'APS, les modèles de planification tactique peuvent inclure des contraintes d'ordonnancement plus ou moins agrégées, qui ne garantissent pas complètement la faisabilité du plan et, même si les informations sont partagées entre les modules, il existe une perte de données, qui dépend du niveau d'agrégation des contraintes liant les modules, et une exploration réduite de l'espace des solutions. L'efficacité d'un APS dépend directement de la qualité des modèles mathématiques et des techniques de résolution utilisées (souvent le recuit simulé, la recherche taboue et les algorithmes génétiques [212]). Nous discutons, dans le Chapitre 2, des travaux modélisant et résolvant des problèmes de planification de la production avec intégration des contraintes d'ordonnancement.

L'intégration des décisions se fait de manière similaire pour connecter la planification de la production avec la planification de la distribution, et l'ordonnancement avec la planification du transport. Dans tous les cas, les décisions déterminées par un APS sont en général réalisables et beaucoup plus cohérentes que celles d'un ERP classique, où la planification de la production est faite à travers la méthode MRP-II, et où les contraintes de distribution ne sont pas considérées. En pratique, on peut trouver des APS intégrés dans des ERP, pour réaliser la planification de la chaîne logistique, tout en partageant des données avec les différents modules de l'ERP [77].

1.5 Ordonnancement de la production

L'étape d'ordonnancement de la production consiste à organiser, à court terme, les différentes ressources de l'atelier de production, de manière à suivre le mieux possible le plan de production déterminé dans l'étape de planification, c'est-à-dire, respecter les délais de production. Il s'agit d'affecter les opérations de production de tous les jobs (ordres de fabrication) aux ressources et de décider de leurs dates de début et de fin. De mauvaises décisions d'ordonnancement

ment peuvent occasionner des délais et des inventaires de produits semi-finis additionnels, ce qui peut déstabiliser la chaîne logistique. Dans des systèmes complexes, avec plusieurs ressources et des produits différents, il est difficile d'estimer la capacité de production à l'avance, et par conséquent, de calculer un plan de production réalisable au niveau ordonnancement. De plus, des incertitudes liées aux activités de maintenance et aux opérations manuelles peuvent aussi conditionner la mise en œuvre des décisions opérationnelles.

1.5.1 Configurations d'ateliers de production

La manière avec laquelle l'ordonnancement est réalisé dépend de la configuration de l'atelier de production, i.e. du nombre de ressources, de leur emplacement et de leur capacité à traiter les différentes opérations. Les types de configurations de ressources que l'on peut retrouver sont les suivantes :

– Atelier à une seule machine

Dans ce type d'ateliers, une seule ressource est disponible pour le traitement de tous les jobs. Donc, soit les produits requièrent uniquement une opération ou les ressources sont polyvalentes (elles peuvent réaliser plusieurs opérations). Ce type d'atelier correspond au problème d'ordonnancement le plus basique et le plus étudié dans la littérature. D'ailleurs, dans certains cas, des problèmes d'ordonnancement complexes peuvent être décomposés et ramenés à l'étude de plusieurs problèmes à une seule machine. Quelques exemples de travaux traitant le problème d'ordonnancement à une seule machine sont : [106], [44], [196], [124], [125] et [126].

– Atelier à machines parallèles

Le principe de ce type d'ateliers est similaire à celui à machine unique, i.e. chaque job passe par une seule ressource, sauf qu'ici il y a plusieurs ressources en parallèles disponibles pour réaliser l'opération. En ordonnancement, le modèle à machines parallèles est donc une généralisation du modèle à machine unique. Nous pouvons distinguer entre deux types d'ateliers à machines parallèles, ceux où les machines sont identiques, pouvant toutes traiter tous les jobs, et ceux où toutes ou une partie des ressources sont différentes, et ne peuvent donc pas traiter tous les jobs, ou elles peuvent les traiter mais avec des durées opératoires et des niveaux de qualité différents. Le problème d'ordonnancement

associé à ce type d'atelier est résolu, entre autres, dans [83] et [191].

– **Atelier de type flow-shop**

Contrairement aux ateliers à une seule machine et à machines parallèles, dans un flow-shop, chaque job requiert plusieurs étapes de transformation (opérations) sur différentes ressources, qui sont placées dans des stations de travail en série. Tous les jobs suivent la même séquence de fabrication et le but est habituellement de définir l'ordre de traitement des jobs. Comme tous les jobs doivent suivre la même séquence, des temps d'attente importants peuvent être générés entre les stations, ce qui peut entraîner un niveau élevé d'inventaire de produits semi-finis. Quelques travaux scientifiques sur le problème d'ordonnement dans un flow-shop sont : [10], [12] et [173].

Une variante du flow-shop est le flow-shop flexible, où à chaque station il y a plusieurs ressources en parallèle capables de traiter la même opération. Donc, il s'agit d'affecter chaque job à une des machines disponibles, station par station. Ceci permet d'accélérer le flux de production et de réduire les inventaires de produits semis-finis. Ce problème est par exemple traité dans [117].

– **Atelier de type job-shop**

Comme dans un flow-shop, le job-shop comporte plusieurs ressources (ou stations de travail) différentes et chaque job doit passer par plusieurs ressources avant d'être transformé en produit fini. La différence réside dans le séquençement des opérations. Dans un job-shop, les séquences entre jobs peuvent être différentes. Par exemple, le job i doit être traité chronologiquement sur les ressources A , B , C , et D ; tandis que le job j doit suivre la séquence (C, B, D, A) .

Des temps d'attentes et des inventaires de produits semis-finis sont générés entre les stations de production, et les temps de réglage (temps de setup) des machines pour passer d'un produit à un autre peuvent être importants. Ceci impose une maîtrise de la capacité de production au moment de planifier les quantités de production, afin de pouvoir programmer pour chaque tâche *ou opération* une date de début et une date de fin cohérentes, i.e. réalisables. Quelques travaux scientifiques sur le problème d'ordonnement dans un job-shop peuvent être consulter dans [119], [25] et [159].

Comme pour le type d'atelier précédent, il existe aussi le job-shop flexible, avec plusieurs ressources polyvalentes en parallèle par station de travail. Donc, non seulement il faut or-

donnancer le mieux possible les opérations, mais aussi déterminer la meilleure affectation aux ressources. Le problème d'ordonnancement du job-shop flexible est aussi connu aussi sous le nom de *Multiprocessor Job-Shop*. Le lecteur intéressé par ce type de problème peut consulter [49].

– **Atelier de type open-shop**

Ce type d'atelier, rarement rencontré dans la pratique, est un système multi-jobs et multi-ressources avec des séquences différentes entre jobs, comme le job-shop. La différence est que la gamme de fabrication n'est pas stricte, c'est-à-dire, l'ordre dans lequel un job doit être traité sur les ressources n'est pas imposé. Autrement dit, toutes les séquences permettant à un job de passer sur les ressources de sa gamme opératoire sont réalisables.

Une description plus détaillée des ateliers de production, du point de vue de l'ordonnancement, est présentée dans [182]. Un état de l'art sur le problème d'ordonnancement associé aux différents ateliers de production peut être consulté dans [15].

1.5.2 Contraintes d'ordonnancement

L'ordonnancement de la production est soumis au respect de certaines contraintes, selon la nature du système de production. Les contraintes les plus étudiées dans la littérature, faisant partie de notre thématique, sont décrites dans ce qui suit. Pour plus de détails sur ces contraintes et d'autres le lecteur peut consulter [181].

– **Contraintes de précédence**

Dans la fabrication de certains produits, il est obligatoire de respecter un ordre spécifique de traitement d'opérations défini par la gamme de fabrication, afin que le produit ait les caractéristiques désirées. De manière similaire, il est aussi possible que la production d'un job i requiert la réalisation anticipée du job j .

– **Contraintes de sélection des machines**

Dans des ateliers à machines parallèles, il se peut que certains produits ne puissent pas être traités sur certaines machines. Ceci restreint le nombre de séquences possibles et

limite donc, indirectement, la capacité.

– **Contraintes de main d’œuvre**

Dans des systèmes de production non-automatisés, la participation d’opérateurs dans le processus de fabrication est requise. Chaque opérateur ayant différentes qualifications et différentes expertises, l’affectation de la main d’œuvre aux machines est aussi importante que le séquençement des opérations. Pour garantir un niveau de qualité désiré ou une productivité élevée, certains systèmes peuvent limiter l’ensemble d’opérateurs disponibles par machine ou par station de travail.

– **Contraintes de séquençement**

Pour garantir la cohérence de l’ordonnancement, des contraintes de séquençement liant les dates de début et de fin de toutes les opérations peuvent être considérées. Elles permettent de respecter l’ordre de traitement des opérations sur chaque machine indépendamment, et sont donc complémentaires aux contraintes de précédence, qui garantissent la cohérence des dates de début et de fin des opérations appartenant à la même gamme de fabrication.

– **Contraintes de maintenance**

Les contraintes de maintenance sont utilisées pour tenir compte des activités de conditionnement, de réparation et d’inspection des objets matériels faisant partie du processus de fabrication, c’est-à-dire, les outils, les machines, les chariots, les entrepôts, etc. Les activités de maintenance sont souvent représentées dans les problèmes d’ordonnancement par des périodes d’indisponibilité.

– **Temps et coûts de réglage dépendant des séquences**

Dans certains systèmes, l’ordre de séquençement des jobs sur une machine a un impact sur le coût et le temps de production, spécifiquement à l’étape de réglage ou lancement (ou *setup*). Ainsi, le coût et le temps de setup, lorsque l’on passe de la production de l’article i à l’article j , ne sont pas les mêmes que lorsque l’on change les paramètres de la machine pour passer du produit i au produit k . Ces contraintes sont assez souvent présentes dans

les systèmes où une diversité importante de produits est fabriquée dans le même atelier.

– **Contraintes de capacité de stockage et de temps d’attente**

Souvent modélisées dans les problèmes de planification, les contraintes de capacité de stockage peuvent aussi être considérées dans les problèmes d’ordonnancement, en décomposant la période longue de planification en plusieurs périodes courtes avec un indicateur du niveau de stock par période. Selon les dates de fin des opérations, le niveau de stock de produits finis et semis-finis peut varier. De manière similaire, les temps d’attente entre stations de travail peuvent différer en fonction des dates de début et de fin. Afin d’éviter des longs temps d’attente et des niveaux de stock ingérables ou indésirables, des bornes (ou limites) sur la capacité de stockage et le temps d’attente sont, dans certains cas, imposées.

– **Préemption**

La préemption est la caractéristique du système permettant d’interrompre le traitement d’une opération, pour en traiter une autre. Dans la plupart de problèmes d’ordonnancement étudiés dans la littérature, la préemption est non autorisée, c’est-à-dire, une fois qu’une opération démarre sur une machine, elle doit être terminée avant de pouvoir affecter une nouvelle tâche à la machine.

– **Contraintes de transport**

Dans un réseau comportant plusieurs sites de production, tenir compte des contraintes de transport peut servir à prendre des meilleures décisions concernant l’ordonnancement. Les contraintes de transport peuvent être : les dates de disponibilité de véhicules, les dates souhaitées de livraison, la capacité des véhicules, etc. Une bonne synchronisation entre l’ordonnancement et le transport peut aider à réduire le coût de stockage de produits finis.

1.5.3 Critères d’optimisation

L’objectif dans un problème d’ordonnancement est de séquencer les opérations sur les ressources le mieux possible, de manière à respecter le plan de production, tout en optimisant un critère. Le critère peut être vu comme un indicateur de la performance de l’activité de production et peut varier selon la nature du système, le comportement de la demande et la

configuration de l'atelier. Les critères liés à notre thématique [181] les plus étudiés sont les suivants :

– **Minimiser le temps total de cycle (*makespan*)**

Connu aussi sous le terme de *makespan*, le temps total de cycle C_{max} est le critère le plus étudié et le plus utilisé en pratique, puisqu'il a un impact direct sur plusieurs autres critères. Le C_{max} est la date de fin la plus grande parmi tous les jobs à fabriquer. Soit C_j la date de fin du job j , alors $C_{max} = \max_j C_j$. Autrement dit, le temps total de cycle, est la date de fin du dernier job à être fabriqué.

– **Maximiser le taux de production**

Souvent donné par la capacité de la machine goulot (machine à faible taux de production qui conditionne la vitesse du processus de fabrication et génère du stock en amont), le taux de production est le principal indicateur de la productivité (nombre de pièces produites par unité de temps) et sa maximisation permet d'améliorer le niveau de service au client (ou niveau de réponse à la demande). Intrinsèquement, maximiser le taux de production permet de minimiser le temps total de cycle, et inversement.

– **Minimiser le retard maximum**

Le retard L_j d'un job j est la différence entre sa date de fin réelle C_j et sa date de fin souhaitée d_j . Le retard maximum L_{max} est donc associé au job dont la valeur $C_j - d_j$ est la plus grande. Un critère similaire est la minimisation du nombre de jobs en retard.

– **Minimiser le temps total de lancement de la production ou de réglage (temps de setup)**

Le temps de réglage d'une machine, avant de lancer la production d'un job, peut varier en fonction de la séquence et peut être élevé. Donc, il peut être plus avantageux d'éviter les changements de réglage, en limitant le nombre de produits différents par machine.

- **Minimiser le coût total de lancement de la production ou de réglage (coût de setup)**

Comme pour le critère précédent, les coûts de setup peuvent aussi être très importants et peuvent varier en fonction de la séquence. Néanmoins, le temps et le coût de setup associés à un même changement d'opération peuvent être indépendants, c'est-à-dire, les deux critères peuvent être en concurrence. Minimiser le coût de setup est un bon choix dans les systèmes où ce coût est élevé par rapport au coût de stockage.

- **Minimiser le stock de produits semi-finis (WIP)**

Le WIP est l'inventaire d'un produit à un état intermédiaire de son processus de fabrication, i.e. le stock de produits pouvant exister entre les différentes opérations de la gamme de fabrication. Dans des systèmes avec temps d'attente élevés entre stations de travail, cet inventaire peut être important, et dans certains cas (produits périssables ou obsolètes, par exemple), il peut être difficile à gérer. Donc, le coût d'inventaire de produits semi-finis peut être important.

- **Minimiser le coût de stockage de produits finis**

Dans des systèmes où les coûts d'inventaire sont élevés, il est indispensable de gérer le flux de production de façon à stocker le moins possible. Le mode de production à la demande peut favoriser la diminution des stocks, mais si les coûts de setup sont importants, il peut être plus avantageux de produire de manière anticipée, et donc de stocker. Par ailleurs, dans le mode de production sur stock, contrôler le taux de production en fonction du niveau de stock de sécurité désiré est une stratégie pour réduire les coûts.

- **Minimiser le temps maximum de livraison**

Dans un système comportant plusieurs sites de fabrication, où chaque site peut fournir des composants aux autres, le temps de livraison peut avoir une influence sur la durée de stockage et sur le niveau de service aux clients. Cependant, ce critère est peu utilisé en ordonnancement, car les sites ont tendance à planifier et à ordonnancer la production sans tenir compte des contraintes des autres sites (contraintes de logistique interne).

– **Minimiser le coût de transport**

Dans un système multi-sites, il est aussi important de tenir compte du coût de transport associé à la livraison. La détermination de routes et l'affectation de produits aux véhicules est souvent un facteur à considérer. Entre le temps et le coût de transport il n'y a pas forcément une relation linéaire, et il peut s'avérer impossible de minimiser le temps et le coût de transport.

– **Minimiser le temps total de flux**

Le temps de flux d'un job i est la différence entre sa date début au plus tôt r_j et sa date de fin C_j . Minimiser le temps de flux permet donc de réduire les temps d'attente, et par conséquent, les inventaires de produits semi-finis.

Tous ces critères sont reliés, et souvent les décisions prises pour en optimiser un ont une influence sur les autres indicateurs. Donc, selon les caractéristiques du système de production (ainsi que le comportement de la demande), des critères différents peuvent être utilisés pour ordonnancer la production. Dans ce travail, nous nous intéressons à l'intégration de la planification et de l'ordonnancement de la production, avec pour objectif de minimiser le coût total. Ce coût est une fonction qui dépend en particulier des coûts de stockage et de setup, et permet donc de trouver un équilibre entre ces deux critères.

Pour de plus amples détails au sujet de l'ordonnancement de la production, le lecteur est invité à consulter [181].

1.6 Conclusion

Nous avons posé dans ce chapitre les bases de notre étude de l'intégration de la planification et de l'ordonnancement de la production dans une chaîne logistique. Ainsi, nous avons décrit la configuration générale d'une chaîne logistique, en expliquant les étapes principales de gestion, les niveaux décisionnels et les contraintes les plus communes, dans le milieu industriel. Nous avons classé les différents types de systèmes de production, selon la nature des flux de matières. Le type de système auquel nous nous intéressons par la suite est un système à flux discrets avec un mode de production par lots et une configuration d'atelier de type job-shop.

D'une manière plus approfondie, nous avons expliqué les étapes de planification et d'ordonnancement de la production dans la gestion de la chaîne logistique. Nous avons décrit les

objectifs et les contraintes impliqués dans chaque étape, ainsi que les approches les plus utilisées pour gérer ces aspects dans un système de production. La principale faiblesse des approches existantes consiste en la difficulté d'intégrer les décisions tactiques et opérationnelles, ce qui rend les solutions tactiques irréalisables ou sous-optimales à court terme, pouvant conduire, dans beaucoup de cas, à des pertes économiques. L'utilisation d'une approche intégrée, à la place d'une méthode hiérarchique, est pourtant un facteur qui peut s'avérer d'une importance cruciale et faire la différence dans le contexte économique actuel, qui connaît de plus en plus une forte concurrence dans la plupart des secteurs industriels.

L'intégration des décisions tactiques et opérationnelles dans l'étape de production repose sur une coordination efficace entre la planification et l'ordonnancement, deux activités très importantes en management de la chaîne logistique. Néanmoins, intégrer les contraintes de planification et d'ordonnancement requiert le développement d'une méthode de résolution dédiée.

Afin de présenter les avantages et les limites des approches qui cherchent à réduire l'incohérence entre la planification et l'ordonnancement, nous passons en revue, dans le Chapitre 2, les travaux scientifiques portant sur ces deux problèmes.

Chapitre 2

État de l'art en planification et ordonnancement

Ce chapitre propose un état de l'art sur les problèmes de dimensionnement de lots avec capacité, i.e., les problèmes de planification tactique qui s'intéressent dans une certaine mesure à la faisabilité du plan de production au niveau opérationnel. Parmi ces problèmes, nous mettons l'accent sur les problèmes intégrant des contraintes d'ordonnancement ou de capacité détaillée dans la formulation mathématique, garantissant des solutions réalisables.

[2.1 Introduction](#)

[2.2 Généralités sur les problèmes d'optimisation](#)

[2.3 Problèmes de dimensionnement de lots](#)

[2.4 Problèmes d'ordonnancement](#)

[2.5 Intégration de la planification et de l'ordonnancement](#)

[2.6 Conclusion](#)

2.1 Introduction

Dans le chapitre précédent, nous avons présenté une vision globale du fonctionnement de la chaîne logistique, en expliquant le rôle des différents acteurs, les enjeux et l'importance d'avoir une stratégie de management intégrant les différentes étapes du cycle de planification (approvisionnement, production, distribution et ventes). Nous avons présenté plus particulièrement les sous-étapes de planification (au niveau tactique) et d'ordonnancement de la production (au niveau opérationnel), dont l'intégration des décisions constitue l'objet de cette thèse.

Nous nous intéressons dans ce chapitre aux problèmes de planification de la production discrets, plus précisément aux problèmes de dimensionnement de lots avec contraintes de capacité, où la prise en compte d'informations sur l'ordonnancement est importante, afin de garantir des solutions réalisables ou d'améliorer la qualité des décisions.

Nous commençons ce chapitre avec des généralités sur les problèmes d'optimisation, dans la Section 2.2, nécessaires à la compréhension des détails sur les problèmes de dimensionnement de lots et d'ordonnancement qui sont présentés par la suite. Dans la Section 2.3, nous décrivons la classification habituelle des problèmes de dimensionnement de lots, et nous détaillons les caractéristiques principales des problèmes avec prise en compte de la capacité de production, qui sont étudiés dans la littérature, ainsi que les travaux les plus importants sur chacun de ces types de problèmes. Ensuite, dans la Section 2.4, nous présentons brièvement les caractéristiques des problèmes d'ordonnancement les plus étudiés dans la littérature. Dans la Section 2.5, nous discutons de l'importance d'intégrer les décisions de planification et d'ordonnancement au niveau tactique, et nous présentons les travaux qui intègrent de manière cohérente ces décisions, selon une classification basée sur la nature de l'approche de résolution. Enfin, nous terminons ce chapitre par une conclusion sur les différentes thématiques traitées.

2.2 Généralités sur les problèmes d'optimisation

Nous avons brièvement discuté, dans le chapitre précédent, la difficulté associée à la résolution de plusieurs problèmes, avec objectifs différents, dans la planification de la chaîne logistique, et particulièrement lorsque l'on veut suivre une approche d'optimisation globale, comme établie par la matrice de planification des logiciels APS. Les problèmes associés aux étapes de planification de la chaîne logistique, dans tous les niveaux de décisions, peuvent être modélisés comme des problèmes d'optimisation, ayant une fonction objectif (par exemple, la minimisation du coût, la minimisation du retard, la minimisation du nombre de véhicules, etc.) et des contraintes (par exemple, la satisfaction de la demande, une capacité de production limitée, des

lots de production avec une taille maximale, des véhicules limités, des fenêtres de temps pour la production ou pour la distribution, etc.). En pratique, avec toutes les données et les contraintes présentes dans les différents types de chaîne logistique, chacun des problèmes modélisant les étapes de planification, devient très difficile à résoudre, cette difficulté étant mesurable par un indicateur connu comme la *complexité*, que nous discutons dans ce qui suit. L'intégration ou la considération des contraintes de plusieurs étapes de planification dans un seul problème augmente encore plus la difficulté de résolution, raison pour laquelle, il est souvent impossible de déterminer la solution optimale d'un problème, et même d'obtenir une solution réalisable. C'est pourquoi les problèmes d'optimisation liés à la chaîne logistique nécessitent de combiner des techniques de recherche opérationnelle et la conception d'algorithmes sophistiqués, capables de résoudre des problèmes complexes dans des temps de calcul raisonnables et avec une utilisation limitée de ressources (processeur et mémoire physique des ordinateurs).

2.2.1 Complexité

La complexité est un indicateur de l'effort de calcul, par rapport au temps d'exécution, nécessaire pour la résolution d'un problème d'optimisation. Une autre mesure de l'effort de calcul est l'utilisation ou la consommation de mémoire physique (RAM dans les ordinateurs) nécessaire pour garder les données des variables, issues de l'exploration de l'espace de solution (ou espace de recherche), dans une méthode d'optimisation. En théorie de la complexité [62], on peut distinguer la complexité de l'algorithme et la complexité du problème.

La complexité du temps de calcul d'un algorithme est donnée par le nombre d'étapes nécessaires pour résoudre le problème. Étant donné que ce nombre peut varier en fonction de l'instance étudiée, la complexité de l'algorithme est définie par rapport au pire cas. La notation O est utilisée pour représenter cette complexité. Ainsi, un algorithme est applicable en temps polynomial si sa complexité est $O(p(n))$, ou $p(n)$ est une fonction polynomiale de la taille de l'instance n (par exemple, le nombre de produits, le nombre de machines ou le nombre de périodes de planification). Donc, si $p(n) = a_k n^k + \dots + a_j n^j + \dots + a_1 n + a_0$, avec $a_k > 0$ et $a_j \geq 0 \forall 1 \leq j \leq k - 1$, l'algorithme a une complexité polynomiale de $O(n^k)$ [212]. Des exemples de temps de calcul polynomiaux sont les complexités $O(n)$, $O(n \log n)$, $O(n^2)$, $O(n^3)$, etc. En revanche, un algorithme est exécutable en temps exponentiel si sa complexité est $O(c^n)$, où c est une constante réelle supérieure à 1.

La complexité d'un problème est équivalente à la complexité du meilleur algorithme capable de le résoudre. S'il existe un algorithme pouvant résoudre le problème en temps polynomial, le problème est dit *facile*. Sinon, le problème est considéré *difficile*. Un problème d'optimisation

peut être réduit à un problème de décision, dont la solution est de type binaire, c'est-à-dire ayant pour réponse *oui* ou *non*. Selon la complexité, les problèmes peuvent être organisés en plusieurs classes [62], dont les plus grandes et importantes sont \mathcal{P} et \mathcal{NP} . Une classe de complexité représente l'ensemble des problèmes qui peuvent être résolus avec une quantité déterminée de ressources. Ainsi, la classe \mathcal{P} fait référence à l'ensemble de tous les problèmes de décision qui peuvent être résolus en temps polynomial, au moyen d'un algorithme déterministe. Un algorithme est déterministe si le nombre d'étapes nécessaires pour résoudre le problème peut être calculé à l'avance. D'autre part, la classe \mathcal{NP} est associée à l'ensemble des problèmes de décision qui peuvent être résolus en temps polynomial par un algorithme non déterministe. La caractéristique d'un algorithme non déterministe est que l'on ne peut pas prédire avec certitude le nombre d'étapes nécessaires pour résoudre le problème.

Une autre classe importante, contenue dans la classe \mathcal{NP} est celle des problèmes \mathcal{NP} -complets. Un problème de décision G est \mathcal{NP} -complet si tous les problèmes de la classe \mathcal{NP} peuvent être ramenés (réduits) en temps polynomial à G . Cela implique que l'on peut construire une instance de G , à partir de n'importe quelle instance d'un problème $H \in \mathcal{NP}$, suivant une fonction polynomiale de la taille de l'instance de H . Le problème d'optimisation correspondant à un problème de décision \mathcal{NP} -complet est dit \mathcal{NP} -difficile, pour lequel la résolution se fait en temps exponentiel, c'est-à-dire que le temps d'exécution augmente de façon exponentielle en fonction de la taille de l'instance. La relation entre les classes mentionnées est illustrée dans la Figure 2.1.

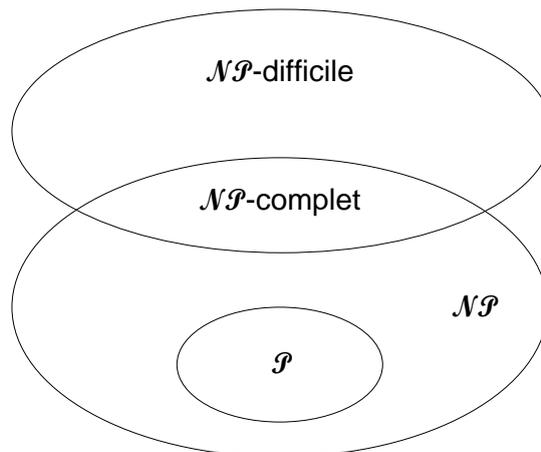


FIGURE 2.1 – Relation entre les principales classes de complexité

Nous pouvons encore distinguer entre problèmes \mathcal{NP} -difficiles au *sens faible* et problèmes \mathcal{NP} -difficiles au *sens fort*. Les premiers sont des problèmes pour lesquels il existe un algorithme

capable de les résoudre en temps polynomial, pas en fonction de la taille de l'instance, mais de la longueur des paramètres (par exemple, les demandes et les durées opératoires). Ce type d'algorithme est appelé *pseudo-polynomial*. S'il n'existe pas un tel algorithme, le problème est \mathcal{NP} -difficile au sens fort.

Un très grand nombre de problèmes d'optimisation associés aux étapes de planification de la chaîne logistique sont \mathcal{NP} -difficiles. C'est le cas, par exemple, des problèmes standard de dimensionnement de lots avec contraintes de capacité, d'ordonnancement avec plus de deux machines, de tournées de véhicules et de localisation d'usines. Dans certains cas, la simple recherche d'une solution réalisable est aussi \mathcal{NP} -difficile.

2.2.2 Modélisation

Nous nous intéressons, dans ce travail, à la modélisation par programmation mathématique. D'autres types de modélisations, que nous ne décrivons pas dans ce travail, sont les modèles d'optimisation combinatoire, les modèles de satisfaction de contraintes et les modèles non analytiques.

En programmation mathématique, les modèles peuvent être classés en : problèmes en nombres réels (continuous programs), problèmes en nombres entiers (IP ou integer programs) et problèmes en nombres entiers mixtes (MIP ou mixed integer programs). Le nom de chaque type de modèle fait référence à la nature des variables de décision utilisées dans la formulation. Ainsi par exemple, les modèles MIP comportent des variables de décision réelles (ou continues) et entières (ou discrètes). Ce type de modélisation peut être vue comme une sous-catégorie de la modélisation IP, car les méthodes de résolution pour des problèmes à nombres réels ne donnent pas des solutions réalisables pour ces modèles, contrairement aux méthodes dédiées pour des problèmes à nombres entiers. Un autre sous-ensemble des modèles IP est celui des modèles 0-1, où les variables de décision prennent uniquement des valeurs binaires (0 ou 1). Les modèles avec uniquement des variables de décision réelles sont plus faciles à résoudre, avec une complexité polynomiale quand la fonction objectif et les contraintes sont linéaires.

La plupart des problèmes de dimensionnement de lots et d'ordonnancement sont modélisés par des programmes à variables mixtes, les variables entières étant souvent binaires, comme la variable de setup Y_{il} qui vaut 1 si la production du produit i est lancée à la période l , ou 0 sinon. En général, l'inclusion de variables binaires dans la formulation mathématique rend plus difficile la résolution du problème. C'est pourquoi, des méthodes de relaxation et de décomposition sont utilisées pour ne pas considérer ou pour fixer ces variables.

Les modèles mathématiques peuvent être linéaires ou non linéaires. Dans le premier cas, la fonction objectif et toutes les contraintes sont linéaires. Le terme de programmation linéaire (LP pour Linear Programming) est associé aux modèles linéaires en nombres réels. Dans les autres cas, on parle de programmation linéaire en nombres entiers (ILP pour Integer Linear Programming) et programmation linéaire en nombres entiers mixtes (MILP pour Mixed Integer Linear Programming). La plupart des problèmes de dimensionnement de lots et d'ordonnement sont formulés comme des modèles linéaires, et plus précisément comme des MILP. Pochet et Wolsey [185] présentent de manière détaillée des formulations MIP, qui sont utilisées pour modéliser un grand nombre de problèmes de dimensionnement de lots étudiés dans la littérature.

Une manière particulière de modéliser les problèmes d'ordonnement est au moyen d'un **graphe disjonctif**, modèle proposé par Roy et Sussmann [194] en 1964, plus particulièrement pour le problème dans un atelier de type job-shop. Un graphe disjonctif $G = (V, C, D)$ est composé d'un ensemble V de sommets représentant les opérations ou tâches de production, d'un ensemble C d'arcs conjonctifs représentant les contraintes de précédence entre les opérations d'une même gamme de fabrication, et d'un ensemble D d'arcs disjonctifs représentant les contraintes de séquençement entre les opérations traitées sur la même ressource. L'objectif est de trouver la meilleure orientation possible pour chaque arc disjonctif, sans créer de cycles, de façon à optimiser un critère pouvant être : la minimisation du temps de cycle, la minimisation du nombre de jobs en retard, etc., comme expliqué dans la Section 1.5 du Chapitre 1. Un exemple de graphe disjonctif est illustré dans la Figure 3.1 dans la Section 3.2 du Chapitre 3.

2.2.3 Méthodes de résolution

La résolution des problèmes d'optimisation est basée sur des techniques de recherche opérationnelle, dont nous distinguons les méthodes exactes et les méthodes approchées. Les méthodes exactes ont pour but de déterminer la solution optimale du problème (optimum global), tandis que les méthodes approchées s'intéressent à l'obtention de la meilleure solution possible, mais dont l'optimalité n'est pas garantie. Les méthodes exactes demandent souvent un effort de calcul très important. C'est pourquoi, les méthodes approchées, dont l'objectif est de considérablement réduire l'effort de calcul en diminuant l'espace des solutions à explorer, sont souvent appliquées pour résoudre des problèmes \mathcal{NP} -difficiles, comme la plupart des problèmes de dimensionnement de lots avec contraintes de capacité.

2.2.3.1 Méthodes exactes

Pour des problèmes linéaires en nombres réels, la méthode la plus utilisée est l'algorithme du **simplexe** [157], proposé par George Dantzig, en 1947. Cette méthode ne donne pas des solutions réalisables pour les problèmes à nombres entiers ou à nombres entiers mixtes, mais elle peut être incorporée dans des méthodes résolvant des problèmes où les contraintes imposant des valeurs entières sont relâchées. Les problèmes de dimensionnement de lots et d'ordonnancement, étant modélisés le plus souvent par des programmes MIP, nous nous intéressons aux méthodes de résolution pour ce type de modèles. Parmi les méthodes exactes les plus utilisées, nous trouvons : la décomposition de Dantzig-Wolfe, la décomposition de Benders, le *branch and bound*, le *branch and cut*, le *branch and price* et la programmation dynamique. D'autres méthodes, que nous ne détaillons pas ici, sont la programmation par contraintes [192] et les méthodes A* [108] et IDA* [139].

La **décomposition de Dantzig-Wolfe** a été proposée par Dantzig et Wolfe [45] en 1960. Cette méthode peut être appliquée aux problèmes avec contraintes compliquées, ayant une structure de bloc angulaire, c'est-à-dire, ceux où l'ensemble de contraintes peut être décomposé en deux sous-ensembles. Un ensemble correspond aux contraintes qui ne lient pas des variables différentes, et l'autre ensemble est associé aux contraintes couplantes (liant plusieurs variables). Le problème original est décomposé en un problème maître et plusieurs sous-problèmes de taille réduite. Cette méthode est une version améliorée de l'algorithme du *simplexe*, utilisant la génération de colonnes [54]. Cette dernière approche est utilisée pour résoudre des problèmes de grande taille, où le problème est initialisé avec un sous-ensemble de colonnes (variables) de taille réduite, et, à chaque itération, une colonne pouvant potentiellement améliorer la solution courante est ajoutée. Pour plus de détails sur la décomposition de Dantzig-Wolfe, le lecteur peut consulter [41]. Les références [52] et [76] sont des exemples d'utilisation de cette méthode dans des problèmes de dimensionnement de lots et d'ordonnancement, respectivement.

La **décomposition de Benders**, proposée par Benders [28] en 1962, est une méthode de décomposition pour des problèmes de grande taille, ayant une structure de bloc, avec des variables compliquées. Contrairement à la décomposition de Dantzig-Wolfe, qui applique la génération de colonnes, la décomposition de Benders ajoute des contraintes progressivement. Pour simplifier la résolution du problèmes, les variables compliquées sont fixées de façon dynamique dans une méthode itérative. Dans les problèmes MILP, la décomposition de Benders est utilisée pour fixer les variables binaires à des valeurs spécifiques réalisables et résoudre le problème LP résultant de façon optimale. Pour plus de détails, le lecteur peut consulter [41]. Parmi les travaux résolvant un problème de dimensionnement de lots au moyen de la décomposition de

Benders, nous pouvons citer [1]. Pour les problèmes d'ordonnancement, le lecteur peut consulter [153].

Le **branch and bound** (B&B), proposé par Land et Doig [142] en 1960, est une méthode de recherche arborescente appliquée en deux étapes de façon itérative. La première étape est la *séparation*, qui consiste à sélectionner les branches de l'espace de solution à explorer, en décomposant un sommet de l'espace des solutions en plusieurs sous-ensembles disjoints de plus petite taille. La deuxième étape est l'*évaluation* et consiste à énumérer systématiquement les solutions candidates, dont les mauvaises sont écartées, sans résoudre de façon détaillée chaque sous-problème. En fait, le choix se base sur les valeurs de la borne inférieure (dans le cas d'un problème de minimisation) ou de la borne supérieure (dans le cas d'un problème de maximisation), en résolvant le sous-problème relâché. À chaque itération, des nouvelles branches sont explorées. Le problème relâché peut par exemple être résolu par une relaxation linéaire, où les contraintes imposant des valeurs binaires à certaines variables sont enlevées, ou par une relaxation Lagrangienne, comme expliqué plus tard dans cette Section. Pour plus de détails sur le B&B, le lecteur peut consulter [35]. Cette technique a été amplement utilisée pour la résolution de problèmes de dimensionnement de lots et d'ordonnancement. À titre d'exemple, nous citons [75] et [53].

Le **branch and cut** (B&C) est une généralisation du B&B. Il suit aussi les étapes de séparation et évaluation, mais avec une variante. La méthode des plans sécants (*cutting plane algorithm*) [236] est utilisée pour définir une nouvelle contrainte, permettant de rendre entières les valeurs de variables relâchées, dont la solution initiale (au moyen de l'algorithme du *simplexe*) est réelle. La génération de plusieurs contraintes peut être nécessaire afin d'obtenir une solution réalisable. Une fois qu'une telle solution est trouvée, les étapes de séparation et évaluation sont appliquées. Le problème est séparé en deux sous-problèmes, à partir de la nouvelle contrainte, et chaque sous-problème est résolu à nouveau par l'algorithme du *simplexe*. La procédure continue de façon itérative jusqu'à converger vers une solution optimale. Si les contraintes sont ajoutées uniquement dans le modèle initial, la méthode prend le nom de *cut and branch* (C&B). Un problème d'ordonnancement résolu par un B&C est présenté dans [190]. La méthode est appliquée pour résoudre un problème de dimensionnement de lots dans [96].

Le **branch and price** (B&P) est aussi une généralisation du B&B. Contrairement au B&C, le *branch and price* n'ajoute pas des contraintes, mais des variables, au moyen de la génération de colonnes. Au début de l'algorithme, un ensemble de variables (colonnes) est enlevé de la modélisation initiale. Des nouveaux problèmes sont résolus au fur et à mesure, en rajoutant des colonnes à chaque itération. Une description détaillée de cette méthode est présentée

dans [26]. Des exemples d'utilisation de cette technique dans des problèmes de planification et d'ordonnement de la production peuvent être consultés dans [52] et [179], respectivement.

La **programmation dynamique**, proposée par Bellman [27], en 1957, est une méthode de « force brute », c'est-à-dire qu'elle explore toutes les alternatives possibles, permettant de trouver la solution optimale par recherche exhaustive. Le problème original est décomposé en plusieurs sous-problèmes plus faciles à résoudre. Le principe de cette méthode est que la solution de chaque sous-problème est gardée en mémoire, de telle sorte que si un sous-problème est composé ou requiert le calcul de la solution d'un sous-problème préalablement résolu, la résolution n'est pas refaite, permettant d'économiser en temps de calcul. Une description plus détaillée de cette méthode peut être consultée dans [149]. À titre d'exemple, la programmation dynamique est utilisée dans [199] pour résoudre un problème de dimensionnement de lots et dans [125] pour résoudre un problème d'ordonnement.

2.2.3.2 Méthodes approchées

Il existe cinq grands groupes de méthodes approchées : les heuristiques basées sur la programmation (modèle) mathématique, les heuristiques Lagrangiennes, les heuristiques de décomposition et d'agrégation, les métaheuristiques et les heuristiques gloutonnes spécifiques au problème. Nous nous basons, dans ce qui suit, sur la classification réalisée par Buschkühl *et al.* [36].

Les **heuristiques basées sur la programmation mathématique** sont conçues à partir de méthodes exactes. Une manière simple de concevoir une telle heuristique consiste à limiter le temps d'exécution ou le nombre d'itérations d'une méthode exacte. C'est le cas de l'heuristique *truncated MIP*, qui limite le temps du B&B ou du B&C. Une autre stratégie consiste à arrêter la procédure lorsque l'écart entre les bornes inférieure et supérieure est inférieure ou égal à une certaine valeur définie avant de démarrer la procédure. Donc, si ces heuristiques suivent le principe d'une méthode exacte, elle ne garantissent pas l'obtention d'une solution optimale. Parmi ces heuristiques, nous trouvons des procédures de résolution basées sur le B&B, le B&C, le B&P, la décomposition de Dantzig-Wolfe et la décomposition de Benders. D'autres heuristiques sont les suivantes :

- *LP and fix*. C'est une méthode itérative où, à chaque itération, les variables binaires sont fixées, et le problème linéaire correspondant est résolu. Cette heuristique est appelée *cut and fix* quand le modèle mathématique est amélioré (réduit) par des coupes ou par une reformulation. Une application de cette méthode sur un problème de dimensionnement

de lots peut être consultée dans [202].

- *Fix and relax* (ou *relax and fix*). Cette méthode décompose le problème en plusieurs sous-problèmes, chacun avec un ensemble de variables binaires fixées, organisées en trois groupes. Le premier groupe est résolu de façon optimale, le deuxième est relâché et, dans le troisième groupe, les variables binaires sont fixées aux valeurs d'une itération précédente. Un travail utilisant cette heuristique pour résoudre un problème de dimensionnement de lots est présenté dans [24].
- *Fix and optimize* (ou *exchange* dans [185]). C'est une variante de la méthode *fix and relax*, avec la différence que le groupe de variables relâchées n'est pas considéré, permettant donc de déterminer une solution réalisable à chaque itération. Cette méthode est utilisée pour résoudre un problème de dimensionnement de lots dans [143].
- Heuristiques d'arrondi. Le but est de réduire la difficulté de résolution du modèle, notamment associée aux variables binaires. Ainsi, le problème est relâché par une relaxation linéaire, où les variables binaires prennent des valeurs continues entre 0 et 1. Ces valeurs sont ensuite arrondies par rapport à un seuil. Ce type d'heuristiques est utilisé pour résoudre un problème de dimensionnement de lots dans [154].

Ces heuristiques ont été largement étudiées pour la résolution de problèmes de dimensionnement de lots avec variables de setup. D'autres stratégies de résolution basées sur la programmation mathématique sont les reformulations et les inégalités valides. Les reformulations consistent à modifier le modèle mathématique, en redéfinissant certaines variables et contraintes, avec le but de rendre le modèle plus facile à résoudre ou de générer de meilleures bornes. Deux reformulations typiques en dimensionnement de lots, spécifiquement pour le problème *Capacitated Lot Sizing Problem*, que nous décrivons plus tard dans ce chapitre, sont : le modèle du plus court chemin [216] et le modèle de localisation de sites de production [207]. Les inégalités valides sont des contraintes qui sont ajoutées pour réduire la taille de l'espace des solutions, permettant d'obtenir des meilleures bornes et de converger plus rapidement vers la solution finale. Il existe trois types d'inégalités valides, celles utilisées pour générer dynamiquement des coupes éliminant les solutions non-entières, celles du B&C et celles du C&B.

Pour plus de détails sur les méthodes basées sur des programmes mathématiques en nombres entiers mixtes (MIP), le lecteur peut consulter [185].

Les **heuristiques Lagrangiennes** sont des méthodes itératives utilisant le principe de la relaxation Lagrangienne, qui décompose, en relâchant certaines contraintes, le problème original en plusieurs sous-problèmes plus faciles à résoudre. Dans cette catégorie, nous trouvons les méthodes suivantes :

- Relaxation Lagrangienne. Les contraintes compliquées sont relâchées et, à chaque itération, les contraintes non respectées sont pénalisées dans la fonction objectif, au moyen de multiplicateurs qui peuvent être mis à jour par différentes méthodes, dont la plus utilisée est celle des sous-gradients [69]. La solution correspond à une borne inférieure, qui évolue itérativement jusqu'à converger vers le dual Lagrangien. L'idée de cette heuristique est donc de proposer une solution qui nécessite peu de modifications pour être optimale, sans avoir à résoudre le problème original, mais des sous-problèmes plus faciles. Des modifications systématiques de la solution peuvent être réalisées tout au long de la procédure, permettant de proposer des solutions réalisables, i.e. des bornes supérieures, qui peuvent être utilisées pour guider l'évolution des multiplicateurs. La relaxation Lagrangienne a été largement utilisée pour la résolution de problèmes de planification (voir par exemple [221]) et d'ordonnement (voir par exemple [37]). Pour plus de détails sur cette méthode, le lecteur peut consulter [146].
- Décomposition Lagrangienne (ou relaxation Lagrangienne augmentée). Elle diffère de la relaxation Lagrangienne par le fait qu'aucune contrainte du problème original n'est relâchée, mais le problème est décomposé en plusieurs sous-problèmes par duplication de variables, chaque sous-problème incluant un sous-ensemble de contraintes. Pour garantir la faisabilité des solutions, les variables dupliquées sont égalisées aux variables originales, au moyen de contraintes couplantes, qui sont relâchées. Une application de cette technique sur un problème d'ordonnement peut être consultée dans [114]. Pour ce qui est des problèmes de dimensionnement de lots, le lecteur peut consulter [218].

Les méthodes de **décomposition** et d'**agrégation** transforment le problème original en plusieurs sous-problèmes de taille réduite et plus faciles à résoudre. D'une part, les approches de décomposition séparent le problème en plusieurs sous-problèmes détaillés. D'autre part, les approches d'agrégation réduisent le niveau de détail du modèle, en rassemblant certaines données (par exemple, la demande ou la capacité). Dans les problèmes de dimensionnement de lots, la décomposition peut être faite à partir des produits [216], du temps (périodes de planification) [2] et des ressources, tandis que l'agrégation se fait seulement à partir des produits [177] et des ressources [32]. La construction d'une solution réalisable se fait au moyen de procédures itératives, permettant de modifier et de faire évoluer les valeurs des variables de décision.

Les **métaheuristiques** sont des procédures utilisées pour guider l'exploration de l'espace des solutions d'un problème complexe, de façon à trouver une solution réalisable, qui offre un bon compromis entre effort de calcul et qualité de la solution. La performance d'une métaheuristique dépend de la combinaison de deux critères opposés : la diversification et l'intensification. Le premier fait référence à l'exploration de différentes branches de l'espace des solutions, et le deuxième correspond à l'exploitation des régions prometteuses. L'intensification est une étape de recherche locale, dans laquelle on cherche à améliorer la qualité de la solution. La diversification permet de sortir de zones très explorées, pour en visiter d'autres, en évitant de rester sur la région d'un optimum local. Les métaheuristiques les plus utilisées dans les problèmes de planification et d'ordonnancement de la production sont les suivantes :

- Recherche locale. L'exploration de l'espace de solution est focalisé sur une région. Des changements systématiques sur certaines variables du problème permettent d'obtenir une nouvelle solution à chaque itération, à partir d'un voisinage de solutions candidates. Ainsi, la solution déterminée à la solution k fait partie des solutions candidates à l'itération $k-1$. La méthode standard s'arrête lorsqu'il n'est plus possible d'améliorer la solution après un certain nombre d'itérations. La solution correspond à un optimum local. Des exemples d'utilisation de la recherche locale pour des problèmes de planification et d'ordonnancement peuvent être consultés dans [101] et [224], respectivement.
- Recuit simulé. C'est une méthode recherche globale, proposée par Kirkpatrick *et al.* [137], en 1983, et par Černý [228], en 1985, basée sur le processus du recuit en métallurgie, qui consiste à augmenter la température d'un métal (typiquement l'acier), jusqu'à ce que la matière atteigne un état spécifique (avant la fusion), et ensuite à le refroidir lentement à température contrôlée, pour que le matériau acquière des propriétés mécaniques spécifiques. L'algorithme simule le mouvement des atomes dans le matériau en fonction de la température, chaque mouvement permettant de définir une nouvelle solution. Le fait d'augmenter la température rend les atomes plus dynamiques (haut niveau d'énergie), et le refroidissement lent les rend statiques (bas niveau d'énergie). Pour chaque solution, il y a un voisinage de solutions candidates, et le but est de minimiser le niveau d'énergie. Néanmoins, pour éviter de rester sur la région d'un optimum local, alterner le refroidissement avec une augmentation de la température permet de diversifier l'exploration. Un travail utilisant le recuit simulé pour résoudre un problème de dimensionnement de lots est présenté dans [176]. Quant aux problèmes d'ordonnancement, une étude appliquant cette technique est présentée dans [226]. Pour plus de détails sur cette méthode, le lecteur

peut consulter [59].

- Recherche taboue. C'est une méthode de recherche globale, proposée par Glover [81], en 1985. Elle combine le principe de la recherche locale (intensification), à travers l'exploration d'un voisinage défini par la solution courante, et la diversification, à travers une structure de *mémoire*, gérée par une *liste taboue*, qui interdit l'accès à certains voisins préalablement explorés, afin d'explorer des nouvelles branches de l'espace des solutions. La liste taboue peut être vue, en effet, comme un vecteur gardant par ordre d'ancienneté les derniers mouvements effectués. Ils existe trois catégories de mémoire : à court terme, à moyen terme et à long terme. La mémoire à court terme est contrôlée par la taille de la liste taboue. Ainsi, une fois qu'un critère d'expiration (souvent le remplissage de la liste taboue) est atteint, l'élément le plus ancien de la liste est libéré, pouvant être réincorporé à des futurs voisinages. La mémoire à moyen terme permet de diriger l'intensification vers des régions prometteuses de l'espace de recherche, et la mémoire à long terme permet de conduire la recherche vers de nouvelles régions, lorsque l'exploration est bloquée dans une région sous-optimale, en tenant compte du critère d'aspiration, qui permet de faire sortir de la liste taboue des mouvements jugés intéressants. Des critères d'aspiration peuvent être : la meilleure solution sur les k dernières itérations, le meilleur voisin, une solution différente des solutions existantes, une solution similaire aux solutions existantes et l'influence sur l'évolution de la solution. Grâce à la mémoire, et contrairement à la recherche locale, les solutions trouvées par la recherche taboue à certaines itérations peuvent être détériorées par rapport à la solution précédente, évitant la convergence vers un optimum local. Le degré de diversification dépend, dans une grande mesure, de la taille de la liste taboue, qui sert à éviter des explorations cycliques. Des exemples de travaux utilisant la recherche taboue pour résoudre des problèmes de planification et d'ordonnancement sont [93] et [232].
- Recherche avec voisinage variable (VNS ou Variable Neighborhood Search) [104]. C'est une méthode de recherche globale, proposée par Mladenović [166], en 1997. Elle se déroule en trois étapes principales : agitation, recherche locale et mouvement. Durant la première étape, une solution du voisinage de la solution courante est sélectionnée au hasard. S'il n'y a pas d'amélioration, la taille du voisinage est augmentée. À partir de la solution voisine, une recherche locale est réalisée. Si celle-ci conduit vers une meilleure solution, le mouvement est effectué, et on construit un nouveau voisinage pour répéter la procédure. Sinon, on revient directement à la première étape. La procédure est appliquée de façon

itérative. Des approches de résolution basée sur le VNS sont présentées dans [20] pour la résolution d'un problème de dimensionnement de lots, et dans [191] pour résoudre un problème d'ordonnement.

- Algorithmes génétiques. Ils se basent sur le principe biologique de la sélection naturelle. La procédure démarre avec une population d'individus, chacun avec un code génétique, qui représente les chromosomes. À chaque itération, une nouvelle génération d'individus est obtenue à partir de trois facteurs stochastiques : la sélection, le croisement et la mutation. Le meilleur chromosome généré est décodé, représentant la meilleure solution. Contrairement aux autres métaheuristiques décrites, les algorithmes génétiques ne comportent pas une étape de recherche locale. Une vue d'ensemble sur des algorithmes génétiques utilisés pour la résolution de problèmes de planification est présentée dans [98]. Un exemple d'application d'un algorithme génétique sur un problème d'ordonnement est présenté dans [180].

D'autres métaheuristiques moins utilisées en planification et ordonnancement sont : l'optimisation par colonies de fourmis [58], qui se base sur le comportement de fourmis durant la recherche d'un chemin entre leur colonie et une source de nourriture, les algorithmes mémétiques [172], qui combinent des heuristiques de recherche locale avec les facteurs de croisement des algorithmes génétiques, la méthode *threshold accepting* [63] qui est une variante du recuit simulé, la recherche adaptative gloutonne (GRASP) [66], qui comporte une étape de construction gloutonne et une étape de recherche locale. Pour des plus amples détails sur les métaheuristiques présentées ici et d'autres, le lecteur peut consulter [212].

Les **heuristiques gloutonnes spécifiques au problème** sont classées en deux catégories : les heuristiques constructives et les heuristiques d'amélioration. Elles ont comme point commun qu'à chaque étape, le meilleur élément (parmi un ensemble d'éléments pouvant constituer une solution) est sélectionné. Le critère de sélection est la valeur de la fonction objectif. Dans les heuristiques constructives [99], la solution initiale est vide et, à chaque étape, un nouvel élément est ajouté à la solution. C'est le cas, par exemple, de *l'algorithme de liste* en ordonnancement, qui suit des règles de priorité (plus longue durée de fabrication ou LPT, plus courte durée de fabrication ou SPT, plus proche délai de livraison ou EDD, etc.) [109] pour construire une solution. En ce qui concerne les heuristiques d'amélioration [74], les solutions sont déterminées à partir de solutions initiales, réalisables ou non réalisables, en réalisant plusieurs mouvements.

2.3 Problèmes de dimensionnement de lots

Comme introduit dans le chapitre précédent, la production par lots est une stratégie pour planifier la production dans les systèmes à flux discrets. Le dimensionnement de lots est normalement réalisé au niveau tactique sur un horizon de planification de plusieurs périodes (en général jours ou semaines), et consiste à décider des quantités de production par produit et par période, afin de satisfaire la demande au moindre coût possible. Les quantités de produit sont appelées « lots » et sont calculées avec le but de regrouper des demandes de différentes périodes pour produire de manière anticipée, si cela permet de réduire les coûts de setup (coûts de lancement de la production) ou de production. Cependant, le fait de produire en avance génère des inventaires et, si les coûts de stockage sont élevés, il peut être plus intéressant de répartir la production sur plusieurs périodes (ce qui induit des setups) et de stocker moins. Le problème de dimensionnement de lots est donc souvent équivalent à trouver un compromis entre les coûts de setup et les coûts de stockage. D'autre part, si les ruptures de stocks sont autorisées, les coûts associés sont à considérer ; et si les ventes perdues sont autorisées, les coûts associés sont aussi à intégrer dans le processus de décision.

Les premières politiques utilisées pour dimensionner les lots consistaient, entre autres, à fixer une taille de lot constante arbitrairement, suivre la demande période par période (sans stockage de produit fini) ou fixer une quantité périodique (dans des scénarios à demande dynamique). Le premier effort pour optimiser les tailles de lots a été proposé par Harris [107] en 1913, avec la création du modèle EOQ (Economic Order Quantity ou quantité économique de commande), qui détermine une taille de lot fixe (constante) optimale en minimisant les coûts de fabrication, de stockage et de commande, dans un système à demande constante, mono-produit et sans contraintes de capacité.

Le premier algorithme permettant d'optimiser le problème de dimensionnement de lots avec demande dynamique a été proposé par Wagner et Whitin [231]. Cet algorithme (dans la suite de ce manuscrit, méthode WW) en temps quadratique ($O(T^2)$, où T est le nombre de périodes) permet de calculer des tailles de lots variables par période pour résoudre le problème mono-produit sans contraintes de capacité, et est à la base de beaucoup de travaux en dimensionnement de lots. Néanmoins, à l'époque de son développement, la puissance de calcul des ordinateurs ne permettait pas de déployer cette méthode. Plusieurs heuristiques ont été proposées pour ce problème, dont celle de Silver et Meal [206]. Plus tard, Wagelmans *et al.* [230], Aggarwal et Park [9], Federgruen et Tzur [65] et Van Hoesel *et al.* [225] ont proposé des versions améliorées de l'algorithme de programmation dynamique WW, de complexité $O(T \log T)$.

L'inconvénient de ces méthodes est que les contraintes de capacité ne sont pas considérées, ce qui implique que les solutions ne sont pas, dans beaucoup de cas, réalisables au niveau opérationnel. Cependant, l'étude des problèmes sans contraintes de capacité a jusqu'à présent suscité beaucoup d'intérêt dans la littérature (voir par exemple [5], [242] et [240]), car ils sont plus faciles à résoudre et parce que des problèmes plus complexes peuvent être décomposés et ramenés à l'étude de problèmes sans contraintes de capacité. Les problèmes avec contraintes de capacité ont été étudiés depuis les années 70, avec le but de réduire l'écart de faisabilité entre les décisions de planification prises à moyen terme et à court terme. Cependant, la grande majorité des modèles développés ne garantissent des solutions réalisables que pour des scénarios très précis, avec des ateliers avec peu de ressources. De plus, la capacité n'est en général pas modélisée de manière exacte, ce qui fait que les solutions sont sous-optimales. Un état de l'art sur les problèmes de dimensionnement de lots avec contraintes de capacité peut être consulté dans [61].

Un des facteurs les plus importants à considérer dans un problème de dimensionnement de lots avec contraintes de capacité est la structure des setups. Comme expliqué par Karimi *et al.* [129] et Amorim *et al.* [22], le fait d'inclure les setups a un impact direct sur l'utilisation de la capacité, et augmente considérablement la difficulté de résolution du modèle. Les auteurs classent les structures de setups en *setups simples* et *setups complexes*, ces derniers étant sous-divisés en trois catégories, auxquelles nous ajoutons les setups débordants, étudiés dans des travaux récents (voir par exemple [161] et [211]).

- **Setup simple** : indépendant de la séquence et des décisions des périodes précédentes.
- **Setup complexe** :
 - **Setup reportable** (*setup carryover*) : si le produit/opération à traiter sur une machine au début de la période l est le même que celui traité sur la même machine à la fin de la période $l - 1$, il n'y a pas de setup.
 - **Setup par famille** (*family setup*) : le coût et le temps de setup entre deux produits de la même famille (*minor setup*) sont plus petits que ceux entre deux produits de familles différentes (*major setup*).
 - **Setup débordant** (*setup crossover*, *period overlapping setup* ou *setup splitting*) : un setup peut être interrompu à la fin d'une période et repris au début de la période suivante, sans générer des coûts ni des temps additionnels.
 - **Setup dépendant de la séquence** : le coût ou le temps de setup générés par un produit sur une machine varient en fonction du produit précédemment traité.

Un exemple illustrant les setups complexes est présenté dans la Figure 2.2, où A , B , C , D correspondent à des produits/opérations à traiter sur une machine, sur un horizon de planification de 5 périodes. Les blocs restants correspondent aux durées de lancement (temps de setup). Nous pouvons clairement identifier un setup reportable, un setup débordant et un setup mineur (par famille). De plus, en comparant le temps de setup pour traiter A dans les périodes 4 et 5, nous observons qu'il s'agit dans ce cas de setups dépendants de la séquence, car le temps de setup de la machine pour produire A varie en fonction du produit précédemment traité (B à la période 3 et C à la période 4). Dans le cas d'un setup simple, les deux durées de lancement seraient identiques.

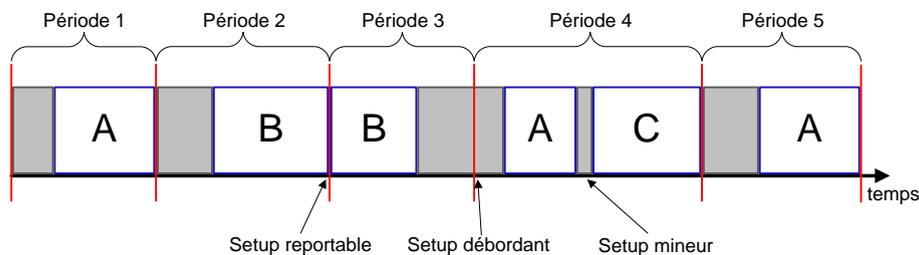


FIGURE 2.2 – Exemple de structures de setups complexes

Les problèmes de dimensionnement de lots peuvent être classés selon différentes caractéristiques. Les caractéristiques les plus utilisées dans la littérature sont : la capacité de production, le comportement de la demande, la longueur des périodes, le nombre de machines, le nombre de produits, le nombre de sites de production et le nombre de niveaux de nomenclature. Un état de l'art sur les problèmes de dimensionnement à un seul produit peut être consulté dans [33]. À partir de la nomenclature, les problèmes de dimensionnement de lots sont classés en : problèmes à un seul niveau et problèmes à plusieurs niveaux. Comme expliqué dans la Section 1.3 du Chapitre 1, le premier groupe correspond aux systèmes de production dans lesquels il n'y a pas de liens entre les produits, c'est-à-dire qu'aucun produit n'est composant d'un autre. En revanche, dans les problèmes multi-niveaux, certains produits sont des composants et d'autres des composés. On parle donc, dans ce dernier cas, de systèmes d'assemblage et de systèmes généraux. Les figures 2.3 et 2.4 présentent une classification des problèmes de dimensionnement de lots à un et plusieurs niveaux, respectivement.

Les problèmes de dimensionnement de lots avec une nomenclature à un seul niveau ont été nettement plus étudiés que les problèmes à plusieurs niveaux. L'avantage des problèmes

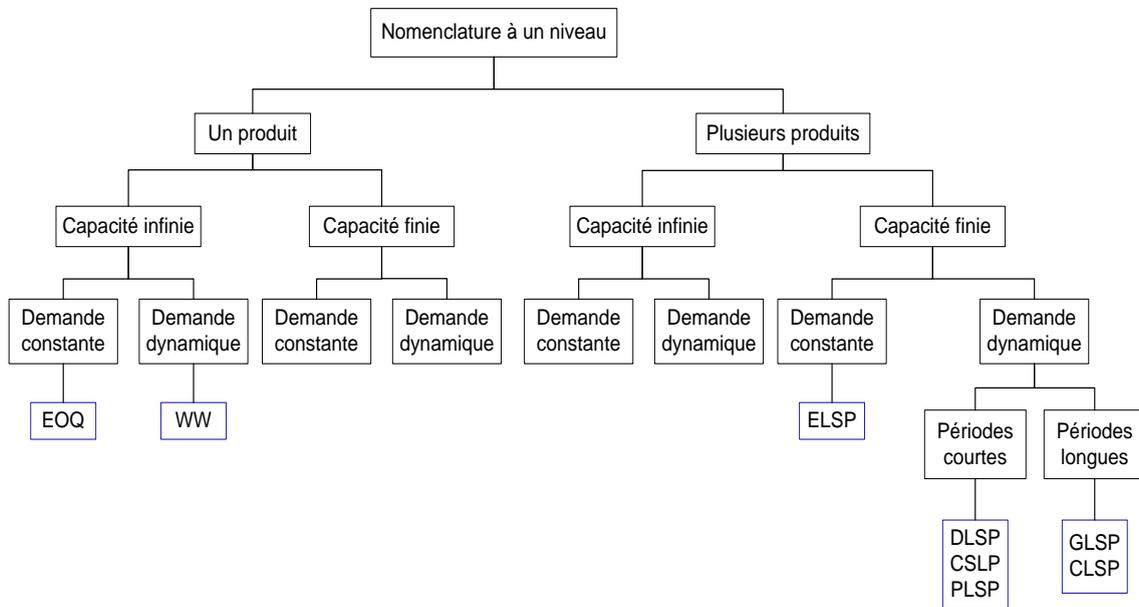


FIGURE 2.3 – Classification des problèmes de dimensionnement de lots à un seul niveau [233]

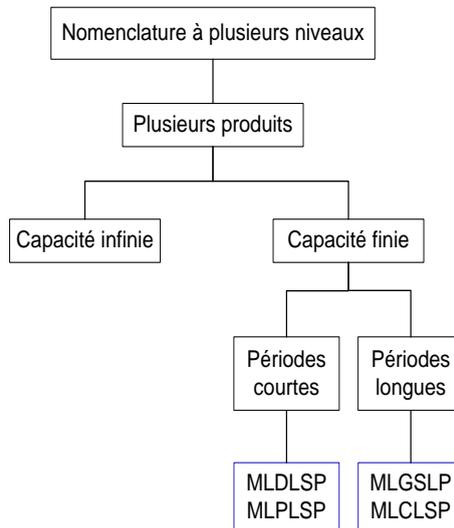


FIGURE 2.4 – Classification des problèmes de dimensionnement de lots à plusieurs niveaux [233]

mono-niveau, du point de vue de l'effort de calcul, est que leur modélisation comporte moins de variables de décisions (et parfois moins de contraintes) et moins de liens entre les variables de décision. Néanmoins, les systèmes de production associés à de tels modèles sont limités. En effet, dans les problèmes mono-niveau, les demandes de chaque produit sont indépendantes de la fabrication des autres produits, car les produits ne sont pas liés entre eux, i.e. aucun produit n'est

composant d'un autre. Cependant, même sous ces hypothèses, la résolution des problèmes de dimensionnement de lots avec contraintes de capacité à un seul niveau est souvent \mathcal{NP} -difficile. La difficulté de résolution est en particulier associée aux contraintes de capacité, et varie selon la configuration de l'atelier de production. Ainsi, les problèmes à une seule ressource sont plus faciles à résoudre que ceux à ressources multiples, car la capacité exacte du système est connue, les contraintes de séquençement détaillées ne sont pas indispensables et les modèles comportent moins de variables binaires. En effet, il n'y a pas de décisions d'affectation des opérations aux ressources à considérer ni de temps d'attente (la machine est toujours occupée), ce qui n'est plus vrai dans le cas multi-ressources. Les problèmes mono-ressource sont de loin les plus étudiés dans la littérature. D'ailleurs, les versions standard des modèles de dimensionnement de lots correspondent à des systèmes de production à une seule machine. Récemment, des extensions à des cas multi-ressources, principalement à des ateliers à machines en parallèle, ont été étudiées. Néanmoins, les hypothèses considérées dans la plupart de ces modèles ne sont applicables à des systèmes plus complexes, comme les ateliers de type flow-shop et job-shop, dont la modélisation nécessite des contraintes d'ordonnancement ou de capacité détaillée. Dans les problèmes de dimensionnement de lots avec contraintes de capacité, des contraintes d'ordonnancement typiquement considérées sont celles modélisant les temps et/ou coûts de setup dépendants des séquences. Un état de l'art sur les problèmes avec setups dépendants des séquences peut être consulté dans [244]. Bien que ces contraintes sont parfois importantes en pratique, elles ne suffisent en général pas pour garantir la faisabilité du plan de production au niveau ordonnancement.

Selon la longueur des périodes, les problèmes de dimensionnement de lots avec contraintes de capacité sont classés dans la littérature en problèmes à courtes périodes et problèmes à longues périodes. Nous nous intéressons, dans le reste de ce chapitre aux problèmes dans lesquels la demande est dynamique, c'est-à-dire varie dans le temps. Il existe tout de même le problème de dimensionnement de lots avec contraintes de capacité et demande constante, appelé *Economic Lot Scheduling Problem* (ELSP), qui est une extension du modèle EOQ au cas multi-produits et est \mathcal{NP} -difficile [115]. Le lecteur intéressé par ce problème peut consulter [187] et [103].

Nous discutons, dans ce qui suit, les principaux travaux scientifiques traitant les problèmes avec contraintes de capacité et à demande dynamique. Nous suivons la classification par longueur des périodes (problèmes à courtes et à longues périodes), en différenciant les travaux par type de nomenclature (mono-niveau et multi-niveaux) et par configuration de l'atelier de production (mono-ressource et multi-ressources).

2.3.1 Problèmes à courtes périodes

Les problèmes de dimensionnement de lots à courtes périodes sont caractérisés par un horizon de planification avec des périodes de courte durée (par exemple des heures ou des jours). D'autres caractéristiques ou hypothèses sont les suivantes :

- Atelier à une seule ressource,
- Limitations dans le nombre de lancements de production par période,
- Limitations dans l'utilisation de la capacité,
- Toutes les périodes ont la même longueur,
- Horizon de planification fini,
- Ordonnancement période par période.

Des généralisations des problèmes à courtes périodes pour des ateliers à plusieurs ressources ont aussi été proposées dans la littérature, principalement pour le cas de machines en parallèle.

Pour éviter d'avoir à décider du séquençement et de l'ordonnancement des opérations, des périodes de courtes durées et des limites sur le nombre de setups par périodes sont fixées. De cette façon, les produits ne sont pas en « compétition » sur l'utilisation des ressources, et le plan de production respecte la capacité. À cause de ces limitations, le nombre de variables binaires devient très important et, par conséquent, seuls quelques problèmes de petite taille (systèmes à une ou deux ressources et peu de produits) peuvent être résolus en des temps de calcul raisonnables. Ces problèmes sont en fait \mathcal{NP} -difficiles.

Le groupe de problèmes à courtes périodes est composé des problèmes ci-dessous :

- Discrete Lot-sizing and Scheduling Problem (DLSP),
- Continuous Setup Lot-sizing Problem (CSLP),
- Proportional Lot-sizing and Scheduling Problem (PLSP).

2.3.1.1 Discrete Lot-sizing and Scheduling Problem

La principale caractéristique de ce problème est qu'au maximum un produit peut être fabriqué par période (i.e. un setup par période). De plus, toute la capacité de la période est utilisée par la production de l'article si l'on décide de le produire. C'est ce que l'on appelle une politique de production de tout ou rien (« all or nothing »). L'horizon de planification est divisé en plusieurs périodes de même longueur. En ce qui concerne la capacité, il y a une contrainte par machine à chaque période. Un état de l'art sur le DLSP peut être consulté dans [70]. La version multi-niveaux du DLSP est connue sous l'appellation « Multi-Level Discrete Lot-sizing and Scheduling Problem » (MLDLSP).

Plusieurs variantes du DLSP ont été étudiées dans la littérature. Pour plus de détails, le lecteur peut consulter le travail de Salomon *et al.* [198], qui décomposent le problème selon :

- Le comportement de la ligne de production,
- Le nombre de machines,
- Le nombre de produits,
- La structure du coût de setup,
- La structure du coût de production,
- La structure du temps de setup.

Les auteurs analysent la complexité des différentes variantes, lorsque l'on veut déterminer soit une solution réalisable, soit une solution optimale. Comme conclusion, trouver la solution optimale est \mathcal{NP} -difficile dans tous les cas, sauf pour le problème à machines parallèles identiques sans coûts de setup, qui est résolu de façon optimale en temps polynomial. Une autre étude sur la complexité du DLSP est présentée par Brüggemann et Jahnke [34], qui corrigent certains résultats trouvés dans [198], concernant les preuves de complexité.

Plusieurs formulations et méthodes de résolution basées sur des modèles MIP ont été proposées pour résoudre le DLSP à une seule machine. Salomon *et al.* [199] développent une méthode exacte pour résoudre le problème avec coûts et temps de setup dépendants des séquences, dénoté DLSPSD. Ils transforment le modèle mathématique en un problème de *voyageur de commerce* ([23]) avec fenêtres de temps. Gicquel *et al.* [79] proposent un modèle MIP pour le problème avec contraintes de coûts et de temps de setups dépendant des séquences. Ils formulent aussi le problème comme celui du voyageur de commerce avec fenêtres de temps et utilisent un algorithme de programmation dynamique pour le résoudre. Les auteurs montrent la similarité du problème avec le problème de séquençement de batchs. Gicquel *et al.* [78] proposent une nouvelle formulation du problème avec coûts de changement dépendant des séquences, en utilisant une structure de produit multi-attributs, un attribut étant une caractéristique physique (par exemple, la couleur, les dimensions, etc.). Les temps de changements ne sont pas considérés. La structure multi-attributs permet de rassembler les produits partageant certains attributs, de façon à utiliser des temps de changement par groupe, ce qui permet de diminuer le nombre de variables du modèle mathématique. Ils utilisent l'approche de Wolsey [237] pour obtenir des inégalités valides permettant de renforcer la modélisation. Des expérimentations sont menées pour comparer la performance des deux approches.

En ce qui concerne les systèmes à plusieurs ressources, Jans et Degraeve [121] proposent un modèle et une procédure de résolution pour le DLSP à machines parallèles chez un fabricant international de pneus. Le problème est décomposé par programmation dynamique en un programme maître, qui minimise le coût total, et un ensemble de sous-problèmes par pneu.

L'approche de résolution est basée sur un algorithme de génération de colonnes. Un travail récent pour un problème avec des machines parallèles identiques a aussi été proposé par Gicquel *et al.* [80]. Les auteurs changent la formulation standard, en remplaçant les variables individuelles associées à chaque machine par des variables agrégées, et ils comparent les formulations à l'aide d'un solveur commercial.

Pour ce qui est du cas multi-niveaux, le MLDLSP a rarement été étudié dans la littérature. Un des seuls travaux est celui de Jordan et Koppelman [123], qui étudie le problème à une seule machine et propose deux façons de modéliser le problème. Le premier modèle correspond au MLDLSP standard (généralisation du DLSP) et le deuxième correspond à un problème de séquençement de batchs (MLBSP ou Multi-Level Batch Sequencing Problem) [122]. Le problème est résolu, en utilisant les deux modèles. Une heuristique de regrets aléatoires [133] est implémentée pour résoudre le modèle MLDLSP, et un *branch and bound* est utilisé pour résoudre le modèle MLBSP. En comparant le temps d'exécution et la qualité des solutions, les auteurs concluent que modéliser le problème comme un MLBSP offre une meilleure performance.

2.3.1.2 Continuous Setup Lot-sizing Problem

Le principe du CSLP est le même que celui du DLSP (on peut produire au maximum un article par période), sauf qu'il n'est pas obligatoire d'utiliser toute la capacité de production de la période pour produire un article (i.e. l'hypothèse « all or nothing » n'est pas considérée). Cette stratégie permet donc de diminuer les tailles des lots, et par conséquent de réduire les niveaux de stock. Donc, les solutions obtenues sont potentiellement meilleures que celles du DLSP. Ce problème, qui est \mathcal{NP} -difficile [73], a aussi été appelé « Deterministic Dynamic Product Cycling Problem ». De manière générale, le CSLP n'a pas généré beaucoup d'intérêt dans la littérature, et à notre connaissance, il n'a jamais été étendu au cas multi-niveaux.

Karmarkar et Schrage [131] décrivent plusieurs formulations utilisées pour modéliser le CSLP à une seule machine sans coûts de setup, et utilisent une relaxation Lagrangienne pour calculer des bornes inférieures, qui sont ensuite utilisées dans un algorithme de *branch and bound*.

Karmarkar *et al.* [130] présentent un modèle mathématique pour le problème à une seule machine avec coûts de démarrage des machines (*startup costs*) et coûts de maintien de la machine en fonctionnement (*reservation costs*). Ils résolvent le problème sans contraintes de capacité par un algorithme de programmation dynamique, et utilisent une procédure de *branch and bound* avec une relaxation Lagrangienne pour résoudre le problème avec contraintes de capacité. Le même problème est résolu par Hindi [112], au moyen d'une recherche taboue. D'autre part, Sandbothe [200] propose une heuristique dédiée pour le problème avec coûts

de setup. Vanderbeck [227] étudie le problème à une seule machine avec coûts de setup et de démarrage de la machine. Il résout le problème à un seul produit avec un algorithme de génération de colonnes, et le problème à plusieurs produits par une procédure de programmation dynamique.

Constantino [42] propose des inégalités valides pour rendre la formulation à une seule machine plus robuste et permettre d'obtenir des bornes inférieures (par relaxation linéaire) de meilleure qualité. Finalement, il résout le problème au moyen d'un algorithme de *branch and cut*.

Almada-Lobo *et al.* [18] ont récemment étudié le CSLP avec coûts et temps de setup dépendant des séquences, dans un atelier multi-ressources. Dans leur travail, les auteurs proposent deux modèles de programmation linéaire en variables mixtes avec des inégalités valides. De Matta et Guignard [50] étudient aussi un problème multi-ressources, mais les temps de setup ne sont pas considérés.

2.3.1.3 Proportional Lot-sizing and Scheduling Problem

Le PLSP est une généralisation (ou variante) du CSLP, dans la mesure où la capacité restante à chaque période peut être utilisée pour fabriquer un deuxième produit, sous la condition que les deux produits soient traités sur la même ressource. Cependant, un seul setup est autorisé par période, c'est-à-dire que le deuxième produit à fabriquer dans la période t est le premier à la période $t + 1$. Le PLSP, dont des extensions sont étudiées par Drexel et Haase [60], permet d'obtenir des meilleures solutions (en termes de coûts) que le CSLP et le DLSP, en utilisant mieux la capacité des périodes [136]. Ce problème a aussi été étudié pour le cas multi-niveaux, et il est nommé « Multi-Level Proportional Lot-sizing and Scheduling Problem » (MLPLSP).

Drexel et Haase [60] présentent un modèle mathématique et une heuristique de regrets aléatoires (*randomized-regret-based biased sampling method*) pour le problème mono-niveau à une seule machine. Suerie [211] présente deux nouveaux modèles, en considérant des setups débordants.

En ce qui concerne les ateliers multi-ressources, Drexel et Haase [60] présentent un modèle mathématique standard. Une extension de cette formulation, en considérant un opérateur commun pour réaliser les opérations de setup est présentée dans [214]. Le problème avec machines parallèles identiques a été étudié par Kaczmarczyk [127], qui propose plusieurs modélisations, en remplaçant les contraintes binaires par des variables entières, permettant d'agréger les machines selon certaines conditions (par exemple, le nombre de machines disponibles pour traiter un produit particulier).

Pour ce qui est des problèmes multi-niveaux, Kimms [132] propose deux heuristiques pour résoudre le MLPLSP mono-ressource. La première est basée sur des regrets aléatoires, comme dans [60], et la deuxième est une recherche taboue qui utilise la notion de graphe disjonctif. Les deux heuristiques construisent les plans de production d'avant en arrière, en affectant au maximum deux produits à chaque période. Stadtler [208] présente un modèle avec des délais d'obtention nuls et un deuxième modèle avec temps de setups débordants et des contraintes de tailles de batches (groupe de produits de la même famille). Il résout un problème de l'industrie pharmaceutique au moyen d'un solveur standard.

Le MLPLSP à plusieurs machines a surtout été traité à travers l'étude de modèles mathématiques, permettant de tenir compte de différentes configurations d'ateliers de production. Un travail de base important est celui de Kimms et Drexl [136], où les auteurs discutent des principales caractéristiques du PLSP, en mettant l'accent sur le MLPLSP multi-ressources, pour lequel un modèle MIP est expliqué en détail. Puis, dans [135], les mêmes auteurs présentent différentes formulations MIP pour le MLPLSP multi-ressources, en faisant varier l'organisation des machines dans l'atelier de production (machines en parallèle, plusieurs machines par produit et machines partiellement renouvelables). Une méthode de résolution basée sur un algorithme génétique est proposée par Kimms [134], qui présente aussi une formulation MIP du problème à plusieurs ressources. Des résultats expérimentaux valident l'intérêt de l'approche face à la méthode hiérarchique classique MRP-II.

2.3.2 Problèmes à longues périodes

Dans les problèmes à longues périodes, la durée des périodes est souvent de l'ordre de quelques semaines ou de quelques mois. La caractéristique générale est que plusieurs produits peuvent être fabriqués par période sur une seule machine. Des extensions à des cas multi-ressources ont aussi été proposées. Contrairement aux problèmes à courtes périodes, qui suivent plus ou moins les mêmes hypothèses, les problèmes à longues périodes diffèrent substantiellement. Les deux problèmes les plus connus sont :

- Le General Lot-sizing and Scheduling Problem (GLSP),
- Et le Capacitated Lot-Sizing Problem (CLSP).

Le GLSP et le CLSP sont des problèmes \mathcal{NP} -difficiles, et ont beaucoup été étudiés car les limitations trouvées dans les problèmes à courtes périodes n'existent plus ou sont moins importantes. Le CLSP est le problème de dimensionnement de lots avec contraintes de capacité le plus étudié dans la littérature, et plusieurs états de l'art sont disponibles ([129], [189], [36]).

2.3.2.1 General Lot-sizing and Scheduling Problem

Dans le GLSP, l'horizon de planification est divisé en un certain nombre de macro-périodes (longues périodes), et chaque macro période est divisée en quelques micro périodes (courtes périodes) de longueur variable. Dans chaque micro-période, il est possible de produire uniquement un article, en utilisant l'hypothèse « all or nothing », comme dans le DLSP. Cependant, ce problème, qui est aussi \mathcal{NP} -difficile [71], requiert encore plus d'effort de calcul que le DLSP, car non seulement les tailles de lots sont déterminées, mais aussi la séquence des lots par macro période. La longueur de chaque micro période est une décision indirecte qui dépend de la taille de lot du produit associé. Dans la version standard, le nombre de micro périodes par macro période est fixe. Une généralisation du GLSP est sa version multi-niveaux, appelée « Multi-Level General Lot-sizing and Scheduling Problem » (MLGLSP).

Plusieurs procédures basées sur des méthodes de recherche locale ont été proposées pour résoudre le GLSP à une seule machine. Fleischmann et Meyr [71] présentent trois heuristiques de recherche locale basées sur la méthode *threshold accepting*. Meyr [162] propose une approche qui combine un algorithme de réoptimisation du problème dual avec la méthode *threshold accepting* et avec un recuit simulé, pour résoudre le problème avec temps de setup dépendant des séquences.

Le problème avec temps et coûts de setup dépendants des séquences a été étudié par Araujo *et al.* [24]. Les auteurs proposent un modèle MIP, avec prise en compte de la rupture de stock, pour un atelier mono-machine dans une fonderie, et ils résolvent le problème par une heuristique de *relax and fix*. L'heuristique est améliorée, en utilisant trois variantes de recherche locale (une heuristique de descente, une recherche de voisinage diminué et un recuit simulé). L'approche suit une structure d'horizon glissant, où seuls les jobs immédiats sont ordonnancés. Un autre cas industriel, modélisé comme un GLSP avec temps et coûts de setup dépendants des séquences, a été étudié par Ferreira *et al.* [68]. Dans leur travail, les auteurs proposent un modèle MIP, une approche de relaxation et plusieurs heuristiques de *relax and fix* pour résoudre le problème de production de boissons. Tiacci et Saetta [220] étudient aussi le problème avec temps et coûts de setup dépendant des séquences, pour lequel ils présentent trois formulations MIP, qui correspondent à des versions simplifiées du modèle détaillé. Les modèles sont basés sur une stratégie d'horizon glissant, où les séquences d'ordonnancement sont considérées uniquement pour la première période. Dans le premier modèle, les contraintes de coûts de setup dépendant des séquences sont considérées uniquement pour la première période ; le deuxième modèle inclut un terme approximatif associé aux futurs coûts de setup, sans détailler la séquence, et le troisième modèle inclut une contrainte additionnelle pour éviter de produire un article à la première période, si son inventaire initial est supérieur aux besoins de la première période. Ferreira *et al.*

[67] étudient un système de production de boissons constitué de deux étages de production. Ils proposent quatre formulations mono-étage basées, pour les deux premières sur le GLSP avec temps et coûts de setup dépendant des séquences, et pour les deux autres sur le problème du voyageur de commerce. Cette dernière stratégie donne des meilleurs résultats.

Toso *et al.* [222] étudient un système de production de produits alimentaires pour animaux. Ils développent deux variantes de la méthode de *relax and fix* et utilisent des formulations alternatives pour résoudre le problème. Les stratégies déployées sont plus avantageuses que la résolution exacte d'un modèle MIP général.

Un problème multi-ressources à été traité par Meyr [163], qui s'intéresse à un atelier à machines parallèles non identiques. L'auteur propose un modèle mathématique et une procédure de résolution qui combine un algorithme de recherche locale et un recuit simulé.

En ce qui concerne le cas à plusieurs niveaux, un des travaux modélisant le MLGLSP est celui de Fandel et Stammen-Hegene [64]. Les auteurs étudient le problème à plusieurs machines, généralisé pour une configuration de type job-shop, avec coûts de setup dépendants des séquences. La formulation proposée permet d'obtenir des solutions réalisables au niveau opérationnel pour des systèmes complexes. Un inconvénient, cependant, est que les délais d'obtention (ou livraison) des produits intermédiaires de la nomenclature ne sont pas considérés. Du point de vue de l'effort de calcul, l'inconvénient majeur est que le modèle mathématique comporte beaucoup de variables avec trois différents types de variables binaires, ce qui rend très difficile la résolution du problème. En effet, comme expliqué par les auteurs, seulement des problèmes de très petite taille (avec peu de produits et peu de périodes) peuvent être résolus. Un autre travail sur le MLGLSP à plusieurs machines est celui de Seeanner et Meyr [202], qui présentent un modèle MIP pour le problème avec coûts de setup dépendant des séquences, coûts de temps morts (inactivité), coûts d'achat externe, coûts de temps additionnel et coûts de stockage de produits semi-finis. Comme le modèle est très difficile à résoudre en un temps raisonnable, même pour des petites instances, les auteurs proposent et comparent plusieurs reformulations du problème, en utilisant trois heuristiques : *truncated MIP*, *LP and fix* et *relax and fix*. La première heuristique offre une meilleure performance, mais le problème reste soluble seulement pour des petites instances. Le même problème est étudié par Seeanner *et al.* [201]. Les auteurs proposent une heuristique combinant les principes de la *recherche de décomposition de voisinage variable* (VNDS) [105], qui est une variante de la métaheuristique VNS, et de la méthode *fix and optimize*. La performance de l'heuristique est comparée avec celle d'un solveur commercial, montrant la pertinence de la méthode, surtout lorsque la taille des problèmes augmente.

2.3.2.2 Capacitated Lot-Sizing Problem

Dans le CLSP il n’y a pas de limitations sur le nombre de setups par période, les périodes ne sont pas divisées en sous-périodes, et n’importe quelle fraction de la capacité disponible par période peut être utilisée. Bitran et Yanasse [31] ont montré que ce problème était \mathcal{NP} -difficile. Parmi les problèmes de dimensionnement de lots avec contraintes de capacité, le CLSP est le problème le plus étudié, car il tient compte de la capacité exacte et que le modèle mathématique est moins complexe. Le CLSP à plusieurs niveaux est nommé dans la littérature « Multi-Level Capacitated Lot-sizing Problem » (MLCLSP). Des états de l’art focalisés sur le CLSP sont réalisés dans [129], [189] et [36].

Comme dans les autres problèmes, le modèle mathématique standard du CLSP ne garantit des solutions réalisables que pour des systèmes de production mono-ressource. Le CLSP à une seule machine ne nécessite pas de décisions de séquençement, à moins que les temps et coûts de lancement de la production varient en fonction des séquences. Différentes extensions au cas multi-ressources ont aussi été proposées dans la littérature. La formulation du CLSP multi-ressources impose la considération des dates de début et de fin des opérations dans les contraintes de capacité, et rend le problème beaucoup plus difficile à résoudre. C’est pourquoi les problèmes multi-ressources étudiés restent typiquement associés aux ateliers à machines parallèles. Les ateliers de type flow-shop et de type job-shop sont très peu étudiés.

Comme pour les autres problèmes, les contraintes d’ordonnancement généralement étudiées dans le CLSP sont celles associées aux temps et/ou aux coûts de setup dépendant des séquences. Contrairement au GLSP, où ces contraintes sont inhérentes au problème (la séquence fait partie des décisions du modèle standard), dans le CLSP, ces contraintes créent une difficulté additionnelle très importante, surtout dans les cas de systèmes avec partage de ressources et à séquençement variable, comme les ateliers de type job-shop, car il n’y a pas de limites sur le nombre de setups par période.

Le CLSP à une seule machine avec des coûts de setup dépendant des séquences, noté CLSD, a été notamment introduit par Haase [100], sans tenir compte des temps de setup. Une méthode de résolution pour ce problème a été proposée par Shim *et al.* [205], en utilisant une heuristique à deux niveaux. Le problème avec temps et coûts de setup dépendants des séquences a été étudié par Haase et Kimms [102]. Dans leur travail, les auteurs présentent un modèle MIP considérant uniquement les séquences efficaces et utilisent un algorithme de *branch and bound* pour résoudre le problème. Une séquence est efficace s’il n’existe pas une autre séquence des mêmes produits avec un coût de setup inférieur. Almada-Lobo *et al.* [17] proposent un nouveau modèle pour le même problème. Kovács *et al.* [140] présentent aussi un autre modèle, et proposent une heuris-

tique pour résoudre des instances de grande taille avec des temps de calcul raisonnables. Gupta et Magnusson [99] étudient le même problème, mais avec setups débordants. Ils présentent une formulation MIP et proposent une heuristique de résolution. Les résultats sont meilleurs quand le nombre de produits est significativement plus grand que le nombre de périodes de l'horizon de planification. Néanmoins, Almada-Lobo *et al.* [19] montrent que ce modèle n'empêche pas la génération de sous-tours déconnectés, et ne garantit donc pas l'obtention de solutions réalisables. Pour éviter ce problème, les auteurs proposent des nouvelles contraintes à ajouter dans le modèle original. Une nouvelle formulation, incluant des setups débordants et dépendant des séquences, est présentée par Menezes *et al.* [161]. Les auteurs comparent ce modèle avec le GLSP, et ils prouvent que leur formulation donne des meilleurs résultats. Almada-Lobo et James [16] étudient aussi le problème avec temps et coûts de setup dépendant des séquences, et proposent deux heuristiques pour résoudre le problème : une recherche taboue et une recherche avec voisinage variable. La recherche taboue a aussi été mise en œuvre par Hindi [113], et la recherche à voisinage variable a été également utilisée par Almada-Lobo *et al.* [20] pour un problème de l'industrie du verre.

Une technique très utilisée pour résoudre le CLSP est la relaxation Lagrangienne. Ainsi, Thizy et Van Wassenhove [219] proposent une heuristique Lagrangienne basée sur la relaxation des contraintes de capacité et utilisant la méthode des sous-gradients. Diaby *et al.* [56] proposent une formulation MIP et des méthodes de résolution basées sur la relaxation Lagrangienne. Les résultats expérimentaux présentent des écarts par rapport à l'optimum inférieurs à 1% avec des temps de calculs petits. Diaby *et al.* [55] proposent une heuristique Lagrangienne pour résoudre des problèmes de grande taille. Millar et Yang [164] proposent deux procédures pour résoudre le CLSP avec rupture de stock, formulé comme un problème de transport : une décomposition Lagrangienne [97] et une relaxation Lagrangienne. Trigeiro *et al.* [223] étudient le CLSP avec temps de setup. Ils présentent une heuristique Lagrangienne, où les contraintes de capacité sont relâchées. Pour construire des solutions réalisables, une heuristique de lissage est implémentée.

Absi et Kedad-Sidhoum [4] étudient le problème avec des coûts de pénurie, qui sont considérés quand la demande est partiellement ou complètement perdue. Ils proposent des inégalités valides et présentent des algorithmes de coupes pour la mise en œuvre de ces inégalités.

Tempelmeier [213] étudie le problème avec demande aléatoire (SCLSP, Stochastic CLSP), où les demandes non satisfaites induisent des coûts de rupture. Une procédure combinant un algorithme de génération de colonnes et l'heuristique ABC_β (présentée dans [217]) est proposée. Helber *et al.* [111] étudient également le SCLSP, mais en incluant des stocks de sécurité. La formulation exacte du problème étant non linéaire, les auteurs proposent deux approximations

sous forme de modèles linéaires. Une heuristique de *fix and optimize* est utilisée pour résoudre les deux modèles.

Goren *et al.* [94] présentent une approche hybride combinant un algorithme génétique avec une heuristique de *fix and optimize* pour résoudre le CLSP avec setups reportables. Le même problème, en ajoutant des setups débordants a été étudié par Mohan *et al.* [170], qui présentent un modèle MIP et réalisent des expérimentations montrant l'importance de considérer cette structure de setup.

En ce qui concerne les problèmes multi-ressources, plusieurs travaux traitant le CLSP avec machines parallèles ont été étudiés dans la littérature (voir par exemple [221] et [156]). Cependant, l'un des seuls travaux à inclure des contraintes d'ordonnancement est celui de Mateus *et al.* [158]. Les auteurs étudient le problème avec des coûts et des temps de setup dépendant des séquences dans un atelier avec machines parallèles non reliées, et ils proposent un modèle mathématique avec ordonnancement détaillé. D'abord, le problème de dimensionnement de lots est résolu de manière exacte, en considérant la capacité agrégée, et le problème d'ordonnancement est ensuite résolu de manière sous-optimale, en utilisant une heuristique GRASP. Un inconvénient est que le problème d'ordonnancement est résolu période par période et, par conséquent, quand la charge de travail dépasse la capacité d'une période, des ruptures de stock sont considérées. Si l'ordonnancement était réalisé sur tout l'horizon de planification, il serait possible d'éviter la rupture de stocks dans plusieurs cas. James et Almada-Lobo [120] proposent une approche combinant plusieurs heuristiques basées sur un modèle MIP (extension du CLSP et du CLSD) pour résoudre le problème à plusieurs machines en parallèle, avec coûts et temps de setups dépendant des séquences. L'algorithme comporte une recherche locale et une étape de modification du voisinage. Ils mettent en évidence que les instances avec coûts de setup élevés sont plus difficiles à résoudre pour toutes les heuristiques. Le problème original est décomposé en plusieurs sous-problèmes pouvant être plus faciles à résoudre. Certaines variables de décision sont fixées à chaque itération. Une première solution est obtenue à l'aide d'une procédure de *relax and fix*. Mohammadi [167] présente un modèle mathématique pour un problème de planification, de chargement et d'ordonnancement avec setups dépendant des séquences et avec une configuration d'atelier de type flow-shop flexible. L'auteur propose des heuristiques basées sur la programmation mathématique pour la résolution du problème et réalise une comparaison avec les solutions optimales obtenues par un solveur. Mohammadi *et al.* [169] étudient le CLSP avec setups dépendants des séquences dans un flow-shop. Un modèle MIP et cinq heuristiques de résolution sont présentés. Les heuristiques sont basées sur des versions simplifiées du modèle et chacune est adaptée pour une taille d'instance différente.

Concernant les nomenclatures à plusieurs niveaux, le MLCLSP est principalement étudié dans le cadre d'ateliers mono-ressource, pour lesquels différents modèles et méthodes de résolutions ont été proposées. Une analyse comparative sur l'efficacité des modèles mathématiques à une seule machine est présentée dans [238]. Hung et Chien [116] proposent un modèle mathématique pour différents types de demandes (demandes en rupture, demandes confirmées, stock de sécurité, prévisions fiables et prévisions incertaines). Ils résolvent différentes instances du problème, en utilisant trois métaheuristiques : une recherche taboue, un recuit simulé et un algorithme génétique. Les résultats sont meilleurs avec la recherche taboue quand la demande suit le modèle des *demandes confirmées*, et le recuit simulé et plus avantageux quand on travaille avec des prévisions. Mohammadi *et al.* [168] proposent deux modèles MIP avec setups dépendant des séquences. Ils proposent aussi quatre heuristiques (deux de type *horizon glissant* et les deux autres de type *fix and relax*). Selon la taille des instances, une heuristique peut être plus favorable que d'autres. Sahling *et al.* [197] proposent une heuristique de type *fix and optimize* [185] pour résoudre le problème avec setups reportables. Almeder [21] propose une approche d'optimisation hybride combinant un algorithme de colonies de fourmis avec une méthode exacte utilisant un solveur standard. Helber et Sahling [110] proposent une approche de *fix and optimize* pour résoudre le MLCLSP avec délais d'obtention positifs, où un ensemble de sous-problèmes sont résolus avec un nombre limité de variables binaires, et Wu *et al.* [239] proposent deux formulations MIP avec prise en compte des ruptures de stock, ainsi qu'une méthode d'optimisation dont les bornes inférieures obtenues à partir de relaxations du modèle sont de bonne qualité.

Le MLCLSP à plusieurs ressources a été rarement étudié. Un des seuls travaux à s'intéresser à ce problème est celui d'Akartunalı et Miller [14], qui présentent une méthode pouvant être adaptée à plusieurs types de problèmes de planification à longues périodes. Ils proposent un modèle mathématique standard (extension du CLSP) et une formulation avec variables d'échelon stock [39]. Un inconvénient de ces modèles est que les contraintes de capacité considérées sont agrégées. La méthode de résolution proposée est basée sur des heuristiques de type *LP and fix* et *relax and fix*, et elle s'avère plus efficace qu'un solveur commercial.

2.4 Problèmes d'ordonnement

L'objectif de ce travail de thèse n'est pas d'étudier en profondeur les problèmes d'ordonnement. Nous sommes intéressés par l'intégration des décisions de planification et d'ordonnement au niveau tactique. Notre problématique principale est par conséquent un problème de planification avec contraintes d'ordonnement ou de capacité détaillées. Dans cette section,

nous proposons tout de même un bref aperçu des problématiques traitées par la communauté scientifique autour des problèmes d'ordonnancement.

Comme discuté dans le chapitre 1, les problèmes d'ordonnancement peuvent être classés selon la configuration de l'atelier de production, selon le critère d'optimisation et selon les contraintes. Un autre aspect permettant de classer ces problèmes est la considération ou pas des setups, dans une ou deux dimensions : temps et/ou coût. Un état de l'art suivant cette classification, et pour différents types d'ateliers de production, peut être consulté dans [15].

Dans la plupart des problèmes d'ordonnancement traités dans la littérature, les setups sont indépendants des séquences. D'une part, les temps de setup peuvent souvent être inclus dans les durées opératoires des tâches de production et, d'autre part, les coûts de setups ne sont pas considérés dans la fonction objectif, les critères d'optimisation étant la plupart du temps liés au temps de fin des tâches. Le critère le plus étudié est le *makespan*. Les contraintes de setups dépendant des séquences sont principalement considérées dans les problèmes intégrant le dimensionnement de lots et l'ordonnancement (voir par exemple [17]), mais les modèles considérés sont souvent à une machine.

Les problèmes d'ordonnancement peuvent aussi être classés en : ordonnancement avec batch [186] et ordonnancement sans batch. Un batch (conteneur, palette, etc.) étant formé par plusieurs jobs de la même famille, le but est d'ordonner les jobs par groupes, en économisant en coûts et en temps de production. Dans l'ordonnancement sans batch, chaque job est séquencé de manière indépendante.

De manière générale, les problèmes d'ordonnancement font partie des problèmes combinatoires les plus difficiles à résoudre. Hormis la considération ou pas de setups et de contraintes spécifiques, la complexité des problèmes d'ordonnancement dépend fortement du nombre de ressources et de jobs (i.e. de la configuration de l'atelier). La plupart des problèmes généraux sont \mathcal{NP} -difficiles, y compris une grande partie des problèmes à une seule machine [147], mais les problèmes les plus difficiles sont ceux associés aux ateliers de type job-shop, car le nombre de séquences possibles augmente de manière exponentielle quand le nombre de ressources ou de jobs augmente de manière linéaire. La difficulté est encore plus importante si le système est flexible (opérations pouvant être réalisées sur plusieurs machines), car non seulement l'ordre de traitement des jobs sur les machines doit être décidé, mais aussi l'affectation des jobs aux machines, ce qui augmente le nombre de variables binaires à considérer dans la modélisation.

Plusieurs types d'approches ont été développées pour la résolution des problèmes d'ordonnancement. Les méthodes exactes les plus utilisées sont le *branch and bound* et la programmation dynamique. Les métaheuristiques les plus appliquées sont la recherche taboue, la recherche à voisinage variable, le recuit simulé et les algorithmes génétiques.

Nous nous intéressons dans les chapitres qui suivent à l'ordonnancement sans batch et avec setups qui sont indépendants des séquences, dans un atelier de type job-shop. Un état de l'art sur les techniques utilisées pour la résolution du problème d'ordonnancement dans un atelier de type job-shop peut être consulté dans [119].

2.5 Intégration de la planification et de l'ordonnancement

L'intégration de la planification et de l'ordonnancement permet de garantir la faisabilité des plans de production décidés au niveau tactique. En traitant la planification et l'ordonnancement de manière séparée, on ne tient pas compte des contraintes de succession qui ont un impact sur la capacité exacte du système, et la détermination d'un plan de production réalisable au niveau opérationnel n'est pas garantie. Il est par conséquent important d'essayer de prendre les deux types de décision en même temps.

Comme expliqué dans la Section 1.4.1 du Chapitre 1, l'approche MRP permet de planifier la production sans tenir compte de la capacité du système. Quand le plan de production est construit, le but est d'ordonner au mieux les activités pour respecter le plan de production qui est supposé être optimal en termes de coût. Le problème apparaît lorsqu'il est impossible de déterminer un ordonnancement respectant les délais de production, dû à une capacité insuffisante du système de production. Dans ce cas, toutes les stratégies envisageables pour respecter le plan de production entraîneront une augmentation du coût global.

MRP-II prend en compte certaines informations sur la capacité de production avec différents niveaux de détails, qui sont désagrégés hiérarchiquement, en construisant plusieurs plans de production jusqu'à obtenir un plan détaillé. Cette procédure d'estimation de la capacité génère une vision limitée des ressources au niveau tactique et conduit vers le calcul d'un plan qui n'utilise pas de manière efficace la capacité, et qui peut être infaisable au niveau ordonnancement. D'où l'importance de suivre une approche qui intègre du mieux possible les contraintes du niveau opérationnel.

Les modèles mathématiques correspondant aux problèmes DLSP, CSLP, PLSP, GLSP, CLSP et ses différentes variantes garantissent des solutions réalisables au niveau opérationnel pour des systèmes de production relativement simples. C'est le cas des ateliers à une seule machine ou deux machines parallèles, ainsi que les systèmes à un seul produit ou plusieurs produits sans gamme de fabrication. Par contre, quand on considère des systèmes de production plus complexes avec différents produits ayant des gammes de fabrication et plusieurs ressources (par exemple, les ateliers de type flow-shop et job-shop), les modèles classiques de planification ne garantissent pas des solutions réalisables. En effet, pour que les plans de production soient

réalisables, il faut prendre en compte des contraintes de capacité détaillées qui incluent des contraintes d’ordonnancement. Il est notamment nécessaire de tenir compte des contraintes de précedence liées aux gammes et aux ressources, permettant de définir les dates de début et de fin des opérations. Les travaux présentés dans la section 2.2, bien qu’ils permettent de trouver des solutions réalisables et de bonne qualité pour des cas d’études particuliers, ne peuvent pas être considérés comme des approches intégrées (considérant des contraintes de capacité détaillée du niveau opérationnel), puisqu’une généralisation de ces méthodes ne garantit pas des solutions réalisables dans des ateliers de production complexes.

Par exemple, les contraintes de coûts et de temps de setup dépendant des séquences, très étudiées dans des travaux récents, sont des contraintes d’ordonnancement qui ne suffisent pas à garantir des solutions réalisables dans les systèmes multi-ressources complexes. Elles permettent uniquement de sélectionner la séquence qui satisfait le mieux un critère de coût ou de temps. Pour garantir des plans de production réalisables, les dates de début et de fin de toutes les opérations doivent être considérées dans le modèle mathématique.

Pour plus de détail sur l’importance d’intégrer les décisions d’ordonnancement dans la planification de la production au niveau tactique, le lecteur peut consulter le travail de Dauzère-Pères et Lasserre [48], qui met en évidence les limites des approches hiérarchiques.

À ce sujet, Maravelias et Sung [155] présentent une classification des stratégies de résolution des problèmes de planification et d’ordonnancement, comme illustré dans la Figure 2.5. Ils définissent trois catégories de méthodes de résolution qui sont :

- Approches hiérarchiques,
- Approches itératives,
- Approches intégrées.

Dans les approches hiérarchiques, le problème de planification est résolu dans un premier temps, sans tenir compte des contraintes détaillées d’ordonnancement. Ainsi, le plan de production n’est pas forcément réalisable au niveau d’ordonnancement, et la solution correspond à une borne inférieure du problème intégré. Dans un deuxième temps, le problème d’ordonnancement est résolu, en fixant les variables de décisions déterminées dans le niveau de planification. Le problème se pose lorsqu’il devient impossible de trouver un ordonnancement réalisable pour suivre le plan de production. De plus, il est probable que l’écart entre la solution obtenue et l’optimum soit très grand.

Dans les approches itératives, il y a un échange systématique d’information entre le niveau résolvant le problème de planification (qui peut contenir ou non des contraintes approximant les contraintes réelles d’ordonnancement) et le niveau dédié à la résolution du problème d’or-

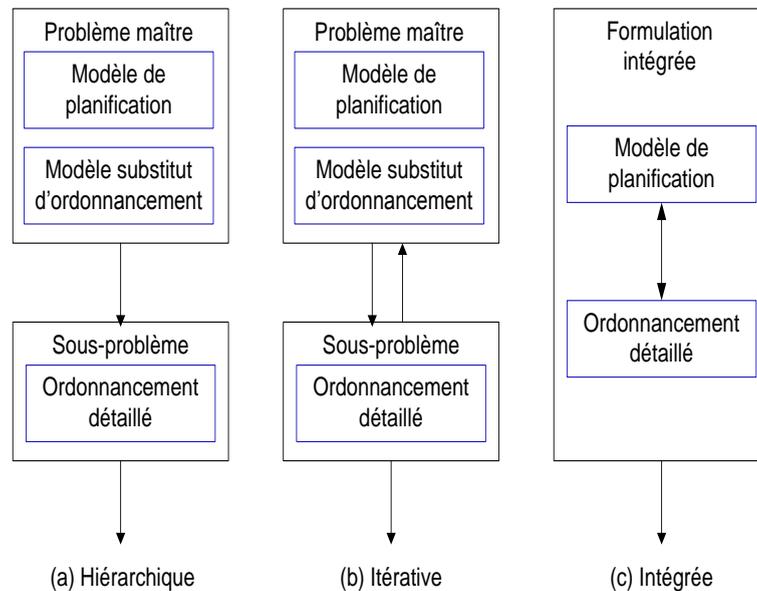


FIGURE 2.5 – Stratégies de résolution du problème intégré de planification et d’ordonnancement [155]

donnancement. Cette stratégie permet d’obtenir une solution réalisable, mais l’optimum global n’est pas garanti.

Dans les approches intégrées, il n’y a pas de séparation entre les problèmes de planification et d’ordonnancement, mais un seul problème est considéré avec une formulation intégrée. Résoudre un tel modèle permet d’obtenir la solution optimale du problème global, mais la complexité du problème étant très importante, il est souvent impossible d’obtenir une solution optimale dans des temps de calcul acceptables, voire même d’obtenir une solution réalisable. Seuls des problèmes de petite taille peuvent être résolus de façon optimale. C’est pourquoi il est nécessaire d’utiliser des méthodes de décomposition ou de relaxation, pour diminuer la complexité du problème dans les approches intégrées. L’optimalité de la solution n’est pas garantie, mais le rapport entre qualité de solution et temps de calcul devient acceptable.

Le but de cette section étant d’étudier les problèmes qui intègrent de manière cohérente les décisions de planification et d’ordonnancement, nous nous focalisons dans ce qui suit sur les approches itératives et les approches intégrées. Le lecteur intéressé par les méthodes hiérarchiques peut consulter [160] et [229].

2.5.1 Approches itératives

Approche de Lasserre [144]

Cette approche est la première à traiter de façon cohérente le problème mono-niveau de planification et d'ordonnancement de la production, et elle est la première méthode à garantir la faisabilité des plans de production au niveau opérationnel. L'approche suit une procédure itérative, où le problème intégré est décomposé en k problèmes de planification avec séquence fixée et k problèmes d'ordonnancement avec tailles de lots fixées, où k est le nombre d'itérations de la procédure de résolution. À chaque itération, un problème de planification et un problème d'ordonnancement sont résolus. Contrairement à d'autres travaux, qui traitent le problème d'ordonnancement période par période, l'ordonnancement est considéré sur tout l'horizon de planification comme un seul problème.

La méthode démarre avec le calcul d'une séquence initiale y , en résolvant le problème d'ordonnancement pour un plan de production dont les tailles de lot sont égales aux demandes. Ensuite, la séquence déterminée y est fixée et le problème de planification avec séquence fixée $P(y)$ est résolu. Puis, un nouveau problème d'ordonnancement (en fixant les tailles de lots calculées dans le plan précédent) est résolu, et un nouveau plan est déterminé. Cette procédure est répétée durant k itérations.

Le schéma général de la procédure itérative, qui converge vers un optimum local, est illustré dans la Figure 2.6.

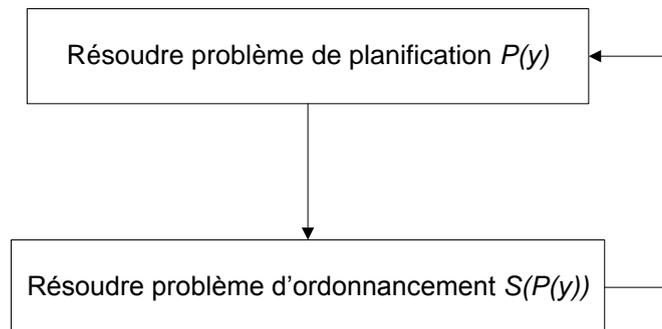


FIGURE 2.6 – Approche itérative de Lasserre [144]

Pour résoudre le problème de dimensionnement de lots à séquence fixée, l'auteur propose un modèle mathématique MIP qui est une extension du CLSP, en tenant compte des contraintes de capacité détaillées associées à la séquence fixée, information suffisante pour garantir la faisabilité du plan au niveau d'ordonnancement. Le modèle tient compte de la rupture des stocks et des

temps et des coûts de setup. Le problème d'ordonnancement est résolu à travers une version modifiée de l'*heuristique par machine goulot* (Shifting Bottleneck heuristic) [6], qui suit un critère de minimisation du *makespan*.

La performance de l'approche a été testée en considérant un atelier de type job-shop avec 6 produit, 6 machines et 3 périodes de planification. Différentes instances ont été générées à partir de plans initiaux différents. Dans la plupart des cas, la solution optimale ou une solution proche de l'optimum a été obtenue après peu d'itérations. Le problème de dimensionnement de lots avec séquence fixée ($P(y)$) a été résolu avec le solveur commercial XPRESS-MP [241].

Approche de Dauzère-Pérès et Lasserre [47]

Les auteurs proposent deux stratégies pour résoudre le problème d'ordonnancement, en suivant l'approche itérative de Lasserre [144]. Le problème d'ordonnancement est encore une fois résolu en utilisant une version modifiée ([46]) de l'heuristique par machine goulot [6], qui a pour but de minimiser le *makespan*.

Les stratégies de résolution du problème d'ordonnancement sont basées sur deux politiques : ordonnancement *global* et ordonnancement *période par période*. Dans l'ordonnancement global, tous les jobs à ordonnancer sur tout l'horizon de planification sont considérés en même temps ; tandis que dans la politique période par période, le problème d'ordonnancement est décomposé en T (nombre de périodes) problèmes d'ordonnancement, chaque problème étant indépendant des autres. La stratégie d'ordonnancement global conduit vers de meilleurs résultats.

Approche de Roux *et al.* [193]

Dans ce travail, un problème multi-niveaux et multi-sites est considéré. Une stratégie de résolution itérative similaire à celle utilisée dans [144] et [47] est appliquée, avec la différence qu'à chaque itération plusieurs problèmes d'ordonnancement sont résolus. En effet, comme il s'agit d'un réseau avec plusieurs sites de production, le problème d'ordonnancement peut être résolu de façon indépendante par site. À chaque itération, pour chaque plan de production généré, le problème d'ordonnancement est résolu avec une métaheuristique proposée dans [49].

Deux modèles mathématiques MIP pour le problème de planification avec séquence fixée et des contraintes de capacité détaillées par site de production sont proposés. Ils peuvent être vus comme des extensions du CLSP. Le premier modèle autorise la rupture de stock pour les produits finis. Par contre, afin d'assurer l'approvisionnement de la demande interne, les composants ne peuvent pas être en rupture de stock. Ainsi, des demandes externes sont considérées uniquement

pour les produits finis. Le deuxième modèle considère que tous les produits, y compris les composants à tous les niveaux, peuvent avoir des demandes externes. Au même titre, la rupture de stock est autorisée pour tous les produits. De nouvelles contraintes sont introduites afin de continuer à garantir la satisfaction de la demande interne.

Dans les deux modèles, les coûts et les temps de setup ne sont pas considérés. La fonction objectif consiste à minimiser la somme des coûts de fabrication, de stockage et de rupture des stocks.

La méthode a été testée sur 4 instances d'un problème d'ordonnancement dans un atelier de type job-shop flexible, avec 5 sites de production et 6 périodes. Les demandes externes ont été considérées uniquement sur les produits finis. Le problème de planification avec séquence fixée a été résolu au moyen de la librairie d'optimisation OSL d'IBM, et les problèmes d'ordonnancement par site ont été résolus en parallèle.

Modèle intégré de Dautère-Pérès et Lasserre [48]

Les auteurs présentent une nouvelle version du modèle intégré proposé dans [144] et [47], permettant maintenant de résoudre des problèmes multi-niveaux et de tenir compte des délais d'obtention des produits. L'aspect multi-niveaux est modélisé en incluant les liens dans la nomenclature (*gozinto factors*) entre les produits et les tailles de lots des produits successeurs dans les contraintes d'équilibre de stocks.

Les auteurs décrivent l'intérêt d'intégrer les contraintes d'ordonnancement dans le problème de planification de la production, afin de garantir des solutions réalisables. Cependant, en tenant compte du fait que le modèle est très difficile à résoudre, ils recommandent l'utilisation de l'approche itérative initialement proposé dans [144] et [47] et étendue au cas multi-niveaux et multi-sites dans [193].

Approche de Ouerfelli *et al.* [175]

Un algorithme de résolution exacte, basé sur la méthode de décomposition de Benders [28], est présenté pour résoudre un problème mono-niveau. Le travail est inspiré de l'approche de Dautère-Pérès et Lasserre [47], avec une différence de conception importante : l'ordonnancement est réalisé période par période. Le problème est décomposé en un problème principal qui fixe la séquence et un problème secondaire qui calcule le plan de production associé à la séquence précédemment fixée. Le problème principal est une version relâchée (éliminant les circuits) du problème d'ordonnancement.

La méthode de Benders est utilisée pour combiner de manière itérative la résolution du problème principal et du problème secondaire. Deux versions de la méthode de Benders (la méthode traditionnelle et la méthode avec des contraintes pour éliminer les circuits) sont comparées avec la résolution du modèle par le solveur commercial IBM ILOG CPLEX.

L'approche a été testée sur 7 instances sur un atelier de type job-shop avec 6 produits et 6 machines et 3 périodes de planification. Les solutions obtenues par IBM ILOG CPLEX sont meilleures dans tous les cas, et les écarts entre les bornes inférieure et supérieure des deux variantes de Benders sont très élevés (273% au minimum). Parmi les deux techniques de Benders, la méthode avec contraintes pour éliminer les circuits offre une meilleure performance. Il faut noter que seules des instances très petites, avec 3 périodes, sont résolues.

Première approche de Li et Ierapetritou [152]

Les auteurs résolvent un problème typique rencontré dans les industries de process. Afin de diminuer la difficulté de résolution du problème intégré, une structure d'horizon glissant est utilisée, afin de résoudre plusieurs sous-problèmes de planification avec capacité agrégée sur tout l'horizon de planification et plusieurs sous-problèmes d'ordonnancement (un par période), permettant ainsi de modifier les décisions tactiques pour obtenir des plans de production réalisables.

À chaque itération, les décisions sont mises à jour. Les tailles de lots obtenues avec l'ordonnancement détaillé remplacent les anciennes tailles de lots déterminées avec capacité agrégée, et les demandes non satisfaites à une période donnée sont reportées à la période suivante. L'algorithme de cette méthode est schématisé dans la Figure 2.7, où p_c , T , P_1 et P_2 correspondent à la période courante, au nombre total de périodes, au problème de planification et au problème d'ordonnancement, respectivement.

Les caractéristiques d'ordonnancement considérées dans le modèle sont :

- Non préemption,
- Capacité de la ressource (bornes inférieure et supérieure),
- Durées opératoires,
- Précédence entre les opérations,
- Équilibre de la matière,
- Capacité de stockage limitée,

Une contribution de ce travail par rapport aux méthodes traditionnelles d'horizon glissant, est que des contraintes de capacité, correspondant à des bornes sur la quantité de production de chaque produit (des inégalités valides), sont incluses dans le modèle de planification. Une

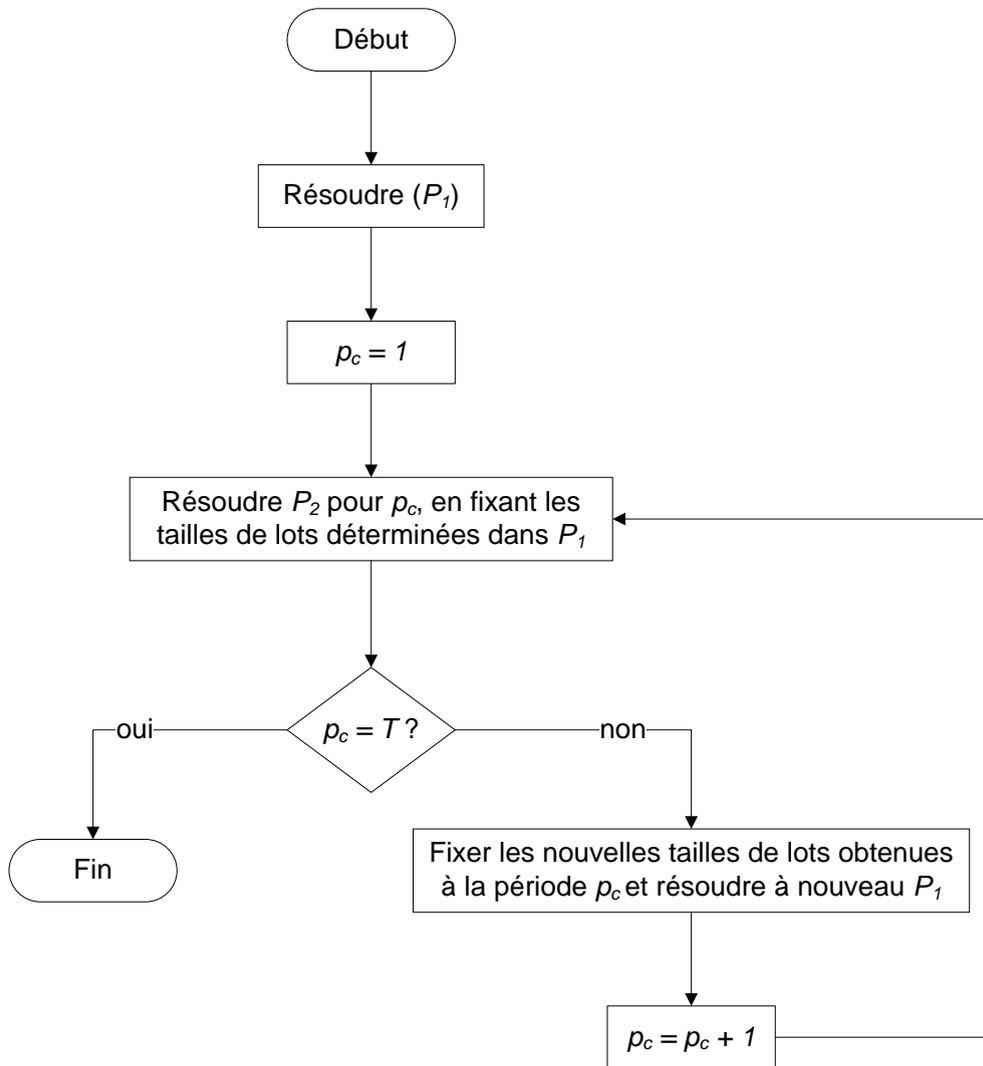


FIGURE 2.7 – Approche avec horizon glissant [152]

comparaison réalisée avec une modélisation sans contraintes de capacité montre que cette information permet d'améliorer de manière significative la qualité des résultats. L'amélioration est due principalement à une grande diminution des quantités en rupture de stock.

Les auteurs proposent aussi un algorithme de programmation paramétrique [183] pour obtenir une information plus précise sur la capacité. Cette technique permet d'obtenir des meilleurs résultats par rapport à la stratégie précédente (avec des contraintes de capacité simples). En considérant un système avec deux produits, ils déterminent la région géométrique de la capacité de production, en maximisant la taille de lot d'un produit et en faisant varier celle de l'autre, et vice versa. L'inconvénient est que lorsque le nombre de produits augmente, construire toutes les

régions de capacité devient très coûteux en termes d'effort de calcul. C'est pourquoi les auteurs proposent une heuristique permettant de décomposer le problème en plusieurs sous-problèmes contenant peu de produits.

L'avantage des procédures avec horizon glissant est que le modèle mathématique est réduit de façon importante, et par conséquent le temps de calcul aussi pour résoudre ce modèle. De plus, dans la pratique, cette stratégie peut permettre de tenir compte des incertitudes pouvant se présenter à court terme, et de recalculer les décisions prises pour le moyen terme. Un inconvénient est que l'écart entre les solutions proposées et les solutions optimales peut être grand.

Deuxième approche de Li et Ierapetritou [150]

Dans ce travail, un problème multi-produits et multi-opérations de l'industrie chimique est étudié. Le problème d'optimisation est modélisé à deux niveaux. Dans le niveau supérieur le problème de planification est formulé, et dans le niveau inférieur plusieurs sous-problèmes d'ordonnancement sont modélisés. Comme le modèle est très difficile à résoudre, une méthode de décomposition est proposée. L'objectif dans le modèle de planification est de minimiser la somme des coûts de stockage, de rupture et de production. Le modèle d'ordonnancement minimise le coût de production. Ce coût est composé d'une partie fixe et d'une partie variable qui dépend des tailles des lots. Les valeurs des variables de décision déterminées au niveau supérieur sont fixées comme paramètres dans le niveau inférieur. Les mêmes contraintes d'ordonnancement que dans [152] sont considérées dans ce travail.

Pour éviter l'infaisabilité des sous-problèmes d'ordonnancement, le modèle d'ordonnancement est reformulé, en incluant un coût de pénalité dans la fonction objectif pour pénaliser les objectifs de production non atteints et des variables d'écart dans les contraintes fixant les objectifs de production, afin de représenter ceux qui ne sont pas atteints.

L'algorithme de résolution suit une méthode de décomposition qui itère entre la résolution d'un problème maître (modèle de planification sans contraintes d'ordonnancement), générant une borne inférieure, et la résolution de l'ensemble des sous-problèmes d'ordonnancement, qui permet d'obtenir une borne supérieure. À chaque itération, le coût de production est mis à jour. Une sous-estimation convexe de la fonction du coût de production est réalisée au moyen d'une relaxation Lagrangienne.

Deux cas d'études sont testés, mettant en évidence la complexité de résolution du problème intégré avec l'approche proposée. Dans le premier cas, l'écart relatif avec la solution de CPLEX est de 25%, et dans le deuxième, l'écart relatif entre la borne inférieure et la borne supérieure

est de 8.5% après 28 heures de calcul.

Troisième approche de Li et Ierapetritou [151]

Les auteurs proposent une méthode de relaxation Lagrangienne augmentée, méthode qui a comme inconvénient la non-séparabilité du problème relâché. C'est pourquoi, les auteurs proposent une méthode d'optimisation à deux niveaux, permettant de décomposer ce problème. Dans le premier niveau, le problème relâché est résolu, en respectant seulement les variables de planification à travers un algorithme itératif. Dans le deuxième niveau, un ensemble de sous-problèmes d'ordonnancement est résolu, en fixant les variables de décision de planification déterminées dans le niveau supérieur. L'implémentation d'une stratégie d'approximation quadratique diagonale [195] est aussi mise en place et une comparaison des performances est réalisée.

L'inconvénient de l'approche à deux niveaux est qu'il est nécessaire d'optimiser un problème quadratique non différentiable à chaque itération, et donc l'effort de calcul devient important. D'autre part, à travers l'approche d'approximation quadratique diagonale, on résout une approximation du problème relâché et il n'est pas possible de garantir l'obtention de l'optimum global. Les comparaisons réalisées permettent de conclure que la méthode d'approximation est meilleure que la stratégie d'optimisation à deux niveaux.

Le modèle utilisé inclut toutes les contraintes de planification et d'ordonnancement dans une seule formulation, avec des variables de planification sur les deux types de contraintes (planification et ordonnancement) et des variables d'ordonnancement uniquement sur les contraintes d'ordonnancement. Pour coupler les deux types de contraintes, une reformulation est faite, en considérant des nouvelles variables servant à définir des contraintes couplantes. Les nouvelles contraintes sont relâchées, et comme expliqué précédemment, le problème est décomposé en un sous-problème de planification et plusieurs sous-problèmes d'ordonnancement. L'algorithme de résolution utilisé est illustré par la Figure 2.8, où λ et μ sont les multiplicateurs Lagrangiens, σ et le paramètre de pénalité, Inv et P sont les variables de tailles de lot et d'inventaire, PP et II sont des variables auxiliaires dupliquant Inv et P , respectivement, f est la valeur objectif du problème dual, g est la valeur de la fonction de consistance, T est le nombre de périodes et ϵ est l'écart de dualité. Les variables auxiliaires permettent de reformuler le problème à partir d'une décomposition.

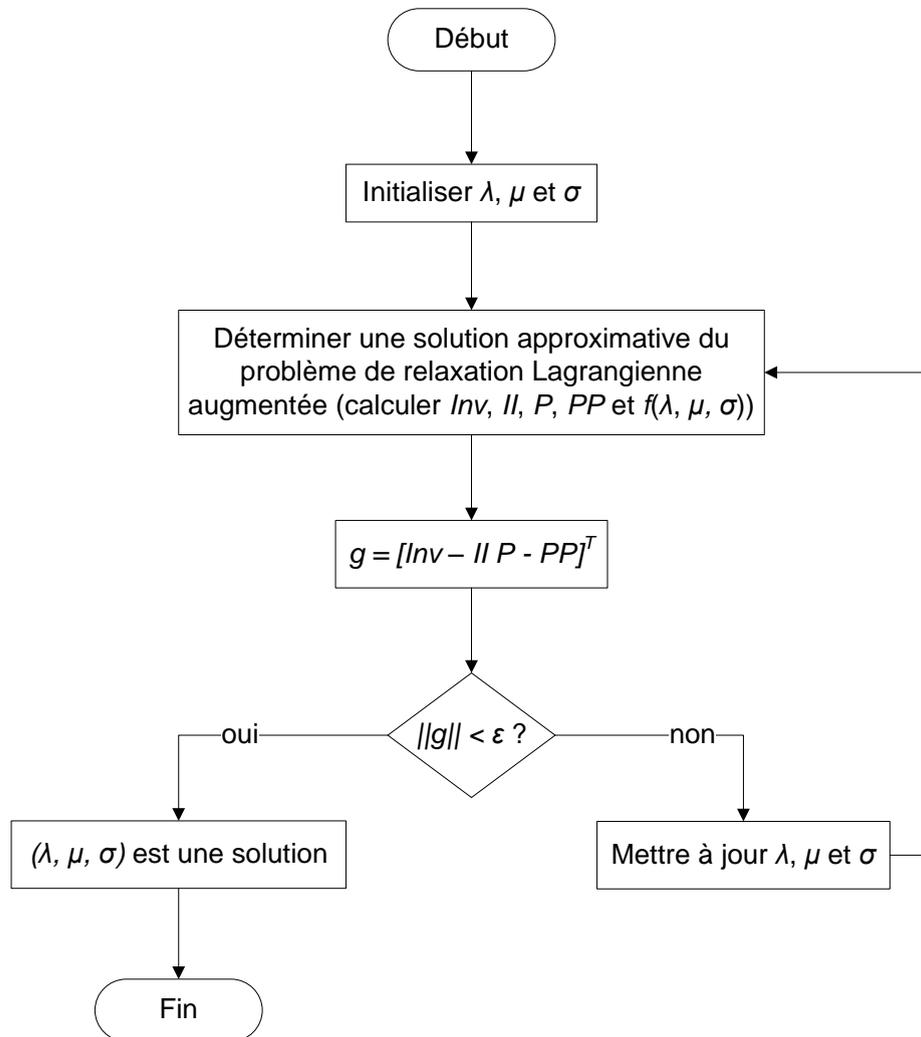


FIGURE 2.8 – Approche avec relaxation Lagrangienne augmentée [151]

2.5.2 Approches intégrées

Approche de Zhang et Yan [243]

Cette approche est basée sur un algorithme génétique hybride, et se focalise sur la résolution du problème de planification et d'ordonnancement dans un atelier de type job-shop. Les règles de priorité LPT (plus longue durée de fabrication) et SPT (plus courte durée de fabrication) pour générer la première séquence sont utilisées, le problème est formulé avec un modèle non-linéaire en variables mixtes et l'ordonnancement est réalisé période par période.

Peu d'instances sont résolues et des comparaisons avec des méthodes exactes n'ont pas été effectuées. Il est par conséquent difficile de conclure sur la performance de l'approche. Des comparaisons ont été réalisées uniquement avec une méthode hiérarchique, où l'on valide la pertinence d'intégrer des décisions.

Approche de Wolosewicz *et al.* [233, 234, 235]

Inspirée des travaux de [47] et [48], cette approche a été proposée dans le but de diminuer les temps de calcul et d'améliorer la qualité des résultats. La méthode itérative devient une méthode intégrée, où le problème d'ordonnancement n'est pas résolu à chaque itération, mais la séquence est modifiée à travers une métaheuristique (recherche taboue ou recuit simulé), permettant potentiellement d'éviter la convergence vers un optimum local et d'explorer de nouveaux espaces de recherche, afin de trouver une solution proche de l'optimum global. Comme dans les travaux précurseurs, l'ordonnancement est considéré sur tout l'horizon de planification.

Une heuristique Lagrangienne pour résoudre le problème de planification avec séquence fixée (voir [234] et [235]) et une recherche taboue pour guider l'évolution de la séquence sont mises en œuvre. Ainsi, les contraintes de capacité détaillées sont relâchées à travers la relaxation Lagrangienne, et l'algorithme dynamique de dimensionnement de lots proposé dans [230] est utilisé pour résoudre un ensemble de problèmes mono-produit sans contraintes de capacité. Pour réparer les contraintes de capacité, une heuristique de lissage de la production est couplée à l'heuristique Lagrangienne, permettant ainsi de calculer des bornes supérieures (plans de production réalisables). Cette heuristique modifie les tailles de lots en déplaçant des quantités de produit entre différentes périodes, avec le but d'éliminer la violation des contraintes de capacité. La difficulté majeure de l'heuristique de lissage est que, comme l'ordonnancement est réalisé sur tout l'horizon de planification, les déplacements de quantités de production réalisés entre deux périodes modifient potentiellement les dates de début et de fin de toutes les opérations, entre et après ces deux périodes et changent ainsi la capacité utilisée dans un très grand nombre de périodes. Néanmoins, comme montré dans [47], l'ordonnancement global est préférable à l'ordonnancement période par période, car la capacité est utilisée plus efficacement et de meilleures solutions sont trouvées.

Les deux versions de cette approche (avec recherche taboue et avec recuit simulé) et l'approche de Dauzère-Pérès et Lasserre [47] sont comparées, en résolvant plusieurs instances (en faisant varier la capacité, les coûts de setup, les demandes et les délais d'obtention) associées à différents ateliers de type job-shop et avec différents horizons de planification. La nouvelle approche avec recherche taboue donne globalement des meilleurs résultats.

La performance de l'approche a été aussi comparée avec la résolution exacte du modèle par le solveur commercial IBM ILOG CPLEX, et une fois de plus l'intérêt d'utiliser cette méthode est validée. Les solutions obtenues avec l'approche intégrée sont globalement meilleures que celles du solveur, surtout pour les grandes instances (avec 10 produits et 10 machines ou 20 produits et 5 machines et 10 ou 20 périodes), pour lesquelles IBM ILOG CPLEX n'obtient pas de solutions réalisables.

Le schéma sur la Figure 2.9 présente l'évolution de l'approche de Lasserre [144], jusqu'à l'approche de Wolosewicz *et al.* [233, 234, 235].

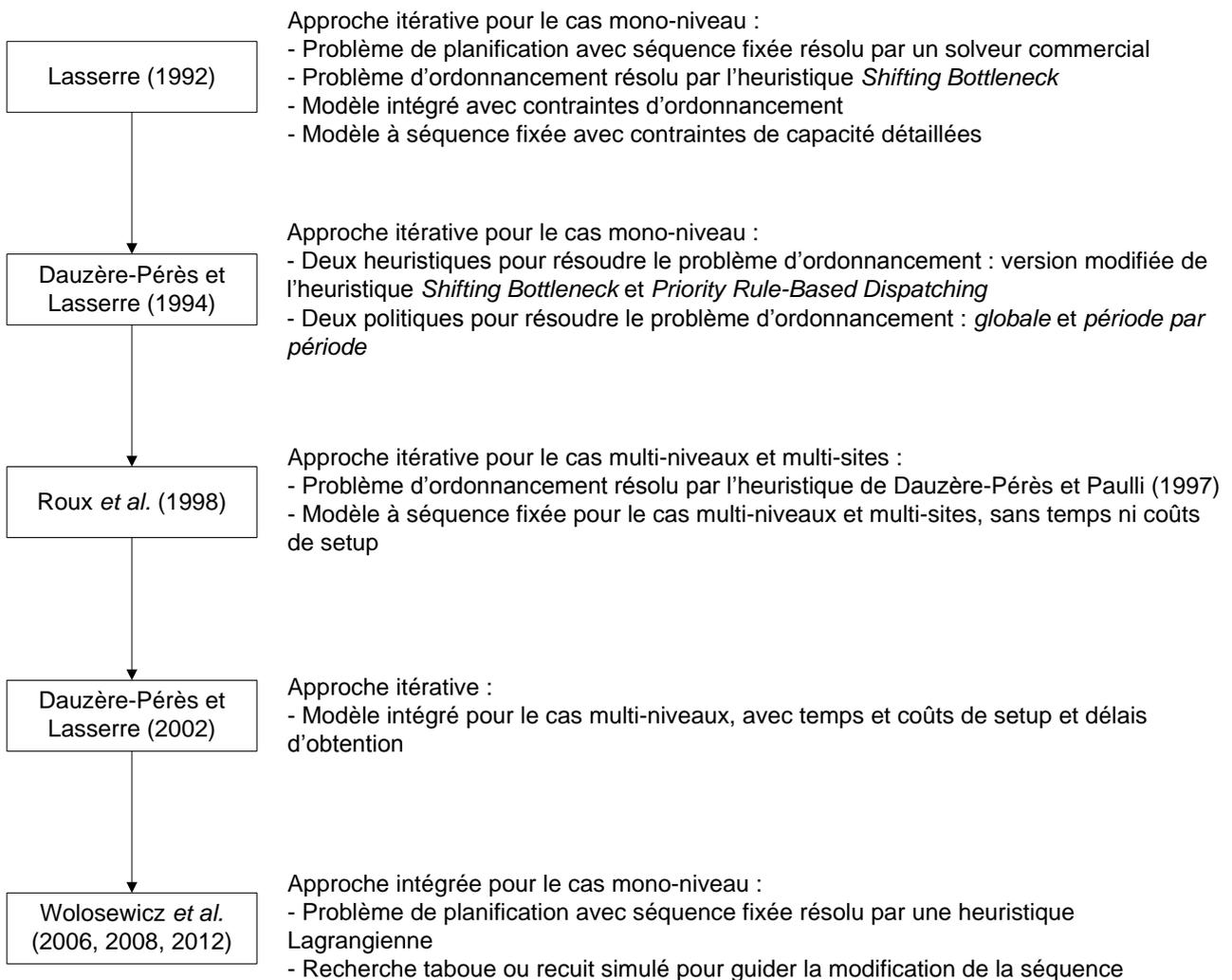


FIGURE 2.9 – Évolution de l'approche de Lasserre [144]

Approche de Shah et Ierapetritou [203]

Les auteurs étudient un problème mono-niveau et multi-sites avec coûts de transport, où plusieurs marchés peuvent être satisfaits par plusieurs sites de production, avec des coûts de transport différents. Une relaxation Lagrangienne augmentée avec une méthode d'approximation diagonale est implémentée pour résoudre le problème intégré. La non-satisfaction de la demande à une période donnée est autorisée dans le modèle, mais elle doit être satisfaite à la période suivante. Des coûts de rupture sont considérés à cet effet. L'objectif du problème intégré est de minimiser la somme des coûts de stockage, de rupture, de production et de transport. Les contraintes d'ordonnancement utilisées dans le modèle sont celles considérées dans [152] et [150].

Comme dans [151], les auteurs montrent que le modèle a une structure de bloc angulaire, où les variables de planification lient les contraintes de planification et d'ordonnancement. Pour décomposer le problème, ils proposent de transformer les variables compliquées (variables de planification) en contraintes compliquées, et de relâcher ces dernières. Dans la reformulation, les variables de planification sont dupliquées, de façon à utiliser des variables différentes pour la planification et l'ordonnancement, et des contraintes couplant ces deux types de variables sont incluses dans le modèle.

Une relaxation Lagrangienne augmentée est appliquée pour décomposer le problème. La nouvelle fonction objectif inclut des multiplicateurs Lagrangiens et un paramètre de pénalité quadratique. L'inconvénient qui apparaît à ce stade est la non-séparabilité des termes bilinéaires associés au paramètre de pénalité. Ce problème est contourné en appliquant une méthode d'approximation quadratique diagonale, qui permet de linéariser ces termes. Ainsi, une nouvelle reformulation est proposée, en décomposant la fonction objectif en deux sous-fonctions, une pour le problème de planification et une autre pour le problème d'ordonnancement. L'approche de résolution itère entre la résolution d'un problème de planification et la résolution de plusieurs sous-problèmes d'ordonnancement. Le modèle MIP du problème intégré original est transformé en un modèle quadratique pour le problème de planification et en des modèles quadratiques à nombres mixtes (MIQP) pour les sous-problèmes d'ordonnancement. L'algorithme converge vers une solution réalisable, mais l'optimalité n'est pas garantie parce que c'est une approximation du problème relâché qui est résolue.

La performance de l'approche a été comparée avec celle d'une méthode exacte utilisant le modèle intégré original et le solveur IBM ILOG CPLEX (version 12.2). Trois types de configurations ou systèmes ont été testés. Sur deux configurations, les résultats sont favorables au solveur standard pour les instances avec un nombre de périodes d'ordonnancement inférieur à

90; et pour l'autre configuration, IBM ILOG CPLEX est plus avantageux sur des instances avec un nombre de périodes inférieur à 45. Pour un nombre de périodes d'ordonnancement plus élevé, l'approche proposée donne de meilleures solutions.

2.6 Conclusion

Nous avons présenté une perspective générale sur la façon dont le lien entre la planification et l'ordonnancement de la production et dans la chaîne logistique est étudié dans la littérature, afin de réduire l'incohérence des décisions hiérarchiques proposées par les approches classiques du type MRP-II, et de déterminer des plans de production tactiques réalisables au niveau opérationnel.

Dans un premier temps, nous avons posé les bases de la complexité, la modélisation et la résolution des problèmes d'optimisation liées aux étapes de planification de la chaîne logistique, et plus précisément aux problèmes de dimensionnement de lots et d'ordonnancement. Ensuite, nous avons présenté une vue d'ensemble des problèmes de dimensionnement de lots multi-produits avec contraintes de capacité et demande dynamique, en utilisant la classification classique selon la longueur des périodes et le nombre de niveaux de nomenclature. Ainsi, nous avons vu que les problèmes à courtes périodes (DLSP, CSLP et PLSP) restreignent l'utilisation de la capacité et considèrent un nombre important de variables discrètes, ce qui rend difficile la résolution de ces problèmes. En raison des hypothèses des modèles, seuls des systèmes de production simples (ateliers à une machine ou à machines parallèles) peuvent être considérés. Parmi les modèles à longues périodes, le GLSP a moins de limitations sur la modélisation de la capacité, mais la difficulté de résolution augmente fortement; tandis que le CLSP modélise la capacité de manière globale, mais il n'est pas adapté à la résolution des systèmes de production complexes, et la prise en compte de contraintes d'ordonnancement entraîne une difficulté supplémentaire très importante. Pour ce qui relève de la prise en compte de la nomenclature, nous avons décrit les travaux les plus représentatifs sur les systèmes mono-niveau et multi-niveaux à une et plusieurs ressources. Peu de travaux ont étudié les problèmes multi-niveaux, et encore moins ceux à plusieurs ressources, malgré leur pertinence dans de nombreux environnements industriels réels. La raison est la complexité des problèmes, notamment avec l'utilisation des modèles mathématiques dont nous avons discuté.

En ce qui concerne les problèmes d'ordonnancement (sans contraintes ni objectifs de planification tactique), nous avons brièvement décrit les principaux groupes de problèmes traités dans la littérature, et nous avons donné des généralités sur la complexité et les méthodes de résolution.

Une attention spéciale a été accordée aux problèmes de planification intégrant des contraintes d'ordonnancement, dans le but de déterminer des solutions réalisables et proches de l'optimum, pour des systèmes impliquant le partage de ressources entre différents produits et nécessitant le traitement de plusieurs opérations. À cet effet, nous avons présenté les approches itératives et les approches intégrées existantes dans la littérature, ainsi que leurs limitations et avantages. Parmi ces travaux, nous retenons l'approche de Wolosewicz *et al.* [233, 234, 235], pour laquelle nous proposons des améliorations significatives dans le Chapitre 3 et une extension aux problèmes multi-niveaux dans le Chapitre 4.

Chapitre 3

Approche intégrée pour des problèmes à un niveau

Ce chapitre présente une méthode d'optimisation permettant de résoudre de manière intégrée le problème de planification et d'ordonnancement de la production mono-niveau, dans des systèmes de production complexes de type job-shop¹. L'approche permet d'obtenir des solutions réalisables proches de l'optimum pour différentes tailles de problèmes, avec des temps de calcul raisonnables.

[3.1 Introduction](#)

[3.2 Modélisation](#)

[3.3 Méthode de résolution](#)

[3.4 Analyses et améliorations](#)

[3.5 Résultats expérimentaux](#)

[3.6 Conclusion](#)

1. Des parties de ce chapitre ont été publiées dans les conférences **ROADEF2011** [11], **IWLS2011** [84], **ROADEF2012** [87], **ILS2012** [92] et **IESM2013** [88] et dans la revue **International Journal of Production Research** [91].

3.1 Introduction

Comme montré dans le chapitre précédent, peu de travaux dans la littérature ont étudié l'intégration des décisions de planification et d'ordonnancement de la production dans des systèmes complexes, afin de déterminer au niveau tactique des plans de production réalisables au niveau opérationnel, i.e. des plans qui respectent les contraintes de capacité détaillées. Parmi les approches présentées, nous nous intéressons à celle de Wolosewicz *et al.* [233, 234, 235], qui est une version évoluée de plusieurs approches et qui, contrairement aux autres travaux, considère le problème d'ordonnancement sur tout l'horizon de planification, permettant ainsi d'utiliser efficacement la capacité de production.

L'approche de Wolosewicz *et al.*, bien qu'elle offre une bonne performance par rapport au solveur commercial IBM ILOG CPLEX, peut être améliorée en termes de qualité de la solution et de temps de calcul, en modifiant plusieurs parties de l'algorithme, comme nous allons le constater dans ce chapitre. En fait, nous proposons, dans ce qui suit, une nouvelle approche basée sur le principe de combinaison d'une heuristique Lagrangienne et d'une recherche taboue pour résoudre le problème intégré dans des systèmes complexes multi-produits, multi-opérations et multi-ressources avec partage de ressources et avec une politique d'ordonnancement sur tout l'horizon de planification.

Il s'agit d'une approche décomposant le problème en un ensemble de sous-problèmes avec séquence fixée, pour diminuer l'effort de calcul, avec une heuristique qui permet d'obtenir des solutions de bonne qualité. Cette stratégie est basée sur une reformulation du modèle intégré de [48], comme introduit dans [234] pour résoudre le problème à séquence fixée.

Nous commençons ce chapitre par une description du problème auquel nous nous intéressons, ainsi que sa modélisation, dans la Section 3.2. Ensuite, dans la Section 3.3, nous expliquons les principes de la méthode de résolution. Dans la Section 3.4, nous décrivons les différentes améliorations qui ont été apportées à la méthode de Wolosewicz *et al.*, ainsi que les différentes analyses réalisées. Puis, dans la Section 3.5, nous comparons les résultats obtenus avec ceux de l'approche de Wolosewicz *et al.* et ceux du solveur IBM ILOG CPLEX. Enfin, nous concluons dans la Section 3.6 sur la pertinence de cette approche.

3.2 Modélisation

3.2.1 Description du problème

Nous nous intéressons à la résolution du problème intégré dans des systèmes de type job-shop. Nous voulons planifier la production de N produits sur T périodes à capacité finie, en utilisant M machines, tout en ordonnant O opérations qui composent les étapes de production (gamme) des différents ordres de fabrication (jobs). Le but est de satisfaire les demandes des différents clients dans les délais (la rupture de stock n'est pas autorisée) au moindre coût, incluant la somme des coûts de production, de stockage et de setup (ou lancement de la production). Le traitement d'une opération ne peut pas être arrêté avant que celle-ci ne soit terminée (la préemption n'est pas autorisée), et une machine ne peut traiter qu'une opération à la fois. Des délais d'obtention sont définis pour chaque produit. La demande est dynamique et connue à l'avance. La capacité de production du système est limitée, et la capacité de stockage est considérée infinie. Les durées unitaires opératoires et de setup des opérations sont connues à l'avance. Les coûts et les temps de setup sont indépendants de la séquence. Les notations utilisées pour définir les données considérées dans ce problème sont les suivantes :

- $D_{i,l}$: demande du produit i à la fin de la période l ,
- c_i^p : coût unitaire de production du produit i ,
- c_i^{inv} : coût unitaire de stockage du produit i ,
- c_i^s : coût unitaire de setup du produit i ,
- L_i : délai d'obtention du produit i ,
- $i(o)$: produit associé à l'opération o ,
- $l(o)$: période associée à l'opération o ,
- $m(o)$: ressource sur laquelle l'opération o doit être traitée,
- p_o^u : durée opératoire de l'opération o par unité du produit $i(o)$,
- s_o^t : temps de setup de l'opération o par unité du produit $i(o)$,
- $r(o)$: date de disponibilité de l'opération o ,
- $d(o)$: date de fin souhaitée de l'opération o ,
- c_l : longueur de la période l (capacité disponible),
- \mathcal{O} : ensemble des opérations,
- \mathcal{A} : ensemble des paires d'opération dans la gamme des produits ($(o, o') \in \mathcal{A}$ signifie que l'opération o précède l'opération o' dans la gamme opératoire),
- \mathcal{L} : ensemble des dernières opérations dans les gammes opératoires,
- \mathcal{F} : ensemble des premières opérations dans les gammes opératoires,
- \mathcal{E} : ensemble des paires d'opérations qui sont traitées sur la même ressource,

- $\mathcal{S}(y)$: séquence des opérations associées à la séquence y ($(o, o') \in \mathcal{S}(y)$ signifie que l'opération o précède l'opération o' dans la séquence d'une ressource).

Du point de vue de la planification, il s'agit de déterminer la taille de lot de chaque produit à chaque période, et du point de vue de l'ordonnancement, il s'agit de déterminer les dates de début et de fin de chaque opération et leur ordre de passage sur les machines.

Le problème d'ordonnancement est modélisé à travers un graphe disjonctif, où les nœuds correspondent aux opérations des jobs et les arcs représentent les contraintes entre les opérations. Les arcs reliant les opérations des gammes de produit ont toujours la même orientation (arcs conjonctifs); tandis que l'orientation des arcs reliant les opérations traitées sur la même ressource doit être définie (arcs disjonctifs). Le problème d'ordonnancement consiste donc à trouver la meilleure orientation pour tous les arcs disjonctifs, i.e. déterminer un graphe conjonctif sans créer des circuits. La Figure 3.1 présente un exemple de graphe disjonctif, ainsi qu'un graphe conjonctif possible pour ce graphe disjonctif.

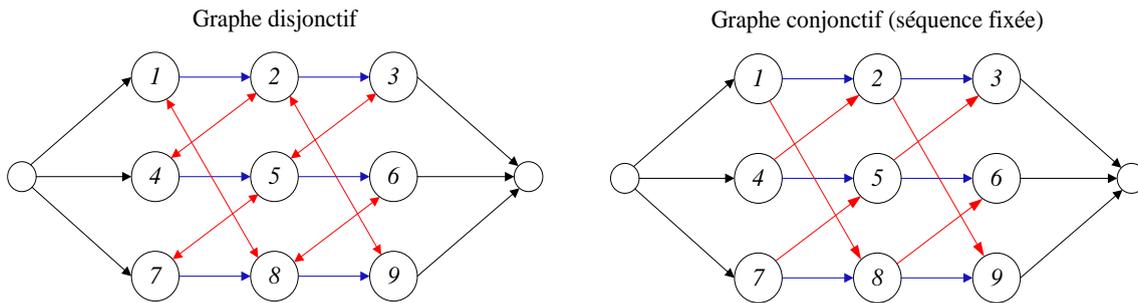


FIGURE 3.1 – Exemple d'un graphe disjonctif et d'un graphe conjonctif associé pour un atelier de type job-shop avec 3 jobs et 3 machines

Dans cet exemple, les opérations 1, 2 et 3 font partie de la gamme de fabrication du job J_1 et doivent être traitées dans l'ordre indiqué par les arcs conjonctifs. De la même façon, les opérations 4, 5 et 6 sont nécessaires pour produire le job J_2 , tandis que les opérations 7, 8 et 9 permettent de produire le job J_3 . Les opérations 1, 8 et 6 doivent être traitées sur la ressource R_1 ; les opérations 4, 2 et 9 sur la ressource R_2 , et les opérations 7, 5 et 3 sur la ressource R_3 .

3.2.2 Modèle intégré

L'objectif étant de minimiser la somme des coûts de planification, les **variables de décision** que nous considérons sont celles qui sont habituellement utilisées dans les modèles de dimensionnement de lots, i.e. *quand produire ?* et *combien produire ?*. Ces variables sont modélisées par les notations suivantes :

- $Y_{i,l}$: variable de setup (= 1 si le produit i est fabriqué à la période l , 0 sinon).
- $X_{i,l}$: taille de lot du produit i disponible à la fin de la période l ,

Les **variables dépendantes** (variant en fonction de Y et X) du problème intégré sont les niveaux de stock des produits et les dates de début des opérations. Elles sont définies par les notations suivantes :

- $I_{i,l}$: inventaire du produit i à la fin de la période l ,
- $t(o)$: date de début de l'opération o ,

Les dates de début des opérations servent notamment à modéliser les contraintes de précédence entre les opérations faisant partie de la même gamme de fabrication et entre celles étant traitées sur la même ressource. Ces contraintes permettent de garantir la cohérence du plan de production au niveau ordonnancement.

Les contraintes de précédence entre les opérations d'un même job étant toujours les mêmes, indépendamment de la séquence, elles sont représentées par l'équation (3.1).

$$t(o') \geq t(o) + p_o^u X_{i(o),l(o)} + s_o^t Y_{i(o),l(o)} \quad \forall (o, o') \in \mathcal{A} \quad (3.1)$$

D'autre part, les contraintes de précédence entre les opérations partageant la même ressource sont modélisées par l'équation (3.2) qui tient compte de la disjonction.

$$\begin{cases} t(o') \geq t(o) + p_o^u X_{i(o),l(o)} + s_o^t Y_{i(o),l(o)} \\ \text{ou} \\ t(o) \geq t(o') + p_{o'}^u X_{i(o'),l(o')} + s_{o'}^t Y_{i(o'),l(o')} \end{cases} \quad \forall (o, o') \in \mathcal{E} \quad (3.2)$$

Si une séquence y est fixée (graphe conjonctif), les contraintes de précédence entre les opérations réalisées sur la même ressource sont définies par l'équation (3.3).

$$t(o') \geq t(o) + p_o^u X_{i(o),l(o)} + s_o^t Y_{i(o),l(o)} \quad \forall (o, o') \in \mathcal{S}(y) \quad (3.3)$$

Un modèle intégré de dimensionnement de lots et d'ordonnancement doit inclure les contraintes (3.1) et (3.2) pour garantir la recherche de la meilleure solution réalisable possible. L'équation (3.3) permet de garantir une solution réalisable, mais pas forcément optimale.

Nous utilisons dans ce chapitre le modèle MIP proposé par Dauzère-Pérès et Lasserre dans [48], mais sans tenir compte de l'aspect multi-niveaux ni de la rupture de stocks. C'est un modèle incluant les contraintes traditionnelles de dimensionnement de lots, avec en plus des contraintes

de capacité et d'ordonnancement détaillées, et en tenant compte des délais d'obtention des produits. Cette modélisation permet donc de planifier la production au niveau tactique, en incluant des décisions opérationnelles. Le modèle est le suivant :

$$\min \sum_{i=1}^N \sum_{l=1}^T (c_i^p X_{il} + c_i^{inv} I_{il} + c_i^s Y_{il}) \quad (3.4)$$

s.c.

$$I_{il} = I_{i,l-1} + X_{il} - D_{il} \quad \forall i, l \quad (3.5)$$

$$t(o') \geq t(o) + p_o^u X_{i(o),l(o)} + s_o^t Y_{i(o),l(o)} \quad \forall (o, o') \in \mathcal{A} \quad (3.1)$$

$$\left\{ \begin{array}{l} t(o) \geq t(o') + p_{o'}^u X_{i(o'),l(o')} + s_{o'}^t Y_{i(o'),l(o')} \\ \text{ou} \\ t(o') \geq t(o) + p_o^u X_{i(o),l(o)} + s_o^t Y_{i(o),l(o)} \end{array} \right. \quad \forall (o, o') \in \mathcal{E} \quad (3.2)$$

$$t(o) + p_o^u X_{i(o),l(o)} + s_o Y_{i(o),l(o)} \leq \sum_{l=1}^{l(o)} c_l \quad \forall o \in \mathcal{L} \quad (3.6)$$

$$t(o) + p_o^u X_{i(o),l(o)} + s_o Y_{i(o),l(o)} \geq \sum_{l=1}^{l(o)-1} c_l \quad \forall o \in \mathcal{L} \quad (3.7)$$

$$t(o) \geq \sum_{l=1}^{l(o)-L_{i(o)}} c_l \quad \forall o \in \mathcal{F} \text{ et } L_{i(o)} > 0 \quad (3.8)$$

$$X_{il} \leq \left(\sum_{k=1}^T D_{ik} \right) Y_{il} \quad \forall i, l \quad (3.9)$$

$$X_{il}, I_{il} \geq 0 \quad \forall i, l \quad (3.10)$$

$$Y_{il} \in \{0, 1\} \quad \forall i, l \quad (3.11)$$

$$t(o) \geq 0 \quad \forall o \quad (3.12)$$

La fonction objectif (3.4) minimise le coût total (coûts de production, de stockage et de setup). Les contraintes (3.5) correspondent aux équations d'équilibre de stock. Les contraintes (3.1) permettent de garantir les contraintes de précédence entre les opérations faisant partie de la même gamme opératoire. Les contraintes (3.2) garantissent les contraintes de précédence entre les opérations partageant la même ressource. Les contraintes (3.6), (3.7) et (3.8) correspondent aux contraintes de capacité. Plus précisément, les contraintes (3.6) forcent la dernière opération de gamme à finir au plus tard à la date de fin de sa période associée ; les contraintes (3.7) forcent

la dernière opération de gamme à finir au plus tôt à la date de début de sa période associée et, les contraintes (3.8) forcent la première opération de gamme à démarrer au plus tôt à la date de début minimale respectant le délai d'obtention. Les contraintes (3.9) permettent de lier les variables de décisions (taille de lot et setup). Les contraintes (3.10) indiquent la non-négativité de la taille de lot et de l'inventaire. Les contraintes (3.11) forcent les variables de setup à prendre des valeurs binaires, et les contraintes (3.12) garantissent la non-négativité des dates de début des opérations.

3.2.3 Modèle avec chemins et séquence fixée

Afin de réduire la difficulté associée à la résolution du modèle intégré, nous utilisons une stratégie qui consiste à résoudre un sous-ensemble de problèmes de planification à séquence fixée. Une première version de cette approche a été proposée dans [233, 234, 235], et nous y avons apporté des modifications jusqu'à obtenir une version améliorée que nous présentons dans la section 3.3. Cette approche est basée sur la résolution d'un modèle utilisant les chemins du graphe conjonctif associé à une séquence fixée.

La première modification consiste à exprimer les contraintes de capacité, non pas en fonction de la date de début des opérations, mais en fonction de la date de disponibilité, c'est-à-dire la date minimale à laquelle une opération peut démarrer. Ces dates sont définies à travers les contraintes (3.13), (3.14), (3.15) et (3.16). Les contraintes (3.13) permettent d'établir que la première opération de gamme doit démarrer à une date garantissant le respect du délai d'obtention. À travers les contraintes (3.14), la dernière opération de chaque gamme est disponible à partir de la date de début de sa période associée (période de production du produit $i(o)$). Les contraintes (3.15) définissent la date de disponibilité pour les opérations comme étant à la fois la première et la dernière dans la gamme de fabrication. La date de disponibilité n'étant pas considérée pour les opérations intermédiaires, les contraintes (3.16) sont utilisées pour formaliser la définition. La différence par rapport au modèle intégré précédent est que la dernière opération de chaque gamme doit démarrer et finir dans sa période associée. Dans le modèle intégré, cette opération peut démarrer à une période différente.

$$r(o) = \sum_{l=1}^{l(o)-L_{i(o)}} c_l \quad \forall o \in \mathcal{F} \text{ et } o \notin \mathcal{L} \quad (3.13)$$

$$r(o) = \sum_{l=1}^{l(o)-1} c_l \quad \forall o \in \mathcal{L} \text{ et } o \notin \mathcal{F} \quad (3.14)$$

$$r(o) = \max \left(\sum_{l=1}^{l(o)-L_{i(o)}} c_l, \sum_{l=1}^{l(o)-1} c_l \right) \quad \forall o \in \mathcal{F} \cap \mathcal{L} \quad (3.15)$$

$$r(o) = 0 \quad \forall o \notin \mathcal{F} \cup \mathcal{L} \quad (3.16)$$

La première opération de chaque job peut démarrer à une période entre 0 et $l(o)$ si $L_{i(o)} = 0$ ou entre $l(o) - L_{i(o)}$ et $l(o)$ si $L_{i(o)} > 0$. La dernière opération de chaque job doit démarrer et terminer dans sa période associée $l(o)$. La date de disponibilité $r(o)$ est connue uniquement pour la première et la dernière opération de chaque job, et une date de fin souhaitée $d(o)$ est imposée uniquement pour la dernière opération de chaque job, comme défini par (3.17). Le respect de la date de fin souhaitée est garanti par (3.25).

$$d(o) = \sum_{l=1}^{l(o)} c_l \quad \forall o \in \mathcal{L} \quad (3.17)$$

D'autres variables dépendantes qui ne font pas partie de la modélisation mathématique, mais qui sont utilisées par le graphe conjonctif pour le calcul des marges de capacité, sont les suivantes :

- $t_e(o)$: date de début au plus tôt de l'opération o ,
- $t_l(o)$: date de début au plus tard de l'opération o ,
- $f(o)$: date de fin de l'opération o ,
- $f_e(o)$: date de fin au plus tôt de l'opération o ,
- $f_l(o)$: date de fin au plus tard de l'opération o .

Les Définitions 1 et 2 permettent de lier les dates de début au plus tôt $t_e(o)$ et plus tard $t_l(o)$ et les dates de fin au plus tôt $f_e(o)$ et au plus tard $f_l(o)$ de l'opération o .

- **Définition 1** : $t_e(o) \leq t(o) \leq t_l(o) \quad \forall o \in \mathcal{O}$.
- **Définition 2** : $f_e(o) \leq f(o) \leq f_l(o) \quad \forall o \in \mathcal{O}$.

En suivant la logique du graphe conjonctif, les dates de début au plus tôt sont déterminées de manière récursive, tout au long de l'horizon de planification, à partir du sommet source O (avec $t_e(O) = 0$), et ensuite, les dates de début au plus tard sont déterminées d'avant en arrière, à partir du sommet puits $*$. Comme illustré sur la Figure 3.2, chaque sommet du graphe conjonctif représente une opération, chaque sommet ayant plusieurs prédécesseurs et successeurs, que nous généralisons dans le Tableau 3.1.

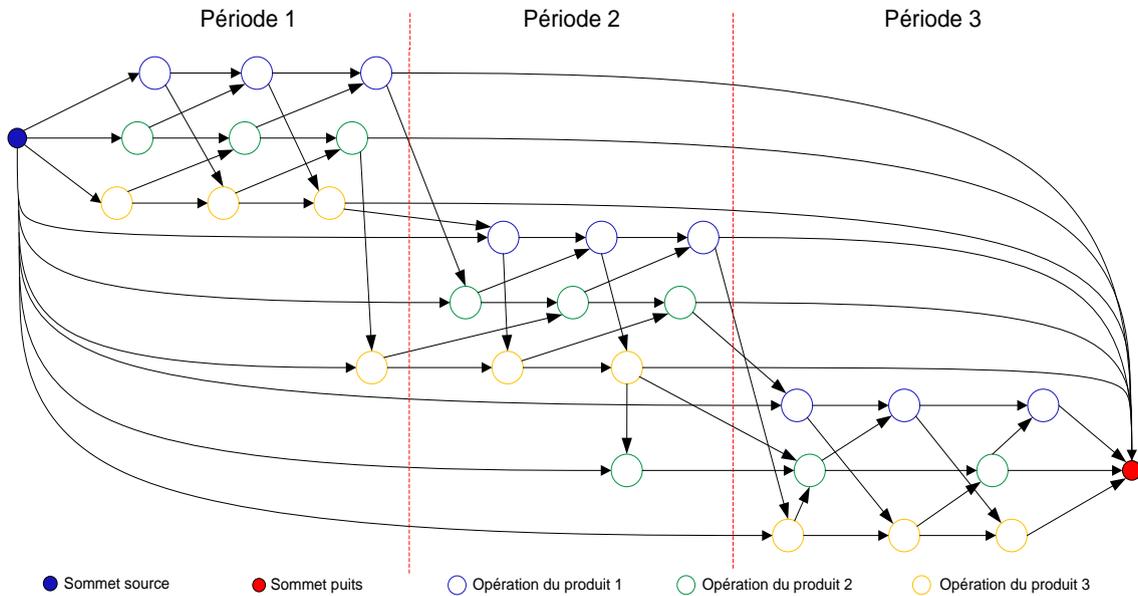


FIGURE 3.2 – Graphe conjonctif pour un atelier de type job-shop avec 3 produits, 3 machines 3 périodes

TABLEAU 3.1 – Prédécesseurs et successeurs dans le graphe conjonctif

Sommet	Prédécesseurs	Successeurs
Source (O)	Aucun	\mathcal{F}
Première opération de gamme ($o \in \mathcal{F}$)	- O - o' avec $(o', o) \in \mathcal{S}(y)$	- o' avec $(o, o') \in \mathcal{A}$ - o'' avec $(o, o'') \in \mathcal{S}(y)$
Opération intermédiaire ($o \notin \mathcal{F} \cup \mathcal{L}$)	- o' avec $(o, o') \in \mathcal{A}$ - o'' avec $(o'', o) \in \mathcal{S}(y)$	- o' avec $(o, o') \in \mathcal{A}$ - o'' avec $(o, o'') \in \mathcal{S}(y)$
Dernière opération de gamme ($o \in \mathcal{L}$)	- o' avec $(o', o) \in \mathcal{A}$ - o'' avec $(o'', o) \in \mathcal{S}(y)$	- o' avec $(o, o') \in \mathcal{S}(y)$ - *
Puits (*)	\mathcal{L}	Aucun

Appelons maintenant $\mathcal{SP}(s)$ l'ensemble de sommets prédécesseurs et $\mathcal{SS}(s)$ l'ensemble de sommets successeurs du sommet $s \in \mathcal{O} \cup \{O, *\}$. Pour chaque prédécesseur n du sommet s ($n \in \mathcal{SP}(s)$), une date fictive $\lambda(n)$ est calculée avec (3.18).

$$\lambda(n) = \begin{cases} r(s) & \text{si } n = O \\ t_e(n) + p_n^u X_{i(n),l(n)} + s_n^t Y_{i(n),l(n)} + \sum_{k=l(n)+1}^T c_l & \text{si } s = * \\ t_e(n) + p_n^u X_{i(n),l(n)} + s_n^t Y_{i(n),l(n)} & \text{sinon} \end{cases} \quad (3.18)$$

À noter que dans ce cas, $r(s)$ est la longueur de l'arc entre $n = O$ et s quand $s \in \mathcal{F}$; $p_n^u X_{i(n),l(n)} + s_n^t Y_{i(n),l(n)} + \sum_{k=l(n)+1}^T c_l$ est la longueur de l'arc entre n et $s = *$ quand $n \in \mathcal{L}$ et, $p_n^u X_{i(n),l(n)} + s_n^t Y_{i(n),l(n)}$ est la longueur de l'arc entre n et s quand $s \notin \mathcal{F} \wedge n \notin \mathcal{L}$.

Les dates de début au plus tôt sont finalement calculées avec (3.19).

$$t_e(s) = \begin{cases} r(s) & \text{si } s \in \mathcal{L} \wedge \max_{n \in \mathcal{SP}(s)} \lambda(n) < r(s) \\ \max_{n \in \mathcal{SP}(s)} \lambda(n) & \text{sinon} \end{cases} \quad \forall s \in \mathcal{O} \cup \{*\} \quad (3.19)$$

De manière similaire, pour chaque successeur n du sommet s ($n \in \mathcal{SS}(s)$), une date fictive $\mu(n)$ est calculée avec (3.20). La date de début au plus tard du sommet puits $t_l(*)$ est fixée comme étant égale à la date de début au plus tôt $t_e(*)$.

$$\mu(n) = \begin{cases} t_l(n) - r(n) & \text{si } s = O \\ t_l(n) - \left(p_s^u X_{i(s),l(s)} + s_s^t Y_{i(s),l(s)} + \sum_{k=l(s)+1}^T c_l \right) & \text{si } n = * \\ t_l(n) - \left(p_s^u X_{i(s),l(s)} + s_s^t Y_{i(s),l(s)} \right) & \text{sinon} \end{cases} \quad (3.20)$$

Les dates de début au plus tard sont finalement déterminées avec (3.21).

$$t_l(s) = \min_{n \in \mathcal{SS}(s)} \mu(n) \quad \forall s \in \mathcal{O} \cup \{O\} \quad (3.21)$$

Connaître les dates de début au plus tôt et au plus tard des opérations permet de calculer les marges de capacité et de gérer des éventuelles modifications du plan de production. La date de début au plus tôt du sommet puits (égale à la date de début au plus tard) correspond à la date de fin du dernier job du plan de production. Donc, **la capacité est violée si** $t_e(*) =$

$t_l(*) > \sum_{l=1}^T c_l$. Par ailleurs, minimiser le *makespan* correspond à minimiser la valeur de $t_e(*)$. La date de fin d'une opération o quelconque peut être déterminée à travers l'équation (3.22).

$$f(o) = t(o) + p_o^u X_{i(o),l(o)} + s_o^t Y_{i(o),l(o)} \quad \forall o \in \mathcal{O} \quad (3.22)$$

Les dates de fin au plus tôt et au plus tard, liées par la Définition 2, peuvent être calculées avec (3.23) et (3.24), respectivement.

$$f_e(o) = t_e(o) + p_o^u X_{i(o),l(o)} + s_o^t Y_{i(o),l(o)} \quad \forall o \in \mathcal{O} \quad (3.23)$$

$$f_l(o) = t_l(o) + p_o^u X_{i(o),l(o)} + s_o^t Y_{i(o),l(o)} \quad \forall o \in \mathcal{O} \quad (3.24)$$

Dans le modèle mathématique avec chemins et séquence fixée, un nouveau type de contraintes est introduit pour gérer la cohérence des liens de précedence entre les opérations et pour intégrer la capacité de production. Cette expression permet de modéliser les contraintes de capacité détaillées, en suivant les informations du graphe conjonctif. Pour chaque chemin c du graphe conjonctif, il existe une contrainte de capacité détaillée. La formulation est la suivante :

$$\min \sum_{i=1}^N \sum_{l=1}^T (c_i^p X_{il} + c_i^{inv} I_{il} + c_i^s Y_{il}) \quad (3.4)$$

s.c.

$$I_{il} = I_{i,l-1} + X_{il} - D_{il} \quad \forall i, l \quad (3.5)$$

$$r(o_c^f) + \sum_{o \in c} (p_o^u X_{i(o),l(o)} + s_o^t Y_{i(o),l(o)}) \leq \sum_{l=1}^{l(o_c^l)} c_l \quad \forall c \in \mathcal{C}(y) \quad (3.25)$$

$$X_{il} \leq \left(\sum_{k=1}^T D_{ik} \right) Y_{il} \quad \forall i, l \quad (3.9)$$

$$X_{il}, I_{il} \geq 0 \quad \forall i, l \quad (3.10)$$

$$Y_{il} \in \{0, 1\} \quad \forall i, l \quad (3.11)$$

La fonction objectif (3.4) et les contraintes d'équilibre des stocks (3.5) du modèle à séquence fixée sont similaires à celles du modèle intégré. La différence réside dans les contraintes (3.25), qui représentent les contraintes de capacité détaillées pour une séquence y , où c représente un chemin et $\mathcal{C}(y)$ est l'ensemble des chemins dans le graphe, et o_c^f et o_c^l sont les première et dernière

opérations du chemin c , respectivement. La différence entre les contraintes (3.25) et (3.3) est que les premières considèrent la capacité des périodes, tandis que les deuxièmes respectent uniquement les contraintes de précédence. Finalement, de la même manière que dans le modèle intégré, les contraintes (3.9) relient les variables de production X aux variables de setup Y , les contraintes (3.10) assurent la non-négativité des variables et les contraintes (3.11) gèrent la valeur des variables binaires de setup.

Ce modèle mathématique est une généralisation du CLSP, avec des contraintes de capacité détaillées qui servent à garantir le respect des contraintes de précédence pour une séquence fixée, en utilisant les informations dérivées du graphe conjonctif. Résoudre un problème de dimensionnement de lots avec des contraintes de capacité agrégées, comme le CLSP, est \mathcal{NP} -difficile pour les ateliers à une seule machine [31]. Donc, utiliser des contraintes de capacité détaillées pour des problèmes multi-ressources avec partage (de type job-shop), rend le problème encore plus difficile à résoudre. Le nombre de chemins dans le graphe conjonctif peut être très important (sa croissance est en effet exponentielle), et satisfaire toutes les contraintes de capacité peut devenir impossible. Notre stratégie de résolution consiste à appliquer une heuristique Lagrangienne pour résoudre le problème de dimensionnement de lots et d'ordonnancement avec séquence fixée, intégrée dans une recherche taboue qui permet d'améliorer la séquence.

3.3 Méthode de résolution

La plupart des problèmes de dimensionnement de lots avec contraintes de capacité sont \mathcal{NP} -difficiles. Si l'on ajoute des contraintes du type (3.1) et (3.2), le problème devient encore plus difficile à résoudre. Par ailleurs, le problème d'ordonnancement dans un atelier de type job-shop multi-périodes est l'un des problèmes les plus difficiles à résoudre (\mathcal{NP} -difficile au sens fort). La combinaison des variables et des contraintes des deux problèmes est donc \mathcal{NP} -difficile. L'implémentation d'une méthode exacte permettant de trouver des solutions optimales pour des problèmes de taille raisonnable paraît impossible. En outre, dans plusieurs cas, les solveurs commerciaux sont même incapables de trouver des solutions réalisables pour des problèmes de ce type. Face à ce constat, nous proposons une approche basée sur des heuristiques et illustrée dans la Figure 3.3. Premièrement, nous résolvons le problème de dimensionnement de lots avec contraintes de capacité détaillées pour une séquence fixée, et deuxièmement nous modifions le graphe conjonctif pour déterminer une nouvelle séquence. Cette procédure est répétée durant un certain nombre d'itérations.

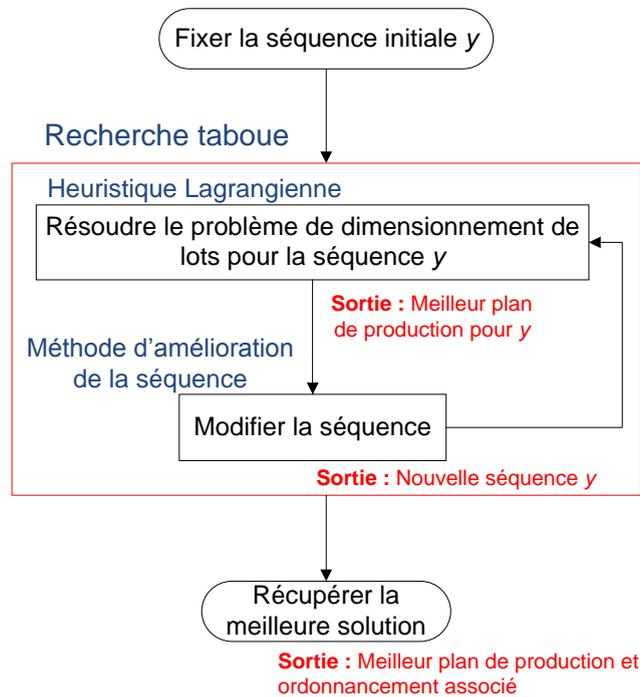


FIGURE 3.3 – Approche intégrée

3.3.1 Heuristique Lagrangienne pour la résolution du problème à séquence fixée

Pour réduire la difficulté de résolution du problème de dimensionnement de lots avec séquence fixée, nous utilisons une heuristique Lagrangienne qui a pour but de : (i) calculer un plan de production optimal pour la séquence fixée avec contraintes de capacité relâchées, et (ii) réparer les contraintes de capacité non respectées afin de construire une solution réalisable.

3.3.1.1 Relaxation Lagrangienne

La stratégie consiste à relâcher les contraintes de capacité, de façon à résoudre un ensemble de problèmes mono-produit à capacité infinie, en utilisant la propriété de Wagner et Whitin [231]. Seuls les coûts de setup et de stockage ont une influence sur le choix des variables de décision.

Pour chaque problème à un produit, nous combinons l'algorithme dynamique de dimensionnement de lots proposé par Wagelmans *et al.* [230], où chaque problème peut être résolu en un temps $O(T \log T)$, avec la technique de la relaxation Lagrangienne, en utilisant la méthode

des sous-gradients ([69]; [146]). La fonction (3.26) est la fonction objectif du problème dual, où β_c correspond au multiplicateur Lagrangien associé au chemin $c \in \mathcal{C}(y)$. Cette fonction est exprimée en termes des variables de décision X et Y et des multiplicateurs Lagrangiens β .

$$\begin{aligned}
F(X, Y, \beta) = \min \sum_{i=1}^N \sum_{l=1}^T \left\{ \left(c_i^s + \sum_{c \in \mathcal{C}(y)} \beta_c \sum_{o \in c} s_o^t \right) Y_{i,l} + \left[c_i^p + \sum_{k=l}^T c_i^{inv} \right. \right. \\
+ \left. \left. \sum_{c \in \mathcal{C}(y)} \beta_c \sum_{o \in c} p_o^u \right] X_{i,l} \right\} - \sum_{i=1}^N \sum_{l=1}^T D_{i,l} \sum_{k=l}^T c_i^{inv} \\
+ \sum_{c \in \mathcal{C}(y)} \beta_c \left[r(o_c^f) - \sum_{l=1}^{l(o_c^l)} c_l \right]
\end{aligned} \tag{3.26}$$

Comme le nombre de chemins $\mathcal{C}(y)$ dans le graphe conjonctif est exponentiel, nous mettons à jour dans la procédure uniquement les multiplicateurs Lagrangiens associés à l'ensemble $\mathcal{CH}(y)$ des chemins les plus violés. Un chemin c appartient à $\mathcal{CH}(y)$ si sa contrainte de capacité associée est la plus violée de tout le graphe à une itération donnée, i.e. si le chemin c est violé et permet de maximiser l'expression $r(o_c^f) + \sum_{o \in c} (p_o^u X_{i(o)l(o)} + s_o^t Y_{i(o)l(o)}) - \sum_{l=1}^{l(o_c^l)} c_l$.

À chaque itération, le coût du plan de production proposé par l'heuristique Lagrangienne est calculé avec (3.26). Puisque des contraintes sont relâchées, ce coût représente une *borne inférieure* $LB(y)$ du problème primal. À la première itération, comme $\beta_c = 0 \forall c$, le plan de production proposé correspond à la solution optimale du problème de dimensionnement de lots sans contraintes de capacité, et le coût associé correspond à la *borne inférieure absolue* (ALB). La plus grande borne inférieure $LB^*(y)$, trouvée par l'heuristique, est la *meilleure borne inférieure* pour la séquence y . À n'importe quelle itération de l'heuristique, si nous fixons $\beta_c = 0 \forall c$, le coût du plan de production est nommé *borne inférieure sans coûts Lagrangiens*, et est noté $LBWLC(y)$.

3.3.1.2 Heuristique de lissage

Comme la relaxation Lagrangienne ne conduit en général pas à des solutions réalisables, une heuristique de lissage de la production est incorporée, avec pour but de valider les contraintes de capacité violées. Cette procédure, qui modifie les tailles de lots X_{il} et les variables de setup Y_{il} , tout au long de l'horizon de planification, suit les étapes suivantes :

1. Sélectionner le job critique (produit critique i et période source l_s),
2. Sélectionner la période destination l_d et la quantité Q_{i,l_s,l_d} du produit i à transférer,
3. Faire le déplacement.

Le job critique est l'ordre de fabrication (i, l_s) , dont le produit critique i à la période source l_s contient la plus grande durée opératoire liée à des opérations sans marge, i.e. sans possibilité d'augmenter les tailles de lots car les dates de début au plus tôt et au plus tard de l'opération sont égales ($t_e(o) = t_l(o)$).

Les mouvements peuvent être réalisés d'avant en arrière ($l_s > l_d$) ou d'arrière en avant ($l_s < l_d$), le choix étant basé sur le moindre coût unitaire parmi tous les mouvements du voisinage, i.e. les différentes périodes destination possibles. Une période est considérée comme une possible période destination si elle ne contient pas d'opérations sans marge. Le coût unitaire C_u est calculé à travers l'équation suivante, où C_{avant} et C_{apres} correspondent aux coûts du plan de production avant et après le transfert, respectivement.

$$C_u = \frac{C_{apres} - C_{avant}}{Q_{i,l_s,l_d}} \quad (3.27)$$

La quantité de transfert Q_{i,l_s,l_d} est déterminée de la manière suivante :

– Si $l_s < l_d$:

1. Nous calculons la quantité maximale possible à déplacer Q_1 , afin de respecter les contraintes d'équilibre de stocks

$$Q_1 = \min_{l_s+1 \leq l < l_d} (I_{i,l_s-1} + X_{i,l_s} - D_{i,l_s}, I_{il}) \quad (3.28)$$

2. Nous calculons la quantité de transfert Q_{i,l_s,l_d} , respectant la condition précédente, la quantité de production disponible X_{il} et la marge de capacité *marge*, cette dernière étant définie par (3.30).

$$Q_{i,l_s,l_d} = \min(Q_1, X_{i,l_s}, \textit{marge}) \quad (3.29)$$

$$\textit{marge} = \min_{o \in (i,l_d)} m_o \quad (3.30)$$

Afin de pouvoir déterminer la marge de capacité, une marge cible m_o est calculée pour chaque opération o du produit critique i à la période destination l_d avec (3.31) si la production du produit i est déjà prévue à la période l_d ($Y_{il} = 1$), ou avec (3.32) sinon.

$$m_o = \frac{t_l(o) - t_e(o)}{\text{sommeDuree}_o + \text{sommeSetup}_o} \quad (3.31)$$

$$m_o = \frac{t_l(o) - t_e(o) - \text{sommeSetup}_o}{\text{sommeDuree}_o + \text{sommeSetup}_o} \quad (3.32)$$

La somme des durées unitaires opératoires sommeDuree_o et la somme des temps de setup sommeSetup_o sont calculées avec (3.33) et (3.34), respectivement, où $o^f(i, l_d)$ est la première opération dans la gamme de fabrication du job (i, l_d) .

$$\text{sommeDuree}_{o(i, l_d)} = \sum_{a=o^f(i, l_d)}^{o(i, l_d)} p_a^u \quad \forall o \in (i, l_d) \quad (3.33)$$

$$\text{sommeSetup}_{o(i, l_d)} = \sum_{a=o^f(i, l_d)}^{o(i, l_d)} s_a^t \quad \forall o \in (i, l_d) \quad (3.34)$$

- Si $l_d < l_s$, les mouvements ne risquent pas de violer les contraintes d'équilibre de stocks, et la quantité de transfert Q_{i, l_s, l_d} est calculée comme suit :

$$Q_{i, l_s, l_d} = \min(X_{i, l_s}, \text{marge}) \quad (3.35)$$

Les trois étapes de l'heuristique de lissage sont répétées jusqu'à ce que le plan de production soit réalisable, ou jusqu'à ce qu'il ne soit plus possible de faire de mouvements, comme dans [235]. Par ailleurs, quand un plan de production réalisable est obtenu, une nouvelle sous-procédure, visant à améliorer la qualité de la solution, est démarrée. Toutes les quantités de production de tous les produits qui peuvent être transférées avec une diminution du coût sont évaluées. Les meilleurs mouvements sont systématiquement effectués, et la procédure s'arrête quand il n'est plus possible de diminuer le coût du plan, comme illustré dans la Figure 3.4.

Dans notre approche, contrairement aux heuristiques de lissage trouvées dans la littérature où les changements sur les tailles de lots dans une période n'ont pas d'impact sur la capacité des autres périodes, les changements sur une période ont une influence sur la capacité de toutes les périodes. Ceci vient du fait que nous résolvons le problème d'ordonnancement sur tout l'horizon de planification, et non période par période, comme c'est le cas dans la littérature. La capacité consommée et les marges à chaque période varient dynamiquement en fonction des mouvements de l'heuristique de lissage, et il par conséquent difficile d'estimer à l'avance les mouvements nécessaires pour respecter les contraintes de capacité. L'avantage est que la

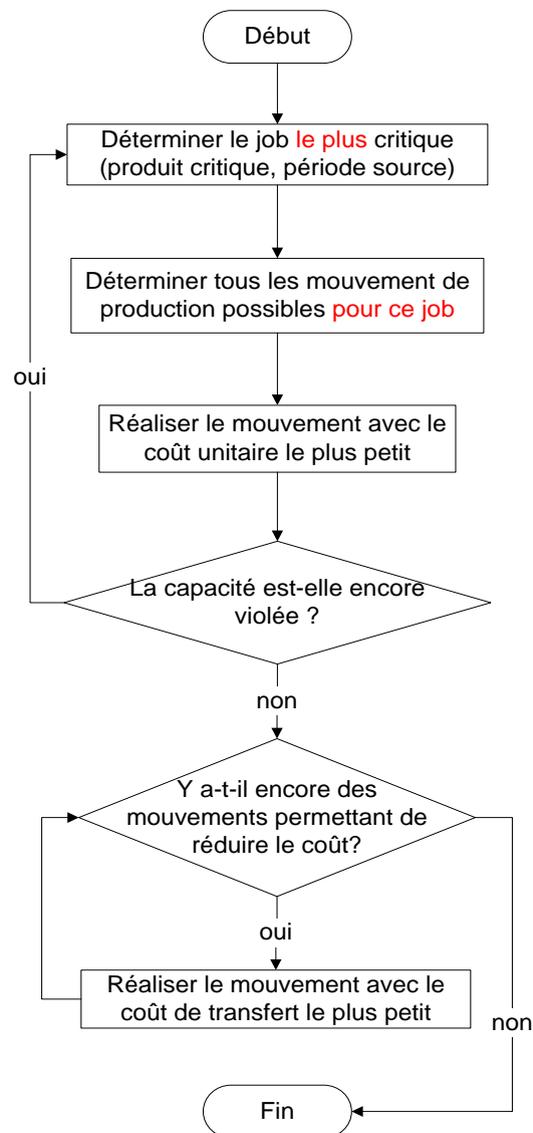


FIGURE 3.4 – Heuristique de lissage de la production

capacité est considérée de manière exacte, permettant une distribution plus efficace des tâches de production.

Le coût du plan final, obtenu à partir de l'heuristique de lissage, correspond à une borne supérieure (UB) du problème intégré de dimensionnement de lots et d'ordonnancement.

3.3.2 Méthode d'amélioration de la séquence

L'heuristique Lagrangienne permet d'obtenir le meilleur plan de production réalisable pour une séquence fixée. Néanmoins, en tenant compte du fait que seulement une instance du graphe disjonctif est considérée, la solution correspond à un optimum local. Pour obtenir ou au moins se rapprocher d'un optimum global, et dans certains cas arriver à trouver une solution réalisable, il est indispensable de modifier la séquence des opérations. C'est pourquoi nous appliquons une méthode d'amélioration de la séquence à l'aide d'une métaheuristique de type recherche taboue [82].

Pour générer une nouvelle séquence, nous changeons l'orientation d'un arc liant deux opérations partageant la même ressource, en prenant comme base le graphe conjonctif courant. Le problème est que le nombre de séquences possibles est exponentiel, et évaluer tous les changements possibles est très coûteux. C'est pourquoi il est très important d'avoir une méthode efficace, permettant de guider la recherche et de changer uniquement l'orientation des arcs les plus prometteurs. Nous appliquons une recherche taboue en utilisant l'information provenant de l'heuristique Lagrangienne. La procédure est résumée sur la Figure 3.5.

3.3.2.1 Principe

Comme il n'est pas possible d'explorer toutes les séquences, nous devons d'abord définir l'espace de recherche de l'heuristique. Nous décidons de restreindre la recherche aux arcs qui ont été violés durant la relaxation Lagrangienne associée à la séquence y , et nous dénotons cet ensemble par $\mathcal{VA}(y)$. Un arc $(o, o') \in \mathcal{S}(y)$ est considéré comme violé s'il est sur un chemin violé, i.e. (o, o') est inclus dans $\mathcal{VA}(y)$ si $(o, o') \in c \in \mathcal{CH}(y)$, où $\mathcal{CH}(y)$ correspond à l'ensemble des chemins les plus violés durant la relaxation Lagrangienne.

Changer l'orientation des arcs appartenant à $\mathcal{VA}(y)$ peut aider à améliorer l'utilisation de la capacité, en diminuant le *makespan*. Cependant, il est difficile d'estimer à l'avance les arcs qui peuvent contribuer à améliorer la qualité de la solution. Par ailleurs, tous les arcs violés ne contribuent pas à la violation de la capacité, et leur impact dépend de la configuration du graphe conjonctif courant. Donc, pour s'adapter à l'évolution du graphe, il est très important de pouvoir changer l'orientation des arcs qui avaient été inversés auparavant. Néanmoins, il n'est pas approprié de les inverser immédiatement après un « mauvais » changement, parce que ceci empêcherait le graphe d'évoluer vers de nouvelles solutions, et la solution finale pourrait converger rapidement vers un optimum local. Il faut ainsi laisser le graphe conjonctif évoluer durant quelques itérations de la recherche taboue, en acceptant des changements d'arcs qui n'améliorent pas forcément la borne supérieure. Cette stratégie peut conduire à l'obtention

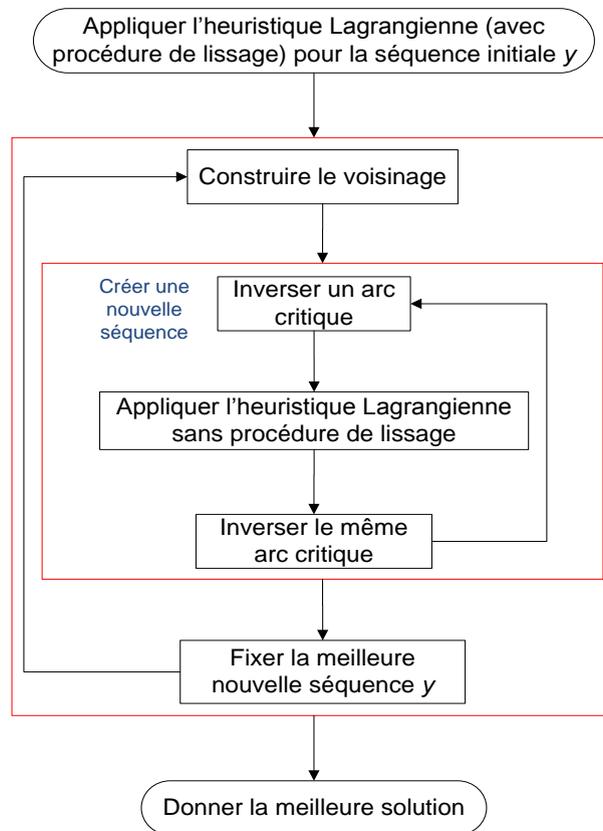


FIGURE 3.5 – Méthode d'amélioration de la séquence

d'une bonne solution, qui ne serait pas atteignable autrement. Dans la recherche taboue, la profondeur de l'exploration de l'espace de recherche est gérée par la taille de la liste taboue. Si un arc $(o, o') \in \mathcal{VA}(y)$ est changé, il est inclus dans la liste taboue, noté \mathcal{TL} , et est enlevé de $\mathcal{VA}(y)$. Ainsi, selon la profondeur d'exploration désirée, la taille maximale de la liste taboue ($size_max$) est fixée. Une fois la liste taboue complètement remplie ($|\mathcal{TL}| > size_max$), l'arc occupant la première position de la liste est enlevé. Définir la taille de la liste taboue est un problème de compromis entre qualité de la solution (profondeur d'exploration) et temps de calcul. Dans l'approche proposée par Wolosewicz *et al.* [233, 234, 235], la taille de la liste taboue était infinie, ce qui ne permettait pas une exploration appropriée de l'espace de recherche. Dans notre approche, la liste taboue est finie. Nous avons réalisé différents tests afin de déterminer la taille la plus appropriée.

3.3.2.2 Voisinage

Comme $|\mathcal{VA}(y)|$ peut être très grand, évaluer tous les arcs violés devient rapidement très coûteux en temps de calcul. Pour augmenter la vitesse de la procédure, nous choisissons de construire un voisinage qui est l'ensemble des arcs critiques $\mathcal{CA}(y)$, avec $\mathcal{CA}(y) \subset \mathcal{VA}(y)$, et nous évaluons non pas l'ensemble des arcs violés, mais l'ensemble des arcs critiques. L'idée est de mesurer l'impact de changer l'orientation de chacun des arcs critiques, pour finalement inverser l'arc qui satisfait le mieux le critère choisi.

Pour sélectionner les arcs critiques, nous définissons tout d'abord un poids $w_{(o,o')} \forall (o, o') \in \mathcal{VA}(y)$, et nous sélectionnons ensuite, tous les arcs $(o, o') \in \mathcal{VA}(y)$, pour lesquels $w_{(o,o')} \geq \alpha w_{max}$, avec $\alpha \in \mathbb{R}$, $0 \leq \alpha \leq 1$ étant un facteur de réduction et $w_{max} = \max_{(o,o') \in \mathcal{VA}(y)} w_{(o,o')}$ étant le poids maximal parmi tous les arcs violés. Ainsi, α est fixé selon la taille d'exploration désirée et les limitations en temps de calcul. Tous les arcs respectant cette condition sont des arcs critiques et font donc partie de $\mathcal{CA}(y)$.

De nombreuses expérimentations ont été réalisées pour déterminer la valeur appropriée de α . Quand α tend vers 0, une très bonne séquence peut systématiquement être choisie, mais le temps de calcul augmente beaucoup, et seulement quelques arcs peuvent être changés en un temps raisonnable. À l'inverse, quand α tend vers 1, des très mauvaises séquences peuvent parfois être choisies, mais comme le voisinage n'est pas très grand, l'exploration peut être rapidement complétée. Cela permet d'appliquer la recherche taboue pendant beaucoup plus d'itérations, ce qui conduit généralement à l'obtention d'une meilleure solution.

De manière similaire, deux méthodes d'affectation du poids des arcs ont été étudiés. Dans la première méthode, $w_{(o,o')} = \sum_{c \in \mathcal{CH}(y); (o,o') \in c} \beta_c \forall (o, o') \in \mathcal{VA}(y)$, c'est-à-dire que nous considérons des arcs avec une grande somme de multiplicateurs Lagrangiens. L'idée est que ces arcs sont potentiellement les plus violés, ou au moins ils appartiennent aux chemins les plus violés. Donc, changer l'orientation d'un de ces arcs est probablement judicieux si on veut réduire la violation de la capacité. La deuxième méthode analysée consiste à calculer le poids d'un arc comme le nombre de chemins violés qui contiennent l'arc. De cette façon, changer l'orientation de l'arc avec le plus grand poids a un impact sur un grand nombre de chemins, ce qui peut contribuer à réduire la violation du graphe de façon importante, en réalisant un seul mouvement. Les deux méthodes ont été comparées à travers plusieurs expérimentations, et même si aucune des deux méthodes n'arrive à dominer l'autre sur toutes les instances étudiées, nous avons décidé d'appliquer la deuxième méthode, car elle conduit à des meilleurs résultats dans la plupart des instances. Plus de détails sont données dans la Section 3.4.

3.3.2.3 Sélection de la nouvelle séquence

Une fois que le voisinage est créé, nous devons choisir un critère pour sélectionner l'arc qui va être finalement inversé. Les trois critères que nous avons analysés sont basés sur : $LB^*(y)$, $LBWLC(y)$ et $UB^*(y)$. L'arc générant la meilleure valeur pour le critère étudié est celui qui est changé.

L'avantage d'utiliser $UB^*(y)$ comme critère de décision est qu'elle permet d'obtenir des meilleures solutions à court terme. Néanmoins, l'inconvénient est que l'heuristique de lissage consomme beaucoup de temps, surtout quand le nombre de périodes est important. Le nombre de séquences changées risque ainsi d'être petit, ne permettant pas d'atteindre une solution de bonne qualité.

D'autre part, utiliser $LB^*(y)$ ou $LBWLC(y)$ permet d'éviter de mettre en œuvre l'heuristique de lissage, et ainsi d'évaluer plus de changements de séquences dans l'exploration globale de l'approche intégrée. Les solutions sont parfois de mauvaise qualité dans l'exploration, mais, dans les mêmes temps de calcul, les solutions obtenues sont meilleures que celles obtenues avec le critère basé sur la borne supérieure.

3.4 Analyses et améliorations

Nous avons mené plusieurs expérimentations pour identifier des parties à améliorer dans l'approche de Wolosewicz *et al.* Une fois les points faibles de la procédure identifiés, nous avons testé plusieurs modifications visant à améliorer les résultats, en termes de qualité de la solution et temps de calcul. Quelques idées ont été retenues dans notre approche ; tandis que les autres ont permis de mieux maîtriser le comportement de la procédure, de connaître les limitations de l'approche et d'avoir une vision sur des possibles améliorations à expérimenter.

Dans ce qui suit, nous détaillons les idées testées sur le problème de planification à séquence fixée et sur la partie amélioration de la séquence.

3.4.1 Résolution du problème à séquence fixée

L'heuristique Lagrangienne permettant de résoudre le problème à séquence fixée comporte trois étapes fondamentales : la résolution du problème mono-produit avec contraintes de capacité relâchées, la construction d'une solution réalisable et la mise à jour des multiplicateurs. Nous avons identifié des points faibles au niveau des deux premières étapes.

En ce qui concerne la résolution du problème mono-produit avec contraintes de capacité relâchées, nous n'avons pas trouvé de faiblesses par rapport à l'application de l'algorithme de programmation dynamique de dimensionnement de lots, mais par rapport au réglage des paramètres de la relaxation Lagrangienne. Ces paramètres ayant une influence directe sur l'évolution des multiplicateurs Lagrangiens et sur la rapidité de l'exploration, leur réglage initial a une forte incidence sur les bornes inférieures trouvées par la procédure.

Au niveau de l'heuristique de lissage, nous avons constaté que, dans certains cas, l'écart entre la solution optimale pour une séquence donnée et la borne supérieure pouvait être important, surtout pour des instances avec coûts de setup élevés par rapport aux coûts de stockage. Nous avons donc étudié plus en détail ces instances. En effet, lorsque les coûts de setup sont importants, la différence entre un plan de production avec k ordres de fabrication et un autre avec $k + 2$ ordres peut être très significative en termes de coûts. Nous avons analysé des alternatives pour améliorer la qualité des bornes supérieures, que nous décrivons dans la sous-section [3.4.1.2](#).

Une partie très importante de la procédure, où nous avons détecté une grande possibilité d'amélioration, est le lien entre l'algorithme de dimensionnement de lots avec contraintes relâchées et l'heuristique de lissage (autrement dit, entre LB et UB).

3.4.1.1 Paramètres de la relaxation Lagrangienne

Les paramètres de la relaxation Lagrangienne sont :

- Le nombre maximum d'itérations de l'heuristique,
- Le nombre maximum d'itérations consécutives sans amélioration,
- Le pas de déplacement initial,
- Le pas de déplacement minimum,
- Le facteur de réduction du pas.

Nous avons testé différents jeux de paramètres sur plusieurs instances, mais les résultats ne permettent pas de trouver les paramètres qui donnent la meilleure solution pour n'importe quel problème. Néanmoins, nous pouvons conclure que le pas de déplacement initial et le pas de déplacement minimum peuvent être fixés à la même valeur pour tous les problèmes sans trop perdre sur la qualité de la solution. D'autre part, concernant le nombre maximum d'itérations de la relaxation Lagrangienne et le nombre maximum d'itérations sans amélioration, on doit garder des valeurs qui génèrent globalement une solution de bonne qualité, car il n'y a pas une combinaison parfaite pour tous les problèmes.

Le paramètre le plus important est le nombre total d'itérations de l'heuristique Lagrangienne. Un grand nombre d'itérations permet d'obtenir une très bonne borne inférieure pour le problème de dimensionnement de lots à séquence fixée. En outre, l'heuristique de lissage de la production est appliquée plusieurs fois, donc plus de bornes supérieures sont générées et la probabilité d'obtenir un bon plan de production réalisable augmente. Cependant, plus le nombre d'itérations de l'heuristique Lagrangienne est important, plus le temps de calcul passé pour résoudre le problème de dimensionnement de lots à séquence fixée augmente. Étant donné que le temps total d'exécution est limité, moins de séquences sont explorées, c'est-à-dire que le nombre de solutions aux problèmes de dimensionnement de lots à séquence fixée est réduit.

Les résultats relatifs à la comparaison de plusieurs nombres maximums d'itérations peuvent être observés sur la Figure 3.6. L'écart sur la qualité EQ , calculé avec (3.36), permet de comparer les solutions obtenues avec les différentes valeurs du paramètre. Plus EQ est petit, plus le plan déterminé se rapproche de la meilleure solution parmi les autres alternatives. Dans cette expression, UB_i est la valeur de la meilleure borne supérieure obtenue en utilisant un nombre maximum d'itérations de i .

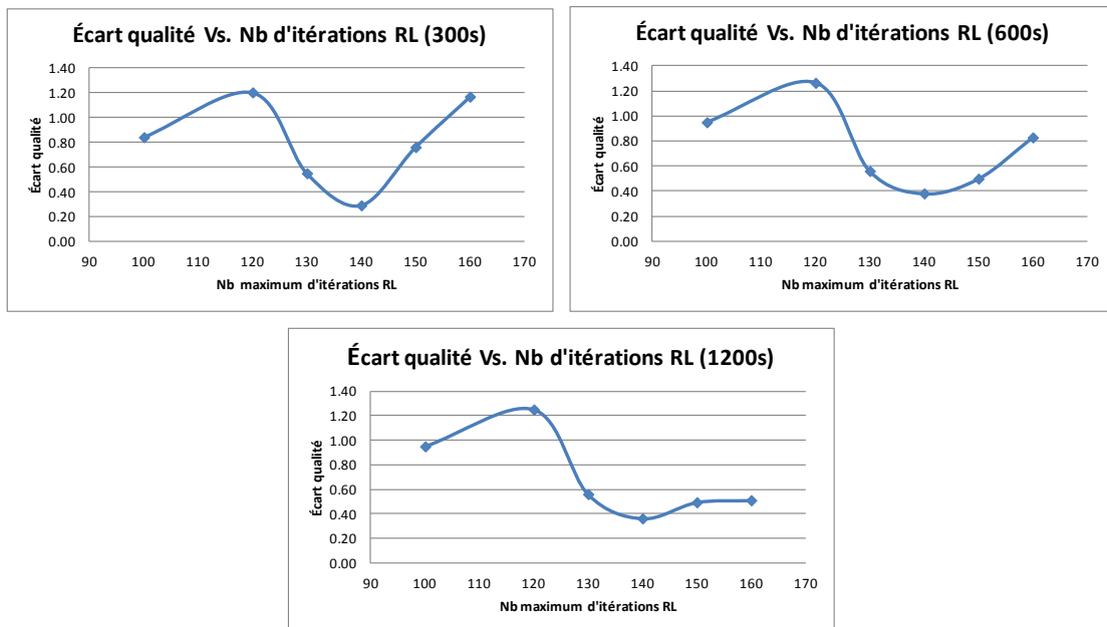


FIGURE 3.6 – Influence du nombre maximum d'itérations de la relaxation Lagrangienne sur le résultat final

$$EQ = \left(\frac{UB_i}{\min_i \{UB_i\}} - 1 \right) \times 100 \quad (3.36)$$

Selon le temps de calcul, l'influence du nombre maximum d'itérations peut être plus ou moins importante. Ainsi, pour un temps de calcul relativement petit, comme 300 secondes, un grand nombre d'itérations (supérieur à 140) peut conduire vers des mauvaises solutions. La raison est que plus de temps que nécessaire est utilisé pour l'heuristique Lagrangienne, laissant moins de temps pour la procédure d'amélioration de la séquence qui s'avère plus importante. D'autre part, avec un temps de calcul 4 fois plus important (1200 secondes), un grand nombre d'itérations a une influence très peu significative sur les résultats, car il y a plus de temps dédié à l'amélioration de la séquence. Globalement, le meilleur nombre maximum d'itérations est égal à 140.

Il est important de remarquer qu'un grand nombre maximum d'itérations peut être un bon choix quand la séquence initiale est proche d'une séquence optimale, i.e. quand peu de changements d'arcs sont nécessaires pour déterminer la solution optimale, ou quand la taille du problème est petite. D'autre part, si le nombre d'itérations est petit, la borne inférieure peut ne pas converger et le nombre de fois où l'heuristique de lissage est appliquée peut ne pas suffire à obtenir une bonne borne supérieure. De plus, des bornes inférieures et supérieures de mauvaise qualité peuvent conduire à des changements d'arcs inappropriés dans la procédure d'amélioration de la séquence. Le problème est de trouver un compromis entre intensification (nombre d'itérations de l'heuristique Lagrangienne) et diversification (nombre d'itérations de la méthode d'amélioration de la séquence), avec l'objectif de trouver le meilleur plan de production réalisable possible en un temps de calcul limité.

3.4.1.2 Heuristique de lissage de la production

L'heuristique de lissage utilisée dans l'approche de Wolosewicz *et al.* a une bonne performance. Elle est appropriée en termes de temps de calcul, car elle évalue les mouvements de production possibles associés à un seul job (le plus critique). Cependant, il n'y a pas de garantie que la borne supérieure obtenue soit optimale, et parfois il y a des écarts significatifs par rapport aux solutions optimales. Pour améliorer l'heuristique de lissage, nous avons analysé les alternatives suivantes :

- Ajouter une **étape d'amélioration**, une fois qu'une solution réalisable est déterminée. Dans [235], l'heuristique originale s'arrête lorsqu'un plan de production réalisable est

- déterminé. La nouvelle alternative consiste à chercher de nouveaux mouvements de production réduisant la valeur de la borne supérieure une fois que la solution est réalisable.
- Appliquer une **heuristique de lissage gloutonne**. Dans [235], un seul job critique est considéré à chaque itération, et toutes les périodes destination possibles et ses quantités de transfert associées sont donc évaluées. La nouvelle idée consiste à considérer tous les jobs critiques à chaque itération et à évaluer tous les mouvements possibles pour chacun d'entre eux. À chaque itération, le mouvement avec le coût unitaire de transfert le plus petit est réalisé. Une version simplifiée de l'algorithme est représentée sur la Figure 3.7.

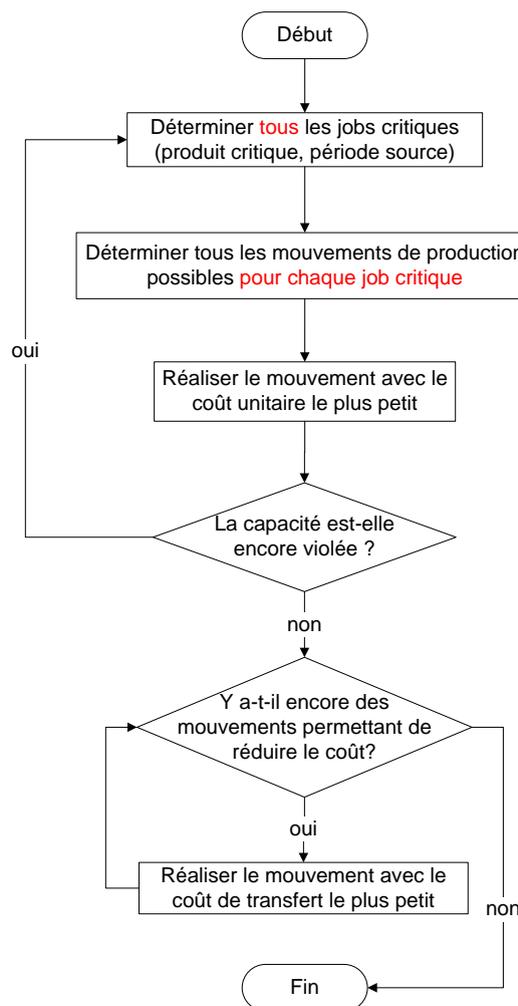


FIGURE 3.7 – Heuristique de lissage gloutonne

Ces deux alternatives supposent une augmentation du temps de calcul de l'heuristique Lagrangienne, réduisant le temps passé dans l'étape de modification de la séquence. Lorsque l'on

résout le problème de dimensionnement de lots à séquence fixée, la première alternative permet d'obtenir au minimum la même qualité de borne supérieure que l'heuristique originale, mais il n'y a pas de dominance lorsque l'on résout le problème intégré. En fait, l'heuristique originale permet de changer plus de séquences, et dans certains cas la solution finale peut être meilleure que celle avec la procédure améliorée. D'autre part, l'heuristique gloutonne peut fournir de très bonnes solutions pour quelques instances, mais elle peut aussi générer des solutions de mauvaise qualité, surtout parce que le nombre de séquences explorées est largement réduit. Sur 153 instances testées, variant différents paramètres pour différentes tailles des problèmes, la procédure originale trouve des meilleures solutions que la procédure gloutonne dans 70.97% des instances. De même, elle permet de changer plus de séquences dans 57.81% des instances.

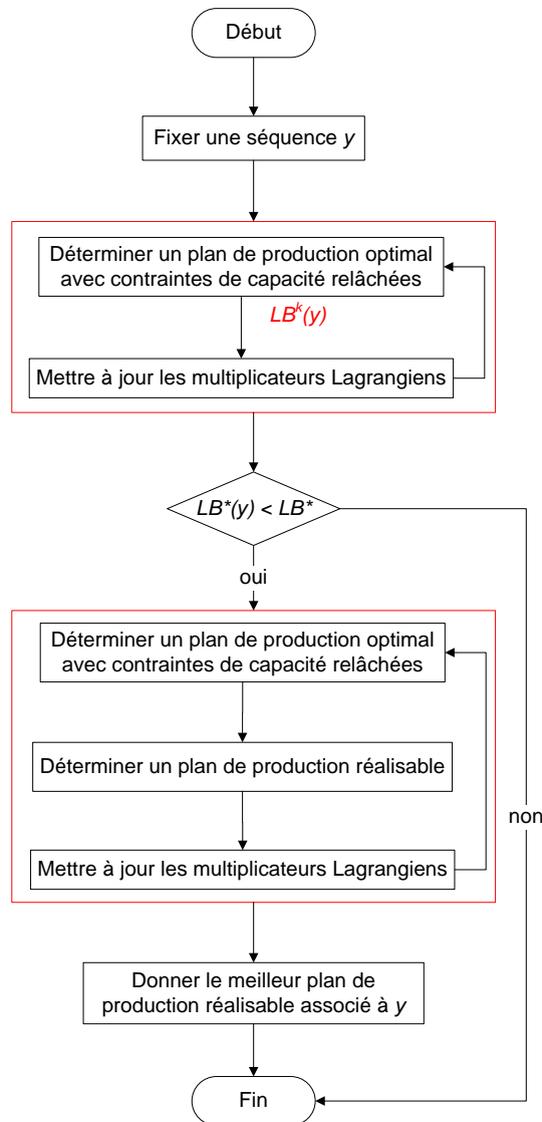
3.4.1.3 Lien entre la relaxation Lagrangienne et la procédure de lissage

Dans l'approche de Wolosewicz *et al.*, la relaxation Lagrangienne est appliquée deux fois pour chaque itération de la recherche taboue. La première a pour objectif d'obtenir une diminution de la meilleure borne inférieure $LB(y)$, et la deuxième est intégrée à l'heuristique de lissage de la production, pour calculer une borne supérieure. La deuxième exécution est déclenchée seulement si $LB(y) < LB^*(y)$, c'est-à-dire, si la borne inférieure est améliorée. Cette procédure est illustrée sur la Figure 3.8.

D'après plusieurs expérimentations réalisées, nous avons constaté que le fait d'obtenir une bonne borne supérieure n'est pas lié à la valeur de la borne inférieure. Autrement dit, une diminution de LB n'entraîne pas une diminution de UB . Donc, dans la procédure de Wolosewicz *et al.*, un ensemble important de bonnes solutions est négligé. Cette stratégie occasionne également des pertes importantes en temps de calcul, car la deuxième relaxation Lagrangienne donne exactement les mêmes résultats que la première. En effet, la séquence est la même et les multiplicateurs Lagrangiens sont réinitialisés à zéro.

Pour cette raison, dans notre approche, une fois qu'une séquence est fixée et pour déterminer un plan de production réalisable, la relaxation Lagrangienne et la procédure de lissage sont toujours combinées. Rappelons que dans la procédure d'amélioration de la séquence, la relaxation Lagrangienne est appliquée sans tenir compte de la procédure de lissage, le but n'étant pas de déterminer une solution réalisable mais de modifier la séquence.

Même si elle est importante, l'heuristique de lissage consomme beaucoup de temps de calcul, et il est par conséquent préférable de ne pas l'appliquer à chaque itération. Le temps de calcul est plus grand quand la capacité est faible, car beaucoup de chemins dans le graphe conjonctif sont violés, et il est nécessaire de modifier les dates de plusieurs nœuds, en réalisant de nom-

FIGURE 3.8 – Heuristique Lagrangienne de Wolosewicz *et al.* [235]

breux transferts de production pour obtenir une solution réalisable. Appliquer plusieurs fois l'heuristique de lissage augmente le temps de calcul de l'heuristique Lagrangienne et réduit le temps disponible pour explorer plus de séquences dans la recherche taboue. Une fois de plus, il s'agit de trouver un compromis entre intensification et diversification. L'objectif est de pouvoir décider quand appliquer l'heuristique de lissage, afin d'obtenir la meilleure borne supérieure possible avec le plus petit temps de calcul possible.

Quand la procédure de lissage est systématiquement appliquée, la meilleure borne supérieure pour chaque séquence est garantie, mais le temps de calcul est élevé. Dans plusieurs cas, le plan

de production proposé par l'algorithme de programmation dynamique de dimensionnement de lots est identique à celui obtenu à une itération précédente, et la borne supérieure associée est par conséquent aussi identique. Éviter de faire plusieurs fois le même calcul permet de réduire le temps de calcul de l'heuristique Lagrangienne.

Dans l'approche de Wolosewicz *et al.*, l'heuristique de lissage est appliquée toutes les 5 itérations, ce qui permet de réduire de manière significative le temps de calcul sur une séquence, mais de bonnes solutions peuvent être manquées par l'algorithme. Pour évaluer les pertes en qualité de solution, nous avons comparé cette alternative avec une stratégie d'application systématique (à chaque itération) de l'heuristique de lissage. Nous avons effectivement constaté que, pour chaque séquence fixée, il y a des écarts significatifs entre les bornes supérieures des deux stratégies. En effet, les solutions déterminées avec un lissage de la production toutes les 5 itérations sont d'une qualité très inférieure pour le problème à séquence fixée. Cependant, étant donné qu'une application systématique de l'heuristique de lissage réduit le temps pour réaliser les itérations de la recherche taboue, la borne supérieure finale (solution du problème intégré), obtenue avec la stratégie de lissage toutes les 5 itérations, est meilleure pour la plupart des instances testées.

Nous avons étudié une deuxième alternative, qui consiste à appliquer l'heuristique de lissage seulement quand $LBWLC^k(y) < UB^*(y)$ à l'itération k . Nous utilisons $LBWLC$ parce que sa valeur est généralement inférieure à UB , i.e. construire une solution réalisable entraîne une augmentation du coût du plan de production. Ceci est dû au fait que $LBWLC$ est déterminé avec des contraintes de capacité relâchées, et donc il y a moins de limitations sur la minimisation du coût. Cette stratégie permet de réduire le temps passé sur l'heuristique Lagrangienne, par rapport au lissage à chaque itération, et la meilleure borne supérieure est obtenue dans tous les cas. De plus, du temps additionnel peut être utilisé pour réaliser plus d'itérations de la recherche taboue, ce qui conduit vers de meilleurs résultats. Néanmoins, la stratégie du lissage toutes les 5 itérations donne des meilleurs résultats, car plus de séquences sont explorées.

Pour augmenter la diversification, nous avons conçu une nouvelle stratégie qui consiste à appliquer l'heuristique de lissage uniquement quand $LBWLC(y) < UB^*(y)$ et $LBWLC(y) \notin \mathcal{B}$, où \mathcal{B} est l'ensemble des 5 dernières valeurs de $LBWLC(y)$. Ce choix est basé sur l'évolution de $LBWLC$. En fait, le plan de production proposé par l'algorithme de programmation dynamique de dimensionnement de lots est souvent le même à différentes itérations. Dans notre méthode de résolution, il est normal d'obtenir un plan qui a été précédemment calculé, dans une fenêtre habituellement (mais pas toujours) inférieure à 5 itérations. Les résultats obtenus avec cette stratégie ont globalement amélioré les solutions des autres alternatives, sauf pour le cas avec construction systématique d'une solution réalisable toutes les 5 itérations. La qualité moyenne

des solutions est, en fait, quasiment la même. Cependant, la nouvelle alternative, que nous appelons *5-LBWL*C, utilise un critère plus cohérent et permet de résoudre plus efficacement le problème à séquence fixée. La nouvelle heuristique Lagrangienne est illustrée sur la Figure 3.9.

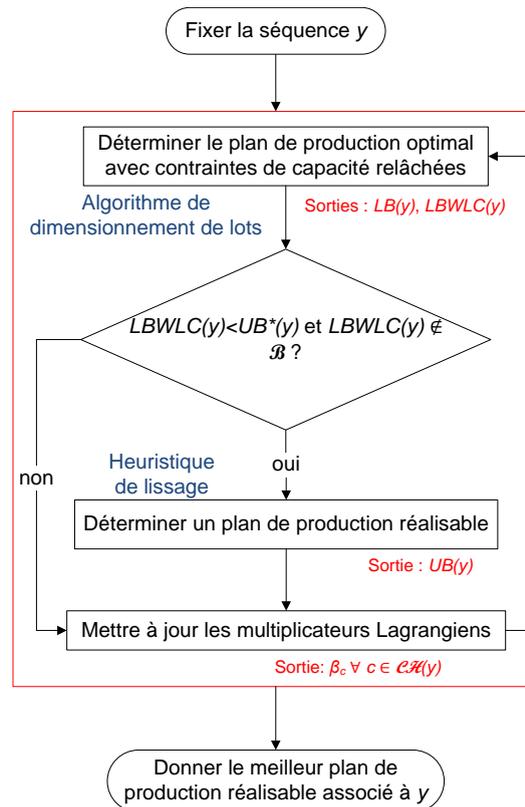


FIGURE 3.9 – Heuristique Lagrangienne modifiée

3.4.2 Amélioration de la séquence

Dans notre approche, la méthode d'amélioration de la séquence est composée principalement de trois étapes : la construction du voisinage, l'exploration du voisinage et la sélection de l'arc à changer. Ces trois parties de l'algorithme diffèrent de la procédure dans l'approche de Wolosewicz *et al.*, schématisée dans la Figure 3.10. Premièrement, dans l'approche de Wolosewicz *et al.*, le voisinage est composé par tous les arcs violés ; tandis que nous définissons un ensemble d'arc critiques pour réduire le nombre de solutions candidates. Deuxièmement, dans l'approche de Wolosewicz *et al.*, il n'y a pas d'exploration de solutions potentielles. Le choix de l'arc à changer est basé sur la plus grande somme de multiplicateurs Lagrangiens (une somme était calculée pour chaque arc). Dans notre approche, illustrée sur la Figure 3.5, nous évaluons

chaque candidat potentiel à travers la relaxation Lagrangienne (sans construction d'une solution réalisable), et nous sélectionnons l'arc avec la plus petite valeur de $LB^*(y)$. Un autre facteur différenciateur est la taille de la liste taboue. Dans l'approche de Wolosewicz *et al.*, la taille de la liste taboue est infinie. Nous avons fixé une taille finie, à partir de différentes expérimentations.

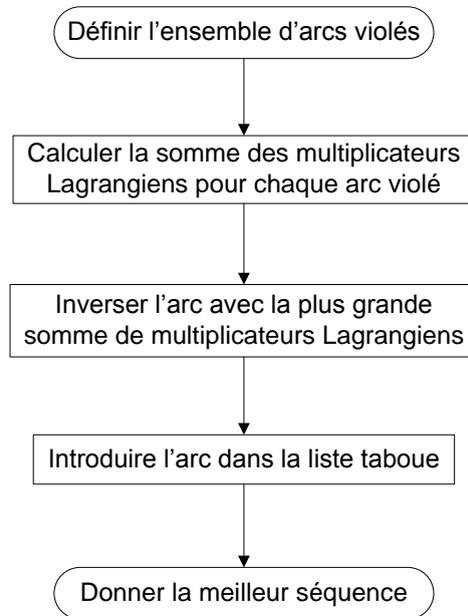


FIGURE 3.10 – Méthode d'amélioration de la séquence dans l'approche de Wolosewicz *et al.*

3.4.2.1 Construction du voisinage

La construction du voisinage des arcs potentiels à inverser comprend deux étapes : l'affectation d'un poids à chaque arc violé et la limitation de l'espace de recherche. Nous avons comparé plusieurs alternatives que nous discutons dans les paragraphes qui suivent.

Affectation du poids aux arcs

Nous avons étudié plusieurs méthodes d'affectation de multiplicateurs Lagrangiens aux arcs faisant partie du voisinage cible. En fait, chaque chemin qui a été violé au moins une fois ($c \in \mathcal{CH}(y)$) a un multiplicateur Lagrangien β_c associé qui vaut zéro quand sa contrainte de capacité est satisfaite, ou qui est supérieur à zéro quand cette dernière est violée. Chacun de ces multiplicateurs est affecté à chaque arc $(o, o') \in c \in \mathcal{CH}(y)$ selon la méthode d'affectation choisie.

– Méthode β_{max}

Cette méthode, qui est utilisée dans l'approche de Wolosewicz *et al.*, consiste à ajouter à chaque arc violé la valeur du multiplicateur Lagrangien de chaque chemin qui le contient. Ainsi, pour chaque arc violé $((o, o') \in \mathcal{VA}(y))$, un poids est calculé avec (3.37).

$$w(o, o')_y = \sum_{c \supset (o, o') \in \mathcal{CH}(y)} \beta_c \quad \forall (o, o') \in \mathcal{CH}(y) \quad (3.37)$$

– Méthode $\beta + 1$

Cette méthode consiste à affecter le multiplicateur Lagrangien du premier chemin contenant l'arc, et ensuite à ajouter le nombre de fois que l'arc violé apparaît dans le parcours d'autres chemins. Le poids est calculé avec (3.38).

$$w(o, o')_y = \beta_{c^1} + \sum_{c \supset (o, o') \in \mathcal{CH}(y); c \neq c^1} 1 \quad \forall (o, o') \in \mathcal{CH}(y) \quad (3.38)$$

De cette façon, les arcs qui appartiennent au plus grand nombre de chemins sont privilégiés. D'autres variantes ont été analysées, mais sans impact significatif sur les résultats. Aucune méthode ne domine complètement les autres, mais $\beta + 1$ conduit globalement vers des meilleurs résultats. C'est pourquoi, cette méthode est utilisée dans notre approche.

En comparant les deux méthodes sur différentes instances d'un problème de grande taille (job-shop avec 20 jobs, 5 opérations par job et 5 machines), l'écart absolu moyen entre les bornes supérieures obtenues avec les deux méthodes est égal à 12.73 et l'écart relatif moyen est égal à 0.02% en faveur de $\beta + 1$. Le nombre moyen de séquences changées est également plus favorable pour $\beta + 1$, ce chiffre augmentant de 7.76%, par rapport à β_{max} . Ce qui est vraiment important, c'est que la combinaison de cette méthode d'affectation de poids avec une nouvelle structure de voisinage conduit à des améliorations très significatives par rapport à l'approche de Wolosewicz *et al.*

Une autre méthode d'affectation, basée sur l'arc du graphe conjonctif avec la plus longue durée parmi les arcs violés, a aussi été testée. Cependant, les résultats ne sont pas compétitifs. L'impact sur l'utilisation de la capacité est important, mais pas toujours dans un bon sens. La plupart du temps, la qualité globale de la borne supérieure est significativement détériorée par rapport aux deux autres méthodes.

Analyse des structures de voisinage

Dans l'approche de Wolosewicz *et al.*, un ensemble d'arc violés $\mathcal{VC}(y)$, contenant tous les arcs (o, o') qui appartiennent à l'ensemble de chemins les plus violés $\mathcal{CH}(y)$ durant la relaxation Lagrangienne, a été défini. Étant donné qu'un arc peut faire partie du parcours de plusieurs chemins, une somme de multiplicateurs Lagrangiens est calculée pour chaque arc $(o, o') \in \mathcal{VC}(y)$, ce qui permet de déterminer les arcs qui potentiellement pourraient avoir un impact significatif sur la violation de la capacité.

L'orientation de l'arc avec la somme de multiplicateurs Lagrangiens la plus grande est inversée et l'arc est introduit dans la liste taboue \mathcal{TL} . Puis, la nouvelle séquence est fixée, le nouveau problème de planification est résolu, et la procédure continue de façon itérative. Un inconvénient de cette procédure est l'absence d'un voisinage permettant d'explorer plusieurs alternatives (solutions candidates). L'avantage est la possibilité de fixer plus de nouvelles séquences, et donc de résoudre plus de problèmes de planification à séquence fixée. Cependant, la méthode converge rapidement vers un optimum local, sans considérer des branches importantes du graphe disjonctif. De plus, même si modifier l'orientation d'un grand nombre d'arcs réduit potentiellement le degré de violation de la capacité, les arcs choisis ne sont pas toujours les meilleurs. Pour cette raison, il est plus intéressant d'avoir un voisinage de mouvements candidats.

Nous avons étudié les types de voisinage suivants :

- **Voisinage complet** : tous les arcs appartenant à tous les chemins de $\mathcal{CH}(y)$ sont considérés. Donc, plus la capacité est faible, plus la taille du voisinage est importante, et par conséquent, l'exploration nécessite plus de temps de calcul. Néanmoins, à chaque itération de la recherche taboue, une meilleure séquence (ou au moins de la même qualité que la séquence précédente) est fixée.
- **Voisinage réduit** : Uniquement les arcs dont le poids est supérieur ou égal à un pourcentage de la valeur du plus grand poids sont considérés. Nous avons testés des facteurs de réduction de 25%, 50%, 75%, 95%, 97% et 100%. L'intérêt de cette structure de voisinage est de pouvoir déterminer l'impact du changement de différents arcs, avec un temps de calcul raisonnable.

Nous avons réalisé des expérimentations pour les instances les plus difficiles (avec coûts de setup importants par rapport aux coûts de stockage). Avec un voisinage complet, une bonne solution convergeant localement, peut être déterminée en un temps de calcul élevé. D'autre part, avec un voisinage réduit, des très bonne solutions sont obtenues en quelques secondes, et plus de branches du graphe disjonctif sont explorées, évitant la convergence locale prématurée. Pour comparer la performance des différents types de voisinage, nous avons défini un ratio

TABLEAU 3.2 – Comparaison de résultats avec différents voisinages

Instance	0% β_{max}		95% β_{max}		97% β_{max}		100% β_{max}		100% $\beta + 1$	
	G_{UB} (%)	R_{QT_i}	G_{UB} (%)	R_{QT_i}	G_{UB} (%)	R_{QT_i}	G_{UB} (%)	R_{QT_i}	G_{UB} (%)	R_{QT_i}
Coût de setup 1	1,61	1,74	1,40	1,96	1,40	1,94	1,27	1,85	0,00	2,00
Coût de setup 2	1,08	1,77	0,33	1,98	0,15	1,95	0,00	1,99	0,06	1,99
Coût de setup 3	2,77	1,86	0,00	1,98	1,92	1,95	0,27	1,99	0,19	1,99
Coût de setup 4	2,80	1,87	0,00	1,95	0,00	1,96	2,53	1,94	0,35	1,96
Coût de setup 5	3,16	1,79	0,15	1,97	2,97	1,91	0,00	1,97	0,38	1,98
Coût de setup 6	6,21	1,67	2,79	1,90	0,00	1,99	2,87	1,91	2,93	1,92
Coût de setup 7	0,17	1,81	3,19	1,93	3,30	1,92	3,08	1,92	0,00	2,00
Moyenne	2,54	1,79	1,12	1,95	1,39	1,94	1,43	1,94	0,56	1,98

UB /temps de calcul. Le Tableau 3.2 présente une comparaison des résultats obtenus avec différents voisinages lorsque l'on résout des instances difficiles (avec grands coûts de setup). Le voisinage complet est appelé « 0% ». L'écart G_{UB} sur UB est calculé par rapport à la meilleure borne supérieure trouvée, et le ratio UB /temps (R_{QT_i}) est déterminé avec (3.39). Le temps d'exécution est fixé à 5 minutes.

$$R_{QT_i} = \frac{1 - \frac{UB_i - \min\{UB_i\}}{\min\{UB_i\}}}{t_i} \times 10 \quad (3.39)$$

Un grand écart sur UB représente une mauvaise solution, et une grande valeur de R_{QT_i} indique que la structure de voisinage proposée conduit à une bonne solution en un temps de calcul petit. La conclusion des expérimentations est que le voisinage avec facteur de réduction de 100% et avec $\beta + 1$ comme méthode d'affectation du poids donne des meilleures solutions que les autres alternatives.

Les résultats montrent qu'il est préférable d'utiliser un voisinage réduit qu'un voisinage complet. Le voisinage à 100% de la valeur du poids maximum s'avère être le meilleur, en prenant comme critère le compromis entre qualité de la borne supérieure et temps de calcul. Un constat important est que la méthode d'affectation du poids $\beta + 1$ donne des résultats significativement meilleurs à ceux de β_{max} lorsque l'on adopte une structure de voisinage réduite et basée sur un critère de sélection plus cohérent, comme LB , $LBWLC$ ou UB .

3.4.2.2 Taille de la liste tabou

Nous avons réalisé plusieurs expérimentations afin de déterminer la taille de la liste tabou la plus convenable. Les résultats sont illustrés sur la Figure 3.11, où « Ratio Qualité moyen » correspond à R_{QT_i} et est calculé avec (3.39). Les tests correspondent à 151 instances obtenues en faisant varier la taille du problème d'ordonnancement, le nombre de périodes et certains paramètres, comme dans la Section 3.5. À part le dernier groupe d'instances, correspondant

à un atelier de type job-shop 20x5 avec 20 périodes, où toutes les tailles aboutissent à une performance identique, les résultats sont très variables. En moyenne, les tailles avec 10 et 15 éléments conduisent à de meilleures solutions.

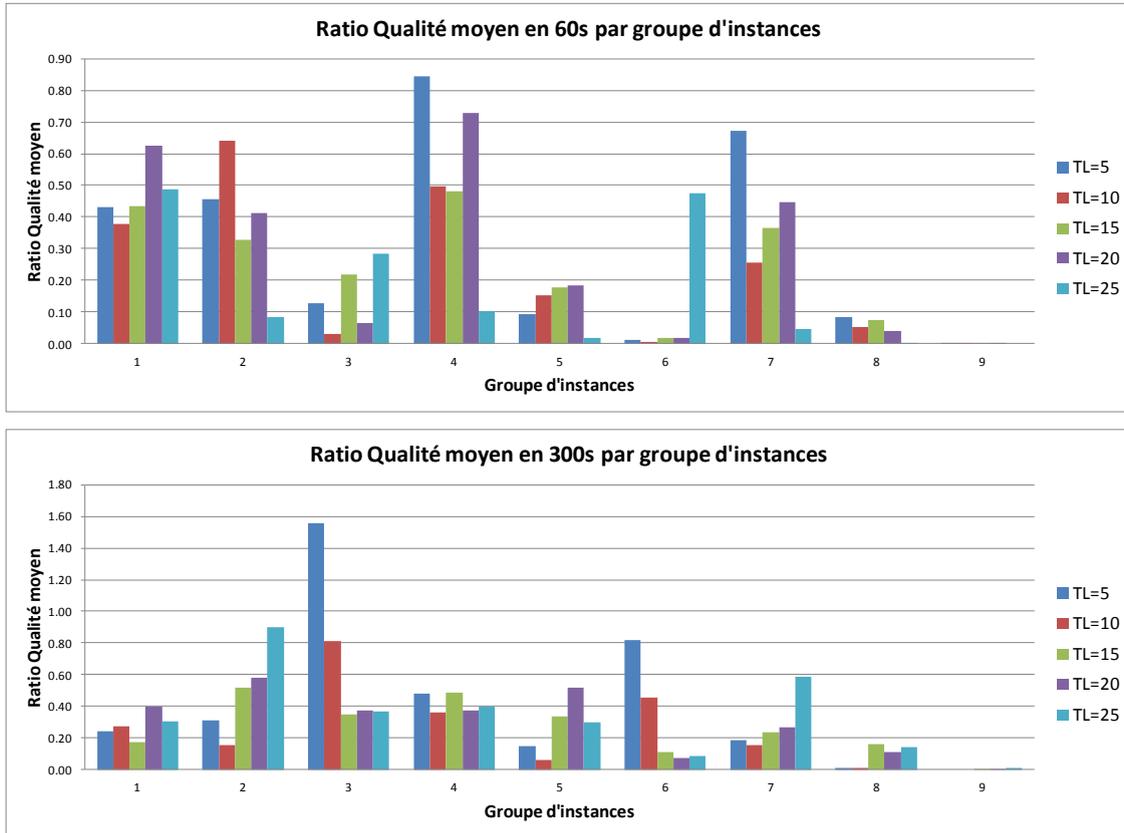


FIGURE 3.11 – Résultats expérimentaux avec plusieurs tailles de liste taboue

3.4.2.3 Sélection de l'arc à changer

Dans l'approche de Wolosewicz *et al.*, la sélection de l'arc à inverser est basée uniquement sur la somme de multiplicateurs Lagrangiens ; une stratégie qui favorise les temps de calcul, mais qui peut conduire à de très mauvaises solutions. C'est pourquoi nous avons analysé d'autres critères de sélection. Comme commenté dans la Section 3.3.2.3, nous avons comparé des stratégies basées sur les valeurs de $LB(y)$, $LBWLC(y)$ et UB . Globalement, la performance de $LBLWLC$ comme critère de sélection n'a pas été satisfaisante. Nous avons par conséquent focalisé l'analyse sur LB et UB .

Comme discuté préalablement, UB guide vers de très bonnes solutions à court terme, mais le temps de calcul de l'heuristique de lissage est très important. Donc, le nombre de séquence changées est limité, ce qui fait que la solution finale du problème intégré n'est pas suffisamment bonne. Sur 151 instances, la méthode guidée par LB a conduit à de meilleures solutions pour 61 instances (39.87%); tandis que UB a permis d'obtenir de meilleurs résultats pour 47 instances (30.72%). L'avantage de LB est aussi par rapport au nombre de séquences changées, qui est en moyenne supérieur de 120.78% en le comparant avec UB .

3.4.2.4 Intensification vs. Diversification

Comme dans beaucoup d'heuristiques, il s'agit d'établir un compromis entre intensification et diversification. L'intensification est utilisée pour améliorer la solution dans un espace de recherche réduit, i.e. on se rapproche d'un optimum local; tandis que la diversification cherche à explorer de nombreuses branches de l'espace de recherche afin de pouvoir potentiellement se rapprocher de l'optimum global. La bonne méthode consiste à équilibrer intensification et diversification. Néanmoins, il n'est pas toujours simple de déterminer le meilleur compromis possible.

Dans notre approche intégrée, nous pouvons identifier différents facteurs de diversification et intensification, comme indiqué dans le Tableau 3.3.

Algorithme/Méthode	Diversification	Intensification
Ordonnancement	Faire plusieurs modifications du graphe disjonctif en même temps	Changer un seul arc par itération
Voisinage	Sous-ensemble des arcs violés	Ensemble complet des arcs violés
Heuristique de lissage	Sans amélioration du coût	Avec amélioration
Critère de sélection du voisin	Borne inférieure	Borne supérieure

TABLEAU 3.3 – Compromis entre diversification et intensification

Dans notre cas, la diversification fait référence à la fixation d'un grand nombre de séquences différentes, et l'intensification correspond à l'exploration détaillée de chaque séquence. Comme le temps de calcul est limité, une grande intensification réduit la diversification, i.e. le nombre de séquence fixées diminue. Néanmoins, une petite intensification (pour augmenter la diversification), peut ne pas être suffisante pour trouver une solution adéquate à chaque problème à séquence fixée. Ainsi par exemple, si nous explorons à chaque itération de la recherche taboue l'ensemble complet d'arcs violés, nous pourrions garantir la meilleure solution pour chaque problème à séquence fixée résolu. Cependant, le temps alloué au changement de séquences sera

fortement réduit, et la solution du problème intégré ne sera sûrement pas intéressante. De la même manière, une heuristique de lissage avec amélioration de UB une fois qu'une solution réalisable est trouvée, donnera au moins un plan de production aussi bon que celui trouvé par l'heuristique sans amélioration. Néanmoins, il est possible que le temps passé à faire des modifications supplémentaires du plan consomme beaucoup de temps (notamment pour des problèmes de grande taille avec un grand nombre de périodes), et que moins de séquences soient changées, ce qui peut conduire à une mauvaise solution du problème intégré.

De manière générale, les expérimentations ont montré qu'il est plus intéressant de diversifier la recherche que de l'intensifier. Cependant, les périodes d'intensification doivent être correctement guidées, afin que la diversification soit efficace. Rien ne sert de résoudre beaucoup de problèmes à séquence fixée si les séquences choisies ne sont pas bonnes ou ne permettent pas de faire évoluer le graphe dans le bon sens.

Des expérimentations pour un atelier de type job-shop avec 20 produits, 5 opérations par produits et 5 machines, avec 5, 10 et 20 périodes, ont montré par exemple, qu'en moyenne et sur une limite de temps de calcul de 60 secondes, l'heuristique de lissage réduisait de 623% le nombre de séquences changées, lorsque l'on utilise la condition 5- $LBWLC$. C'est une raison de plus pour éviter d'appliquer l'heuristique de lissage à chaque itération de l'heuristique Lagrangienne, et pour privilégier LB au lieu de UB comme critère de sélection de l'arc à changer. Néanmoins, dans ce dernier cas, il faut s'assurer que la borne inférieure a été correctement calculée, c'est-à-dire que le nombre d'itérations de la relaxation Lagrangienne et les autres paramètres assurent une convergence correcte de LB .

Une opportunité d'amélioration de notre approche actuelle est sur la génération de séquences. Le changement de la séquence par la simple inversion d'un arc étant une procédure intensive, il peut être plus avantageux de réaliser plusieurs changements d'arcs en même temps, ou que la nouvelle séquence ne soit pas directement dérivée de la séquence précédente.

3.5 Résultats expérimentaux

Dans cette section, nous comparons notre approche avec l'approche de Wolosewicz *et al.* [233, 234, 235] et avec le solveur standard IBM ILOG CPLEX Optimization Studio 12.3. L'objectif est de mesurer la performance de ces trois alternatives, en analysant la relation entre qualité de la solution et temps de calcul. Les deux versions de l'approche intégrée ont été implémentées sur Microsoft Visual C++ 2010 Express, et toutes les expérimentations ont été réalisées avec un ordinateur présentant les caractéristiques suivantes : système d'exploitation de 64 bits (Microsoft Windows 7), processeur Intel core i7 4 Quad @2.0GHz et 8GB de mémoire RAM.

Les problèmes étudiés correspondent aux configurations de type job-shop décrites dans le Tableau 3.4. Nous considérons des horizons de planification de 5, 10 et 20 périodes, pour représenter des problèmes de petite, moyenne et grande taille. Les données ont été générées aléatoirement, et 153 instances ont été créées en faisant varier : la capacité, les coûts de setup et les demandes. Le Tableau 3.5 présente les paramètres généraux (par défaut) utilisés pour générer les instances. Les paramètres à faire varier ont été choisis pour leur importance dans le problème. Ainsi, comme discuté dans les Chapitres 1 et 2, la capacité est un facteur très important en planification de la chaîne logistique. La plupart du temps, la capacité n'est pas gérée correctement dans les systèmes ERP, et par conséquent les plans de production tactiques ne sont pas réalisables au niveau opérationnel. C'est pourquoi, nous considérons des instances avec ratios de capacité petits. Un autre facteur important est le rapport entre les coûts de stockage et les coûts de setup. Des coûts de stockage élevés entraînent le lancement de la production plus souvent pour éviter des grands niveaux d'inventaires ; tandis que des coûts de setup élevés génèrent l'effet inverse. Nous étudions des instances avec des coûts de setup élevés par rapport aux coûts de stockage. Dans ce cas, trouver une solution réalisable est déjà difficile, car des grandes tailles de lot (pour éviter des setups) sont restreintes par une petite capacité. Finalement, faire varier les demandes permet de considérer différents niveaux de capacité, et par conséquent différentes décisions de setup faisant varier les niveaux de stocks.

TABLEAU 3.4 – Problèmes étudiés

Job-Shop	Description
6x6	6 produits, 6 opérations par produit, 6 ressources
10x10	10 produits, 10 opérations par produit, 10 ressources
20x5	20 produits, 5 opérations par produit, 5 ressources

TABLEAU 3.5 – Paramètres généraux pour la génération d'instances

Job-Shop	T	cap	c_i^s	$D_{i,l}$	L_i	c_i^p	c_i^{inv}
6x6	5	0,46	30	[3, 8]	3	4	1
	10	0,46	30	[3, 8]	3	4	1
	20	0,46	30	[2, 5]	3	4	1
10x10	5	0,44	30	[3, 8]	3	4	1
	10	0,50	30	[3, 8]	3	4	1
	20	0,58	20	[2, 5]	3	4	1
20x5	5	0,44	30	[3, 8]	3	4	1
	10	0,48	30	[3, 8]	3	4	1
	20	0,54	25	[5, 10]	3	4	1

Le paramètre cap correspond à un ratio de capacité, qui est utilisé pour modifier une capacité de base, permettant de fabriquer tous les produits à la période où ils sont demandés, en utilisant une seule ressource. Ce ratio permet donc de mesurer l'impact, en termes de difficulté, sur le calcul d'une borne supérieure quand on augmente ou diminue la capacité de chaque période. Ainsi, la capacité c_l du système à la période l est proportionnelle à $0 < cap < 1$, et est calculée avec (3.40).

$$c_l = cap \sum_{o \in l} (p_o^u X_{i(o),l(o)} D_{i(o),l(o)} + s_o^t Y_{i(o),l(o)}) \quad \forall l \quad (3.40)$$

Dit autrement, c_l est le temps total requis pour satisfaire la demande totale de la période l , en utilisant uniquement une ressource, multiplié par un facteur de réduction.

Les coûts sont exprimés en unités monétaires, les délais d'obtention sont donnés en périodes et les demandes varient entre les différentes périodes, selon l'intervalle considéré (par exemple, en considérant les paramètres généraux, dans le job-shop 6x6 avec $T = 5$, les demandes varient entre 3 et 5 unités). Pour chaque groupe d'instances, nous fixons les paramètres généraux (Tableau 3.5) et nous faisons varier, soit les coûts de setup, soit les demandes, soit le ratio de capacité par période.

Dans ce qui suit, $UB1$, $UB2$ et CPL correspondent aux solutions déterminées avec notre approche, l'approche de Wolosewicz *et al.* et le solveur IBM ILOG CPLEX, respectivement. G_1 est l'écart entre $UB1$ et $UB2$, G_2 est l'écart entre $UB1$ et CPL et G_3 est l'écart entre $UB2$ et CPL . Les écarts sont calculés avec les équations (3.41), (3.42) et (3.43). PV est la valeur du paramètre qui varie et LBA est la valeur de la borne inférieure absolue (donnée à titre indicatif). Les coûts de setup sont particulièrement élevés par rapport à la valeur par défaut, afin d'analyser les instances les plus difficiles, c'est-à-dire les cas où le ratio entre les coûts de setup et les coûts de stockage est élevé. Les expérimentations ont été réalisées avec des limites de temps de calcul de 60 secondes et 300 secondes.

$$G_1 = \frac{UB2 - UB1}{UB1} \times 100 \quad (3.41)$$

$$G_2 = \frac{UB1 - CPL}{CPL} \times 100 \quad (3.42)$$

$$G_3 = \frac{UB2 - CPL}{CPL} \times 100 \quad (3.43)$$

3.5.1 Expérimentations avec un atelier de type job-shop 6x6

Les solutions obtenues avec les deux approches et le solveur standard sont présentées dans le Tableau 3.6. Les instances sont classées par nombre de périodes et par paramètre qui varie. Les résultats illustrés correspondent aux bornes supérieures et aux écarts entre les solutions des procédures, pour des limites de temps de calcul de 60 secondes et 300 secondes. Nous comparons, dans ce qui suit, la performance de notre approche avec celle du solveur standard et avec celle de l'approche de Wolosewicz *et al.*. Les écarts entre les solutions de l'approche de Wolosewicz *et al.* et celles du solveur sont présentés à titre indicatif.

3.5.1.1 Comparaison entre notre approche et le solveur IBM ILOG CPLEX

Avec 5 périodes de planification et en 60 secondes de temps de calcul, IBM ILOG CPLEX est capable de résoudre de façon optimale 7 des 17 instances. D'autre part, notre approche trouve la solution optimale pour une instance (*Défaut*), et les écarts correspondants aux autres résultats sont globalement favorables au solveur standard, mais ils ne sont pas significatifs (l'écart moyen est égal à 0.38%). De plus, sur 4 instances, notre approche donne de meilleures solutions. Avec un temps de calcul élargi à 300 secondes, le solveur arrive à trouver la solution optimale de 13 instances. Nos solutions restent encore proches, avec un écart maximum de 5.39% (instance *Demande 1*).

En augmentant le nombre de périodes de 5 à 10, nous observons qu'il devient plus difficile pour le solveur de résoudre les différentes instances, avec une limite de temps de 60 secondes. En effet, il n'est pas capable de trouver des solutions réalisables pour 9 instances (53%), et hormis l'instance *Capacité 2*, toutes les solutions obtenues avec notre approche sont meilleures que celles fournies par IBM ILOG CPLEX. Pour l'instance *Demande 4*, notre approche trouve la solution optimale. Avec un temps de calcul de 300 secondes, le solveur arrive à trouver une solution réalisable pour toutes les instances avec 6 solutions optimales. Dans ce cas, des écarts importants (entre 6.93% et 14.79%) avec notre approche, en faveur du solveur, sont constatés pour les instances avec grands coûts de setup. Cependant, pour l'instance *Coût de setup 7*, un grand écart (6.71%) est en faveur de notre approche. Pour les autres instances, les écarts sont petits (inférieurs à 1%), avec 3 solutions en faveur de notre approche.

La difficulté de résolution augmente encore lorsque le nombre de périodes est égal à 20. Comme pour le cas avec 10 périodes, quand le temps de calcul est limité à 60 secondes, IBM ILOG CPLEX n'est pas capable de trouver des solutions réalisables pour 9 instances (majoritairement celles avec coûts de setup élevés). De plus, les écarts entre les solutions du solveur et celles de notre approche, sont largement favorables à notre méthode, avec des valeurs entre

TABLEAU 3.6 – Résultats pour le job-shop 6x6

T	Instance	PV	LBA	Résultats après 60 secondes						Résultats après 300 secondes					
				UB1	UB2	CPL	G ₁	G ₂	G ₃	UB1	UB2	CPL	G ₁	G ₂	G ₃
5	Défaut	Connu	1190	1201*	1201*	1201*	0.00	0.00	0.00	1201*	1201*	1201*	0.00	0.00	0.00
	Capacité 1	0.42	1190	1233	1271	1261	3.08	-2.22	0.79	1233	1271	1207	3.08	2.15	5.30
	Capacité 2	0.45	1190	1217	1225	1224	0.66	-0.57	0.08	1207	1225	1204	1.49	0.25	1.74
	Capacité 3	0.48	1190	1201	1221	1198*	1.67	0.25	1.92	1200	1221	1198*	1.75	0.17	1.92
	Capacité 4	0.50	1190	1198	1215	1195*	1.42	0.25	1.67	1198	1215	1195*	1.42	0.25	1.67
	Coût de setup 1	80	1529	1807	1880	1804	4.04	0.17	4.21	1807	1880	1801*	4.04	0.33	4.39
	Coût de setup 2	100	1649	2047	2168	2044	5.91	0.15	6.07	2047	2168	2041*	5.91	0.29	6.22
	Coût de setup 3	120	1769	2287	2309	2281	0.96	0.26	1.23	2287	2309	2281*	0.96	0.26	1.23
	Coût de setup 4	140	1889	2527	2678	2527	5.98	0.00	5.98	2527	2678	2521*	5.98	0.24	6.23
	Coût de setup 5	180	2129	3032	3343	3003	10.26	0.97	11.32	3004	3343	3001*	11.28	0.10	11.40
	Coût de setup 6	200	2249	3257	3447	3258	5.83	-0.03	5.80	3247	3447	3258	6.16	-0.34	5.80
	Coût de setup 7	220	2369	3487	3526	3484	1.12	0.09	1.21	3487	3526	3484	1.12	0.09	1.21
	Demande 1	[1, 5]	696	803	809	761*	0.75	5.52	6.31	802	809	761*	0.87	5.39	6.31
Demande 2	[1, 10]	906	972	1015	956*	4.42	1.67	6.17	972	1015	956*	4.42	1.67	6.17	
Demande 3	[4, 8]	1202	1258	1346	1220*	7.00	3.11	10.33	1258	1346	1220*	7.00	3.11	10.33	
Demande 4	[10, 15]	2233	2247	2255	2324	0.36	-3.31	-2.97	2247	2255	2242*	0.36	0.22	0.58	
Demande 5	[5, 15]	1760	1787	1798	1785*	0.62	0.11	0.73	1787	1798	1785*	0.62	0.11	0.73	
10	Défaut	Connu	2347	2410	2443	-	1.37	-∞	-∞	2373	2443	2378	2.95	-0.21	2.73
	Capacité 1	0.47	2347	2420	2482	-	2.56	-∞	-∞	2388	2482	2366*	3.94	0.93	4.90
	Capacité 2	0.48	2347	2400	2422	2375	0.92	1.05	1.98	2373	2422	2364*	2.06	0.38	2.45
	Capacité 3	0.53	2347	2368	2470	2692	4.31	-12.04	-8.25	2363	2470	2359*	4.53	0.17	4.71
	Capacité 4	0.54	2347	2366	2370	2427	0.17	-2.51	-2.35	2363	2370	2364	0.30	-0.04	0.25
	Coût de setup 1	80	3038	3579	3825	-	6.87	-∞	-∞	3563	3825	3332	7.35	6.93	14.80
	Coût de setup 2	100	3278	4075	4231	-	3.83	-∞	-∞	4001	4231	3620	5.75	10.52	16.88
	Coût de setup 3	120	3518	4694	4542	4786	-3.24	-1.92	-5.10	4375	4542	3996	3.82	9.48	13.66
	Coût de setup 4	140	3730	5009	5089	-	1.60	-∞	-∞	4829	5072	4488	5.03	7.60	13.01
	Coût de setup 5	180	4005	6098	5962	-	-2.23	-∞	-∞	5673	5962	4942	5.09	14.79	20.64
	Coût de setup 6	200	4125	6416	6461	-	0.70	-∞	-∞	6127	6417	5420	4.73	13.04	18.39
	Coût de setup 7	220	4245	6101	6349	-	4.06	-∞	-∞	6101	6349	6540	4.06	-6.71	-2.92
	Demande 1	[1, 5]	1426	1472	1526	-	3.67	-∞	-∞	1472	1526	1466*	3.67	0.41	4.09
Demande 2	[1, 10]	2331	2352	2376	2398	1.02	-1.92	-0.92	2352	2376	2342*	1.02	0.43	1.45	
Demande 3	[4, 8]	2403	2435	2439	3222	0.16	-24.43	-24.30	2429	2439	2426	0.41	0.12	0.54	
Demande 4	[10, 15]	4227	4227*	4269	4257	0.99	-0.70	0.28	4227*	4269	4227*	0.99	0.00	0.99	
Demande 5	[6, 15]	3591	3608	3622	3619	0.39	-0.30	0.08	3608	3622	3610	0.39	-0.06	0.33	
20	Défaut	Connu	3136	3357	3430	-	2.17	-∞	-∞	3302	3430	-	3.88	-∞	-∞
	Capacité 1	0.50	3136	3333	3351	5183	0.54	-35.69	-35.35	3299	3351	3269	1.58	0.92	2.51
	Capacité 2	0.54	3136	3247	3305	5299	1.79	-38.72	-37.63	3203	3305	3334	3.18	-3.93	0.87
	Capacité 3	0.59	3136	3204	3196	5203	-0.25	-38.42	-38.57	3196	3196	4984	0.00	-35.87	-35.87
	Capacité 4	0.61	3136	3224	3199	5248	-0.78	-38.57	-39.04	3193	3199	3276	0.19	-2.53	-2.35
	Coût de setup 1	40	3385	3539	3590	4246	1.44	-16.65	-15.45	3521	3590	3645	1.96	-3.40	-1.51
	Coût de setup 2	50	3624	3832	3851	7075	0.50	-45.84	-45.57	3810	3851	3864	1.08	-1.40	-0.34
	Coût de setup 3	65	3924	4316	4413	-	2.25	-∞	-∞	4306	4384	5702	1.81	-24.48	-23.11
	Coût de setup 4	80	4213	4759	4818	-	1.24	-∞	-∞	4697	4818	4765	2.58	-1.43	1.11
	Coût de setup 5	90	4393	4988	4912	-	-1.52	-∞	-∞	4988	4912	4954	-1.52	0.69	-0.85
	Coût de setup 6	100	4564	5319	5361	-	0.79	-∞	-∞	5284	5217	5243	-1.27	0.78	-0.50
	Coût de setup 7	120	4875	5812	5929	-	2.01	-∞	-∞	5812	5929	5622	2.01	3.38	5.46
	Demande 1	[1, 5]	2734	2833	2780	-	-1.87	-∞	-∞	2818	2780	2800	-1.35	0.64	-0.71
Demande 2	[1, 10]	4048	4097	4096	-	-0.02	-∞	-∞	4097	4096	4110	-0.02	-0.32	-0.34	
Demande 3	[4, 8]	4946	4982	4965	7110	-0.34	-29.93	-30.17	4982	4965	5012	-0.34	-0.60	-0.94	
Demande 4	[6, 15]	8601	-	-	-	-	-	-	-	-	-	-	-	-	
Demande 5	[5, 15]	7231	7254	7265	9015	0.15	-19.53	-19.41	7241	7265	7249	0.33	-0.11	0.22	

- Pas de solutions réalisable.

* Solution optimale.

-∞ Pas de solution réalisable avec IBM ILOG CPLEX (écart indéterminé).

TABLEAU 3.7 – Écarts relatifs pour le job-shop 6x6 avec 60 secondes de temps de calcul

T	$G_1(\%)$			$G_2(\%)$			$G_3(\%)$		
	Moy	Min	Max	Moy	Min	Max	Moy	Min	Max
5	3.18	0.00	10.26	0.38	-3.31	11.32	3.58	-2.97	11.32
10	1.60	-3.24	6.87	-5.35	-24.43	1.98	-4.82	-24.30	1.98
20	0.51	-1.87	2.25	-32.92	-45.84	-15.45	-32.65	-45.57	-15.45

16.65% et 45.84%. Avec un temps de calcul de 300 secondes, le solveur arrive à trouver des solutions réalisables pour presque toutes les instances (à l'exception des instances *Défaut* et *Demande 5*), et les écarts diminuent considérablement. Néanmoins, la plupart des instances sont mieux résolues avec notre approche.

3.5.1.2 Comparaison entre l'approche de Wolosewicz *et al.* et notre approche

Avec un temps de calcul limité à 60 secondes, pour les instances avec $T = 5$, notre approche améliore toutes les solutions de l'approche de Wolosewicz *et al.*, sauf pour la première instance, où les deux méthodes trouvent la solution optimale. Avec $T = 10$, uniquement deux instances (11.76%) sont mieux résolues avec l'approche de Wolosewicz *et al.*. Les 15 restantes (88.24%) sont dominées par notre approche. Finalement, avec $T = 20$, 6 instances (35.29%) sont dominées par l'approche de Wolosewicz *et al.*, 10 (58.82%) par notre approche et une instance n'est pas résolue. Les écarts relatifs moyens, minimum et maximum sont présentés dans le Tableau 3.7. Au total, l'approche de Wolosewicz *et al.* est plus favorable sur 8 instances (15.69%); tandis que notre approche donne des meilleures solutions sur 41 instances (80.39%).

Si nous incrémentons le temps de calcul à 300 secondes, pour les instances avec $T = 5$, notre approche trouve à nouveau des meilleures solutions que l'approche de Wolosewicz *et al.* pour toutes les instances, sauf pour la première, où une solution optimale est trouvée avec les 2 procédures. Avec $T = 10$, notre approche domine sur toutes les instances, et avec $T = 20$, l'approche de Wolosewicz *et al.* gagne sur 6 instances (35.29%), la notre sur 10 (58.82%), il y a une parité sur une instance et une instance n'est résolue avec aucune des deux approches, ni par IBM ILOG CPLEX. Les écarts relatifs des expérimentations à 300 secondes sont présentés dans le Tableau 3.8. Au total, l'approche de Wolosewicz *et al.* donne des meilleurs résultats sur 6 instances (11.76%); tandis que notre approche domine sur 43 instances (84.31%).

3.5.2 Expérimentations avec un atelier de type job-shop 10x10

Dans la sous-section précédente, nous avons étudié des instances associées à des problèmes de taille relativement petite (job-shop 6x6). Dans ce qui suit, nous nous intéressons à des

TABLEAU 3.8 – Écarts relatifs pour le job-shop 6x6 avec 300 secondes de temps de calcul

T	$G_1(\%)$			$G_2(\%)$			$G_3(\%)$		
	Moy	Min	Max	Moy	Min	Max	Moy	Min	Max
5	3.32	0.00	11.28	0.84	-0.34	5.39	4.19	0.00	11.40
10	3.30	0.30	7.35	3.40	-6.71	14.79	6.88	-2.92	20.64
20	0.88	-1.52	3.88	-4.51	-35.87	3.38	-3.87	-35.87	5.46

problèmes de moyenne taille, et nous utilisons une configuration 10x10, comme expliqué par le Tableau 3.4. Les résultats des expérimentations sont affichés dans le Tableau 3.9. À nouveau, 51 instances sont étudiées ; elles sont classées par nombre de périodes de planification (5, 10 et 20) et par type de paramètre varié (capacité, coût de setup et demande).

3.5.2.1 Comparaison entre notre approche et le solveur IBM ILOG CPLEX

Résoudre le job-shop 10x10 avec 5, 10 et 20 périodes pose plus de problèmes pour IBM ILOG CPLEX. En effet, le solveur n'est pas capable de trouver des solutions réalisables, ni en 60 secondes ni en 300 secondes. Des expérimentations additionnelles avec un temps de calcul de 600 secondes ont montré le même comportement. Une explication est la grande sollicitation en termes de mémoire. À l'inverse, l'approche intégrée arrive à résoudre toutes les instances, exceptée l'instance *Demande 2* avec $T = 10$.

Ces résultats mettent en évidence la complexité de résolution du problème avec des méthodes exactes, même en utilisant un solveur commercial et une grande capacité de calcul. Notre approche est donc appropriée pour trouver des solutions réalisables pour des problèmes de moyenne taille, en un temps de calcul raisonnable.

3.5.2.2 Comparaison entre l'approche de Wolosewicz *et al.* et notre approche

Avec un temps de calcul limité à 60 secondes, notre approche domine toutes les instances à 5 périodes. Avec 10 périodes, l'approche de Wolosewicz *et al.* est meilleur pour 5 instances (29.41%), la notre pour 11 (64.71%), et une instance n'est pas résolue. Avec 20 périodes, 6 instances (35.29%) sont mieux résolues avec l'approche de Wolosewicz *et al.*, 10 avec la notre (58.82%), et il y a une parité sur une instance. Au total, l'approche de Wolosewicz *et al.* est plus performante sur 11 instances (21.57%) et notre approche domine sur 38 instances (74.51%).

En augmentant le temps de calcul à 300 secondes, toutes les instances avec $T = 5$ sont à nouveau mieux résolues avec notre approche. Avec $T = 10$, 3 instances (17.65%) sont mieux résolues avec l'approche de Wolosewicz *et al.*, 13 avec la notre (76.47%), et une instance reste sans solution. Finalement, avec $T = 20$, l'approche de Wolosewicz *et al.* domine sur 5 instances

TABLEAU 3.9 – Résultats pour le job-shop 10x10

T	Instance	PV	LBA	Résultats après 60 secondes						Résultats après 300 secondes					
				UB1	UB2	CPL	G ₁	G ₂	G ₃	UB1	UB2	CPL	G ₁	G ₂	G ₃
5	Défaut	Connu	1961	1980	1999	-	0.96	-∞	-∞	1980	1999	-	0.96	-∞	-∞
	Capacité 1	0.42	1961	1986	1999	-	0.65	-∞	-∞	1984	1999	-	0.76	-∞	-∞
	Capacité 2	0.50	1961	1968	1986	-	0.91	-∞	-∞	1968	1986	-	0.91	-∞	-∞
	Capacité 3	0.46	1961	1977	1985	-	0.40	-∞	-∞	1974	1985	-	0.56	-∞	-∞
	Capacité 4	0.48	1961	1976	2002	-	1.32	-∞	-∞	1974	2002	-	1.42	-∞	-∞
	Coût de setup 1	80	2525	2924	3021	-	3.32	-∞	-∞	2913	3021	-	3.71	-∞	-∞
	Coût de setup 2	100	2725	3341	3621	-	8.38	-∞	-∞	3341	3621	-	8.38	-∞	-∞
	Coût de setup 3	120	2925	3642	3823	-	4.97	-∞	-∞	3637	3823	-	5.11	-∞	-∞
	Coût de setup 4	140	3125	4090	4228	-	3.37	-∞	-∞	4090	4228	-	3.37	-∞	-∞
	Coût de setup 5	180	3525	4857	5032	-	3.60	-∞	-∞	4843	5032	-	3.90	-∞	-∞
	Coût de setup 6	200	3725	5245	5467	-	4.23	-∞	-∞	5105	5467	-	7.09	-∞	-∞
	Coût de setup 7	220	3925	5614	5838	-	3.99	-∞	-∞	5614	5838	-	3.99	-∞	-∞
	Demande 1	[1, 5]	1194	1310	1313	-	0.23	-∞	-∞	1308	1313	-	0.38	-∞	-∞
	Demande 2	[1, 5]	1623	1654	1656	-	0.12	-∞	-∞	1654	1656	-	0.12	-∞	-∞
	Demande 3	[2, 4]	1216	1320	1355	-	2.65	-∞	-∞	1320	1355	-	2.65	-∞	-∞
Demande 4	[1, 2]	750	942	975	-	3.50	-∞	-∞	918	975	-	6.21	-∞	-∞	
Demande 5	[5, 15]	3154	3171	3179	-	0.25	-∞	-∞	3169	3179	-	0.32	-∞	-∞	
10	Défaut	Connu	3867	3909	3939	-	0.77	-∞	-∞	3898	3939	-	1.05	-∞	-∞
	Capacité 1	0.42	3867	3998	3954	-	-1.10	-∞	-∞	3951	3954	-	0.08	-∞	-∞
	Capacité 2	0.44	3867	3944	3939	-	-0.13	-∞	-∞	3937	3939	-	0.05	-∞	-∞
	Capacité 3	0.46	3867	3929	3937	-	-0.23	-∞	-∞	3920	3928	-	0.20	-∞	-∞
	Capacité 4	0.48	3867	3911	3924	-	0.33	-∞	-∞	3911	3924	-	0.33	-∞	-∞
	Coût de setup 1	80	5035	5518	5597	-	1.43	-∞	-∞	5452	5397	-	-1.01	-∞	-∞
	Coût de setup 2	110	5635	6490	6525	-	0.54	-∞	-∞	6376	6454	-	1.22	-∞	-∞
	Coût de setup 3	130	6033	7047	7199	-	2.16	-∞	-∞	6995	7108	-	1.62	-∞	-∞
	Coût de setup 4	150	6353	7733	7628	-	-1.36	-∞	-∞	7661	7628	-	-0.43	-∞	-∞
	Coût de setup 5	180	6685	8547	9092	-	6.38	-∞	-∞	8396	9092	-	8.29	-∞	-∞
	Coût de setup 6	200	6885	9196	9395	-	2.16	-∞	-∞	8999	9395	-	4.40	-∞	-∞
	Coût de setup 7	220	7085	9759	10002	-	2.49	-∞	-∞	9590	9831	-	2.51	-∞	-∞
	Demande 1	[1, 5]	2215	2397	2420	-	0.96	-∞	-∞	2367	2420	-	2.24	-∞	-∞
	Demande 2	[1, 6]	1982	-	-	-	-	-	-	-	-	-	-	-	-
	Demande 3	[1, 5]	1482	1653	1693	-	2.42	-∞	-∞	1640	1646	-	0.37	-∞	-∞
Demande 4	[1, 3]	1285	1322	1309	-	-0.98	-∞	-∞	1316	1309	-	-0.53	-∞	-∞	
Demande 5	[5, 15]	5764	5780	5789	-	0.16	-∞	-∞	5780	5789	-	0.16	-∞	-∞	
20	Défaut	Connu	4769	4788	4788	-	0.00	-∞	-∞	4782	4788	-	0.13	-∞	-∞
	Capacité 1	0.45	4769	4856	4845	-	-0.23	-∞	-∞	4848	4823	-	-0.52	-∞	-∞
	Capacité 2	0.49	4769	4824	4843	-	0.39	-∞	-∞	4819	4843	-	0.50	-∞	-∞
	Capacité 3	0.50	4769	4819	4816	-	-0.06	-∞	-∞	4813	4816	-	0.06	-∞	-∞
	Capacité 4	0.52	4769	4818	4799	-	-0.39	-∞	-∞	4816	4799	-	-0.35	-∞	-∞
	Coût de setup 1	40	5701	5893	5902	-	0.15	-∞	-∞	5881	5876	-	-0.09	-∞	-∞
	Coût de setup 2	55	6293	6621	6644	-	0.35	-∞	-∞	6611	6523	-	-1.33	-∞	-∞
	Coût de setup 3	80	7102	7690	7641	-	-0.64	-∞	-∞	7690	7641	-	-0.64	-∞	-∞
	Coût de setup 4	90	7402	8188	8227	-	0.48	-∞	-∞	8160	8227	-	0.82	-∞	-∞
	Coût de setup 5	100	7693	8715	8693	-	-0.25	-∞	-∞	8618	8692	-	0.86	-∞	-∞
	Coût de setup 6	120	8213	9558	9740	-	1.90	-∞	-∞	9558	9740	-	1.90	-∞	-∞
	Coût de setup 7	125	8333	9875	10004	-	1.31	-∞	-∞	9793	9800	-	0.07	-∞	-∞
	Demande 1	[1, 2]	2379	2419	2416	-	-0.12	-∞	-∞	2413	2416	-	0.12	-∞	-∞
	Demande 2	[0, 9]	5489	5562	5577	-	0.27	-∞	-∞	5557	5577	-	0.36	-∞	-∞
	Demande 3	[1, 5]	3535	3616	3650	-	0.94	-∞	-∞	3601	3629	-	0.78	-∞	-∞
Demande 4	[0, 9]	5762	5825	5897	-	0.77	-∞	-∞	5809	5887	-	1.34	-∞	-∞	
Demande 5	[0, 9]	7047	7163	7187	-	0.34	-∞	-∞	7123	7187	-	0.90	-∞	-∞	

- Pas de solution réalisable.

* Solution optimale.

-∞ Pas de solution réalisable avec IBM ILOG CPLEX (écart indéterminé).

TABLEAU 3.10 – Écarts relatifs pour le job-shop 10x10

T	$G_1(\%)$ après 60 secondes			$G_1(\%)$ après 300 secondes		
	Moy	Min	Max	Moy	Min	Max
5	2,52	0.12	8.38	2.93	0.12	8.38
10	1.00	-1.36	6.38	1.28	-1.01	8.29
20	0.31	-0.64	1.90	0.29	-1.33	1.90

(29.41%) et la notre sur les 12 instances restantes (70.59%). Les écarts relatifs sont présentés dans le Tableau 3.10. Au total, l'approche de Wolosewicz *et al.* est plus efficace sur 8 instances (15.69%); tandis que la notre domine sur 42 instances (82.35%).

3.5.3 Expérimentations avec un atelier de type job-shop 20x5

Nous nous intéressons finalement à la résolution de problèmes de grande taille, correspondant à un atelier de type job-shop avec 20 produits, 5 opérations par produit et 5 ressources (configuration 20x5). Des instances considérant des variations sur les paramètres et les horizons de planification précédemment étudiés, sont également analysées dans ce cas. Les résultats sont présentés dans le Tableau 3.11.

3.5.3.1 Comparaison entre notre approche et le solveur IBM ILOG CPLEX

De manière similaire à ce que nous avons constaté pour le job-shop 10x10, résoudre le job-shop 20x5 suppose un effort de calcul très important. IBM ILOG CPLEX est en effet incapable de déterminer des solutions réalisables pour toutes les instances. Les résultats sont corroborés avec une limite de temps de calcul de 600 secondes. Avec notre approche, des plans de production réalisables sont trouvés pour toutes les instances. Une fois de plus, les limitations des solveurs commerciaux et la pertinence de notre approche sont mises en évidence. L'intégration des décisions de planification et d'ordonnancement de la production est très importante en management de la chaîne logistique, mais le problème issu d'une modélisation intégrée est très complexe à résoudre et nécessite la conception de méthodes dédiées, comme l'approche que nous proposons.

3.5.3.2 Comparaison entre l'approche de Wolosewicz *et al.* et notre approche

Avec les deux limites de temps de calcul (60 secondes et 300 secondes) notre approche domine celle de Wolosewicz *et al.* sur toutes les instances, avec toutes les longueurs d'horizon de

TABLEAU 3.11 – Résultats pour le job-shop 20x5

T	Instance	PV	LBA	Résultats après 60 secondes						Résultats après 300 secondes					
				UB1	UB2	CPL	G ₁	G ₂	G ₃	UB1	UB2	CPL	G ₁	G ₂	G ₃
5	Défaut	Connu	3921	4023	4035	-	0.30	-∞	-∞	4023	4035	-	0.30	-∞	-∞
	Capacité 1	0.42	3921	4046	4052	-	0.15	-∞	-∞	4046	4052	-	0.15	-∞	-∞
	Capacité 2	0.50	3921	3990	4000	-	0.25	-∞	-∞	3990	4000	-	0.25	-∞	-∞
	Capacité 3	0.46	3921	4022	4038	-	0.40	-∞	-∞	4022	4038	-	0.40	-∞	-∞
	Capacité 4	0.48	3921	3999	4005	-	0.15	-∞	-∞	3999	4005	-	0.15	-∞	-∞
	Coût de setup 1	80	5055	6059	6173	-	1.88	-∞	-∞	6035	6127	-	1.52	-∞	-∞
	Coût de setup 2	100	5455	6929	7006	-	1.11	-∞	-∞	6839	7006	-	2.44	-∞	-∞
	Coût de setup 3	120	5855	7572	7788	-	2.85	-∞	-∞	7572	7788	-	2.85	-∞	-∞
	Coût de setup 4	140	6255	8497	8586	-	1.05	-∞	-∞	8449	8586	-	1.62	-∞	-∞
	Coût de setup 5	180	7055	10074	10529	-	4.52	-∞	-∞	9902	10529	-	6.33	-∞	-∞
	Coût de setup 6	200	7455	11039	11057	-	0.16	-∞	-∞	10871	11057	-	1.71	-∞	-∞
	Coût de setup 7	220	7855	11687	11872	-	1.58	-∞	-∞	11463	11868	-	3.53	-∞	-∞
	Demande 1	[1, 5]	2413	2727	2833	-	3.89	-∞	-∞	2704	2833	-	4.77	-∞	-∞
	Demande 2	[1, 10]	4056	4165	4187	-	0.53	-∞	-∞	4164	4171	-	0.17	-∞	-∞
Demande 3	[4, 8]	4057	4210	4334	-	2.95	-∞	-∞	4205	4334	-	3.07	-∞	-∞	
Demande 4	[10, 15]	7153	7212	7222	-	0.14	-∞	-∞	7212	7222	-	0.14	-∞	-∞	
Demande 5	[5, 15]	6230	6311	6353	-	0.67	-∞	-∞	6311	6353	-	0.67	-∞	-∞	
10	Défaut	Connu	7614	7732	7761	-	0.38	-∞	-∞	7730	7761	-	0.40	-∞	-∞
	Capacité 1	0.42	7614	7814	7846	-	0.41	-∞	-∞	7774	7846	-	0.93	-∞	-∞
	Capacité 2	0.44	7614	7773	7873	-	1.29	-∞	-∞	7773	7873	-	1.29	-∞	-∞
	Capacité 3	0.46	7614	7771	7781	-	0.13	-∞	-∞	7744	7781	-	0.48	-∞	-∞
	Capacité 4	0.50	7614	7726	7764	-	0.49	-∞	-∞	7719	7764	-	0.58	-∞	-∞
	Coût de setup 1	80	9968	10939	11280	-	3.12	-∞	-∞	10852	11280	-	3.94	-∞	-∞
	Coût de setup 2	100	10768	12159	12606	-	3.68	-∞	-∞	12097	12606	-	4.21	-∞	-∞
	Coût de setup 3	120	11568	13492	13960	-	3.47	-∞	-∞	13402	13960	-	4.16	-∞	-∞
	Coût de setup 4	150	12566	15352	16017	-	4.33	-∞	-∞	15173	16017	-	5.56	-∞	-∞
	Coût de setup 5	180	13233	16851	17759	-	5.39	-∞	-∞	16779	17759	-	5.84	-∞	-∞
	Coût de setup 6	200	13633	18409	19388	-	5.32	-∞	-∞	18068	19388	-	7.31	-∞	-∞
	Coût de setup 7	220	14033	19449	20572	-	5.77	-∞	-∞	19253	20572	-	6.85	-∞	-∞
	Demande 1	[1, 5]	4692	4894	4919	-	0.51	-∞	-∞	4886	4919	-	0.68	-∞	-∞
	Demande 2	[1, 10]	7393	7522	7569	-	0.62	-∞	-∞	7517	7569	-	0.69	-∞	-∞
Demande 3	[4, 8]	7926	8062	8094	-	0.79	-∞	-∞	8062	8126	-	0.79	-∞	-∞	
Demande 4	[10, 15]	14363	14418	14440	-	0.15	-∞	-∞	14415	14440	-	0.17	-∞	-∞	
Demande 5	[5, 15]	11951	11981	11985	-	0.03	-∞	-∞	11981	11985	-	0.03	-∞	-∞	
20	Défaut	Connu	18272	18323	18335	-	0.07	-∞	-∞	18317	18335	-	0.03	-∞	-∞
	Capacité 1	0.45	18272	18352	18362	-	0.05	-∞	-∞	18352	18362	-	0.05	-∞	-∞
	Capacité 2	0.48	18272	18339	18372	-	0.18	-∞	-∞	18339	18372	-	0.18	-∞	-∞
	Capacité 3	0.50	18272	18340	18372	-	0.17	-∞	-∞	18340	18372	-	0.17	-∞	-∞
	Capacité 4	0.52	18272	18330	18346	-	0.09	-∞	-∞	18330	18346	-	0.09	-∞	-∞
	Coût de setup 1	5	14100	14100*	14100*	-	0.00	-∞	-∞	14100*	14100*	-	0.00	-∞	-∞
	Coût de setup 2	20	17463	17480	17495	-	0.09	-∞	-∞	17479	17495	-	0.09	-∞	-∞
	Coût de setup 3	50	21443	21677	21706	-	0.13	-∞	-∞	21677	21703	-	0.12	-∞	-∞
	Coût de setup 4	100	26001	27107	27116	-	0.03	-∞	-∞	27096	27116	-	0.07	-∞	-∞
	Coût de setup 5	150	29504	31942	31970	-	0.09	-∞	-∞	31942	31970	-	0.09	-∞	-∞
	Coût de setup 6	200	32497	36865	37218	-	0.96	-∞	-∞	36539	37218	-	1.86	-∞	-∞
	Coût de setup 7	250	35314	40692	41086	-	0.97	-∞	-∞	40273	41086	-	2.02	-∞	-∞
	Demande 1	[5, 7]	15217	15263	15267	-	0.03	-∞	-∞	14263	15267	-	0.03	-∞	-∞
	Demande 2	[1, 2]	5517	5753	5774	-	0.37	-∞	-∞	5737	5765	-	0.49	-∞	-∞
Demande 3	[1, 3]	6527	6715	6770	-	0.82	-∞	-∞	6715	6770	-	0.82	-∞	-∞	
Demande 4	[3, 6]	12131	12194	12196	-	0.02	-∞	-∞	12192	12196	-	0.03	-∞	-∞	
Demande 5	[4, 8]	15552	15576	15596	-	0.13	-∞	-∞	15576	15596	-	0.13	-∞	-∞	

- Pas de solution réalisable.

* Solution optimale.

-∞ Pas de solution réalisable avec IBM ILOG CPLEX (écart indéterminé).

TABLEAU 3.12 – Écarts relatifs pour le job-shop 20x5

T	$G_1(\%)$ après 60 secondes			$G_1(\%)$ après 300 secondes		
	Moy	Min	Max	Moy	Min	Max
5	1.33	0.14	4.52	1.77	0.14	6.33
10	2.11	0.03	5.77	2.58	0.03	7.31
20	0.25	0.00	0.97	0.37	0.00	2.02

planification (5, 10 et 20 périodes). Notre approche est ainsi fortement dominante avec une solution de meilleure qualité sur 50 instances (98.04%) pour 60 secondes et 300 secondes de temps de calcul. Les écarts relatifs sont présentés dans le Tableau 3.12. Ces résultats mettent donc en évidence la pertinence de toutes les améliorations apportées sur la méthode de Wolosewicz *et al.*, que nous avons décrites dans ce chapitre.

3.6 Conclusion

Nous avons présenté une nouvelle approche intégrée pour résoudre le problème intégré de dimensionnement de lots et d'ordonnancement en garantissant des solutions réalisables au niveau opérationnel, contrairement aux approches traditionnelles où les décisions sont prises de manière hiérarchique. La plupart des approches existantes peuvent résoudre des problèmes mono-ressource, car les contraintes de capacité sont agrégées. De plus, à cause des limitations sur l'utilisation de la capacité, les solutions sont souvent sous-optimales.

Dans notre approche, des contraintes de capacité détaillées sont considérées pour générer des solutions réalisables dans des systèmes de production complexes. De plus, l'ordonnancement est effectué sur tout l'horizon de planification, et non pas période par période, ce qui permet d'utiliser la capacité totale d'une manière efficace. Le problème de dimensionnement de lots est résolu pour une séquence fixée avec une heuristique Lagrangienne, et une procédure d'amélioration de la séquence est appliquée pour faire évoluer l'ordonnancement. Ces deux étapes sont connectées de manière itérative à travers une recherche taboue, qui permet d'explorer plusieurs branches de l'espace des solutions.

Cette nouvelle approche intégrée pour la résolution de problèmes avec production mono-niveau, est le fruit de plusieurs améliorations que nous avons apportées sur le travail précurseur de Wolosewicz *et al.* [233, 234, 235]. Des analyses d'autres idées étudiées qui n'ont pas permis d'obtenir de meilleurs résultats ont été également présentées. Les modifications effectuées sur la méthode de résolution ont permis d'obtenir de meilleurs résultats pour la grande majorité des instances testées, et globalement pour tous les groupes d'instances étudiées avec toutes les tailles d'horizon de planification. Par rapport au solveur commercial IBM ILOG CPLEX, l'écart

sur les solutions est encore plus favorable avec notre approche. Il est important de remarquer que le solveur n'est pas capable d'obtenir des solutions réalisables pour les systèmes étudiés avec des horizons de planification comportant 10 et 20 périodes. D'où l'importance de proposer des méthodes de résolution dédiées et la pertinence, dans ce sens, de notre approche intégrée.

Cette approche représente une avancée importante en matière d'intégration de décisions dans la gestion de la chaîne logistique. Elle permet en effet d'intégrer des flux verticaux associés à l'étape de production entre les niveaux tactique et opérationnel. Néanmoins, d'autres activités importantes sont encore à considérer, afin de pouvoir déterminer des solutions qui optimisent globalement la chaîne logistique. C'est pourquoi nous proposons, dans le chapitre suivant, une extension de l'approche au cas de systèmes multi-niveaux. Des flux de l'activité d'approvisionnement sont intégrés (flux horizontaux), notamment pour lier les besoins de production entre les composants et composés de la nomenclature, qui peuvent être fabriqués sur des sites de production différents.

Chapitre 4

Approche intégrée pour des problèmes à plusieurs niveaux

Ce chapitre étend l'approche intégrée de résolution de problèmes mono-niveau de planification et d'ordonnancement de la production au cas de nomenclatures à plusieurs niveaux, dans des systèmes de production complexes de type job-shop¹. Cette méthode peut être adaptée à des scénarios mono-site et multi-sites et donc au contexte général d'approvisionnement dans une chaîne logistique. Les solutions proposées par l'approche sont réalisables au niveau opérationnel, et sont avantageuses en termes de qualité et de temps de calcul par rapport aux solveurs commerciaux.

[4.1 Introduction](#)

[4.2 Évolution du modèle](#)

[4.3 Méthode de résolution](#)

[4.4 Résultats expérimentaux](#)

[4.5 Conclusion](#)

1. Des parties de ce chapitre ont été publiées dans les conférences **MOSIM2012** [86], **IWLS2012** [85], **ROADEF2013** [89] et **IWLS2013** [90].

4.1 Introduction

Dans le chapitre précédent, nous avons présenté une nouvelle approche intégrée pour résoudre des problèmes de planification et d'ordonnancement de la production, dans des systèmes de production à un seul niveau. Nous nous intéressons ici à la résolution des problèmes à plusieurs niveaux. La différence réside dans le fait que des contraintes de production liées à la nomenclature (besoins en composants entre produits) entrent en jeu dans la problématique. Ainsi, non seulement les contraintes de capacité doivent être satisfaites afin de garantir des solutions réalisables au niveau ordonnancement, mais aussi nous devons garantir une cohérence entre les différents niveaux de production.

La synchronisation des niveaux de production est assurée en pratique, dans la plupart des cas, à travers la logique MRP qui consiste à décider des tailles de lots de façon hiérarchique, niveau par niveau, en commençant par les produits les plus agrégés (produits finis ou produits sans successeurs). De cette façon, la taille du lot d'un produit est contrainte par les tailles des lots de ses successeurs directs dans la nomenclature. Bien que le calcul MRP garantit le respect des besoins en matières de tous les produits, il n'existe aucune garantie sur la qualité de la solution, l'optimalité n'étant pas le but de la procédure. Ceci et le fait que la capacité n'est pas intégrée correctement dans le processus de décision conduisent souvent à des coûts supplémentaires non négligeables pour la gestion de la chaîne logistique.

Face à ce constat, nous proposons dans ce qui suit une approche intégrée pour résoudre le problème multi-niveaux de dimensionnement de lots et d'ordonnancement, dans des systèmes de production complexes avec configurations d'atelier de type job-shop. Nous commençons dans la Section 4.2 par introduire les nouvelles notations nécessaires pour la modélisation des problèmes multi-niveaux et par présenter l'évolution du modèle mathématique par rapport au cas mono-niveau. Ensuite, dans la Section 4.3, nous proposons une méthode de résolution. Dans la Section 4.4, nous évaluons la performance de l'approche en réalisant des comparaisons avec le solveur commercial IBM ILOG CPLEX. Pour finir, nous présentons les conclusions dans la Section 4.5.

4.2 Évolution du modèle

4.2.1 Description du problème

Comme dans le chapitre précédent, nous nous intéressons à la résolution du problème intégré de planification et d'ordonnancement de la production dans des ateliers multi-ressources de type

job-shop. Cependant, nous considérons ici des structures de nomenclature établissant des liens ou des besoins de production entre les produits, c'est-à-dire des problèmes multi-niveaux. De manière plus générale, ceci peut correspondre à des chaînes logistiques comprenant différents sites de production, chaque site fabricant des composants différents pour satisfaire les demandes internes d'autres sites, ainsi que les demandes externes.

De manière similaire à l'approche pour des problème mono-niveau, le but est de planifier la production de N produits sur R machines et T périodes en déterminant les dates de début et de fin des O opérations nécessaires pour la fabrication des produits. Les hypothèses considérées pour l'approche du chapitre précédent restent valables. Néanmoins, nous considérons que les produits peuvent avoir besoin de certains composants (d'autres produits) dans leur processus de fabrication.

Le problème d'ordonnancement est à nouveau représenté par un graphe disjonctif, où les opérations sont liées par des contraintes de précédence associées aux gammes de fabrication et à l'utilisation des ressources. Ce graphe reste inchangé par rapport au graphe d'un problème mono-niveau puisqu'il n'y a pas d'arcs de précédence entre les couples *composant-composé*.

En plus des données décrites dans le chapitre précédent, nous utilisons par la suite les données suivantes :

- d_{il} : demande externe du produit i à la fin de la période l ,
- D_{il} : demande totale du produit i à la fin de la période l ,
- e_i : coût unitaire d'échelon stock du produit i ,
- $\mathcal{DS}(i)$: ensemble des successeurs directs du produit i dans la nomenclature,
- $\mathcal{AS}(i)$: ensemble de tous les successeurs du produit i dans la nomenclature,
- $\mathcal{DP}(i)$: ensemble des prédécesseurs directs du produit i dans la nomenclature,
- $\mathcal{AP}(i)$: ensemble de tous les prédécesseurs du produit i dans la nomenclature,
- g_{ij} : nombre d'unités du produit i requises pour produire une unité du produit j ($g_{ij} = 0$ si $j \notin \mathcal{DS}(i)$),
- n_{ij} : nombre de chemins liant les produits i et j dans la nomenclature,
- p_{ij}^n : quantité totale du produit i nécessaire pour produire une unité du produit j , en suivant le chemin n dans la nomenclature,
- $P^n(i, j)$: $n^{\text{ème}}$ chemin reliant le produit i et son successeur j ,
- $K^n(i, j)$: somme des délais d'obtention entre les produits i et $j \in \mathcal{AS}(i)$ à travers le chemin $P^n(i, j)$ dans la nomenclature,
- M_i : somme maximale des délais d'obtention du produit i ,
- $T_i + L_i$: période de production au plus tard du produit i ,
- B : nombre de niveaux de la nomenclature,

- $\mathcal{BL}(b)$: ensemble des produits appartenant au niveau b de la nomenclature.

Les **variables de décision** du problème intégré restent les mêmes (tailles des lots X_{il} et setups Y_{il}), mais nous utilisons en plus les **variables dépendantes** suivantes :

- E_{il} : échelon stock du produit i à la fin de la période l .

Ces notations ont été également utilisées par Afentakis *et al.* [8], sans prise en compte des délais d'obtention, et par Afentakis et Gavish [7], Clark et Armentano [39, 40] et Berretta *et al.* [30], avec des délais d'obtention non nuls, pour l'implémentation d'une formulation basée sur l'échelon stock. Ce concept a été introduit par Clark et Scarf [38] et nous l'expliquons plus tard dans cette section. Une différence importante entre la formulation des travaux précédents et la notre est que nous comptabilisons la production X_{il} à la période où le produit est terminé (la production de i peut commencer au plus tôt au début de la période $l - L_i + 1$ et doit se terminer à la période l). Dans les autres travaux, la production X_{il} démarre à la période l et est disponible pour consommation à la période $l + L_i$.

Définissons d'abord les notions de bases dans un contexte multi-niveaux. Suivant une nomenclature, les différents produits d'un système de production sont organisés par niveau, selon leurs liens de dépendance. Ainsi par exemple, il peut y avoir une nomenclature avec un nombre de niveaux $B = 5$, les produits appartenant au niveau 1 ($\mathcal{BL}(1)$) étant les produits finis (sans successeurs), les produits appartenant au niveau 5 ($\mathcal{BL}(5)$) étant les composants basiques (produits sans prédécesseurs) et, $\mathcal{BL}(b) \forall 1 < b < 5$ représentant les composants intermédiaires. Les produits peuvent être liés à travers de multiples chemins, n_{ij} étant le nombre de chemins entre les produits i et $j \in \mathcal{AS}(i)$.

Nous pouvons donc calculer le **nombre total d'unités** p_{ij}^n du produit i nécessaire pour fabriquer une unité du produit j , en suivant le chemin $P^n(i, j)$, avec l'expression suivante :

$$p_{ij}^n = \prod_{k \in P^n(i, j) - j} g_{k, s^n(k)} \quad (4.1)$$

Un facteur important à prendre en compte dans un problème multi-niveaux est le délai d'obtention. En fait, pour fabriquer le composant i à la période l , nous devons garantir que tous les prédécesseurs directs ($\mathcal{DP}(i)$) de i sont disponibles dès le début de la période $l - L_i + 1$. Nous commençons par calculer la **somme des délais** $K^n(i, j)$ pour obtenir les composants du produit j à partir du produit i , en suivant le chemin $P^n(i, j)$.

$$K^n(i, j) = \sum_{k \in P^n(i, j) - j} L_k \quad (4.2)$$

Nous devons aussi déterminer la **somme maximale de délais** M_i pour obtenir tous les composants du produit i . Une manière récursive d'obtenir cette valeur est donnée par l'équation suivante :

$$M_i = \begin{cases} 0, & \text{si } i \text{ est un composant basique} \\ \max_{j \in \mathcal{DP}(i)} (M_j + L_j), & \text{si } i \text{ n'est pas un composant basique} \end{cases} \quad (4.3)$$

Le produit i peut être fabriqué à partir de la période $M_i + L_i$ (période de fin de production), et au plus tard à la période $T_i + L_i$, T_i étant calculée de manière récursive avec l'équation suivante pour le cas d'une nomenclature avec un seul produit fini représenté par l'indice 1 ([39]).

$$T_i = M_1 - \max_{n=1, \dots, n_{i1}} \{K^n(i, 1)\} \quad (4.4)$$

Nous étendons l'expression au cas général avec j produits finis, comme suit :

$$T_i = \min_{j \in \mathcal{AS}(i); j \in \mathcal{BL}(1)} \left\{ M_j - \max_{n=1, \dots, n_{ij}} [K^n(i, j)] \right\} \quad (4.5)$$

$T_i + L_i$ est en effet la **dernière période** à laquelle le produit i peut être fabriqué, afin de pouvoir produire $j \in \mathcal{AS}(i)$ à la période $M(j) + L_j$. Suivre une stratégie de production à la période $T_i + L_i \forall i \in \{1, \dots, N\}$ correspond à une logique de planification descendante (comme dans l'approche MRP), c'est-à-dire les composants sont produits à la période qui permet de fabriquer un produit successeur, et pas avant, évitant ainsi de générer de niveaux de stocks inutiles. Clark et Armentano [39, 40] et Berretta *et al.* [30] utilisent $T_i + 1$ comme période de démarrage de la production du produit $i \in \{1, \dots, N\}$ (dans notre cas il s'agit de $T_i + L_i$, car nous comptabilisons la production à la période de fin). Si cette condition n'est pas nécessaire pour assurer l'obtention de la solution optimale, car il suffit de permettre la fabrication du produit i à partir de $M_i + 1$ ($M_i + L_i$ dans notre cas), elle permet de borner l'horizon de planification de chaque produit, réduisant la taille de l'espace des solutions. Néanmoins, il peut s'avérer indispensable d'autoriser la production à partir de $M_i + L_i$ comme période initiale pour la production de i , car il peut être impossible, dans certains cas, de produire à la période $T_i + L_i$, tout en respectant la capacité. Les périodes comprises entre $M_i + L_i$ et $T_i + L_i - 1$ donnent une marge de manœuvre pour la répartition de la production à cet effet. De plus, s'il existe une demande externe d_{il} pour le produit i à la période l , avec $M_i + L_i \leq l < T_i + L_i$, elle doit être satisfaite.

Nous considérons comme hypothèse que $I_{i,0} = 0 \forall i \in \{1, \dots, N\}$. Donc, $X_{i,l} = 0 \forall l < M_i + L_i, \forall i \in \{1, \dots, N\}$, i.e. la production du produit i n'est possible qu'à partir de la période $M_i + L_i$.

Exemple : calcul de $p_{ij}^n, K^n(i, j), M_i$ et T_i

Nous présentons sur la Figure 4.1 un exemple de nomenclature à 4 niveaux et 7 produits. Chaque nœud correspond à un produit i , et les chiffres sur les arcs correspondent aux valeurs de $g_{ij} \forall i, \forall j \in \mathcal{DS}(i)$. Nous observons qu'il existe 3 chemins pour obtenir le produit 1 à partir du produit 5, qui sont : $P^1(5, 1)$, $P^2(5, 1)$ et $P^3(5, 1)$. Nous pouvons donc calculer les valeurs de p_{ij}^n comme suit : $p_{51}^1 = 3 \times 1 = 3$, $p_{51}^2 = 3 \times 2 = 6$ et $p_{51}^3 = 3 \times 1 = 3$. Nous pouvons également calculer les sommes des délais d'obtention liées à ces trois chemins comme suit : $K^1(5, 1) = 2 + 4 = 6$, $K^2(5, 1) = 2 + 1 = 3$ et $K^3(5, 1) = 2 + 2 = 4$. La même logique peut être utilisée pour obtenir les autres valeurs de p_{ij}^n et $K^n(i, j)$.

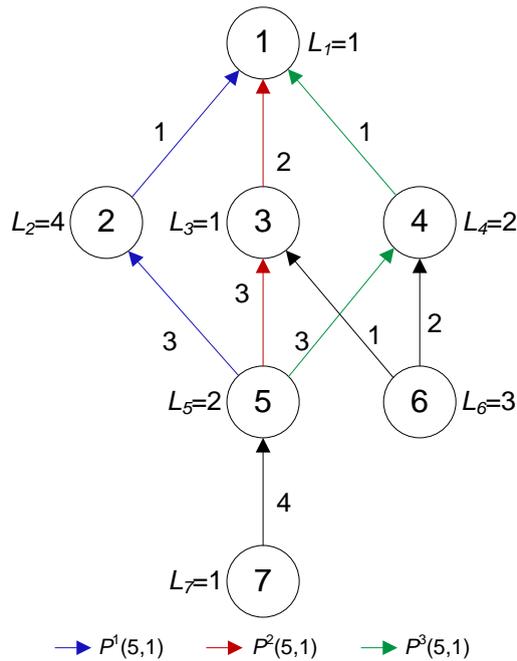


FIGURE 4.1 – Exemple de nomenclature

Les délais maximum sont obtenus comme suit :

- $M_7 = 0, M_6 = 0,$
- $M_5 = M_7 + L_7 = 1,$

- $M_4 = \max(M_5 + L_5, M_6 + L_6) = 3$,
- $M_3 = \max(M_5 + L_5, M_6 + L_6) = 3$,
- $M_2 = M_5 + L_5 = 3$ et
- $M_1 = \max(M_2 + L_2, M_3 + L_3, M_4 + L_4) = 7$.

Ainsi, la production du produit 1 ne peut être réalisée qu'à partir de la période 8 ($M_1 + L_1$), à moins qu'il y ait suffisamment de stock de composants directs à une période précédente. Le calcul de $T_i \forall i$ est présenté ci-dessous.

- $T_1 = M_1 - 0 = 7$,
- $T_2 = M_1 - K^1(2, 1) = 3$,
- $T_3 = M_1 - K^1(3, 1) = 6$,
- $T_4 = M_1 - K^1(4, 1) = 5$,
- $T_5 = M_1 - \max\{K^1(5, 1), K^2(5, 1), K^3(5, 1)\} = 1$,
- $T_6 = M_1 - \max\{K^1(6, 1), K^2(6, 1)\} = 2$ et
- $T_7 = M_1 - \max\{K^1(7, 1), K^2(7, 1), K^3(7, 1)\} = 0$

Constatons que la production des produits 1, 2, 5 et 7 peut être réalisée uniquement aux périodes 8, 7, 3 et 1 respectivement ; tandis que celle des produits 3, 4 et 6 peut être réalisée entre les périodes 4 et 7, 5 et 7 et 3 et 5 respectivement.

Échelon stock et demande d'échelon

L'**échelon stock** $E_{i,l}$ du produit i à la période l est défini dans la littérature comme la quantité de matière de i , existante à l , en tant que produit fini ou en tant que composant, pour satisfaire la demande externe $d_{i,l}$ et interne $\sum_{j \in \mathcal{DS}(i)} g_{ij} D_{j,l+L_j}$. En revanche, l'inventaire $I_{i,l}$ est la quantité de matières vraiment disponible. Il varie en fonction des tailles de lots des successeurs directs $\mathcal{DS}(i)$, et non pas en fonction de la demande interne, comme défini par l'équation suivante :

$$I_{i,l} = I_{i,l-1} + X_{il} - d_{i,l} - \sum_{j \in \mathcal{DS}(i)} g_{ij} X_{j,l+L_j} \quad (4.6)$$

Il y a deux façons d'exprimer l'échelon stock du produit i à la période l , soit en fonction de l'inventaire de tous les successeurs de i avec (4.7), soit de manière récursive en fonction de l'échelon stock des successeurs directs de i avec (4.8).

$$E_{il} = I_{i,l} + \sum_{j \in \mathcal{AS}(i)} \sum_{n=1}^{n_{ij}} p_{ij}^n I_{j,l+L_j+K^n(i,j)} \quad (4.7)$$

$$E_{il} = I_{i,l} + \sum_{j \in \mathcal{DS}(i)} g_{ij} E_{j,l+L_j} \quad (4.8)$$

Comme l'inventaire I_{il} et l'échelon stock E_{il} ne sont pas équivalents, le coût unitaire de stockage du produit i n'est pas égal au coût unitaire d'échelon stock de i . Ce dernier, noté e_i , est défini par l'équation suivante :

$$e_i = c_i^{inv} - \sum_{j \in \mathcal{DP}(i)} c_j^{inv} g_{ji} \quad (4.9)$$

La **demande d'échelon** ou demande totale du produit i à la période l est égale à la somme de ses demandes externes et internes. La définition récursive de la demande est modélisée avec (4.10), et l'expression non-récursive avec (4.11).

$$D_{il} = d_{i,l} + \sum_{j \in \mathcal{DS}(i)} g_{ij} D_{j,l+L_j} \quad (4.10)$$

$$D_{il} = d_{i,l} + \sum_{j \in \mathcal{AS}(i)} \sum_{n=1}^{n_{ij}} p_{ij}^n D_{j,l+L_j+K^n(i,j)} \quad (4.11)$$

4.2.2 Modèle intégré avec notation classique

Le modèle mathématique que nous considérons pour le problème multi-niveaux intégré est celui proposé par Dauzère-Pérès et Lasserre [48], avec la différence que nous ne considérons pas la rupture de stocks. La différence entre cette formulation et celle que nous avons employée pour la modélisation du problème mono-niveau réside dans les contraintes d'équilibre des stocks. Pour tenir compte des liens de production entre les produits, le paramètre g_{ij} permettant de calculer la demande interne de chaque produit est introduit. La formulation du problème intégré est la suivante :

$$\min \sum_{i=1}^N \sum_{l=1}^T (c_i^p X_{il} + c_i^{inv} I_{il} + c_i^s Y_{il}) \quad (3.4)$$

s.c.

$$I_{il} = I_{i,l-1} + X_{il} - d_{il} - \sum_{j \in \mathcal{DS}(i)} g_{ij} X_{j,l+L_j} \quad \forall i, l \quad (4.12)$$

$$t(o') \geq t(o) + p_o^u X_{i(o),l(o)} + s_o^t Y_{i(o),l(o)} \quad \forall (o, o') \in \mathcal{A} \quad (3.1)$$

$$\begin{cases} t(o) \geq t(o') + p_{o'}^u X_{i(o'),l(o')} + s_{o'}^t Y_{i(o'),l(o')} \\ \text{ou} \\ t(o') \geq t(o) + p_o^u X_{i(o),l(o)} + s_o^t Y_{i(o),l(o)} \end{cases} \quad \forall (o, o') \in \mathcal{E} \quad (3.2)$$

$$t(o) + p_o^u X_{i(o),l(o)} + s_o Y_{i(o),l(o)} \leq \sum_{l=1}^{l(o)} c_l \quad \forall o \in \mathcal{L} \quad (3.6)$$

$$t(o) + p_o^u X_{i(o),l(o)} + s_o Y_{i(o),l(o)} \geq \sum_{l=1}^{l(o)-1} c_l \quad \forall o \in \mathcal{L} \quad (3.7)$$

$$t(o) \geq \sum_{l=1}^{l(o)-L_{i(o)}} c_l \quad \forall o \in \mathcal{F} \text{ et } L_{i(o)} > 0 \quad (3.8)$$

$$X_{il} \leq \left(\sum_{k=1}^T D_{ik} \right) Y_{il} \quad \forall i, l \quad (3.9)$$

$$X_{il}, I_{il} \geq 0 \quad \forall i, l \quad (3.10)$$

$$Y_{il} \in \{0, 1\} \quad \forall i, l \quad (3.11)$$

$$t(o) \geq 0 \quad \forall o \quad (3.12)$$

$$X_{il} = 0 \quad \forall i; \forall l < M_i + L_i \quad (4.13)$$

De nombreuses contraintes du modèle sont les mêmes que pour le problème à un seul niveau (voir Section 3.2 du Chapitre 3). Nous avons uniquement modifié les contraintes d'équilibre des stocks (4.12), pour tenir compte de l'aspect multi-niveaux, et ajouté les contraintes (4.13), qui garantissent qu'il n'est pas possible de déclencher la production d'un produit i à une période l si la somme maximale des délais d'obtention du produit i n'est pas supérieure à l .

Comme dans le cas des problèmes à un seul niveau, il est très compliqué de résoudre un tel modèle. De plus, les nouvelles variables de décision ($X_{jl} \forall j \in \mathcal{DS}(i), \forall l$) introduisent une

difficulté supplémentaire, car elles empêchent d'utiliser la propriété de Wagner et Whitin pour la décomposition du problème en plusieurs sous-problèmes mono-produit.

4.2.3 Modèle intégré avec échelon stock

Pour réduire le nombre de variables de décision et rendre le problème décomposable par produit, nous utilisons les variables d'échelon stock et de demande d'échelon, qui permettent de remplacer les variables d'inventaire et les tailles de lot des produits successeurs respectivement. Pour garantir l'approvisionnement en composants, des contraintes de nomenclature liant l'échelon stock de chaque produit avec ceux de ses successeurs sont aussi ajoutées.

$$\min \sum_{i=1}^N \sum_{l=1}^T (c_i^p X_{il} + e_i E_{il} + c_i^s Y_{il}) \quad (4.14)$$

s.c.

$$E_{il} = E_{i,l-1} + X_{il} - D_{il} \quad \forall i, l \quad (4.15)$$

$$\sum_{j \in \mathcal{DS}(i)} g_{ij} E_{j,l+L_j} - E_{il} \leq 0 \quad \forall i, l \quad (4.16)$$

$$t(o') \geq t(o) + p_o^u X_{i(o),l(o)} + s_o^t Y_{i(o),l(o)} \quad \forall (o, o') \in \mathcal{A} \quad (3.1)$$

$$\begin{cases} t(o) \geq t(o') + p_{o'}^u X_{i(o'),l(o')} + s_{o'}^t Y_{i(o'),l(o')} \\ \text{ou} \\ t(o') \geq t(o) + p_o^u X_{i(o),l(o)} + s_o^t Y_{i(o),l(o)} \end{cases} \quad \forall (o, o') \in \mathcal{E} \quad (3.2)$$

$$t(o) + p_o^u X_{i(o),l(o)} + s_o Y_{i(o),l(o)} \leq \sum_{l=1}^{l(o)} c_l \quad \forall o \in \mathcal{L} \quad (3.6)$$

$$t(o) + p_o^u X_{i(o),l(o)} + s_o Y_{i(o),l(o)} \geq \sum_{l=1}^{l(o)-1} c_l \quad \forall o \in \mathcal{L} \quad (3.7)$$

$$t(o) \geq \sum_{l=1}^{l(o)-L_{i(o)}} c_l \quad \forall o \in \mathcal{F} \text{ et } L_{i(o)} > 0 \quad (3.8)$$

$$X_{il} \leq \left(\sum_{k=1}^T D_{ik} \right) Y_{il} \quad \forall i, l \quad (3.9)$$

$$X_{il}, E_{il} \geq 0 \quad \forall i, l \quad (4.17)$$

$$Y_{il} \in \{0, 1\} \quad \forall i, l \quad (3.11)$$

$$t(o) \geq 0 \quad \forall o \quad (3.12)$$

$$X_{il} = 0 \quad \forall i; \forall l < M_i + L_i \quad (4.13)$$

La fonction objectif du modèle est représentée par (4.14), qui correspond à la minimisation du coût total, i.e. la somme des coûts de production, d'échelon stock et de setup. Les contraintes d'équilibre des stocks sont modélisées par (4.15). Les contraintes (4.16) correspondent aux contraintes de nomenclature ou de type BOM (Bill Of Materials). Les contraintes (4.17) forcent les variables de tailles de lots et d'échelon stock à prendre des valeurs strictement positives. Les autres contraintes sont identiques à celles du modèle utilisant les notations classiques de la Section 3.2 du Chapitre 3.

4.2.4 Modèle avec chemins, échelon stock et séquence fixée

Nous adoptons la même stratégie que nous avons utilisée pour la résolution des problèmes à un seul niveau, qui consiste à décomposer le problème en un sous-ensemble de problèmes de planification à séquence fixée. Néanmoins, cette fois-ci nous utilisons les notations d'échelon stock, ce qui conduit à la formulation suivante :

$$\min \sum_{i=1}^N \sum_{l=1}^T (c_i^s Y_{il} + c_i^p X_{il} + e_i E_{il}) \quad (4.14)$$

s.a.

$$E_{i,l-1} + X_{il} - E_{i,l} = D_{i,l} \quad \forall i, l \quad (4.15)$$

$$\sum_{j \in \mathcal{DS}(i)} g_{ij} E_{j,l+L_j} - E_{il} \leq 0 \quad \forall i, l \quad (4.16)$$

$$r(o_c^f) + \sum_{o \in c} (p_o^u X_{i(o)l(o)} + s_o^t Y_{i(o)l(o)}) \leq \sum_{l=1}^{l(o_c^f)} c_l \quad \forall c \in \mathcal{C}(y) \quad (3.25)$$

$$X_{il} \leq \left(\sum_{k=l}^T D_{ik} \right) Y_{il} \quad \forall i, l \quad (3.9)$$

$$X_{il}, E_{il} \geq 0 \quad \forall i, l \quad (4.17)$$

$$Y_{il} = \{0, 1\} \quad \forall i, l \quad (3.11)$$

$$X_{il} = 0 \quad \forall i; \forall l < M_i + L_i \quad (4.13)$$

Hormis la nouvelle fonction objectif (4.14), les nouvelles contraintes d'équilibre de stocks (4.15) et de nomenclature (4.16), nous retrouvons les contraintes de capacité détaillées (3.25) introduites dans la Section 3.2.3 du Chapitre 3, qui permettent d'assurer la faisabilité du plan de production au niveau ordonnancement. Le modèle est complété par les contraintes de production (3.9) liant les variables de décision X et Y , les contraintes de non-négativité (4.17), les contraintes (3.11) forçant les variables de setup à être binaires et les contraintes (4.13) empêchant de fabriquer un produit si son approvisionnement en composants n'est pas garanti.

4.3 Méthode de résolution

Dans les problèmes mono-niveau, la difficulté majeure est au niveau de la validation des contraintes de capacité détaillées, dont le nombre est exponentiel. La relaxation de ces contraintes

conduit à des plans de production qui ne sont pas forcément réalisables, et un lissage de la production est dans la plupart des cas nécessaire afin de les rendre réalisables. En ce qui concerne le contexte des problèmes multi-niveaux, un nouvel obstacle apparaît : la validation des contraintes de nomenclature. Tout comme le cas mono-niveau, le problème multi-niveaux de dimensionnement de lots et d'ordonnancement est \mathcal{NP} -difficile. À cause de la difficulté de résolution, très peu de travaux de dimensionnement de lots avec contraintes de capacité détaillées ont été traités dans la littérature. Particulièrement, les problèmes à plusieurs ressources ont été rarement étudiés, et encore moins ceux qui intègrent des décisions d'ordonnancement.

Nous utilisons la même stratégie de résolution que pour les problèmes à un seul niveau, c'est-à-dire l'application itérative d'une heuristique Lagrangienne pour la résolution du problème de planification avec séquence fixée combinée à une méthode d'amélioration de la séquence.

4.3.1 Heuristique Lagrangienne pour la résolution du problème avec séquence fixée

Nous relâchons les contraintes de nomenclature en plus de celles de capacité, idée aussi mise en œuvre par Tempelmeier et Derstroff [215], et nous utilisons la méthode de sous-gradients pour la mise à jour des multiplicateurs Lagrangiens. Une nouvelle fois, nous combinons la relaxation Lagrangienne avec l'algorithme de programmation dynamique de dimensionnement de lots, proposé par Wagelmans *et al.* [230], ce qui fait que les valeurs de variables de décision varient en fonction des valeurs des multiplicateurs Lagrangiens. Étant donné que deux types de contraintes sont relâchés, le plan de production proposé à chaque itération n'est pas, dans la plupart des cas, réalisable au niveau ordonnancement. La solution correspond à une borne inférieure $LB(y)$ du problème de dimensionnement de lots avec séquence fixée y . Afin de valider les contraintes non satisfaites, nous proposons plusieurs stratégies de lissage de la production, permettant de déterminer un plan de production réalisable, dont le coût correspond à une borne supérieure UB du problème intégré.

Dans ce qui suit, nous décrivons les principales parties de l'heuristique Lagrangienne. Une version simplifiée de l'algorithme est illustrée dans la Figure 4.2.

4.3.1.1 Déterminer un plan de production optimal avec des contraintes de capacité et de nomenclature relâchées

Les multiplicateurs Lagrangiens associés aux contraintes de capacité sont mis à jour de la même manière que pour les problèmes à un seul niveau (voir Section 3.3 du Chapitre 3), c'est-à-

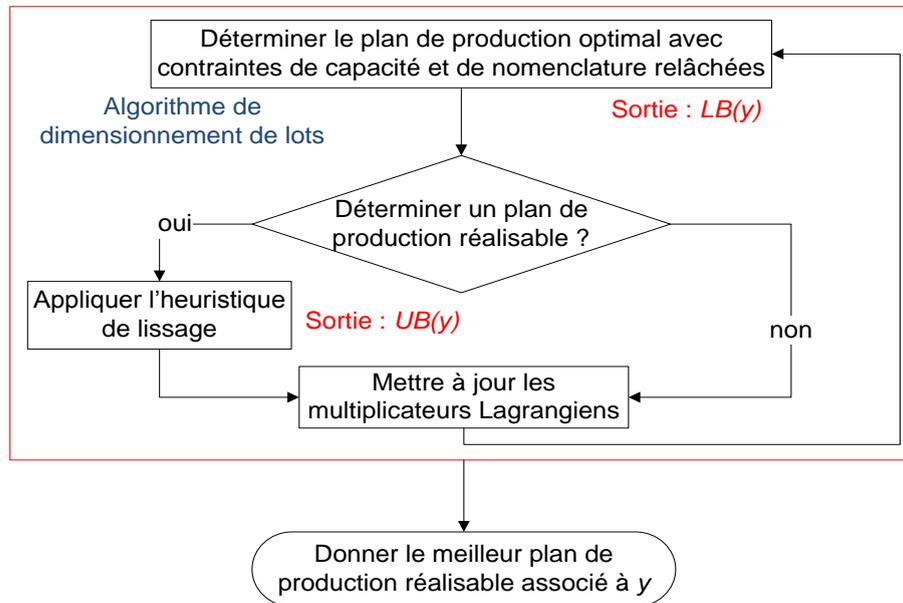


FIGURE 4.2 – Heuristique Lagrangienne pour des problèmes de dimensionnement de lots à plusieurs niveaux

dire qu'à chaque itération, uniquement les chemins les plus violés $\mathcal{CH}(y)$ sont considérés. En ce qui concerne les contraintes de nomenclature, un multiplicateur Lagrangien $\gamma_{i,l}$ est défini pour chaque job (i, l) , et ils sont tous mis à jour à chaque itération.

Le coût du plan de production obtenu à la première itération correspond à la *borne inférieure absolue* (ALB) du problème intégré, qui est calculée quand tous les multiplicateurs Lagrangiens sont à zéro, c'est-à-dire que le plan de production résout de façon optimale le problème mono-niveau de dimensionnement de lots sans contraintes de capacité. La plus grande borne inférieure obtenue par l'heuristique Lagrangienne correspond à la meilleure borne inférieure ($LB^*(y)$) pour le problème avec séquence y fixée. La valeur de $LB(y)$ est obtenue en évaluant la fonction objectif du problème dual ci-dessous :

$$\begin{aligned}
F(X, Y, \beta, \gamma) = \min \sum_{i=1}^N \sum_{l=1}^T & \left\{ \left(c_i^s + \sum_{c \in \mathcal{C}(y)} \beta_c + \sum_{o \in c} s_o^t \right) Y_{il} \right. \\
& + \left[c_i^p + \sum_{k=l}^T (e_i - \gamma_{ik}) + \sum_{c \in \mathcal{C}(y)} \beta_c \sum_{o \in c} p_o^u \right] X_{il} \left. \right\} \\
& + \sum_{c \in \mathcal{C}(y)} \beta_c \left[r(o_c^f) - \sum_{l=1}^{l(o_c^l)} c_l \right] - \sum_{i=1}^N \sum_{l=1}^T \sum_{k=1}^T (e_i - \gamma_{ik}) D_{il} \\
& + \sum_{i=1}^N \sum_{l=1}^T \sum_{j \in \mathcal{DS}(i)} \sum_{k=l}^T \gamma_{il} g_{ij} X_{j,l+L_j} \\
& + \sum_{i=1}^N \sum_{l=1}^T \sum_{j \in \mathcal{DS}(i)} \sum_{k=l}^T \gamma_{il} g_{ij} D_{j,l+L_j}
\end{aligned} \tag{4.18}$$

Le simplification de la fonction objectif relâchée jusqu'à obtenir (4.18), qui dépend uniquement des variables de décision, est détaillée dans l'Annexe A de ce manuscrit. À partir de cette expression, nous pouvons identifier le coût dynamique $C_Y(\beta)$ à utiliser dans l'algorithme de Wagelmans *et al.* [230] pour faire varier les décisions de setup. Ce coût est calculé avec l'équation suivante :

$$C_Y(\beta) = c_i^s + \sum_{c \in \mathcal{C}(y)} \beta_c \sum_{o \in c} s_o^t \tag{4.19}$$

Pour obtenir le coût dynamique $C_X(\beta, \gamma)$ modifiant les tailles de lots, une transformation de la fonction objectif relâchée est nécessaire. En fait, d'après (4.18), les tailles de lot des produits successeurs ont une influence sur la solution du problème dual. Les décisions concernant les tailles de lots ne peuvent donc pas être prises de manière isolée (sans tenir compte des liens de nomenclature). Pour que les décisions soient cohérentes, l'information liant composants et composés doit être considérée dans le coût dynamique sur la taille de lot de chaque produit. Nous pouvons réécrire la fonction objectif du problème dual comme suit :

$$\begin{aligned}
F(X, Y, \beta, \gamma) = \min \sum_{i=1}^N \sum_{l=1}^T \left\{ \left(c_i^s + \sum_{c \in \mathcal{C}(y)} \beta_c + \sum_{o \in c} s_o^t \right) Y_{il} \right. \\
+ \left[c_i^p + \sum_{k=l}^T (e_i - \gamma_{ik}) + \sum_{c \in \mathcal{C}(y)} \beta_c \sum_{o \in c} p_o^u + \sum_{j \in \mathcal{DP}(i)} \sum_{k=l-L_j}^{T-L_j} \gamma_{jk} g_{ji} \right] X_{il} \Big\} \\
+ \sum_{c \in \mathcal{C}(y)} \beta_c \left[r(o_c^f) - \sum_{l=1}^{l(o_c^l)} c_l \right] - \sum_{i=1}^N \sum_{l=1}^T \sum_{k=1}^T (e_i - \gamma_{ik}) D_{il} \\
+ \sum_{i=1}^N \sum_{l=1}^T \sum_{j \in \mathcal{DS}(i)} \sum_{k=l}^T \gamma_{il} g_{ij} D_{j,l+L_j}
\end{aligned} \tag{4.20}$$

À partir de cette dernière expression, nous pouvons identifier $C_X(\beta, \gamma)$, défini par l'équation suivante :

$$C_X(\beta, \gamma) = c_i^p + \sum_{k=l}^T (e_i + \gamma_{ik}) + \sum_{c \in \mathcal{C}(y)} \beta_c \sum_{o \in c} p_o^u + \sum_{j \in \mathcal{DP}(i)} \sum_{k=l-L_j}^{T-L_j} \gamma_{jk} g_{ji} \tag{4.21}$$

4.3.1.2 Heuristique de lissage de la production

Le but de cette procédure est de réaliser tous les mouvements nécessaires pour réparer les contraintes de capacité et de nomenclature qui sont violées à une itération donnée de la relaxation Lagrangienne. Les objectifs de production sont modifiés, en déplaçant la production vers des périodes qui ont de la marge de capacité. Le coût du nouveau plan de production correspond à une *borne supérieure* (*UB*). Plusieurs heuristiques de lissage ont été conçues et testées et sont présentées dans ce que suit.

L'algorithme utilisé pour satisfaire les contraintes de capacité est le même que pour les problèmes mono-niveau (voir section 3.3.1.2 du Chapitre 3), mais la quantité de transfert peut être limitée au respect des contraintes de nomenclature. En revanche, pour satisfaire les contraintes de nomenclature, nous proposons trois algorithmes différents : un algorithme glouton, un algorithme descendant et un algorithme ascendant. Le couplage de la procédure réparant les contraintes de capacité et celle réparant les contraintes de nomenclature est possible au moyen de 5 stratégies. La première stratégie, nommée **SP1**, et schématisée dans la Figure 4.3, consiste à satisfaire en premier complètement les contraintes de capacité, sans violer ou sans augmenter le degré de violation (avec restriction) des contraintes de nomenclature. Une fois que les

contraintes de capacité ont toutes été validées, les contraintes de nomenclature sont réparées sans violer (avec restriction) les contraintes de capacité. Des restrictions sont donc imposées sur les deux types de mouvement.

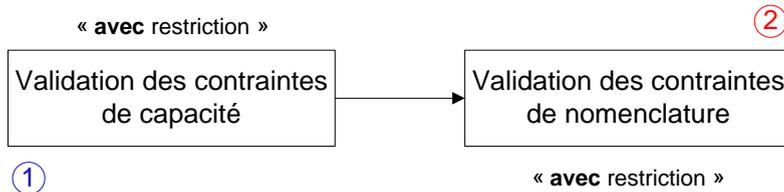


FIGURE 4.3 – Stratégie de lissage SP1

Inversement, avec la stratégie **SP2**, illustrée sur la Figure 4.4, nous cherchons à valider en premier toutes les contraintes de nomenclature, sans violer ou sans augmenter le degré de violation des contraintes de capacité et, une fois que les contraintes de nomenclature sont satisfaites, nous cherchons à valider les contraintes de capacité, sans violer les autres contraintes.

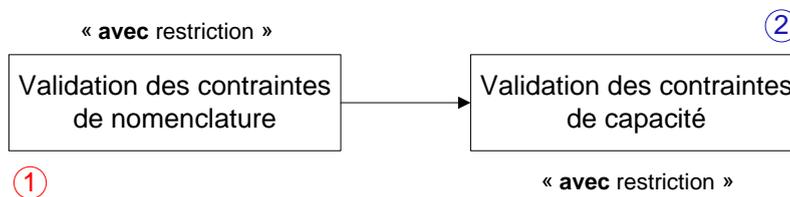


FIGURE 4.4 – Stratégie de lissage SP2

Nous avons constaté que la condition de non-violation de l'autre type de contrainte pouvait rendre très restrictif les mouvements possibles. C'est pourquoi nous avons testé deux nouvelles procédures, nommées **SP3** et **SP4**, qui sont illustrées sur les Figures 4.5 et 4.6, et qui autorisent la violation (si nécessaire) du deuxième type de contrainte (sans restriction). Puis, une fois que toutes les contraintes du premier type sont satisfaites, on cherche à valider les autres contraintes, cette fois-ci avec la condition de non-violation (avec restriction) pour ne pas revenir à des solutions non-réalisables.

Pour diminuer le degré de restriction des mouvements, nous proposons la stratégie **SP5** qui comporte deux étapes principales (Figure 4.7). La première étape a pour but de permettre la réalisation des mouvements sans restriction, c'est-à-dire que l'on cherche à satisfaire les contraintes de nomenclature sans se préoccuper des contraintes de capacité et vice-versa. Dans

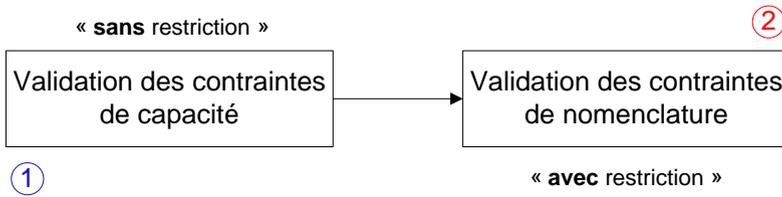


FIGURE 4.5 – Stratégie de lissage SP3

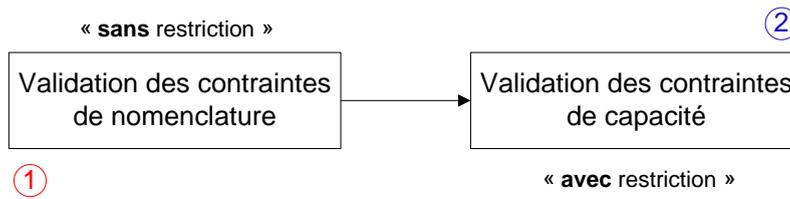


FIGURE 4.6 – Stratégie de lissage SP4

la deuxième étape, la restriction de non-violation des contraintes est utilisée pour chercher à construire une solution réalisable.

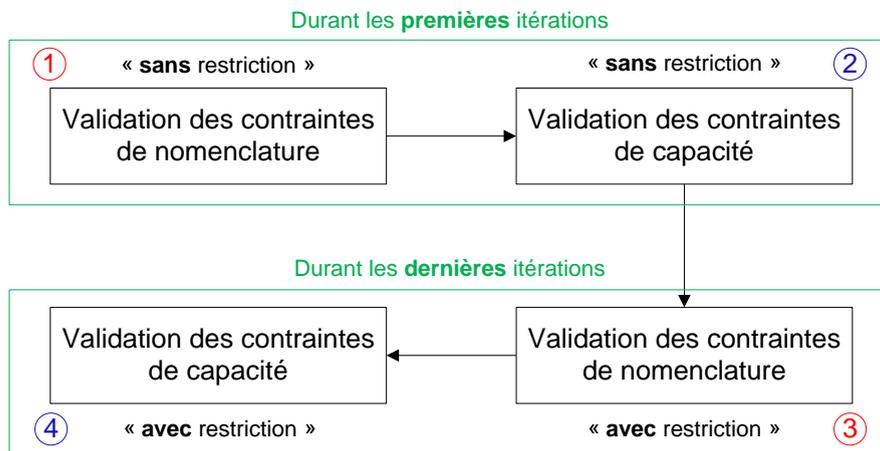


FIGURE 4.7 – Stratégie de lissage SP5

Algorithme glouton

Cette procédure, illustrée sur la Figure 4.8, consiste à identifier à chaque itération la contrainte la plus violée et à réaliser le mouvement diminuant le degré de violation avec le

coût unitaire le plus petit. Pour satisfaire une contrainte de nomenclature, il est possible de modifier la taille de lot du produit critique, en déplaçant une quantité de production d'avant en arrière dans l'horizon de planification (vers une période plus tard), ou de modifier la taille des lots de certains successeurs, en réalisant un mouvement de production d'arrière en avant (vers une période plus récente). Pour ce faire, tous les mouvements possibles sont testés. Puis, le mouvement avec le coût unitaire le plus petit est effectué. La procédure s'arrête une fois que toutes les contraintes de nomenclature sont respectées, ou lorsqu'il n'est plus possible de faire des mouvements.

La **quantité de transfert** est calculée comme suit. D'abord, définissons l_c , l_s et l_d comme étant la période critique (celle où se trouve le produit critique), la période source (d'où l'on enlève de la production) et la période destination (vers où l'on transfère la production) respectivement. Dans la procédure réparant les contraintes de capacité, $l_c = l_s$; tandis que dans la procédure réparant les contraintes de nomenclature, l_c et l_s ne sont pas forcément les mêmes.

La quantité de transfert cherchant à valider les contraintes de capacité, **sans respecter** les contraintes de nomenclature (SP3), est déterminée de la façon suivante :

– Si $l_s < l_d$:

1. Nous calculons avec (4.22) la quantité maximale $Q_1(i)$ du produit critique i qui peut être déplacée sans causer une rupture d'échelon stock entre les périodes concernées par le transfert.

$$Q_1(i) = \max \left[0, \min_{l_s+1 \leq l < l_d} (E_{i,l_s-1} + X_{i,l_s} - D_{i,l_s}, E_{il}) \right] \quad (4.22)$$

2. Nous calculons avec (4.23) la quantité de transfert Q_{i,l_s,l_d} du produit i , de la période l_s vers la période l_d , respectant la condition précédente, la quantité limite de production et la marge de capacité (déterminée avec (3.30)).

$$Q_{i,l_s,l_d} = \min (Q_1(i), X_{i,l_s}, \text{marge}) \quad (4.23)$$

- Si $l_s > l_d$, l'équilibre d'échelon stock est toujours respecté. La quantité de transfert Q_{i,l_s,l_d} doit simplement garantir que la quantité limite de production et la marge de capacité ne sont pas dépassées.

$$Q_{i,l_s,l_d} = \min (X_{i,l_s}, \text{marge}) \quad (4.24)$$

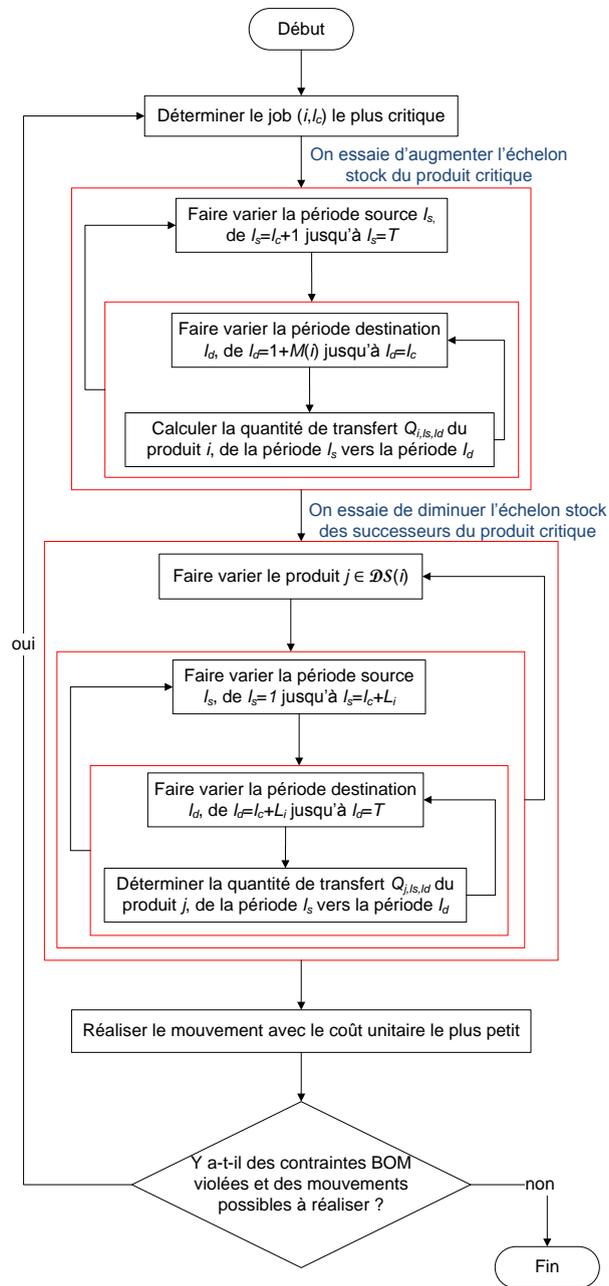


FIGURE 4.8 – Algorithme glouton pour satisfaire les contraintes de nomenclature

La quantité de transfert réparant les contraintes de nomenclature, **sans respecter** les contraintes de capacité (SP4), est déterminée de la façon suivante.

– Si $l_s < l_d$ (déplacement d'un successeur $j \in \mathcal{DS}(i)$ du produit critique i) :

1. Nous calculons avec (4.25) la quantité maximale $Q_2(j)$ de j qu'il est possible à transférer sans violer la contrainte de nomenclature liant i et $\mathcal{DS}(i)$ entre l_s et l_d .

$$Q_2(j) = \max \left[0, \min_{l_s \leq l < l_d} \left(E_{j,l} - \sum_{z \in \mathcal{DS}(j)} g_{jz} E_{z,l+L_z} \right) \right] \quad (4.25)$$

2. Nous calculons avec (4.26) la quantité maximale $Q_1(j)$ qui peut être déplacée de façon à ne pas créer de rupture d'échelon stock de j sur aucune des périodes comprises dans l'intervalle $[l_s + 1, l_d - 1]$.

$$Q_1(j) = \max \left[0, \min_{l_s+1 \leq l < l_d} (E_{j,l_s-1} + X_{j,l_s} - D_{j,l_s}, E_{j,l}) \right] \quad (4.26)$$

3. Finalement, nous calculons avec (4.27) la quantité de transfert Q_{j,l_s,l_d} , du produit j , de la période l_s vers la période l_d , qui respecte ces deux conditions.

$$Q_{j,l_s,l_d} = \min (Q_1(j), Q_2(j), X_{j,l_s}) \quad (4.27)$$

– Si $l_s > l_d$ (déplacement du produit critique i) :

1. Pour chaque prédécesseur j du produit i , nous calculons avec (4.28) la quantité maximale $Q^j(i)$ de i qui peut être déplacée vers la période destination, sans créer de rupture de stock sur j .

$$Q^j(i) = \max \left(0, \frac{E_{j,l_d-L_i}}{g_{ji}} - E_{i,l_d} - \sum_{z \in \mathcal{DS}(j); z \neq i} \frac{g_{jz}}{g_{ji}} E_{z,l_d} \right), \forall j \in \mathcal{DP}(i) \quad (4.28)$$

2. Nous calculons avec (4.29) la quantité de transfert Q_{i,l_s,l_d} , du produit i , de la période l_s vers la période l_d , respectant la condition précédente et la quantité limite de production.

$$Q_{i,l_s,l_d} = \min_{j \in \mathcal{DP}(i)} (Q^j(i), X_{i,l_s}) \quad (4.29)$$

Si l'on souhaite **respecter les contraintes** qui ne font pas l'objet de validation (dans les étapes « sans restriction »), qu'il s'agisse de la capacité (SP2) ou de la nomenclature (SP1), la quantité de transfert Q_{j,l_s,l_d} du produit j , de la période l_s vers la période l_d , est déterminée comme suit :

– Si $l_s < l_d$:

$$Q_{j,l_s,l_d} = \min(Q_1(j), Q_2(j), X_{j,l_s}, marge) \quad (4.30)$$

où $j \in \mathcal{DS}(i)$, avec i étant le produit critique, si le but est de réparer les contraintes de nomenclature. En revanche, si le but est de réparer les contraintes de capacité, j est le produit critique.

– Si $l_s > l_d$:

$$Q_{j,l_s,l_d} = \min_{j \in \mathcal{DP}(i)} (Q^j(i), X_{i,l_s}, marge) \quad (4.31)$$

où i est le produit critique.

Algorithme descendant

Cette procédure suit une structure décisionnelle similaire à la construction du plan MRP, c'est-à-dire que les tailles de lots des produits sont modifiées pour satisfaire les objectifs de production des successeurs directs. La procédure est réalisée de haut en bas, niveau par niveau, et seuls les prédécesseurs peuvent être modifiés, comme illustré sur la Figure 4.9. Cette stratégie permet de construire une solution respectant les contraintes de nomenclature plus rapidement que l'algorithme glouton.

La façon de calculer la quantité de transfert pour réparer les contraintes de capacité reste la même. En ce qui concerne la validation des contraintes de nomenclature, les mouvements de production sont réalisés d'avant en arrière ($l_s > l_d$) comme suit :

1. Nous calculons avec (4.32) la quantité $Q_3(i)$ du produit i nécessaire pour satisfaire la contrainte de nomenclature liant i et $\mathcal{DS}(i)$.

$$Q_3(i) = \max \left(0, \sum_{j \in \mathcal{DS}(i)} g_{ij} E_{j,l_d+L_j} - E_{i,l_d} \right) \quad (4.32)$$

2. Nous calculons la quantité de production Q_{i,l_s,l_d} du produit i à déplacer de la période l_s vers la période l_d , avec (4.33) si l'on interdit la violation des contraintes de capacité, ou avec (4.34) sinon.

$$Q_{i,l_s,l_d} = \min(Q_3(i), X_{i,l_s}, marge) \quad (4.33)$$

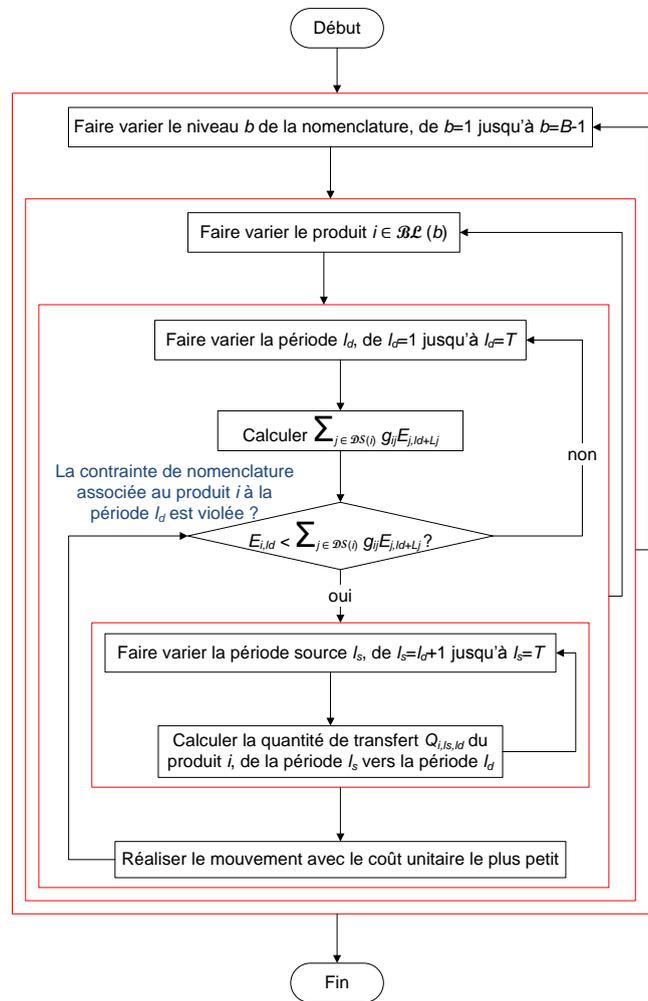


FIGURE 4.9 – Algorithme descendant pour satisfaire les contraintes de nomenclature

$$Q_{i,l_s,l_d} = \min(Q_3(i), X_{i,l_s}) \tag{4.34}$$

Algorithme ascendant

Dans cette procédure, présentée sur la Figure 4.10, les mouvements sont effectués de bas en haut, niveau par niveau, en modifiant uniquement les tailles de lots des successeurs directs. Contrairement à l'algorithme descendant, où les décisions sur les prédécesseurs peuvent être facilement subordonnées aux décisions sur les successeurs, dans une logique de bas en haut, les modifications sur un successeur doivent tenir compte des décisions sur tous ses prédécesseurs.

Les quantités à transférer pour satisfaire les contraintes de capacité sont calculées comme expliqué précédemment. En ce qui concerne la validation des contraintes de nomenclature, les transferts de production sont réalisés d'arrière en avant ($l_s < l_d$) comme suit :

1. Nous calculons avec (4.35) la quantité $Q_4(j)$ de $j \in \mathcal{DS}(i)$ nécessaire à déplacer pour réparer la contrainte de nomenclature associée au produit critique i à la période critique l_c .

$$Q_4(j) = \max \left(0, \frac{\sum_{z \in \mathcal{DS}(i)} g_{iz} E_{z,l_s} - E_{i,l_s - L_z}}{g_{ij}} \right) \quad (4.35)$$

où $l_s = l_c + L_i$.

2. Nous calculons avec (4.36) la quantité maximale $Q_5(j)$ de $j \in \mathcal{DS}(i)$ à transférer pour ne pas créer de rupture d'échelon stock.

$$Q_5(j) = \max \left[0, \min_{l_s+1 \leq p < l_d} (E_{j,l_s-1} + X_{j,l_s} - D_{j,l_s}, E_{j,p}) \right] \quad (4.36)$$

où $l_s < l_d \leq T$.

3. La quantité de transfert Q_{j,l_s,l_d} du produit j de la période l_s vers la période l_d est finalement calculée avec (4.37) si l'on interdit la violation des contraintes de capacité, ou avec (4.38) sinon.

$$Q_{j,l_s,l_d} = \min (Q_4(j), Q_5(j), X_{j,l_s}, \text{marge}) \quad (4.37)$$

$$Q_{j,l_s,l_d} = \min (Q_4(j), Q_5(j), X_{j,l_s}) \quad (4.38)$$

4.3.2 Méthode d'amélioration de la séquence

De la même manière que pour les problèmes à un seul niveau, nous utilisons ici une méthode pour améliorer la séquence suivant l'algorithme présenté dans la Section 3.3.2 du Chapitre 3. Nous nous intéressons donc aux changements de l'orientation des arcs du graphe conjonctif les plus violés durant la relaxation Lagrangienne, en utilisant l'information provenant des multiplicateurs Lagrangiens associés aux contraintes de capacité. L'objectif est de trouver une nouvelle séquence, permettant de mieux répartir la production, pour réduire le degré de violation des

contraintes de capacité. Nous ne nous intéressons donc pas pour l'instant à une sélection de la séquence basée sur les contraintes de nomenclature.

Nous connaissons l'information relative au degré de violation des contraintes de nomenclature, mais il n'y a pas d'arcs dans le graphe conjonctif liant les produits par rapport à la nomenclature. Il est difficile alors de prévoir l'impact du changement d'un arc sur les contraintes de nomenclature. C'est pourquoi nous construisons le voisinage uniquement par rapport à l'information concernant la capacité. Cependant, pour des futurs travaux, nous envisageons la construction de voisinages autour des liens composant-composé dont les contraintes de nomenclature sont violées.

La procédure de construction du voisinage et les critères de sélection de l'arc à changer à chaque itération restent donc identiques à ceux de l'approche pour des problèmes mono-niveau.

4.4 Résultats expérimentaux

Afin de tester l'efficacité de l'approche intégrée pour résoudre des problèmes multi-niveaux, nous comparons sa performance à celle du solveur commercial IBM ILOG CPLEX Optimization Studio 12.6, et nous comparons également les différentes stratégies et heuristiques de lissage présentées dans la section précédente. Pour ce faire, nous réalisons des expérimentations résolvant le problème de dimensionnement de lots avec séquence fixée et le problème intégré. L'approche intégrée a été implémentée sur Microsoft Visual C++ 2010 Express, et les expérimentations ont été réalisées avec un ordinateur présentant les caractéristiques suivantes : système d'exploitation de 64 bits (Microsoft Windows 7), processeur Intel core i7 4 Quad @2.0GHz et 8GB de mémoire RAM.

Les systèmes étudiés correspondent à des ateliers de type job-shop, avec 6 produits, 6 opérations par produit et 6 machines (6x6) et avec 10 produits, 10 opérations par produit et 10 machines (10x10). Nous considérons des horizons de planification avec 5, 10 et 20 périodes et des demandes externes sur tous les produits. De nombreuses expérimentations ont été réalisées en faisant varier la capacité par période, permettant de mesurer l'impact de ce paramètre au niveau ordonnancement.

4.4.1 Expérimentations avec des ateliers de type job-shop 6x6

Nous étudions 4 nomenclatures différentes avec 6 produits, chacun ayant besoin du traitement de 6 opérations sur 6 machines différentes. Les paramètres généraux utilisés pour construire les instances sont affichés dans le Tableau 4.1. Ces paramètres restent fixes pour

toutes les expérimentations, et nous faisons varier la valeur du ratio de capacité cap (défini dans la Section 3.5 du Chapitre 3) pour chaque instance, permettant d’obtenir des longueurs de période c_l différentes. Dans ce cas, la demande utilisée pour la génération de la capacité (3.40) est la demande d’échelon.

Nomenclature	c_i^s	d_{il}	L_i	c_i^p	c_i^{inv}
6x6-1	60	[0, 15]	1	4	[1, 15]
6x6-2	100	[0, 15]	1	4	[1, 25]
6x6-3	100	[0, 15]	1	4	[1, 15]
6x6-4	60	[0, 15]	1	4	[1, 15]

TABLEAU 4.1 – Paramètres généraux associés aux ateliers de type job-shop 6x6

4.4.1.1 Nomenclature 6x6-1

Dans la nomenclature 6x6-1 (Figure 4.11), nous avons trois niveaux et six produits, avec un produit sans successeurs et deux produits sans prédécesseurs. Dans le Tableau 4.2, nous présentons les solutions obtenues par le solveur IBM ILOG CPLEX, avec une limite de temps de 60 secondes, pour 13 instances où varient le ratio cap et le nombre de périodes T . Nous affichons la valeur de la borne inférieure absolue ALB , la solution du problème de dimensionnement de lots pour une séquence fixée y donnée $CPL1$, et la solution du problème intégré $CPL2$. Nous constatons que le solveur commercial arrive à résoudre de façon optimale toutes les instances.

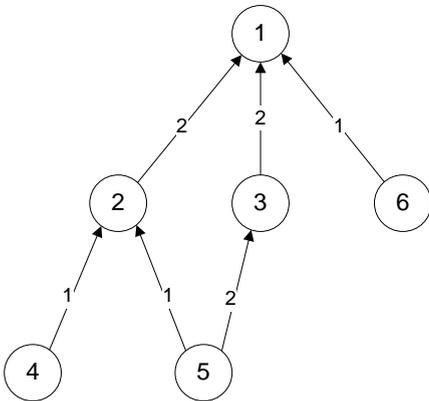


FIGURE 4.11 – Nomenclature 6x6-1

Instance	cap	T	ALB	$CPL1$	$CPL2$
1	0.60	5	4474	4724*	4715*
2	0.65			4715*	4715*
3	0.70			4690*	4690*
4	0.75			4690*	4690*
5	0.80			4690*	4690*
6	0.65	10	9965	10550*	10441*
7	0.70			10447*	10438*
8	0.75			10438*	10438*
9	0.80			10416*	10416*
10	0.65	20	22796	24113*	24074*
11	0.70			23960*	23970*
12	0.75			23908*	23874*
13	0.80			23904*	23874*

TABLEAU 4.2 – Solutions obtenues avec IBM ILOG CPLEX pour la nomenclature 6x6-1

Application de l'algorithme glouton

Les résultats de notre approche pour le problème à séquence fixée et pour le problème intégré, en utilisant l'algorithme glouton pour la validation des contraintes de nomenclature, sont présentés dans le Tableau 4.3. G est l'écart (en pourcentage) entre la borne supérieure UB obtenue par notre approche et la solution $CPL1$ pour le problème à séquence fixée et $CPL2$ pour le problème intégré fournie par le solveur, t est le temps de calcul exprimé en secondes pour le problème à séquence fixée et A est le pourcentage d'amélioration de la borne supérieure du problème à séquence fixée, lorsque l'on résout le problème intégré avec une limite de temps de 60 secondes.

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	$t(s)$	UB	G	$t(s)$	UB	G	$t(s)$	UB	G	$t(s)$	UB	G	$t(s)$
1	4725	0.02	0.30	4724*	0.00	0.23	4832	2.29	0.40	4724*	0.00	0.22	4832	2.29	0.39
2	4715*	0.00	1.10	4715*	0.00	0.33	4715*	0.00	0.38	4715*	0.00	0.32	4715*	0.00	0.45
3	4690*	0.00	0.34	4690*	0.00	0.32	4690*	0.00	0.34	4690*	0.00	0.32	4690*	0.00	0.33
4	4690*	0.00	0.33	4690*	0.00	0.32	4690*	0.00	0.34	4690*	0.00	0.31	4690*	0.00	0.34
5	4690*	0.00	0.33	4690*	0.00	0.31	4690*	0.00	0.34	4690*	0.00	0.32	4690*	0.00	0.34
6	10637	0.82	8.21	10624	0.70	2.65	10646	0.91	1.16	10650	0.95	3.07	10646	0.91	2.11
7	10448	0.01	3.29	10519	0.69	1.42	10575	1.23	1.78	10519	0.69	1.36	10575	1.23	3.04
8	10440	0.02	1.07	10438*	0.00	0.70	10438*	0.00	1.45	10438*	0.00	0.72	10438*	0.00	1.14
9	10438	0.21	0.76	10490	0.71	1.00	10438	0.21	0.88	10490	0.71	1.05	10438	0.21	0.98
10	24410	1.23	40.96	24390	1.15	23.12	24398	1.18	24.79	24390	1.15	17.37	25028	3.79	49.78
11	24262	1.26	19.16	24205	1.02	21.80	24398	1.83	24.93	24205	1.02	17.12	24179	0.91	45.19
12	24141	0.97	22.83	24077	0.71	21.06	24130	0.93	24.41	24057	0.62	17.14	24062	0.64	47.30
13	24170	1.11	23.20	24178	1.15	26.56	24166	1.10	28.06	24069	0.69	21.11	24072	0.70	48.55

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A												
1	4720	0.11	0.11	4724	0.19	0.00	4832	2.48	0.00	4724	0.19	0.00	4832	2.48	0.00
2	4715*	0.00	0.00	4715*	0.00	0.00	4715*	0.00	0.00	4715*	0.00	0.00	4715*	0.00	0.00
3	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00
4	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00
5	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00
6	10532	0.87	0.99	10532	0.87	0.87	10646	1.96	0.00	10581	1.34	0.65	10646	1.96	0.00
7	10438*	0.00	0.10	10438*	0.00	0.77	10575	1.31	0.00	10438*	0.00	0.77	10438*	0.00	1.30
8	10438*	0.00	0.02	10438*	0.00	0.00	10438*	0.00	0.00	10438*	0.00	0.00	10438*	0.00	0.00
9	10438	0.21	0.00	10438	0.21	0.50	10438	0.21	0.00	10438	0.21	0.50	10438	0.21	0.00
10	24338	1.10	0.29	24384	1.29	0.02	24398	1.35	0.00	24390	1.31	0.00	25028	3.96	0.00
11	24243	1.14	0.08	24199	0.96	0.02	24229	1.08	0.69	24199	0.96	0.02	24179	0.87	0.00
12	24133	1.08	0.03	24057	0.77	0.08	24110	0.99	0.08	24057	0.77	0.00	24062	0.79	0.00
13	24062	0.79	0.45	24033	0.67	0.60	24166	1.22	0.00	24069	0.82	0.00	24072	0.83	0.00

TABLEAU 4.3 – Résultats pour la nomenclature 6x6-1 avec l'algorithme glouton

En ce qui concerne le problème à séquence fixée, les solutions avec toutes les stratégies de lissage sont bonnes, avec un écart maximum de 3.79% pour SP5. Notre approche trouve aussi plusieurs solutions optimales (dont la plupart pour $T = 5$), et les temps de calcul sont faibles, excepté pour $T = 20$, où plusieurs itérations de l'heuristique de lissage sont nécessaires afin

de construire une solution réalisable. Parmi toutes les stratégies de lissage, SP4 propose une meilleure performance globale. SP1 présente le meilleur écart moyen ($\bar{G} = 0.44\%$), mais la différence avec SP4 ($\bar{G} = 0.45\%$) n'est pas significative. De plus, SP4 domine sur l'écart-type moyen ($\sigma_{\bar{G}} = 0.46$) et présente l'écart maximum le plus petit ($G_{max} = 1.15\%$). En termes de temps de calcul, c'est aussi SP4 qui domine globalement. La stratégie la moins performante est SP5, aussi bien en qualité de solution qu'en temps de calcul.

Si nous comparons le fait d'autoriser ou d'interdire la violation des contraintes qui ne sont pas validées à la première étape, nous constatons qu'entre SP2 et SP4 (procédures commençant par satisfaire les contraintes de nomenclature) il n'y a pas une différence significative sur la qualité des solutions (SP4 domine légèrement), mais l'impact est plus visible sur le temps de calcul. Le fait d'autoriser la violation des contraintes de capacité permet de construire un plan de production respectant les contraintes de nomenclature plus rapidement, et cet avantage est maintenu jusqu'à la fin de l'exécution. Entre SP1 et SP3, la différence sur la qualité des solutions est un peu plus importante, SP3 offrant une performance moins intéressante, avec un écart moyen de 0.74% . Quant au temps de calcul, le fait de permettre la violation des contraintes de nomenclature durant la validation des contraintes de capacité a globalement une influence positive.

Concernant le problème intégré, c'est la stratégie SP2 qui génère globalement la plus grande amélioration des bornes supérieures. SP2 domine SP1 en performance, avec un écart moyen de 0.38% contre 0.41% pour SP1. La stratégie SP4 offre aussi une bonne performance, avec $\bar{G} = 0.43\%$. De plus, 6 solutions optimales sont obtenues avec SP1, SP2, SP4 et SP5, et 5 avec SP3.

Application de l'algorithme descendant

Résolvons maintenant les mêmes instances, en utilisant l'algorithme descendant pour réparer les contraintes de nomenclature. Les résultats sont présentés dans le Tableau 4.4.

Nous observons que la qualité des solutions est fortement détériorée par rapport à l'approche utilisant l'algorithme glouton. En fait, les bornes supérieures affichées correspondent aux solutions initiales où $X_{il} = D_{il} \forall i, l$, car l'heuristique de lissage est incapable, dans la plupart des cas, de construire une solution réalisable. Ce comportement a deux causes. Premièrement, à travers les stratégies commençant par satisfaire les contraintes de nomenclature (SP2 et SP4), une solution respectant ces contraintes est rapidement construite en suivant la logique MRP. Le plan de production résultant privilégie fortement les décisions associées aux produits finis, et l'importance des décisions descend niveau par niveau jusqu'à la planification des composants

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)
1	5032	8.64	4.04	5032	8.64	0.43	5032	8.64	2.57	5032	8.64	0.25	5032	8.64	0.48
2	5032	8.88	3.58	5032	8.88	0.43	5032	8.88	3.00	5032	8.88	0.30	5032	8.88	0.46
3	5032	10.41	0.95	5032	10.41	0.41	5032	10.41	0.99	5032	10.41	0.28	5032	10.41	0.42
4	5032	10.89	1.20	5032	10.89	0.48	5032	10.89	1.20	5032	10.89	0.30	5032	10.89	0.47
5	5032	12.23	1.24	4732	5.54	0.46	5032	12.23	1.31	4696	4.74	0.28	4732	5.54	0.43
6	11136	8.04	9.48	11136	8.04	1.23	11136	8.04	14.57	11136	8.04	1.12	11136	8.04	1.47
7	11136	8.03	29.06	11136	8.03	1.78	11136	8.03	30.26	11136	8.03	1.58	11136	8.03	1.95
8	11136	8.01	16.83	11136	8.01	1.67	11136	8.01	24.69	11136	8.01	1.50	11136	8.01	1.84
9	11136	7.92	25.47	11136	7.92	1.83	11136	7.92	26.60	11087	7.45	1.47	11136	7.92	1.77
10	25028	5.94	95.45	25028	5.94	7.71	25028	5.94	123.52	25028	5.94	8.30	25028	5.94	17.31
11	25028	5.78	84.77	25028	5.78	7.55	25028	5.78	105.51	25028	5.78	9.22	25028	5.78	13.16
12	25028	5.74	84.78	25028	5.74	7.63	25028	5.74	107.98	25028	5.74	11.09	25028	5.74	13.36
13	25028	5.69	71.36	25028	5.69	8.67	25028	5.69	98.78	25028	5.69	11.39	25028	5.69	10.97

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A												
1	5032	6.72	0.00	5032	6.72	0.00	5032	6.72	0.00	5032	6.72	0.00	5032	6.72	0.00
2	5032	6.72	0.00	5032	6.72	0.00	5032	6.72	0.00	5032	6.72	0.00	5032	6.72	0.00
3	5032	7.29	0.00	5032	7.29	0.00	5032	7.29	0.00	5032	7.29	0.00	5032	7.29	0.00
4	5032	7.29	0.00	5032	7.29	0.00	5032	7.29	0.00	5032	7.29	0.00	5032	7.29	0.00
5	4789	2.11	4.83	4696	0.13	0.76	5032	7.29	0.00	4696	0.13	0.00	4732	0.90	0.00
6	11136	6.66	0.00	11136	6.66	0.00	11136	6.66	0.00	11136	6.66	0.00	11136	6.66	0.00
7	11136	6.69	0.00	11136	6.69	0.00	11136	6.69	0.00	11136	6.69	0.00	11136	6.69	0.00
8	11136	6.69	0.00	11136	6.69	0.00	11136	6.69	0.00	11082	6.17	0.48	11136	6.69	0.00
9	11136	6.91	0.00	11136	6.91	0.00	11136	6.91	0.00	11052	6.11	0.32	11136	6.91	0.00
10	25028	3.96	0.00	25028	3.96	0.00	25028	3.96	0.00	25028	3.96	0.00	25028	3.96	0.00
11	25028	4.41	0.00	25028	4.41	0.00	25028	4.41	0.00	25028	4.41	0.00	25028	4.41	0.00
12	25028	4.83	0.00	25028	4.83	0.00	25028	4.83	0.00	25028	4.83	0.00	25028	4.83	0.00
13	25028	4.83	0.00	25028	4.83	0.00	25028	4.83	0.00	25028	4.83	0.00	25028	4.83	0.00

TABLEAU 4.4 – Résultats pour la nomenclature 6x6-1 avec l’algorithme descendant

de base qui sont complètement subordonnés aux autres composants. Par conséquent, le plan de production est très chargé sur peu de périodes afin de satisfaire les besoins des successeurs. Comme le volume de production est naturellement plus important pour les composants faisant partie du niveau le plus bas de la nomenclature, la capacité dans les premières périodes devient fortement violée. Il est ensuite très difficile de réparer les contraintes de capacité sans violer les contraintes de nomenclature. Deuxièmement, si l’on commence par satisfaire les contraintes de capacité, un plan respectant ces contraintes est construit, mais il devient problématique de trouver une solution respectant la nomenclature sans violer la capacité. La procédure est en effet très peu flexible, contrairement à l’algorithme glouton qui teste plusieurs mouvements.

Cet algorithme a été conçu pour privilégier la satisfaction des contraintes de nomenclature, ce qui explique la différence sur les temps de calcul. Les stratégies qui valident en premier les contraintes de nomenclature (SP2 et SP4) consomment beaucoup moins de temps que celles qui commencent par réparer les contraintes de capacité (SP1 et SP3).

Concernant le problème intégré, très peu d'améliorations sont effectuées par rapport au problème avec séquence fixée. Même si la séquence est changée, la logique descendante restreint l'application d'un lissage efficace. Une amélioration importante est néanmoins obtenue sur l'instance 5 avec SP1. SP4 s'avère être la stratégie la plus performante avec un écart moyen final de 5.52% sur la valeur de la solution optimale.

Comme l'algorithme descendant a un comportement similaire pour toutes les autres nomenclatures, nous ne présentons pas les résultats lui étant associé dans le reste de ce chapitre. Néanmoins, ils peuvent être consultés dans l'Annexe B.

Application de l'algorithme ascendant

Les résultats changent grandement lorsque les contraintes de nomenclature sont réparées au moyen de l'algorithme ascendant (logique opposée à MRP), comme nous pouvons l'observer dans le Tableau 4.5.

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)
1	4725	0.02	1.72	4724*	0.00	0.26	4832	2.29	0.28	4724*	0.00	0.25	4832	2.29	0.36
2	4715*	0.00	0.78	4715*	0.00	0.31	4715*	0.00	0.36	4715*	0.00	0.30	4715*	0.00	0.37
3	4690*	0.00	0.41	4690*	0.00	0.24	4690*	0.00	0.27	4690*	0.00	0.24	4690*	0.00	0.31
4	4690*	0.00	0.41	4690*	0.00	0.25	4690*	0.00	0.27	4690*	0.00	0.24	4690*	0.00	0.31
5	4690*	0.00	0.41	4690*	0.00	0.24	4690*	0.00	0.26	4690*	0.00	0.25	4690*	0.00	0.3
6	10637	0.82	5.44	10637	0.82	3.61	10646	0.91	0.76	10637	0.82	4.26	10646	0.91	1.20
7	10500	0.51	0.91	10519	0.69	0.98	10627	1.72	0.87	10519	0.69	0.90	10627	1.72	1.62
8	10490	0.50	0.56	10509	0.68	0.79	10490	0.50	0.72	10509	0.68	0.75	10509	0.68	0.85
9	10492	0.73	0.63	10509	0.89	0.81	10490	0.71	0.53	10492	0.73	0.77	10490	0.71	0.81
10	24394	1.17	14.26	24394	1.17	5.12	24394	1.17	6.94	24394	1.17	3.71	24394	1.17	15.32
11	24367	1.70	5.19	24310	1.46	5.16	24394	1.81	7.64	24310	1.46	3.76	24316	1.49	15.86
12	24308	1.67	5.00	24308	1.67	5.45	24308	1.67	6.16	24331	1.77	3.87	24307	1.67	13.68
13	24290	1.61	5.71	24288	1.61	5.82	24274	1.55	6.54	24293	1.63	4.23	24273	1.54	18.36

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A	UB	G	A	UB	G	A	UB	G	A	UB	G	A
1	4720	0.11	0.11	4719	0.08	0.11	4832	2.48	0.00	4724	0.19	0.00	4832	2.48	0.00
2	4715*	0.00	0.00	4715*	0.00	0.00	4715*	0.00	0.00	4715*	0.00	0.00	4715*	0.00	0.00
3	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00
4	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00
5	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00	4690*	0.00	0.00
6	10557	1.11	0.75	10532	0.87	0.99	10646	1.96	0.00	10637	1.96	0.00	10646	1.96	0.00
7	10490	0.50	0.10	10490	0.50	0.28	10490	0.50	1.29	10519	1.99	0.00	10490	0.50	1.29
8	10490	0.50	0.00	10490	0.50	0.18	10490	0.50	0.00	10438*	0.00	0.68	10490	0.50	0.18
9	10490	0.71	0.02	10490	0.71	0.18	10490	0.71	0.00	10492	0.74	0.00	10490	0.71	0.00
10	24388	1.30	0.02	24388	1.30	0.02	24394	1.33	0.00	24394	1.33	0.00	24394	1.33	0.00
11	24310	1.42	0.23	24306	1.40	0.02	24394	1.77	0.00	24310	1.69	0.00	24312	1.43	0.02
12	24308	1.82	0.00	24269	1.65	0.16	24307	1.81	0.00	24331	2.18	0.00	24285	1.72	0.09
13	24278	1.69	0.05	24268	1.65	0.08	24268	1.65	0.02	24293	2.18	0.00	24268	1.65	0.02

TABLEAU 4.5 – Résultats pour la nomenclature 6x6-1 avec l'algorithme ascendant

Cette méthode permet de répartir la production des produits des niveaux les plus bas tout au long de l'horizon de planification, sans être contraint par les tailles de lots des successeurs, ce qui fait que la charge de travail est mieux équilibrée. En effet, en tenant compte du fait que la plupart des unités produites correspondent aux composants basiques, il convient de décider de leurs tailles de lots et de leurs périodes de production avant que les décisions soient très contraintes par la nomenclature et que l'on soit obligé de les regrouper dans peu de périodes. Un autre avantage est que les temps de calculs sont très faibles.

La qualité des solutions est légèrement détériorée par rapport à l'algorithme glouton pour les instances avec 10 et 20 périodes. Néanmoins, les temps de calcul moyens sont fortement réduits (entre 65.47% et 71.07% de réduction). La stratégie de lissage proposant la meilleure qualité de solution est SP1, avec $\bar{G} = 0.67\%$. SP2 et SP4 offrent tout de même une bonne performance avec $\bar{G} = 0.69\%$, cette dernière garantissant la meilleure performance par rapport au temps de calcul avec $\bar{t} = 1.81$ secondes.

Pour ce qui est de la condition d'autorisation ou d'interdiction de violation des contraintes de capacité durant la réparation des contraintes de nomenclature (SP2 et SP4), il n'y a quasiment pas de différence sur la qualité des solutions. En revanche, si l'on souhaite satisfaire en premier les contraintes de capacité, il convient d'autoriser la violation des contraintes de nomenclature. Concernant le temps de calcul, il vaut mieux autoriser la violation du deuxième type de contraintes dans la première étape de validation, c'est-à-dire privilégier SP3 et SP4 au lieu de SP1 et SP2 respectivement.

Concernant la résolution du problème intégré, c'est la stratégie SP2 qui offre globalement la meilleure performance avec $\bar{G} = 0.67\%$, contre un écart moyen de 0.70% pour SP1.

4.4.1.2 Nomenclature 6x6-2

Considérons maintenant la nomenclature 6x6-2 (Figure 4.12), où nous avons 3 niveaux et 2 produits par niveau. Chaque produit est composant des deux produits du niveau immédiatement supérieur. Il y a donc deux produits sans prédécesseurs et deux produits sans successeurs. Les solutions obtenues par le solveur commercial sont affichées dans le Tableau 4.6. Le problème à séquence fixée est résolu de façon optimale sur toutes les instances ; tandis que le problème intégré est résolu de façon optimale uniquement sur les instances avec 5 et 10 périodes.

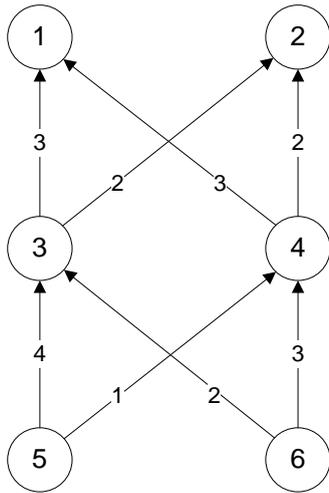


FIGURE 4.12 – Nomenclature 6x6-2

Instance	cap	T	ALB	CPL1	CPL2
1	0.60	5	10802	11884*	11884*
2	0.65			11884*	11884*
3	0.70			11884*	11884*
4	0.75			11884*	11884*
5	0.80			11884*	11884*
6	0.60	10	1913	29370*	29370*
7	0.65			29370*	29370*
8	0.70			29370*	29331*
9	0.85			29312*	29294*
10	0.80			29275*	29263*
11	0.60	20	14812	66553*	66548
12	0.65			66548*	66548
13	0.70			66536*	67073
14	0.75			66495*	66556
15	0.80			66458*	66537

TABLEAU 4.6 – Solutions obtenues avec IBM ILOG CPLEX pour la nomenclature 6x6-2

Application de l’algorithme glouton

Les résultats associés à l’algorithme glouton sont présentés dans le Tableau 4.7. Notre approche présente une très bonne performance avec les procédures qui satisfont en premier les contraintes de nomenclature (SP2, SP4, et SP5), avec $\bar{G} = 0.37\%$ pour SP2 et $\bar{G} = 0.06\%$ pour SP4 et SP5. Avec ces deux dernières stratégies, notre approche résout de façon optimale toutes les instances avec $T = 5$, ainsi que trois instances avec $T = 10$ et une instance avec $T = 20$. Les autres solutions restent très proches de l’optimum avec un écart type moyen de 0.1%. Les stratégies SP1 et SP3, qui satisfont en premier la capacité, offrent une performance un peu moins intéressante, les écarts moyens étant de 1.35% et de 1.22% respectivement. Ces deux stratégies ont des problèmes notamment sur les instances avec $T = 5$. Le temps de calcul est très variable entre toutes les stratégies, mais en moyenne SP1 est la plus rapide avec $\bar{t} = 6.11$ secondes.

Les écarts moyens par rapport aux solutions obtenues par le solveur sont encore meilleurs pour le problème intégré. De plus, les instances 13 et 14 sont mieux résolues avec notre approche qu’avec le solveur. SP4 et SP5 restent les meilleures stratégies avec un écart moyen final de 0.01%.

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)												
1	12268	3.23	0.82	11884*	0.00	23.28	12268	3.23	0.84	11884*	0.00	19.93	11884*	0.00	0.84
2	12268	3.23	1.32	11884*	0.00	20.46	12268	3.23	1.00	11884*	0.00	18.81	11884*	0.00	1.08
3	11998	0.96	1.03	11998	0.96	19.94	11998	0.96	1.02	11884*	0.00	18.65	11884*	0.00	1.14
4	12268	3.23	1.33	11884*	0.00	21.28	12268	3.23	1.03	11884*	0.00	18.94	11884*	0.00	1.15
5	12132	2.09	1.65	11884*	0.00	20.37	12268	3.23	1.06	11884*	0.00	19.47	11884*	0.00	1.26
6	29370*	0.00	22.23	29370*	0.00	4.41	29370*	0.00	41.07	29370*	0.00	4.03	29370*	0.00	16.37
7	29708	1.15	32.54	29380	0.03	3.32	29398	0.10	14.91	29370*	0.00	2.68	29370*	0.00	14.79
8	29708	1.15	2.53	29370*	0.00	3.04	29581	0.72	5.77	29370*	0.00	2.03	29370*	0.00	8.95
9	29489	0.60	5.62	29370	0.20	2.94	29489	0.60	2.80	29370	0.20	2.32	29370	0.20	5.83
10	29708	1.48	5.24	29370	0.32	2.89	29561	0.98	7.28	29370	0.32	3.93	29370	0.32	5.62
10	66621	0.10	1.42	67227	1.01	3.48	66621	0.10	1.41	66621	0.10	0.93	66621	0.10	0.84
11	67024	0.72	8.95	67024	0.72	14.16	67020	0.71	23.76	66548*	0.00	14.37	66548*	0.00	35.46
12	67024	0.73	2.79	67024	0.73	9.55	66740	0.31	23.00	66548	0.02	5.36	66548	0.02	12.54
13	67024	0.80	2.23	67024	0.80	5.28	66820	0.49	13.95	66548	0.08	6.68	66548	0.08	13.03
14	67024	0.85	2.01	67024	0.85	7.34	66765	0.46	8.59	66548	0.14	9.89	66548	0.14	18.97

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A												
1	11884*	0.00	3.13	11884*	0.00	0.00	11884*	0.00	3.13	11884*	0.00	0.00	11884*	0.00	0.00
2	12268	3.23	0.00	11884*	0.00	0.00	12268	3.23	0.00	11884*	0.00	0.00	11884*	0.00	0.00
3	11998	0.96	0.00	11998	0.96	0.00	11998	0.96	0.00	11884*	0.00	0.00	11884*	0.00	0.00
4	12268	3.23	0.00	11884*	0.00	0.00	12268	3.23	0.00	11884*	0.00	0.00	11884*	0.00	0.00
5	12132	2.09	0.00	11884*	0.00	0.00	12268	3.23	0.00	11884*	0.00	0.00	11884*	0.00	0.00
6	29370*	0.00	0.00	29370*	0.00	0.00	29370*	0.00	0.00	29370*	0.00	0.00	29370*	0.00	0.00
7	29708	1.15	0.00	29380	0.03	0.00	29398	0.10	0.00	29370*	0.00	0.00	29370*	0.00	0.00
8	29489	0.54	0.74	29370	0.13	0.00	29421	0.31	0.54	29370	0.13	0.00	29370	0.13	0.00
9	29489	0.67	0.00	29370	0.26	0.00	29489	0.67	0.00	29370	0.26	0.00	29370	0.26	0.00
10	29708	1.52	0.00	29370	0.37	0.00	29557	1.00	0.01	29370	0.37	0.00	29370	0.37	0.00
11	66621	0.11	0.00	67227	1.02	0.00	66621	0.11	0.00	66621	0.11	0.00	66621	0.11	0.00
12	67024	0.72	0.00	66798	0.38	0.34	66827	0.42	0.29	66548*	0.00	0.00	66548*	0.00	0.00
13	67024	-0.07	0.00	67024	-0.07	0.00	66740	-0.50	0.00	66548	-0.78	0.00	66548	-0.78	0.00
14	67024	0.70	0.00	67024	0.70	0.00	66740	0.28	0.12	66548	-0.01	0.00	66548	-0.01	0.00
15	67024	0.73	0.00	66958	0.63	0.10	66753	0.32	0.02	66548	0.02	0.00	66548	0.02	0.00

TABLEAU 4.7 – Résultats pour la nomenclature 6x6-2 avec l’algorithme glouton

Algorithme ascendant

Les résultats associés à l’algorithme ascendant sont présentés dans le Tableau 4.8. Cette méthode offre la même performance en qualité de solution que l’algorithme glouton pour SP4 et SP5, mais améliore le temps de calcul. De plus, les solutions avec SP1, SP2 et SP3 sont améliorées, tout comme les temps de calcul.

En ce qui concerne le problème intégré, les écarts moyens avec IBM ILOG CPLEX diminuent, SP4 et SP5 étant à nouveau les stratégies les plus performantes, avec un écart moyen de 0.01%, et permettant d’obtenir 8 solutions optimales. Comme avec l’algorithme glouton, les solutions correspondant aux instances 13 et 14 sont meilleures que celles du solveur.

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)
1	12020	1.14	0.84	11884*	0.00	7.24	12268	3.23	0.78	11884*	0.00	7.79	11884*	0.00	0.69
2	12223	2.85	1.19	11884*	0.00	6.63	12268	3.23	0.80	11884*	0.00	7.06	11884*	0.00	0.98
3	11998	0.96	0.81	11998	0.96	6.55	11998	0.96	0.80	11884*	0.00	7.02	11884*	0.00	1.00
4	12268	3.23	1.21	11884*	0.00	6.63	12268	3.23	0.78	11884*	0.00	7.11	11884*	0.00	1.01
5	12132	2.09	1.60	11884*	0.00	6.69	12268	3.23	0.80	11884*	0.00	7.05	11884*	0.00	1.05
6	29370*	0.00	15.25	29370*	0.00	1.56	29370*	0.00	13.49	29370*	0.00	2.82	29370*	0.00	5.75
7	29708	1.15	7.62	29380	0.03	1.56	29398	0.10	7.52	29370*	0.00	1.60	29370*	0.00	6.96
8	29708	1.15	2.74	29370*	0.00	1.97	29581	0.72	2.26	29370*	0.00	1.31	29370*	0.00	5.01
9	29489	0.60	6.65	29370	0.20	1.50	29489	0.60	1.86	29370	0.20	1.44	29370	0.20	4.18
10	29708	1.48	6.40	29370	0.32	1.42	29561	0.98	4.60	29370	0.32	1.37	29370	0.32	3.24
11	66621	0.10	1.64	67699	1.72	1.69	66621	0.10	1.62	66621	0.10	0.78	66621	0.10	0.92
12	67024	0.72	7.27	66732	0.28	4.38	66786	0.36	19.72	66548*	0.00	7.07	66548*	0.00	13.19
13	67024	0.73	4.41	66685	0.22	5.41	66754	0.33	7.36	66548	0.02	7.28	66548	0.02	6.57
14	67024	0.80	4.10	66623	0.19	4.04	66738	0.37	6.71	66548	0.08	6.81	66548	0.08	11.78
15	67024	0.85	2.60	67024	0.85	1.53	66941	0.73	2.75	66548	0.14	2.66	66548	0.14	6.76

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A												
1	11884*	0.00	1.13	11884*	0.00	0.00	11884*	0.00	3.13	11884*	0.00	0.00	11884*	0.00	0.00
2	12223	2.85	0.00	11884*	0.00	0.00	12268	3.23	0.00	11884*	0.00	0.00	11884*	0.00	0.00
3	11998	0.96	0.00	11998	0.96	0.00	11998	0.96	0.00	11884*	0.00	0.00	11884*	0.00	0.00
4	12268	3.23	0.00	11884*	0.00	0.00	12268	3.23	0.00	11884*	0.00	0.00	11884*	0.00	0.00
5	12132	2.09	0.00	11884*	0.00	0.00	12268	3.23	0.00	11884*	0.00	0.00	11884*	0.00	0.00
6	29370*	0.00	0.00	29370*	0.00	0.00	29370*	0.00	0.00	29370*	0.00	0.00	29370*	0.00	0.00
7	29708	1.15	0.00	29380	0.03	0.00	29398	0.10	0.00	29370*	0.00	0.00	29370*	0.00	0.00
8	29489	0.54	0.74	29370	0.13	0.00	29416	0.29	0.56	29370	0.13	0.00	29370	0.13	0.00
9	29489	0.67	0.00	29370	0.26	0.00	29489	0.67	0.00	29370	0.26	0.00	29370	0.26	0.00
10	29708	1.52	0.00	29370	0.37	0.00	29525	0.90	0.12	29370	0.37	0.00	29370	0.37	0.00
11	66621	0.11	0.00	67227	1.02	0.70	66621	0.11	0.00	66621	0.11	0.00	66621	0.11	0.00
12	67024	0.72	0.00	66685	0.21	0.07	66716	0.25	0.10	66548*	0.00	0.00	66548	0.00	0.00
13	67024	-0.07	0.00	66609	-0.69	0.11	66754	-0.48	0.00	66548	-0.78	0.00	66548	-0.78	0.00
14	67024	0.70	0.00	66563	0.01	0.09	66680	0.19	0.09	66548	-0.01	0.00	66548	-0.01	0.00
15	67024	0.73	0.00	66685	0.22	0.51	66750	0.32	0.29	66548	0.02	0.00	66548	0.02	0.00

TABLEAU 4.8 – Résultats pour la nomenclature 6x6-2 avec l’algorithme ascendant

4.4.1.3 Nomenclature 6x6-3

Pour ce qui est de la nomenclature 6x6-3 (Figure 4.13), nous avons 2 niveaux et 6 produits, avec 3 produits par niveau. Chaque produit du niveau inférieur est composant de tous les produits du niveau supérieur. Les solutions obtenues par le solveur commercial sont présentées dans le Tableau 4.9. La solution optimale est trouvée dans tous les cas, sauf pour les instances 11 et 12 associées au problème intégré.

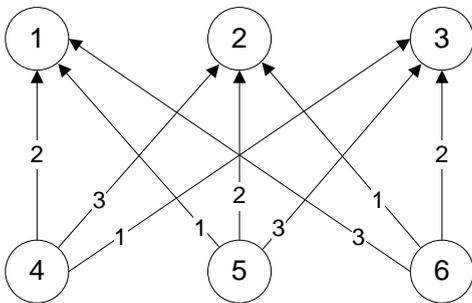


FIGURE 4.13 – Nomenclature 6x6-3

Instance	cap	T	ALB	CPL1	CPL2
1	0.60	5	6311	6437*	6335*
2	0.65			6385*	6335*
3	0.70			6351*	6335*
4	0.75			6335*	6335*
5	0.80			6335*	6335*
6	0.60	10	13510	13891*	13685*
7	0.65			13800*	13585*
8	0.70			13731*	13548*
9	0.85			13665*	13546*
10	0.80			13615*	13546*
11	0.60	20	28301	28979*	28722
12	0.65			28760*	28465
13	0.70			28582*	28373*
14	0.75			28544*	28365*
15	0.80			28509*	28365*

TABLEAU 4.9 – Solutions obtenues avec IBM ILOG CPLEX pour la nomenclature 6x6-3

Application de l'algorithme glouton

Les résultats des expérimentations avec l'algorithme glouton sont présentés dans le Tableau 4.10. Notre approche propose de bonnes solutions pour toutes les instances, mais les écarts avec les solutions optimales sont un peu plus élevés que pour les nomenclatures précédentes. En qualité de solution, ce sont les stratégies validant en premier les contraintes de nomenclature qui offrent une meilleure performance, avec $\bar{G} = 1.15\%$ pour SP2 et $\bar{G} = 1.12\%$ pour SP4. Cependant, le temps de calcul est en moyenne 65% plus petit avec SP1, qui présente un écart moyen sur la solution optimale de 1.45%. La stratégie de lissage SP5 est celle qui consomme le plus de temps de calcul (13.35 secondes par instance en moyenne), et ses solutions sont globalement meilleures que celles de SP1 et SP3.

Le fait de commencer par satisfaire les contraintes de nomenclature prime sur le fait d'autoriser ou d'interdire la violation des contraintes de capacité. C'est pourquoi il n'y a pas une différence significative entre SP2 et SP4. En revanche, si l'on commence par valider les contraintes de capacité, l'autorisation ou l'interdiction de violer les contraintes de nomenclature devient important. Il est en fait préférable d'interdire la violation de ces contraintes, afin de faciliter leur validation postérieure. C'est pour cela que SP1 est plus performante que SP3, aussi bien en qualité de solution ($\Delta\bar{G} = 7.64\%$) qu'en temps de calcul ($\Delta\bar{t} = 51.29\%$).

Concernant la résolution du problème intégré, les plus grandes améliorations (1.08% en moyenne) sont obtenues avec SP1, grâce au faible temps de calcul passé sur chaque séquence,

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)												
1	6523	1.34	0.95	6481	0.68	12.22	6491	0.84	0.71	6481	0.68	11.94	6489	0.81	1.17
2	6462	1.21	0.60	6434	0.77	10.33	6464	1.24	0.60	6434	0.77	10.11	6443	0.91	0.70
3	6355	0.06	0.58	6354	0.05	11.09	6464	1.78	0.58	6354	0.05	10.82	6440	1.40	0.62
4	6450	1.82	0.50	6424	1.40	11.33	6464	2.04	0.50	6424	1.40	11.06	6424	1.40	0.47
5	6424	1.40	0.59	6424	1.40	9.43	6424	1.40	0.57	6424	1.40	9.23	6424	1.40	0.53
6	14083	1.38	8.28	14065	1.25	49.18	14095	1.47	1.85	14065	1.25	44.67	14101	1.51	2.96
7	14005	1.49	2.93	14011	1.53	14.74	14017	1.57	1.03	14011	1.53	12.49	13963	1.18	1.89
8	13975	1.78	0.94	13976	1.78	7.12	13998	1.94	0.57	13976	1.78	6.64	13962	1.68	1.26
9	13877	1.55	0.38	13841	1.29	5.90	13841	1.29	1.49	13841	1.29	5.31	13818	1.12	2.00
10	13850	1.73	0.34	13840	1.65	19.94	13841	1.66	0.43	13840	1.65	19.06	13840	1.65	0.53
11	30144	4.02	18.84	29438	1.58	20.03	29563	2.02	69.72	29303	1.12	30.38	29489	1.76	104.13
12	29181	1.46	19.68	29175	1.44	15.49	29307	1.90	27.20	29175	1.44	18.03	29209	1.56	43.12
13	28813	0.81	7.24	28878	1.04	5.14	29113	1.86	22.06	28878	1.04	6.64	29064	1.69	26.24
14	28826	0.99	3.05	28729	0.65	1.91	28977	1.52	9.00	28729	0.65	2.34	28851	1.08	27.05
15	28700	0.67	5.74	28699	0.67	4.25	28814	1.07	8.75	28699	0.67	5.21	28814	1.07	17.57

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A												
1	6345	0.16	2.73	6393	0.92	1.36	6463	2.02	0.43	6393	0.92	1.36	6460	1.97	0.45
2	6388	0.84	1.15	6359	0.38	1.17	6424	1.40	0.62	6359	0.38	1.17	6443	1.70	0.00
3	6355	0.32	0.00	6354	0.30	0.00	6424	1.40	0.62	6354	0.30	0.00	6424	1.40	0.25
4	6355	0.32	1.47	6354	0.30	1.09	6424	1.40	0.62	6354	0.30	1.09	6424	1.40	0.00
5	6424	1.40	0.00	6424	1.40	0.00	6424	1.40	0.00	6424	1.40	0.00	6424	1.40	0.00
6	13985	2.19	0.70	13890	1.50	1.24	13948	1.92	1.04	14065	2.78	0.00	14101	3.04	0.00
7	13766	1.33	1.71	13789	1.50	1.58	13749	1.21	1.91	13805	1.62	1.47	13805	1.62	1.13
8	13662	0.84	2.24	13675	0.94	2.15	13675	0.94	2.31	13689	1.04	2.05	13651	0.76	2.23
9	13639	0.69	1.72	13639	0.69	1.46	13639	0.69	1.46	13639	0.69	1.46	13639	0.69	1.30
10	13639	0.69	1.52	13639	0.69	1.45	13639	0.69	1.46	13639	0.69	1.45	13639	0.69	1.45
11	29474	2.62	2.22	29265	1.89	0.59	29563	2.93	0.00	29303	2.02	0.00	29489	2.67	0.00
12	29172	2.48	0.03	29167	2.47	0.03	29306	2.95	0.00	29060	2.09	0.39	29209	2.61	0.00
13	28784	1.45	0.10	28777	1.42	0.35	28935	1.98	0.61	28777	1.42	0.35	29064	2.44	0.00
14	28684	1.12	0.49	28702	1.19	0.09	28834	1.65	0.49	28659	1.04	0.24	28851	1.71	0.00
15	28679	1.11	0.07	28596	0.81	0.36	28814	1.58	0.00	28596	0.81	0.36	28642	0.98	0.60

TABLEAU 4.10 – Résultats pour la nomenclature 6x6-3 avec l’algorithme glouton

et SP2 devient la stratégie la plus performante avec $\bar{G} = 1.09\%$. Les améliorations moins significatives sont obtenues avec SP5, à cause d’un temps de calcul élevé pour résoudre le problème avec séquence fixée.

Application de l’algorithme ascendant

Les résultats obtenus avec l’algorithme ascendant sont présentés dans le Tableau 4.11. La méthode offre une très bonne performance par rapport aux deux autres méthodes. En terme de qualité de solution, elle est un peu moins compétitive que l’algorithme glouton, les écarts moyens étant légèrement détériorés (entre 1.91% et 8.93% d’augmentation par rapport aux écarts de l’algorithme glouton). En temps de calcul, l’algorithme ascendant est significativement plus

rapide avec toutes les stratégies de lissage (entre 30.62% et 88.19% de réduction), excepté avec SP1, où le temps moyen augmente de 25.90%.

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)
1	6523	1.34	1.24	6489	0.81	0.45	6491	0.84	0.70	6489	0.81	0.41	6489	0.81	0.99
2	6462	1.21	0.72	6462	1.21	0.36	6464	1.24	0.66	6462	1.21	0.36	6443	0.91	0.66
3	6401	0.79	0.67	6400	0.77	0.31	6464	1.78	0.67	6400	0.77	0.32	6440	1.40	0.61
4	6450	1.82	0.65	6424	1.40	0.27	6464	2.04	0.61	6424	1.40	0.31	6424	1.40	0.49
5	6424	1.40	0.79	6424	1.40	0.25	6424	1.40	0.75	6424	1.40	0.28	6424	1.40	0.55
6	14083	1.38	11.79	14040	1.07	5.79	14095	1.47	1.70	14040	1.07	4.84	14101	1.51	3.53
7	14005	1.49	1.27	13967	1.21	1.23	14017	1.57	1.06	13967	1.21	1.15	13963	1.18	1.97
8	13975	1.78	1.08	13976	1.78	1.26	13998	1.94	0.73	13976	1.78	1.18	13962	1.68	1.13
9	13877	1.55	0.68	13841	1.29	1.05	13841	1.29	1.23	13841	1.29	1.03	13818	1.12	1.66
10	13850	1.73	0.62	13840	1.65	0.97	13841	1.66	0.58	13840	1.65	0.97	13840	1.65	0.48
11	30144	4.02	26.42	29445	1.61	4.66	29489	1.76	38.56	29423	1.53	5.51	29489	1.76	96.30
12	29181	1.46	21.26	29175	1.44	3.38	29307	1.90	14.18	29175	1.44	5.71	29205	1.55	26.33
13	28815	0.82	12.65	28852	0.94	1.59	29184	2.11	11.12	28852	0.94	2.37	29139	1.95	13.57
14	28778	0.82	4.04	28769	0.79	0.91	29025	1.69	7.11	28769	0.79	1.40	28946	1.41	5.20
15	28720	0.74	5.13	28774	0.93	0.97	28903	1.38	3.45	28774	0.93	1.47	28889	1.33	6.16

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A	UB	A										
1	6391	0.88	2.02	6393	0.92	1.48	6463	2.02	0.43	6450	1.82	0.60	6460	1.97	0.45
2	6424	1.40	0.59	6440	1.66	0.34	6424	1.40	0.62	6404	1.09	0.90	6424	1.40	0.29
3	6401	1.04	0.00	6400	1.03	0.00	6424	1.40	0.62	6400	1.03	0.00	6424	1.40	0.25
4	6424	1.40	0.40	6424	1.40	0.00	6424	1.40	0.62	6381	0.73	0.67	6424	1.40	0.00
5	6424	1.40	0.00	6424	1.40	0.00	6424	1.40	0.00	6381	0.73	0.67	6424	1.40	0.00
6	14083	2.91	0.00	14040	2.78	0.00	14095	3.00	0.00	13984	2.18	0.40	14101	3.04	0.00
7	13855	1.99	1.07	13789	1.50	1.27	14003	3.08	0.10	13961	2.77	0.04	13786	1.48	1.27
8	13841	2.16	0.96	13675	0.94	2.15	13824	2.04	1.24	13946	2.94	0.21	13683	1.00	2.00
9	13825	2.06	0.37	13639	0.69	1.46	13793	1.82	0.35	13841	2.18	0.00	13679	0.98	1.01
10	13775	1.69	0.54	13639	0.69	1.45	13678	0.97	1.18	13770	1.65	0.51	13639	0.69	1.45
11	29459	2.57	2.27	29442	2.51	0.01	29489	2.67	0.00	29367	2.25	0.19	29487	2.66	0.01
12	29127	2.33	0.19	29035	2.00	0.48	29252	2.76	0.19	29118	2.29	0.20	29205	2.60	0.00
13	28766	1.39	0.17	28761	1.37	0.32	29033	2.33	0.52	28852	1.93	0.00	29064	2.44	0.26
14	28732	1.29	0.16	28729	1.28	0.14	28987	2.19	0.13	28769	1.93	0.00	28900	1.89	0.16
15	28706	1.20	0.05	28630	0.93	0.50	28711	1.22	0.66	28774	1.46	0.00	28671	1.08	0.75

TABLEAU 4.11 – Résultats pour la nomenclature 6x6-3 avec l’algorithme ascendant

Il n’y a pas de différence significative lorsque l’on autorise ou interdit la violation des contraintes de capacité pour satisfaire celles de nomenclature, ni sur la qualité de solution ni sur le temps de calcul. SP2 et SP4 proposent toutes les deux un écart moyen de 1.22% et des temps moyens de calcul de 1.56 secondes et 1.82 secondes respectivement. En revanche, lorsque l’on valide les contraintes de nomenclature, il est préférable d’interdire la violation des contraintes de nomenclature, SP1 présentant un écart moyen de 1.49% et celui de SP3 étant de 1.60%. SP5 est la stratégie qui consomme le plus de temps avec $\bar{t} = 10.64$ secondes.

Pour ce qui est du problème intégré, la stratégie SP2 est la plus performante et celle qui génère les améliorations les plus significatives, avec $\bar{G} = 1.41\%$ et $\bar{A} = 0.64\%$. L’avantage par

rapport à SP4 est dû à une résolution plus rapide du problème à séquence fixée, permettant d'explorer plus de séquences.

4.4.1.4 Nomenclature 6x6-4

Nous nous intéressons maintenant à un cas d'étude un peu différent. C'est la nomenclature 6x6-4 (Figure 4.14), qui comporte deux produits finis, chacun avec uniquement deux prédécesseurs différents, sans liens entre eux. La nomenclature peut donc être divisée en deux sous-nomenclatures. Comme pour les deux cas d'études précédents, le solveur commercial détermine les solutions optimales pour toutes les instances, excepté celles du problème intégré avec 20 périodes, comme détaillé dans le Tableau 4.12.

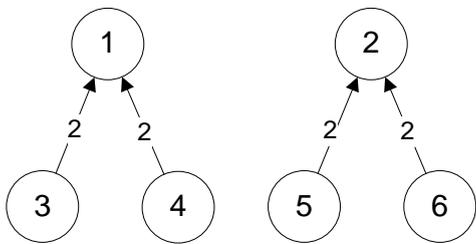


FIGURE 4.14 – Nomenclature 6x6-4

Instance	cap	T	ALB	CPL1	CPL2
1	0.60	5	3349	3565*	3523*
2	0.65			3544*	3523*
3	0.70			3527*	3523*
4	0.75			3523*	3523*
5	0.80			3523*	3523*
6	0.60	10	6497	7497*	7437*
7	0.65			7455*	7427*
8	0.70			7446*	7407*
9	0.85			7401*	7357*
10	0.80			7357*	7357*
11	0.60	20	12867	16108*	16139
12	0.65			16064*	16133
13	0.70			16058*	16107
14	0.75			16058*	16096
15	0.80			16042*	16122

TABLEAU 4.12 – Solutions obtenues avec IBM ILOG CPLEX pour la nomenclature 6x6-4

Application de l'algorithme glouton

Les solutions obtenues avec l'algorithme glouton sont affichées dans le Tableau 4.13. Toutes les stratégies offrent une performance similaire, avec un léger avantage pour SP4, qui présente un écart moyen sur la solution optimale de 1.21% et un temps de calcul moyen de 3.74 secondes. La stratégie la moins compétitive est SP3, avec $\bar{G} = 1.63\%$ et $\bar{t} = 4.89$ secondes.

La stratégie SP2 devient la plus performante lorsque l'on résout le problème intégré, avec un écart moyen de 0.93%. Cependant, SP4 offre aussi une bonne performance, avec $\bar{G} = 0.97\%$. Les plus grandes améliorations de la borne supérieure sont obtenues avec SP3, avec un écart

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)
1	3586	0.59	0.81	3607	1.18	6.16	3639	2.08	0.52	3607	1.18	6.02	3586	0.59	0.67
2	3561	0.48	0.59	3561	0.48	3.85	3586	1.19	0.66	3561	0.48	3.82	3584	1.13	0.67
3	3551	0.68	0.53	3551	0.68	3.00	3551	0.68	0.58	3551	0.68	2.96	3551	0.68	0.56
4	3551	0.79	0.58	3551	0.79	2.62	3551	0.79	0.64	3551	0.79	2.59	3551	0.79	1.00
5	3551	0.79	0.63	3551	0.79	2.75	3551	0.79	0.69	3551	0.79	2.71	3551	0.79	0.78
6	7635	2.09	0.64	7603	1.66	1.51	7727	3.32	1.90	7603	1.66	1.37	7603	1.66	1.79
7	7656	2.70	0.76	7589	1.80	1.47	7609	2.07	1.64	7589	1.80	1.35	7592	1.84	1.19
8	7588	1.91	1.36	7580	1.80	1.61	7611	2.22	1.10	7580	1.80	1.44	7587	1.89	1.26
9	7506	1.42	0.62	7513	1.51	1.53	7559	2.13	0.86	7513	1.51	1.41	7544	1.93	1.00
10	7463	1.44	0.45	7463	1.44	1.33	7463	1.44	0.55	7463	1.44	1.25	7463	1.44	0.67
11	16417	1.92	11.94	16344	1.47	8.06	16392	1.76	20.19	16354	1.53	7.37	16331	1.38	17.81
12	16252	1.17	10.92	16300	1.47	7.38	16333	1.67	13.20	16224	1.00	6.01	16224	1.00	15.23
13	16243	1.15	8.90	16243	1.15	6.60	16313	1.59	9.94	16293	1.46	5.67	16293	1.46	11.06
14	16239	1.13	9.08	16239	1.13	7.12	16287	1.43	11.75	16233	1.09	5.38	16233	1.09	11.17
15	16186	0.90	8.77	16261	1.37	6.70	16239	1.23	9.08	16191	0.93	6.82	16185	0.89	9.19

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A	UB	G	A	UB	G	A	UB	G	A	UB	G	A
1	3551	0.79	0.98	3551	0.79	1.55	3551	0.79	2.42	3551	0.79	1.55	3551	0.79	0.98
2	3551	0.79	0.28	3551	0.79	0.28	3551	0.79	0.98	3551	0.79	0.28	3551	0.79	0.92
3	3551	0.79	0.00	3551	0.79	0.00	3551	0.79	0.00	3551	0.79	0.00	3551	0.79	0.00
4	3551	0.79	0.00	3538	0.43	0.37	3551	0.79	0.00	3538	0.43	0.37	3551	0.79	0.00
5	3551	0.79	0.00	3538	0.43	0.37	3551	0.79	0.00	3538	0.43	0.37	3551	0.79	0.00
6	7598	2.16	0.48	7581	1.94	0.29	7610	2.33	1.51	7581	1.94	0.29	7584	1.98	0.25
7	7596	2.28	0.78	7544	1.58	0.59	7559	1.78	0.66	7544	1.58	0.59	7559	1.78	0.43
8	7544	1.85	0.58	7513	1.43	0.88	7588	2.44	0.30	7513	1.43	0.88	7559	2.05	0.37
9	7463	1.44	0.57	7463	1.44	0.67	7463	1.44	1.27	7463	1.44	0.67	7463	1.44	1.07
10	7463	1.44	0.00	7463	1.44	0.00	7463	1.44	0.00	7463	1.44	0.00	7463	1.44	0.00
11	16286	0.91	0.80	16252	0.70	0.56	16294	0.96	0.60	16280	0.87	0.45	16322	1.13	0.06
12	16233	0.62	0.12	16224	0.56	0.47	16233	0.62	0.61	16224	0.56	0.00	16224	0.56	0.00
13	16233	0.78	0.06	16233	0.78	0.06	16213	0.66	0.61	16233	0.78	0.37	16279	1.07	0.09
14	16186	0.56	0.33	16233	0.85	0.04	16233	0.85	0.33	16233	0.85	0.00	16233	0.85	0.00
15	16116	-0.04	0.43	16116	-0.04	0.89	16116	-0.04	0.76	16185	0.39	0.04	16185	0.39	0.00

TABLEAU 4.13 – Résultats pour la nomenclature 6x6-4 avec l’algorithme glouton

moyen de 1.10%. On notera aussi que la dernière instance est mieux résolue avec SP1, SP2 et SP3 qu’avec le solveur commercial.

Application de l’algorithme ascendant

Enfin, les expérimentations avec l’algorithme ascendant sont présentées dans le Tableau 4.14. Cette fois-ci la différence avec l’algorithme glouton est un peu plus significative. La qualité de la solution est détériorée avec des écarts moyens par rapport à l’optimum variant entre 2.61% et 2.73%, ce qui reste raisonnable, mais représente une augmentation entre 64.42% et 125.62% par rapport aux écarts moyens de l’algorithme glouton. En ce qui concerne le temps de calcul, une réduction importante est obtenue avec SP2, SP3 et SP4 (66.91%, 27.40% et

50.27% respectivement), et une augmentation est observée avec SP1 et SP5 (18.57% et 10.12% respectivement). La meilleure borne supérieure est obtenue en moyenne avec SP5, mais elle présente le temps de calcul moyen le plus élevé, à cause des instances avec 20 périodes. Le meilleur rapport entre qualité de solution et temps de calcul correspond à SP2.

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)												
1	3586	0.59	1.69	3586	0.59	0.37	3633	1.91	1.01	3586	0.59	0.35	3586	0.59	0.69
2	3586	1.19	1.46	3628	2.37	0.36	3586	1.19	1.64	3628	2.37	0.39	3628	2.37	0.68
3	3586	1.67	1.11	3617	2.55	0.37	3586	1.67	1.06	3617	2.55	0.39	3586	1.67	0.74
4	3616	2.64	3.86	3568	1.28	0.37	3568	1.28	3.74	3568	1.28	0.41	3568	1.28	0.69
5	3568	1.28	3.26	3568	1.28	0.45	3568	1.28	3.15	3568	1.28	0.49	3568	1.28	0.94
6	7711	3.10	1.57	7717	3.18	1.22	7770	3.89	2.09	7717	3.18	1.41	7717	3.18	1.86
7	7620	2.21	4.74	7733	3.73	1.08	7711	3.43	1.06	7732	3.72	1.21	7725	3.62	2.93
8	7705	3.48	1.21	7644	2.66	1.05	7687	3.24	0.94	7644	2.66	1.16	7680	3.14	1.87
9	7684	3.82	1.21	7666	3.58	1.04	7669	3.62	0.98	7666	3.58	1.21	7663	3.54	1.83
10	7657	4.08	0.82	7651	4.00	0.98	7657	4.08	0.67	7651	4.00	1.13	7654	4.04	1.27
11	16713	3.76	9.13	16598	3.04	2.75	16671	3.50	8.70	16608	3.10	4.06	16582	2.94	16.87
12	16566	3.13	9.89	16581	3.22	2.70	16531	2.91	8.21	16585	3.24	3.69	16550	3.03	14.20
13	16523	2.90	9.57	16539	3.00	2.70	16469	2.56	6.60	16610	3.44	3.91	16529	2.93	13.13
14	16469	2.56	8.84	16469	2.56	2.68	16515	2.85	7.24	16539	3.00	4.36	16514	2.84	11.91
15	16527	3.02	8.76	16469	2.66	2.26	16485	2.76	6.10	16514	2.94	3.76	16479	2.72	11.99

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A												
1	3586	1.79	0.00	3586	1.79	0.00	3586	1.79	1.29	3586	2.27	0.00	3586	1.79	0.00
2	3586	1.79	0.00	3628	2.98	0.00	3586	1.79	0.00	3573	1.42	1.52	3586	1.79	1.16
3	3568	1.28	0.50	3586	1.79	0.86	3568	1.28	0.50	3563	1.14	1.49	3568	1.28	0.50
4	3568	1.28	1.33	3568	1.28	0.00	3568	1.28	0.00	3523	0.00	1.26	3568	1.28	0.00
5	3568	1.28	0.00	3568	1.28	0.00	3568	1.28	0.00	3523	0.00	1.26	3568	1.28	0.00
6	7711	3.68	0.00	7634	2.65	1.08	7721	3.82	0.63	7686	3.35	0.40	7696	3.48	0.27
7	7620	2.60	0.00	7659	3.12	0.96	7620	2.60	1.18	7685	3.47	0.61	7659	3.12	0.85
8	7674	3.60	0.40	7620	2.88	0.31	7588	2.44	1.29	7644	4.10	0.00	7588	2.44	1.20
9	7684	4.44	0.00	7588	3.14	1.02	7614	3.49	0.72	7666	4.54	0.00	7588	3.14	0.98
10	7569	2.88	1.15	7543	2.53	1.41	7565	2.83	1.20	7651	4.54	0.00	7543	2.53	1.45
11	16622	2.99	0.54	16504	2.26	0.57	16512	2.31	0.95	16608	3.37	0.00	16540	2.48	0.25
12	16532	2.47	0.21	16483	2.17	0.59	16530	2.46	0.01	16585	3.38	0.00	16521	2.41	0.18
13	16469	2.25	0.33	16457	2.17	0.50	16469	2.25	0.00	16610	4.10	0.00	16527	2.61	0.01
14	16469	2.32	0.00	16469	2.32	0.00	16469	2.32	0.28	16539	3.46	0.00	16514	2.60	0.00
15	16467	2.14	0.36	16469	2.15	0.00	16436	1.95	0.30	16514	3.25	0.00	16458	2.08	0.13

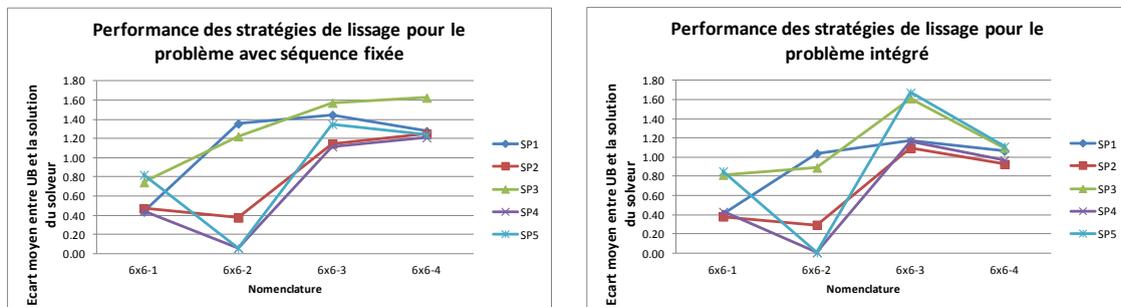
TABLEAU 4.14 – Résultats pour la nomenclature 6x6-4 avec l'algorithme ascendant

En ce qui concerne le problème intégré, plusieurs améliorations sont réalisées avec toutes les stratégies. La plus grande amélioration moyenne est obtenue avec SP2, qui surpasse SP5 comme méthode la plus performante. Constatons aussi que les instances 4 et 5 sont résolues de façon optimale avec SP4.

4.4.1.5 Bilan des expérimentations avec des job-shops 6x6

En tenant compte des 4 nomenclatures étudiées, des 5 stratégies de lissage et des 3 algorithmes de satisfaction des contraintes de nomenclature, nous pouvons décrire le comportement global de l'approche somme suit. Premièrement, considérons l'algorithme glouton, pour lequel une synthèse des résultats est présentée dans la Figure 4.15. Nous constatons que :

- La nomenclature 6x6-1 est relativement facile à résoudre avec toutes les stratégies de lissage,
- Les nomenclatures 6x6-3 et 6x6-4 sont les plus difficiles à résoudre, indépendamment de la stratégie de lissage,
- La difficulté de résolution du problème associé à la nomenclature 6x6-2 varie significativement selon la stratégie de lissage utilisée, SP4 et SP5 étant les stratégies les plus adéquates,
- Il est plus avantageux de satisfaire les contraintes de nomenclature avant celles de capacité. En effet, SP2 et SP4 offrent globalement la meilleure performance, les stratégies SP1 et SP3 n'étant pas compétitives,
- Il n'y a pas de différence significative entre le fait d'interdire ou d'autoriser la violation des contraintes de capacité, lorsque l'on cherche à satisfaire les contraintes de nomenclature. Hormis le cas de la nomenclature 6x6-2, pour laquelle SP4 est significativement meilleure, SP2 et SP4 offrent toujours une performance similaire,
- Si l'on satisfait les contraintes de capacité avant celles de nomenclature, il est préférable d'interdire la violation des contraintes de nomenclature. En effet, excepté le cas de la nomenclature 6x6-2, SP1 est plus performante que SP3.



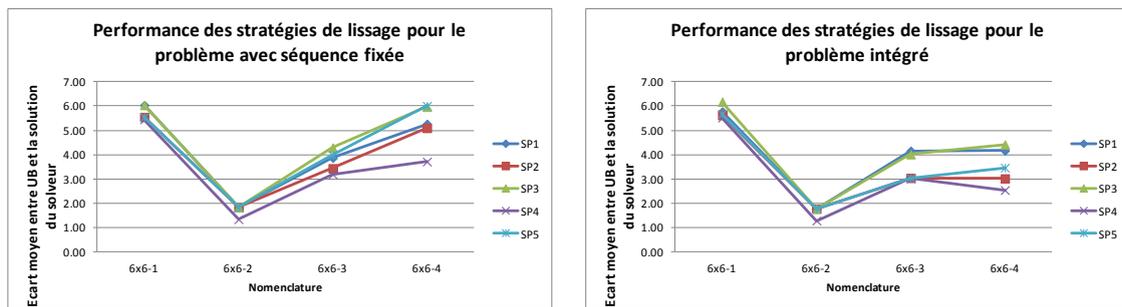
(a) Résultats pour le problème à séquence fixée

(b) Résultats pour le problème intégré

FIGURE 4.15 – Synthèse des résultats pour les nomenclatures 6x6 avec l'algorithme glouton

Considérons maintenant l'algorithme descendant, pour lequel une synthèse des résultats est présentée dans la Figure 4.16. Nous observons que :

- Les nomenclatures 6x6-1 et 6x6-4 sont les plus difficiles à résoudre. Néanmoins, des améliorations significatives des solutions associées à la nomenclature 6x6-4 sont réalisées en résolvant le problème intégré,
- La nomenclature 6x6-2 est la plus facile à résoudre,
- Il est plus avantageux de satisfaire les contraintes de nomenclature avant celles de capacité, SP2 et SP4 étant les stratégies de lissage les plus performantes,
- De meilleurs résultats sont obtenus lorsque l'on autorise la violation des contraintes de capacité pendant la validation des contraintes de nomenclature, SP4 étant plus performante que SP2,
- Il n'y a pas de différence significative entre le fait d'interdire ou d'autoriser la violation des contraintes de nomenclature pendant la validation des contraintes de capacité. SP1 et SP3 proposent des résultats similaires.



(a) Résultats pour le problème à séquence fixée

(b) Résultats pour le problème intégré

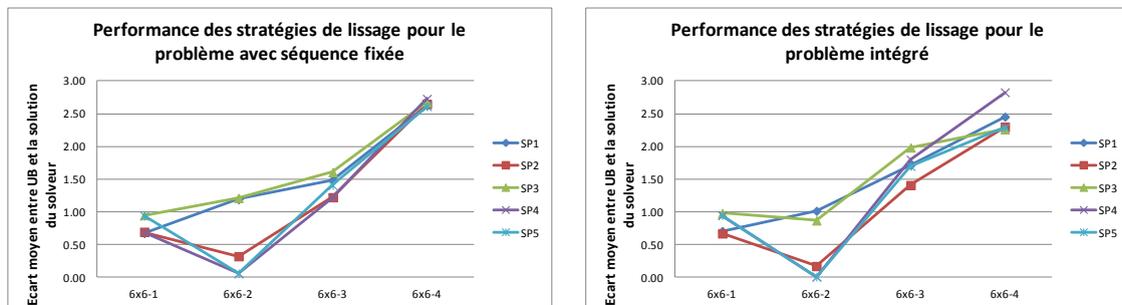
FIGURE 4.16 – Synthèse des résultats pour les nomenclatures 6x6 avec l'algorithme descendant

En ce qui concerne l'application de l'algorithme ascendant, nous présentons une synthèse des résultats dans la Figure 4.17. Nous pouvons constater que :

- Les nomenclatures 6x6-3 et 6x6-4 sont les plus difficiles à résoudre, indépendamment de l'heuristique de lissage,
- Le problème associé à la nomenclature 6x6-2 est facile à résoudre avec les stratégies satisfaisant en premier les contraintes de nomenclature (SP2 et SP4), mais relativement difficile avec SP1 et SP3,
- Les stratégies qui commencent par satisfaire les contraintes de nomenclature offrent globalement une performance supérieure à celles des stratégies validant en premier les

contraintes de capacité. Cependant, SP1 et SP3 sont meilleures que SP4 pour le problème associé à la nomenclature 6x6-4.

- Hormis la résolution du problème à séquence fixée, pour lequel la stratégie SP4 est légèrement meilleure que SP2 (grâce à l'impact de la solution pour la nomenclature 6x6-2), il est préférable d'interdire la violation des contraintes de capacité lorsque l'on satisfait les contraintes de nomenclature. SP2 offre une performance globale meilleure que celle de SP4,
- Si les contraintes de capacité sont satisfaites avant celles de nomenclature, il est globalement préférable d'interdire la violation des contraintes de nomenclature, SP1 étant plus performante que SP3. Cependant, les écarts sont réduits en résolvant le problème intégré.



(a) Résultats pour le problème à séquence fixée

(b) Résultats pour le problème intégré

FIGURE 4.17 – Synthèse des résultats pour les nomenclatures 6x6 avec l'algorithme ascendant

Pour finir, comparons la performance des trois algorithmes proposés. Dans la Figure 4.18, nous présentons les écarts moyens entre les bornes supérieures et les solutions du solveur commercial pour l'ensemble des nomenclatures 6x6 et nous différencions chacune des stratégies de lissage. Ainsi, nous observons que :

- L'algorithme glouton offre la meilleure performance, mais l'algorithme ascendant est compétitif,
- L'algorithme descendant n'est pas approprié,
- SP2 et SP4 sont globalement les stratégies les plus performantes, indépendamment de l'algorithme de validation des contraintes de nomenclature utilisé.

4.4.2 Expérimentations avec des ateliers de type job-shop 10x10

Nous considérons maintenant trois nomenclatures associées à un atelier de type job-shop 10x10, qui est par définition un problème de plus grande taille et plus difficile à résoudre que

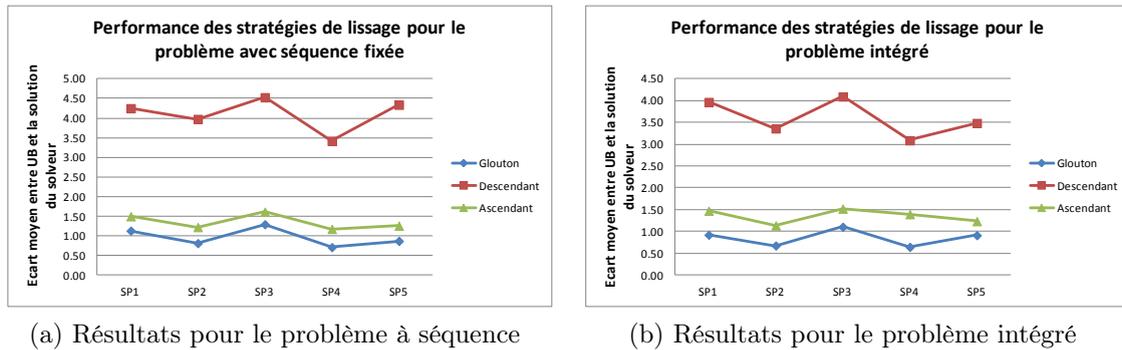


FIGURE 4.18 – Comparaison des algorithmes de validation des contraintes de nomenclature pour les nomenclatures 6x6

l'atelier de type job-shop 6x6. Nous nous focalisons sur l'analyse des expérimentations réalisées avec les algorithmes glouton et ascendant, l'algorithme descendant n'étant pas compétitif. Les résultats associés à l'algorithme descendant peuvent être consultés dans l'Annexe B.

Les paramètres généraux utilisés pour construire les instances sont détaillés dans le Tableau 4.15.

Nomenclature	c_i^s	d_{il}	L_i	c_i^p	c_i^{inv}
10x10-1	5	[0, 15]	1	4	1
10x10-2	100	[0, 15]	1	4	[1, 15]
10x10-3	100	[0, 15]	1	4	[1, 15]

TABLEAU 4.15 – Paramètres généraux associés aux ateliers de type job-shop 10x10

4.4.2.1 Nomenclature 10x10-1

Dans la nomenclature 10x10-1 (Figure 4.19), nous avons 4 niveaux et 10 produits, avec un seul produit fini et deux composants basiques. Cette fois-ci, le solveur commercial ne résout de façon optimale que 8 instances sur 12 du problème à séquence fixée, avec $T = 5$ et $T = 10$. Des solutions sous-optimales sont proposées pour les instances avec 20 périodes. Pour ce qui est du problème intégré, des solutions réalisables, dont 3 optimales, sont proposées pour les instances avec 5 périodes. Pour les autres instances, aucune solution réalisable n'est obtenue, mettant ainsi en évidence le niveau de difficulté plus élevé que pour le cas des nomenclatures à 6 produits. Les résultats sont présentés dans le Tableau 4.16.

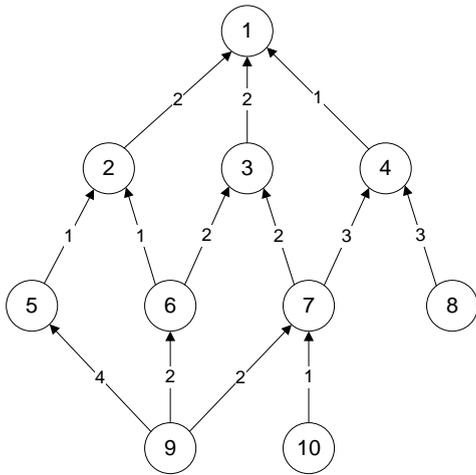


FIGURE 4.19 – Nomenclature 10x10-1

Instance	cap	T	ALB	CPL1	CPL2
1	0.90	5	11137	12910*	12910*
2	0.95			12910*	12910*
3	1.00			12910*	12910*
4	1.05			12910*	12928
5	0.85	10	19939	38497*	-
6	0.90			38492*	-
7	0.95			38476*	-
8	1.00			38456*	-
9	0.85	20	-23411	89188	-
10	0.90			89175	-
11	0.95			89101	-
12	1.00			89116	-

TABLEAU 4.16 – Solutions obtenues avec IBM ILOG CPLEX pour la nomenclature 10x10-1

Application de l’algorithme glouton

Les résultats obtenus par notre approche, grâce à l’implémentation de l’algorithme glouton, sont présentés dans le Tableau 4.17. Des solutions réalisables sont proposées pour toutes les instances, avec 4 solutions optimales (pour $T = 5$). Cependant, ces solutions correspondent au plan initial suivant la demande période par période, l’heuristique de lissage ne parvenant pas à obtenir de meilleures solutions avec aucune des stratégies. Ceci n’est pas réellement un problème pour ce cas d’étude précis, étant donné que les écarts entre les solutions obtenues par le solveur (optimales pour la plupart des cas) et les solutions initiales déterminées par l’approche ne sont pas très significatifs. L’écart est en moyenne égal à 0.18% avec toutes les stratégies. En ce qui concerne les temps de calcul, 4 stratégies sont très coûteuses (123.51 secondes, 53.57 secondes, 54.08 secondes et 221.75 secondes avec SP2, SP3, SP4 et SP5 respectivement). Seule SP1 présente un temps de calcul moyen raisonnable ($\bar{t} = 7.99$ secondes).

Aucune amélioration de la borne supérieure n’est obtenue en résolvant le problème intégré, mais la quatrième instance est mieux résolue par l’approche que par le solveur.

Application de l’algorithme ascendant

Les résultats avec l’algorithme ascendant sont présentés dans le Tableau 4.18. Contrairement aux deux autres méthodes, l’algorithme ascendant permet d’obtenir des solutions meilleures que le plan initial, principalement avec SP2, SP4 et SP5, ce qui fait améliorer légèrement la performance globale de l’approche par rapport aux algorithmes glouton et descendant. L’écart

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)
1	12910*	0.00	2.77	12910*	0.00	2.09	12910*	0.00	2.77	12910*	0.00	2.05	12910*	0.00	2.01
2	12910*	0.00	1.50	12910*	0.00	1.51	12910*	0.00	1.50	12910*	0.00	1.44	12910*	0.00	1.47
3	12910*	0.00	1.18	12910*	0.00	2.44	12910*	0.00	1.10	12910*	0.00	2.62	12910*	0.00	1.11
4	12910*	0.00	1.10	12910*	0.00	1.07	12910*	0.00	1.09	12910*	0.00	1.35	12910*	0.00	1.10
5	38588	0.24	1.53	38588	0.24	3.43	38588	0.24	3.83	38588	0.24	2.24	38588	0.24	5.60
6	38588	0.25	2.17	38588	0.25	12.49	38588	0.25	15.00	38588	0.25	6.18	38588	0.25	14.66
7	38588	0.29	4.25	38588	0.29	6.91	38588	0.29	5.20	38588	0.29	4.12	38588	0.29	6.77
8	38588	0.34	47.12	38588	0.34	55.06	38588	0.34	22.41	38588	0.34	25.00	38588	0.34	74.35
9	89368	0.20	8.57	89368	0.20	286.55	89368	0.20	157.34	89368	0.20	152.00	89368	0.20	678.85
10	89368	0.22	8.55	89368	0.22	415.65	89368	0.22	160.52	89368	0.22	165.89	89368	0.22	647.87
11	89368	0.30	11.88	89368	0.30	402.18	89368	0.30	138.44	89368	0.30	163.08	89368	0.30	789.71
12	89368	0.28	5.25	89368	0.28	292.72	89368	0.28	133.58	89368	0.28	123.01	89368	0.28	437.44

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A												
1	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00
2	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00
3	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00
4	12910	-0.14	0.00												
5	38588	-	0.00	38588	-	0.00	38588	-	0.00	38588	-	0.00	38588	-	0.00
6	38588	-	0.00	38588	-	0.00	38588	-	0.00	38588	-	0.00	38588	-	0.00
7	38588	-	0.00	38588	-	0.00	38588	-	0.00	38588	-	0.00	38588	-	0.00
8	38588	-	0.00	38588	-	0.00	38588	-	0.00	38588	-	0.00	38588	-	0.00
9	89368	-	0.00	89368	-	0.00	89368	-	0.00	89368	-	0.00	89368	-	0.00
10	89368	-	0.00	89368	-	0.00	89368	-	0.00	89368	-	0.00	89368	-	0.00
11	89368	-	0.00	89368	-	0.00	89368	-	0.00	89368	-	0.00	89368	-	0.00
12	89368	-	0.00	89368	-	0.00	89368	-	0.00	89368	-	0.00	89368	-	0.00

TABLEAU 4.17 – Résultats pour la nomenclature 10x10-1 avec l’algorithme glouton

moyen \bar{G} avec toutes les stratégies est de 0.17%. Cette amélioration est renforcée par une réduction du temps de calcul, par rapport à l’algorithme glouton, entre 41.16% et 77.88%. La stratégie la plus lente est à nouveau SP5, avec un temps de calcul moyen de 49.05 secondes, et la méthode la plus rapide est SP1, avec $\bar{t} = 3.95$ secondes.

Concernant le problème intégré, une légère amélioration est observée sur l’instance 6 avec la stratégie SP1, mais SP4 et SP5 présentent globalement la meilleure performance. Nous ne pouvons pas conclure sur la qualité globale de l’approche, car la plupart des instances ne sont pas résolues par le solveur. Cependant, notre approche garantit des solutions réalisables pour toutes les instances, contrairement à IBM ILOG CPLEX.

4.4.2.2 Nomenclature 10x10-2

Dans la nomenclature 10x10-2 (Figure 4.20), nous avons 3 niveaux et 10 produits, avec 3 produits sans successeurs et 3 produits sans prédécesseurs. Les résultats obtenus avec le solveur commercial sont présentés dans le Tableau 4.19. La solution optimale est trouvée pour les

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)
1	12910*	0.00	2.03	12910*	0.00	2.59	12910*	0.00	3.76	12910*	0.00	2.57	12910*	0.00	1.97
2	12910*	0.00	1.43	12910*	0.00	1.87	12910*	0.00	3.75	12910*	0.00	1.88	12910*	0.00	1.42
3	12910*	0.00	0.99	12910*	0.00	2.65	12910*	0.00	1.00	12910*	0.00	2.74	12910*	0.00	1.02
4	12910*	0.00	0.99	12910*	0.00	1.30	12910*	0.00	0.99	12910*	0.00	2.63	12910*	0.00	1.00
5	38579	0.21	1.89	38579	0.21	2.81	38579	0.21	4.11	38579	0.21	2.52	38579	0.21	3.98
6	38588	0.25	1.71	38579	0.23	4.02	38579	0.23	6.83	38579	0.23	3.54	38579	0.23	6.44
7	38588	0.29	2.29	38579	0.27	2.28	38588	0.29	2.98	38579	0.27	2.19	38579	0.27	2.88
8	38588	0.34	7.88	38579	0.32	6.07	38588	0.34	5.97	38579	0.32	6.02	38579	0.32	15.23
9	89368	0.20	8.39	89368	0.20	143.58	89368	0.20	120.29	89362	0.20	66.45	89362	0.20	173.29
10	89368	0.22	8.49	89368	0.22	113.82	89368	0.22	83.24	89362	0.21	53.09	89362	0.21	138.97
11	89368	0.30	5.84	89362	0.29	77.91	89368	0.30	69.13	89362	0.29	49.53	89362	0.29	137.17
12	89368	0.28	5.52	89362	0.28	69.19	89368	0.28	76.21	89362	0.28	38.27	89362	0.28	105.25

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A												
1	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00
2	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00
3	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00
4	12910	-0.14	0.00												
5	38579	-	0.00	38579	-	0.00	38579	-	0.00	38579	-	0.00	38579	-	0.00
6	38579	-	0.02	38579	-	0.00	38579	-	0.00	38579	-	0.00	38579	-	0.00
7	38588	-	0.00	38579	-	0.00	38588	-	0.00	38579	-	0.00	38579	-	0.00
8	38588	-	0.00	38579	-	0.00	38588	-	0.00	38579	-	0.00	38579	-	0.00
9	89368	-	0.00	89368	-	0.00	89368	-	0.00	89362	-	0.00	89362	-	0.00
10	89368	-	0.00	89368	-	0.00	89368	-	0.00	89362	-	0.00	89362	-	0.00
11	89368	-	0.00	89362	-	0.00	89368	-	0.00	89362	-	0.00	89362	-	0.00
12	89368	-	0.00	89362	-	0.00	89368	-	0.00	89362	-	0.00	89362	-	0.00

TABLEAU 4.18 – Résultats pour la nomenclature 10x10-1 avec l'algorithme ascendant

instances avec 5 et 10 périodes du problème avec séquence fixée, et des solutions réalisables sont proposées pour les instances avec $T = 20$. Le problème intégré est considérablement plus difficile à résoudre. Uniquement les instances avec 5 périodes sont résolues, dont 2 de façon optimale.

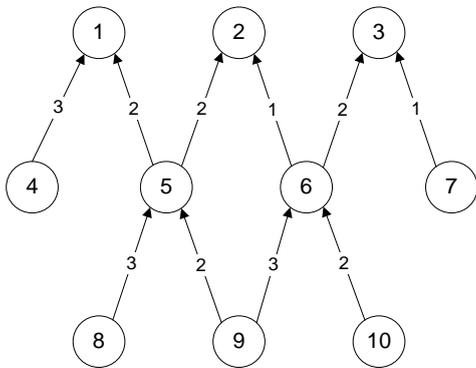


FIGURE 4.20 – Nomenclature 10x10-2

Instance	cap	T	ALB	CPL1	CPL2
1	0.60	5	11302	12555*	12558
2	0.65			12427*	12381*
3	0.70			12427*	12457
4	0.75			12381*	12381
5	0.80			12381*	12248*
6	0.60	10	23447	30572*	-
7	0.65			30506*	-
8	0.70			30546*	-
9	0.85			30329*	-
10	0.80			30258*	-
11	0.60	20	32953	66187	-
12	0.65			66022	-
13	0.70			65973	-
14	0.75			65692	-
15	0.80			65562	-

TABLEAU 4.19 – Solutions obtenues avec IBM ILOG CPLEX pour la nomenclature 10x10-2

Application de l’algorithme glouton

Les résultats avec l’algorithme glouton sont présentés dans le Tableau 4.20. Les écarts entre les solutions de notre approche et les solutions optimales sont plus importants que dans les cas d’études précédents, mettant en évidence la difficulté de ce type de nomenclature. L’algorithme glouton ne permet pas d’obtenir de solutions réalisables pour les instances avec $T = 10$ et $T = 20$, la solution proposée étant le plan initial. La qualité des solutions des différentes stratégies de lissage est différentiable à partir des instances avec 5 périodes. Les stratégies qui valident en premier les contraintes de nomenclature permettent une nouvelle fois d’obtenir globalement des meilleures solutions. SP5 offre le meilleur écart moyen \bar{G} , qui est de 4.52%, et SP2 et SP4 présentent un écart de 4.56%. D’autre part, SP1 et SP3 offrent des écarts moyens de 5.39% et 5.21% respectivement. En terme de temps de calcul, la meilleure stratégie est SP4 avec $\bar{t} = 6.21$ secondes, et la stratégie la plus lente est SP3 avec $\bar{t} = 11.83$ secondes. Si nous considérons un rapport entre qualité de solution et temps de calcul, c’est SP4 qu’il convient d’utiliser.

Concernant le problème intégré, la borne supérieure de l’instance 5 est améliorée avec SP3 et SP4, et SP5 reste la stratégie la plus performante avec un écart moyen de 3.61%, en considérant uniquement les instances avec $T = 5$, car le solveur commercial ne résout pas les autres instances.

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)												
1	13476	7.34	2.75	13476	7.34	2.83	13476	7.34	3.00	13476	7.34	2.61	13476	7.34	2.95
2	12654	1.83	3.61	12688	2.10	3.99	12668	1.94	3.08	12688	2.10	3.63	12668	1.94	3.77
3	12784	2.87	3.19	12688	2.10	3.08	12784	2.87	3.09	12688	2.10	2.79	12638	1.70	2.81
4	13476	8.84	3.24	12638	2.08	3.16	13145	6.17	2.98	12638	2.08	2.75	12638	2.08	2.79
5	13476	8.84	2.89	12844	3.74	2.89	13476	8.84	2.84	12844	3.74	2.73	12845	3.75	2.79
6	32060	4.87	15.26	32060	4.87	12.64	32060	4.87	17.69	32060	4.87	8.98	32060	4.87	17.60
7	32060	5.09	2.26	32060	5.09	3.60	32060	5.09	3.54	32060	5.09	2.47	32060	5.09	2.68
8	32060	4.96	2.53	32060	4.96	3.66	32060	4.96	3.78	32060	4.96	2.58	32060	4.96	2.96
9	32060	5.71	7.93	32060	5.71	13.10	32060	5.71	12.67	32060	5.71	7.56	32060	5.71	14.26
10	32060	5.96	5.41	32060	5.96	6.00	32060	5.96	5.73	32060	5.96	3.97	32060	5.96	10.59
11	69112	4.42	12.10	69112	4.42	21.69	69112	4.42	36.34	69112	4.42	11.79	69112	4.42	19.51
12	69112	4.68	9.96	69112	4.68	8.81	69112	4.68	9.23	69112	4.68	5.43	69112	4.68	8.96
13	69112	4.76	14.15	69112	4.76	20.99	69112	4.76	24.49	69112	4.76	11.59	69112	4.76	22.76
14	69112	5.21	19.96	69112	5.21	19.58	69112	5.21	24.05	69112	5.21	11.60	69112	5.21	22.04
15	69112	5.41	19.56	69112	5.41	21.50	69112	5.41	25.01	69112	5.41	12.71	69112	5.41	26.72

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A	UB	G	A	UB	G	A	UB	G	A	UB	G	A
1	13476	7.31	0.00	13476	7.31	0.00	13476	7.31	0.00	13476	7.31	0.00	13476	7.31	0.00
2	12654	2.20	0.00	12688	2.48	0.00	12668	2.32	0.00	12688	2.48	0.00	12668	2.32	0.00
3	12784	2.63	0.00	12688	1.85	0.00	12784	2.63	0.00	12688	1.85	0.00	12638	1.45	0.00
4	13476	8.84	0.00	12638	2.08	0.00	13145	6.17	0.00	12638	2.08	0.00	12638	2.08	0.00
5	13476	10.03	0.00	12844	4.87	0.00	13149	7.36	2.43	12806	4.56	0.30	12845	4.87	0.00
6	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00
7	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00
8	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00
9	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00
10	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00
11	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00
12	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00
13	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00
14	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00
15	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00

TABLEAU 4.20 – Résultats pour la nomenclature 10x10-2 avec l’algorithme glouton

Application de l’algorithme ascendant

Les résultats avec l’algorithme ascendant sont présentés dans le Tableau 4.21. Des solutions différentes du plan initial sont trouvées pour toutes les instances avec toutes les stratégies validant la capacité après la nomenclature (SP2, SP4 et SP5). La stratégie SP1 bloque sur les instances 1, 4, 5, 7, 8 et 14. La stratégie SP3 bloque sur les instances 5 et 7.

La performance de l’approche avec l’algorithme ascendant est donc meilleure que celles des deux autres algorithmes. L’écart moyen sur la solution du solveur commercial est amélioré de 16.51%, 42.54%, 29.94%, 48.25% et 44.47%, par rapport à l’algorithme glouton, avec SP1, SP2, SP3, SP4 et SP5 respectivement. La meilleure qualité est obtenue avec SP4, avec $\bar{G} = 2.36\%$, et la pire avec SP1, avec $\bar{G} = 4.50\%$. La performance est aussi améliorée par rapport aux temps

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)
1	13476	7.34	3.63	12713	1.26	2.14	13072	4.12	3.62	12713	1.26	2.25	12903	2.77	2.81
2	12638	1.70	3.93	12638	1.70	2.17	12638	1.70	3.68	12638	1.70	2.30	12638	1.70	5.33
3	12638	1.70	3.83	12638	1.70	2.09	12638	1.70	3.53	12638	1.70	2.24	12638	1.70	2.80
4	13476	8.84	3.90	12688	2.48	2.03	13045	5.36	3.49	12638	2.08	2.26	12638	2.08	2.73
5	13476	8.84	3.85	12806	3.43	1.97	13476	8.84	3.43	12638	2.08	2.21	12638	2.08	2.68
6	31448	2,87	16,41	31207	2,08	6,07	31448	2,87	10,45	31204	2,07	6,70	31445	2,86	13,09
7	32060	5.09	2.21	31226	2.36	1.83	32060	5.09	2.21	31173	2.19	1.76	31173	2.19	2.05
8	32060	4.96	2.14	31226	2.23	2.11	31238	2.27	2.28	31173	2.05	1.81	31173	2.05	2.31
9	31516	3.91	2.73	31315	3.25	2.62	31315	3.25	2.61	31173	2.78	2.50	31173	2.78	2.52
10	31512	4.14	2.86	31476	4.03	2.50	31533	4.21	2.57	31173	3.02	2.22	31173	3.02	2.64
11	68198	3.04	7.17	67789	2.42	8.08	67877	2.55	14.37	67789	2.42	7.03	67789	2.42	10.90
12	68166	3.25	6.46	67847	2.76	4.82	67941	2.91	5.76	67789	2.68	4.19	67789	2.68	6.04
13	67850	2.85	7.36	67847	2.84	7.07	67850	2.85	10.34	67789	2.75	6.20	67789	2.75	12.08
14	69112	5.21	9.13	67850	3.29	6.60	67850	3.29	9.80	67789	3.19	6.18	67789	3.19	11.78
15	67986	3.70	9.01	67850	3.49	7.33	68010	3.73	10.19	67789	3.40	6.89	67789	3.40	13.86

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A	UB	G	A	UB	G	A	UB	G	A	UB	G	A
1	12948	3.11	3.92	12663	0.84	0.39	12933	2.99	1.06	12713	1.23	0.00	12903	2.75	0.00
2	12638	2.08	0.00	12638	2.08	0.00	12638	2.08	0.00	12638	2.08	0.00	12638	2.08	0.00
3	12638	1.45	0.00	12638	1.45	0.00	12638	1.45	0.00	12638	1.45	0.00	12638	1.45	0.00
4	13476	8.84	0.00	12688	2.48	0.00	13045	5.36	0.00	12638	2.08	0.00	12638	2.08	0.00
5	13476	10.03	0.00	12806	4.56	0.00	13045	6.51	3.20	12638	3.18	0.00	12638	3.18	0.00
6	31448	-	0.00	31207	-	0.00	31448	-	0.00	31204	-	0.00	31441	-	0.01
7	32060	-	0.00	31226	-	0.00	32060	-	0.00	31173	-	0.00	31173	-	0.00
8	31105	-	2.98	31145	-	0.26	31226	-	0.04	31173	-	0.00	31173	-	0.00
9	31315	-	0.64	31305	-	0.03	31305	-	0.03	31173	-	0.00	31173	-	0.00
10	31465	-	0.15	31384	-	0.29	31309	-	0.71	31173	-	0.00	31173	-	0.00
11	67931	-	0.39	67789	-	0.00	67873	-	0.01	67789	-	0.00	67789	-	0.00
12	67850	-	0.46	67847	-	0.00	67852	-	0.13	67789	-	0.00	67789	-	0.00
13	67850	-	0.00	67789	-	0.09	67850	-	0.00	67789	-	0.00	67789	-	0.00
14	67850	-	1.83	67850	-	0.00	67850	-	0.00	67789	-	0.00	67789	-	0.00
15	67986	-	0.00	67789	-	0.09	67931	-	0.12	67789	-	0.00	67789	-	0.00

TABLEAU 4.21 – Résultats pour la nomenclature 10x10-2 avec l’algorithme ascendant

de calcul qui diminuent en moyenne de 32.21%, 59.72%, 50.21%, 39.13% et 42.65%, avec SP1, SP2, SP3, SP4 et SP5 respectivement. Le plus petit temps moyen de calcul (3.78 secondes) est obtenu avec SP4 et le plus grand avec SP5 (6.24 secondes). En dernier constat, on notera que la performance globale est améliorée lorsque l’on autorise la violation du deuxième type de contrainte, aussi bien en temps de calcul qu’en qualité de solution.

Concernant le problème intégré, plusieurs améliorations sont obtenues avec SP1, SP2 et SP3, mais SP4 reste la méthode la plus performante, avec un écart moyen par rapport aux solutions du solveur pour $T = 5$ de 2.00%. L’approche intégrée est globalement plus avantageuse qu’IBM ILOG CPLEX, étant donné que ce dernier ne trouve pas des solutions optimales pour les instances avec 10 et 20 périodes.

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)
1	9344	14.02	2.85	9344	14.02	2.91	9344	14.02	2.93	9344	14.02	2.82	8481	3.49	2.81
2	9344	14.26	3.03	9344	14.26	3.07	9344	14.26	3.07	9344	14.26	2.99	9344	14.26	2.98
3	8080*	0.00	3.19	8080*	0.00	3.18	8080*	0.00	3.28	8080*	0.00	3.04	8080*	0.00	2.95
4	8080*	0.00	3.93	8080*	0.00	3.94	8080*	0.00	3.95	8080*	0.00	3.77	8080*	0.00	3.67
5	9344	15.64	4.21	9344	15.64	4.23	9344	15.64	4.23	8080*	0.00	3.76	8080*	0.00	3.67
6	20412	9.98	3.16	20412	9.98	3.39	20412	9.98	3.43	20412	9.98	2.42	20412	9.98	3.50
7	20412	10.78	3.39	20412	10.78	3.49	20412	10.78	3.59	20412	10.78	2.49	20412	10.78	3.92
8	20412	10.39	3.18	20412	10.39	3.58	20412	10.39	3.63	20412	10.39	2.49	20412	10.39	3.64
9	20412	11.15	6.77	20412	11.15	7.34	20412	11.15	7.26	20412	11.15	5.17	20412	11.15	6.99
10	20412	11.53	3.12	20412	11.53	3.24	20412	11.53	3.19	20412	11.53	2.37	20412	11.53	3.00
11	44088	10.01	5.51	44088	10.01	20.67	44088	10.01	28.79	44088	10.01	17.53	44088	10.01	20.75
12	44088	10.21	6.48	44088	10.21	12.45	44088	10.21	16.00	44088	10.21	10.02	44088	10.21	9.78
13	44088	10.35	5.58	44088	10.35	7.49	44088	10.35	7.12	44088	10.35	5.67	44088	10.35	7.55
14	44088	10.72	4.65	44088	10.72	5.46	44088	10.72	5.77	44088	10.72	4.74	44088	10.72	5.73
15	44088	10.39	3.32	44088	10.39	4.03	44088	10.39	4.10	44088	10.39	3.79	44088	10.39	5.77

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A	UB	G	A	UB	G	A	UB	G	A	UB	G	A
1	9344	15.64	0.00	8429	4.32	9.79	8481	4.96	9.24	9344	15.64	0.00	8481	4.96	0.00
2	9344	15.64	0.00	8429	4.32	9.79	9344	15.64	0.00	9344	15.64	0.00	9344	15.64	0.00
3	8080*	0.00	0.00	8080*	0.00	0.00	8080*	0.00	0.00	8080*	0.00	0.00	8080*	0.00	0.00
4	8080*	0.00	0.00	8080*	0.00	0.00	8080*	0.00	0.00	8080*	0.00	0.00	8080*	0.00	0.00
5	9344	15.64	0.00	8080*	0.00	13.53	9344	15.64	0.00	8080*	0.00	0.00	8080*	0.00	0.00
6	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00
7	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00
8	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00
9	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00
10	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00
11	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00
12	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00
13	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00
14	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00
15	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00

TABLEAU 4.23 – Résultats pour la nomenclature 10x10-3 avec l’algorithme glouton

Concernant le problème intégré, des fortes améliorations sont réalisées avec SP2 pour les instances avec 5 périodes, permettant de réduire l’écart moyen par rapport aux solutions du solveur (pour $T = 5$) à 1.73%. SP2 est par conséquent la méthode la plus performante. Des comparaisons avec le solveur ne peuvent pas être effectuées pour les instances avec 10 et 20 périodes, car celui-ci ne trouve même pas de solutions réalisables.

Application de l’algorithme ascendant

Enfin, les résultats avec l’algorithme ascendant sont présentés dans le Tableau 4.24. Une fois de plus, cet algorithme permet d’éliminer le problème de blocage et réussit à déterminer des solutions meilleures que le plan initial avec toutes les stratégies, hormis SP1, qui bloque sur la

première et la deuxième instance avec $T = 5$, la quatrième instance avec $T = 10$ ainsi que la première, la deuxième et la cinquième instances avec $T = 20$. Les solutions sont significativement améliorées avec toutes les stratégies, les écarts moyens diminuant de 25.00%, 57.33%, 56.43%, 54.82% et 51.09% avec SP1, SP2, SP3, SP4 et SP5 respectivement, par rapport aux écarts de l'algorithme glouton. L'écart moyen le plus petit est de 4.02% et est obtenu avec SP5, mais SP4 offre aussi une bonne performance avec $\bar{G} = 4.03\%$. Les temps moyens de calcul sont aussi réduits (entre 7.79% et 34.58%), sauf pour SP1, où l'on passe de 4.16 secondes à 5.45 secondes. La stratégie la plus rapide est SP2 avec $\bar{t} = 3.86$ secondes, et la plus lente est SP3 avec $\bar{t} = 5.80$ secondes. En considérant qualité de solution et temps de calcul, SP4 est la stratégie la plus performante.

Instance	Problème à séquence fixée														
	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)
1	9344	14.02	3.73	8394	2.43	1.78	8394	2.43	3.21	8394	2.43	1.92	8394	2.43	2.92
2	9344	14.26	3.59	8394	2.64	1.83	8394	2.64	3.35	8394	2.64	2.00	8394	2.64	3.17
3	8352	3.37	3.50	8352	3.37	1.81	8352	3.37	3.17	8352	3.37	1.95	8352	3.37	3.33
4	8352	3.37	4.36	8352	3.37	1.73	8352	3.37	4.09	8352	3.37	1.83	8352	3.37	3.89
5	8605	6.50	4.36	8605	6.50	1.75	8605	6.50	4.12	8352	3.37	1.86	8352	3.37	3.94
6	19431	4.69	1.54	19351	4.26	4.09	19431	4.69	2.07	19351	4.26	4.32	19333	4.16	3.26
7	19219	4.31	1.74	19198	4.20	4.27	19317	4.84	2.17	19198	4.20	4.47	19198	4.20	3.73
8	19274	4.23	1.62	19192	3.79	4.23	19197	3.82	2.17	19192	3.79	4.52	19177	3.71	3.29
9	20412	11.15	2.85	19151	4.29	4.15	19151	4.29	2.84	19151	4.29	4.44	19151	4.29	7.14
10	19278	5.33	1.54	19151	4.64	4.17	19151	4.64	1.65	19151	4.64	4.52	19151	4.64	3.09
11	44088	10.01	5.15	42011	4.83	6.08	41973	4.73	10.76	41935	4.64	5.17	41935	4.64	13.98
12	44088	10.21	5.26	41876	4.68	6.91	41909	4.76	7.41	41876	4.68	6.55	41869	4.66	10.20
13	41869	4.80	5.56	41872	4.80	5.37	41869	4.80	5.26	41872	4.80	5.47	41869	4.80	6.97
14	41949	5.35	18.97	41872	5.16	4.84	41916	5.27	17.18	41872	5.16	5.11	41869	5.15	5.63
15	44088	10.39	18.02	41872	4.84	4.85	41892	4.89	17.56	41872	4.84	5.08	41869	4.83	5.41

Instance	Problème intégré														
	SP1			SP2			SP3			SP4			SP5		
	UB	G	A	UB	G	A	UB	G	A	UB	G	A	UB	G	A
1	8394	3.89	10.17	8394	3.89	0.00	8394	3.89	0.00	8394	3.89	0.00	8394	3.89	0.00
2	8402	3.99	10.08	8394	3.89	0.00	8394	3.89	0.00	8388	3.81	0.07	8394	3.89	0.00
3	8352	3.37	0.00	8352	3.37	0.00	8352	3.37	0.00	8352	3.37	0.00	8352	3.37	0.00
4	8352	3.37	0.00	8352	3.37	0.00	8352	3.37	0.00	8352	3.37	0.00	8352	3.37	0.00
5	8352	3.37	2.94	8605	6.50	0.00	8352	3.37	2.94	8352	3.37	0.00	8352	3.37	0.00
6	19320	-	0.57	19295	-	0.29	19338	-	0.48	19252	-	0.51	19295	-	0.20
7	19197	-	0.11	19177	-	0.11	19177	-	0.72	19180	-	0.09	19177	-	0.11
8	19180	-	0.49	19177	-	0.08	19177	-	0.10	19137	-	0.29	19163	-	0.07
9	19456	-	4.68	19151	-	0.00	19151	-	0.00	19037	-	0.60	19151	-	0.00
10	19278	-	0.00	19151	-	0.00	19151	-	0.00	19151	-	0.00	19151	-	0.00
11	42190	-	4.31	41967	-	0.10	41965	-	0.02	41925	-	0.02	41935	-	0.00
12	44088	-	0.00	41876	-	0.00	41908	-	0.00	41675	-	0.48	41869	-	0.00
13	41869	-	0.00	41872	-	0.00	41869	-	0.00	41872	-	0.00	41869	-	0.00
14	41949	-	0.00	41872	-	0.00	41889	-	0.06	41578	-	0.70	41869	-	0.00
15	41869	-	5.03	41869	-	0.01	41869	-	0.05	41748	-	0.30	41869	-	0.00

TABLEAU 4.24 – Résultats pour la nomenclature 10x10-3 avec l'algorithme ascendant

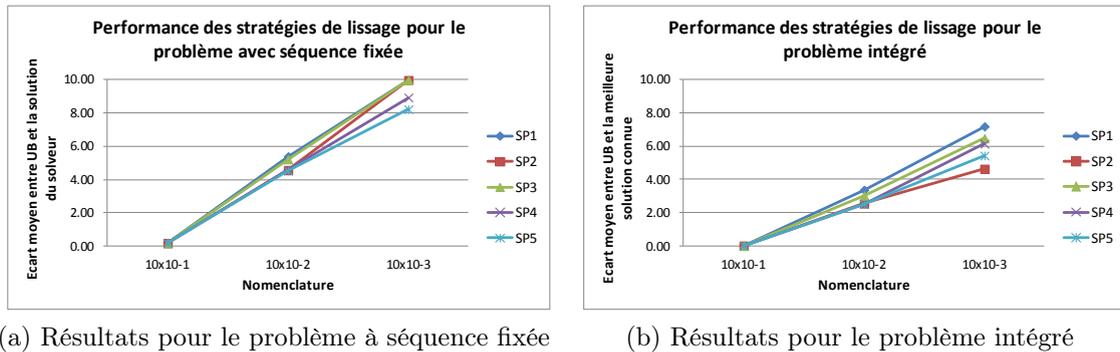
Concernant le problème intégré, des fortes améliorations sont réalisées avec SP1, mais la qualité globale des solutions est toujours la moins intéressante. Les meilleures solutions sont globalement obtenues avec SP4 et, comme pour les autres nomenclatures à 10 produits, l'approche intégrée est plus avantageuse que le solveur commercial, car ce dernier ne résout pas la plupart des instances.

4.4.2.4 Bilan des expérimentations avec des job-shops 10x10

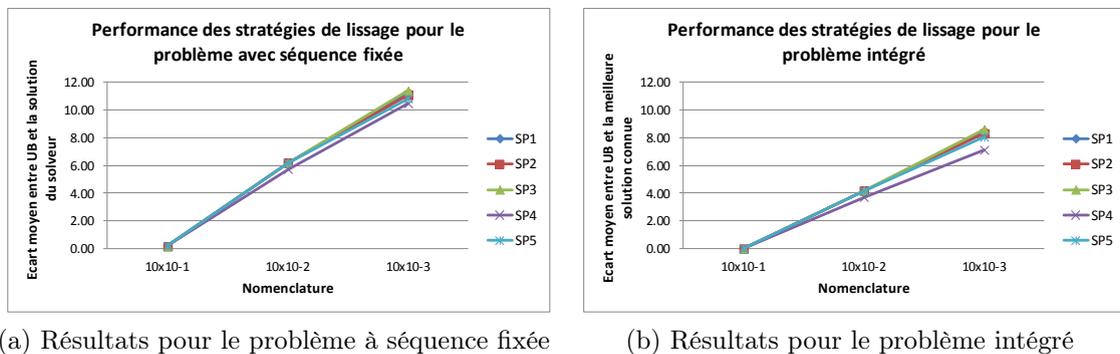
Nous avons résolu le problème de planification et d'ordonnancement intégré associé à 3 nomenclatures différentes, avec 5 stratégies de lissage et 3 algorithmes pour la satisfaction des contraintes de nomenclature. Nous pouvons conclure sur le comportement global de l'approche somme suit. Considérons d'abord l'algorithme glouton, pour lequel une synthèse des résultats est présentée dans la Figure 4.22. Comme IBM ILOG CPLEX ne trouve pas de solutions réalisables pour la plupart des instances, les écarts moyens associés au problème intégré ont été calculés par rapport à la meilleure solution connue, parmi tous les algorithmes et stratégies de lissage. Nous pouvons constater que :

- Bien que le problème associé à la nomenclature 10x10-1 est difficile à résoudre, le plan initial correspond à une solution proche de l'optimum dans le cas avec séquence fixée. Pour ce qui est du problème intégré, aucune stratégie ne permet d'améliorer le plan initial,
- Les écarts les plus importants en qualité de solution sont obtenus avec la nomenclature 10x10-3,
- Les stratégies qui commencent par satisfaire les contraintes de nomenclature donnent de meilleurs résultats. SP5 est la stratégie la plus performante pour le problème avec séquence fixée et SP2 est plus avantageuse pour le problème intégré.
- Mis à part la nomenclature 10x10-3, il n'y a pas de différence entre le fait d'interdire ou d'autoriser la violation des contraintes de capacité pendant la validation des contraintes de nomenclature. En effet, SP2 et SP4 offrent la même performance. Cependant, avec la nomenclature 10x10-3, le problème avec séquence fixée est mieux résolu avec SP2, tandis que SP4 donne de meilleures solutions pour le problème intégré.
- Si l'on satisfait en premier les contraintes de capacité, il est préférable d'autoriser la violation des contraintes de nomenclature. SP3 offre une meilleure performance que SP1 pour le problème intégré.

Considérons maintenant les résultats obtenus avec l'algorithme descendant, dont une synthèse est présentée dans la Figure 4.23. À nouveau, les écarts moyens associés au problème intégré ont été calculés par rapport à la meilleure solution connue, parmi tous les algorithmes et stratégies de lissage. Nous constatons que :

FIGURE 4.22 – Synthèse des résultats pour les nomenclatures 10×10 avec l’algorithme glouton

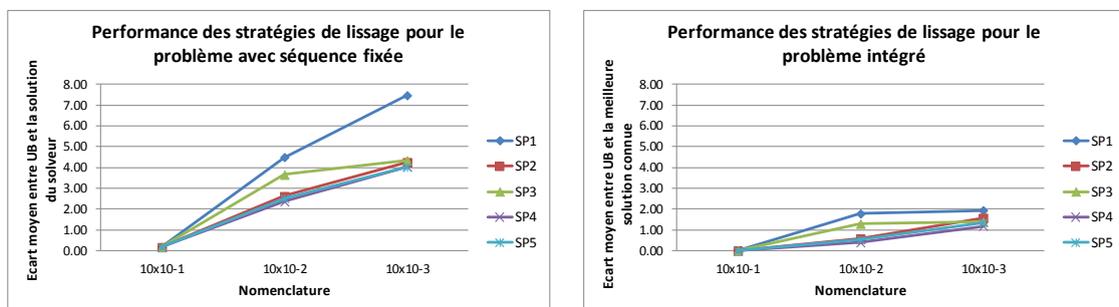
- Il n’y a quasiment pas de différence entre les performances des stratégies de lissage, car la solution finale correspond la plupart du temps au plan initial,
- Les écarts les plus importants sont générés en résolvant le problème associé à la nomenclature 10×10^{-3} ,
- SP4 est la stratégie de lissage la plus performante.

FIGURE 4.23 – Synthèse des résultats pour les nomenclatures 10×10 avec l’algorithme descendant

En ce qui concerne l’application de l’algorithme ascendant, une synthèse des résultats est présentée dans la Figure 4.24. Les écarts moyens associés au problème intégré ont été calculés par rapport à la meilleure solution connue, parmi tous les algorithmes et stratégies de lissage. Nous pouvons tirer les conclusions suivantes :

- Des solutions différentes du plan initial sont obtenues avec toutes les stratégies de lissage,

- Les écarts sur la qualité de solution les plus importants sont générés avec la nomenclature 10×10^{-3} , mais ils sont réduits en résolvant le problème intégré,
- Il est plus avantageux de satisfaire en premier les contraintes de nomenclature, SP2, SP4 et SP5 étant les stratégies de lissage les plus performantes,
- Il est préférable d'autoriser la violation des contraintes de capacité pendant la validation des contraintes de nomenclature, SP4 étant supérieure à SP2,
- Si l'on commence par satisfaire les contraintes de capacité, il est plus avantageux d'autoriser la violation des contraintes de nomenclature. La performance de SP3 est significativement supérieure à celle de SP1, particulièrement avec la nomenclature 10×10^{-3} .



(a) Résultats pour le problème à séquence fixée

(b) Résultats pour le problème intégré

FIGURE 4.24 – Synthèse des résultats pour les nomenclatures 10×10 avec l'algorithme ascendant

Finalement, une synthèse des résultats permettant de comparer les 3 algorithmes de validation des contraintes de nomenclature est présentée dans la Figure 4.25. Nous présentons les écarts moyens entre les bornes supérieures et les meilleures solutions trouvées pour l'ensemble des nomenclatures 10×10 . Nous pouvons constater que :

- L'algorithme ascendant offre globalement la meilleure performance et permet de trouver des solutions réalisables différentes du plan initial,
- L'algorithme glouton et l'algorithme descendant ne sont pas appropriés, car ils ne permettent pas de toujours trouver des solutions réalisables. La plupart du temps, les solutions proposées correspondent au plan initial,
- Il est plus avantageux de satisfaire en premier les contraintes de nomenclature, SP2, SP4 et SP5 étant les stratégies de lissage les plus performantes,
- SP1 est globalement la stratégie de lissage la moins performante, et il ne faut pas l'utiliser avec l'algorithme ascendant. Cependant, l'écart est réduit en résolvant le problème intégré.

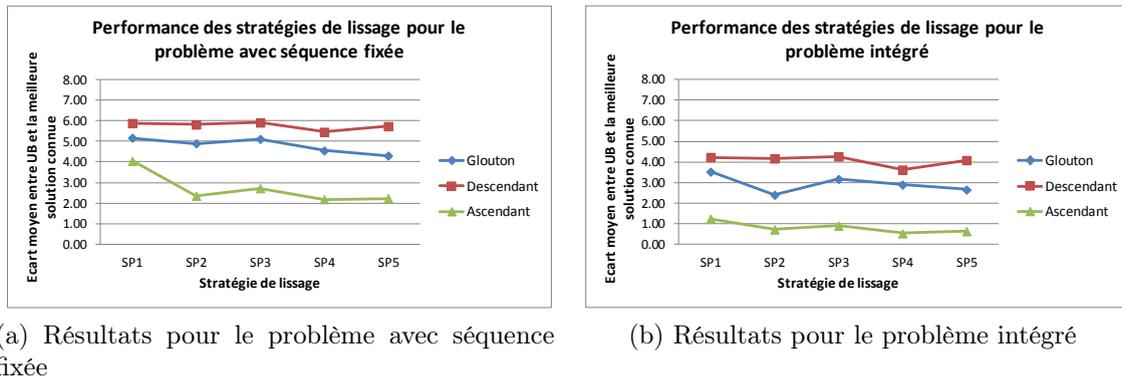


FIGURE 4.25 – Comparaison des algorithmes de validation des contraintes de nomenclature pour les nomenclatures 10x10

4.5 Conclusion

Nous avons présenté une extension de l'approche intégrée permettant de résoudre des problèmes de dimensionnement de lots et d'ordonnancement dans des systèmes de production à plusieurs niveaux, configurés comme des ateliers de type job-shop. Cette méthode permet de trouver des solutions réalisables tenant compte de la capacité détaillée au niveau opérationnel et des liens de nomenclature entre les produits composants et composés.

De manière similaire à l'approche pour des problèmes mono-niveau, nous avons implémenté une heuristique Lagrangienne pour résoudre le problème de planification à séquence fixée et une méthode d'amélioration de la séquence guidée par une recherche taboue. Nous avons utilisé une nouvelle formulation basée sur le concept d'échelon stock, afin de réduire la difficulté de résolution du problème. Cette stratégie nous a permis de continuer à utiliser la propriété de Wagner-Whitin. Comme de nouvelles contraintes (nomenclature) sont relâchées, nous avons proposé plusieurs stratégies de lissage afin de satisfaire les deux types de contraintes. Nous avons également proposé plusieurs algorithmes de satisfaction des contraintes de nomenclature.

Une analyse de la performance de l'approche intégrée avec l'utilisation des différentes stratégies de lissage et des différents algorithmes a été menée, en la comparant à la performance du solveur standard IBM ILOG CPLEX Optimization Studio 12.6. Sur des problèmes de taille relativement petite, la performance de notre approche et celle du solveur sont très bonnes. Le modèle intégré est résolu de façon optimale par le solveur et nos solutions ont des écarts très petits par rapport à l'optimum. Pour des problèmes de grande taille, le solveur n'est pas capable

de résoudre la plupart des instances et même de trouver des solutions réalisables, tandis que notre approche détermine des solutions réalisables pour toutes les instances.

Parmi les algorithmes que nous avons proposés pour la validation des contraintes de nomenclature, l'algorithme glouton est le plus efficace pour les nomenclatures de petite taille (6 produits) ; tandis que l'algorithme ascendant (logique opposée à MRP) s'avère plus efficace pour les nomenclatures de grande taille (10 produits). En analysant les stratégies de lissage, nous avons constaté qu'il est plus avantageux de satisfaire les contraintes de nomenclature avant celles de capacité et qu'il est préférable, dans la plupart des cas, d'autoriser la violation des contraintes de capacité pendant la validation des contraintes de nomenclature.

Conclusion Générale et Perspectives

Conclusion Générale

Dans cette thèse, nous nous sommes intéressés à la résolution du problème intégré de planification et d'ordonnancement de la production, dans des ateliers de production complexes, pour faire face à l'incohérence des décisions en management de la chaîne logistique. La planification et l'ordonnancement étant résolus habituellement de manière séparée et hiérarchique, les approches existantes conduisent, la plupart du temps, à des plans de production irréalisables. En outre, très peu d'approches sont dédiées à des ateliers complexes.

Cette problématique nécessitant une étude approfondie, nous avons présenté, en premier lieu, les principes et les limites des solutions logicielles existantes pour le management de la chaîne logistique, et plus particulièrement pour la gestion de la production. Nous avons constaté que l'approche MRP-II et les systèmes de planification avancée (APS) ne permettaient pas de résoudre efficacement le problème de planification et d'ordonnancement, les solutions proposées engendrant souvent des plans de production coûteux. Pourtant, des progiciels de gestion intégrée (ERP) basés sur ces concepts sont largement utilisés par les entreprises pour la gestion de la chaîne logistique.

En second lieu, nous avons dressé une vue d'ensemble sur les problèmes de dimensionnement de lots avec contraintes de capacité étudiés dans la littérature. Un soin particulier a été apporté aux problèmes incluant des contraintes d'ordonnancement, permettant d'assurer la faisabilité des plans de production tactiques. Nous avons constaté que les ateliers multi-ressources nécessitant des contraintes de capacité détaillées comme le job-shop étaient très peu étudiés. C'est particulièrement le cas pour les problèmes multi-niveaux de dimensionnement de lots avec contraintes de capacité. Une des raisons principales est la difficulté de résolution des problèmes de planification et d'ordonnancement qui peuvent être très complexes, même pris séparément.

Face à ces constats, nous avons proposé, dans un premier temps, une approche permettant de résoudre de manière intégrée le problème de dimensionnement de lots et d'ordonnancement dans des systèmes de production à un seul niveau et dans des ateliers complexes. L'approche utilise

une heuristique Lagrangienne pour la résolution du problème de planification avec séquence fixée et une méthode d'amélioration de la séquence basée sur une recherche taboue. L'approche résulte d'améliorations menées sur une succession de travaux qui apportaient une solution satisfaisante aux besoins de cohérence entre la planification et l'ordonnancement.

Dans un deuxième temps, nous avons étendu l'approche intégrée au cas des nomenclatures à plusieurs niveaux. Nous avons fait évoluer la formulation mathématique pour tenir compte de nouvelles contraintes liées aux structures de produits à plusieurs niveaux, et nous avons proposé plusieurs stratégies et algorithmes de lissage de la production.

Des expérimentations numériques ont permis de valider la pertinence de notre approche intégrée, aussi bien dans le cas des problèmes à un seul niveau que dans le cas multi-niveaux. Elle constitue ainsi une alternative sérieuse aux approches existantes et une solution efficace pour l'optimisation intégrée des décisions en planification et ordonnancement dans la chaîne logistique.

Perspectives

L'efficacité de l'approche intégrée a été démontrée. Néanmoins, des améliorations des algorithmes de résolution sont toujours possibles. Nous envisageons notamment de considérer de nouvelles structures de voisinage pour la recherche taboue, par exemple en changeant plusieurs arcs disjonctifs en même temps ou en régénérant complètement de nouvelles séquences. La procédure actuelle, permettant de définir une nouvelle séquence à partir du changement d'orientation d'un des arcs disjonctifs, conditionne le temps d'obtention d'une solution de bonne qualité. En effet, selon la taille du problème et la qualité de la séquence initiale, plusieurs changements peuvent être nécessaires avant de converger vers une bonne solution. Une alternative serait de changer un arc disjonctif par machine à chaque itération de la recherche taboue. Une autre alternative, permettant de reconstruire une nouvelle séquence, serait d'utiliser l'heuristique *Shifting Bottleneck* [6] pour résoudre le problème d'ordonnancement à partir du meilleur plan de production trouvé après un certain nombre d'itérations sans améliorations.

Dans le cas précis de l'approche pour des problèmes multi-niveaux, des efforts d'améliorations peuvent être conduits au niveau de l'heuristique de lissage. De nouveaux mouvements qui déplacent plusieurs produits en même temps est une voie à explorer. Les algorithmes de lissage proposés permettent de déplacer la production d'un successeur ou d'un prédécesseur à la fois, mais pas les deux en même temps. Nous pourrions calculer, en fonction de la marge sur les contraintes d'équilibres des stocks et de capacité, une quantité à déplacer vers l'arrière pour le produit critique (prédécesseur) et des quantités à déplacer vers l'avant pour un ou plusieurs de

ses successeurs en même temps. Cela permettrait de mieux répartir la production sur l'horizon de planification et d'éviter des blocages (situations où il n'y a plus de mouvements possibles) prématurés.

L'application de l'approche et son adaptation à des problèmes comportant plusieurs sites de production est envisagée à court terme. Nous avons développé une approche pour résoudre des problèmes multi-niveaux. En pratique, ce type de problème peut être rencontré dans des systèmes avec plusieurs sites de production, où chaque site fabrique un des composants de la nomenclature. Nous pouvons tenir compte de ce scénario dans notre approche en considérant un problème d'ordonnancement différent par site de production. Très peu de changements sur l'approche seraient nécessaires, et il serait possible de modifier la séquence de chaque site de production à chaque itération de la recherche taboue. Il serait intéressant, dans ce cas, d'utiliser le calcul parallèle pour résoudre les problèmes d'ordonnancement de chaque site.

Nous réfléchissons également à la considération de nouvelles contraintes pratiques, comme l'intégration de temps et de coûts de setup dépendant des séquences. Ces considérations sont souvent intégrées dans les problèmes de dimensionnement de lots à une seule machine ou à machines parallèles avec une seule station de travail. Dans des problèmes complexes associés aux ateliers de type flow-shop ou job-shop cette structure de setup génère une difficulté très importante, car des nouvelles variables et contraintes doivent être ajoutées. Néanmoins, étant donné que dans une partie des systèmes de production rencontrés en pratique, la séquence des opérations a un impact sur les coûts et les temps de setup, il serait pertinent d'étudier cette caractéristique.

Le travail réalisé dans cette thèse est une contribution importante en management de la chaîne logistique, mais les opportunités et les challenges ne s'arrêtent pas à ce stade. Une des pistes à développer dans un futur travail, et dont nous proposons une première modélisation dans l'Annexe C, est l'intégration de la planification de la distribution. En effet, nous avons considéré jusqu'à présent l'intégration de flux verticaux dans le processus de décisions d'une chaîne logistique. Il est aussi important de considérer l'intégration de flux horizontaux, permettant notamment de lier production et distribution. Pour combler ce besoin, nous avons commencé à étudier le problème intégré de dimensionnement de lots, d'ordonnancement et de distribution dans des chaînes logistiques multi-sources (plusieurs fournisseurs) et multi-sites (plusieurs clients internes) et des ateliers de production de type job-shop. Cette nouvelle problématique fait que les coûts de production, de setup et de stockage dépendent du site de production et que des coûts de transport sont à considérer.

Enfin, une autre idée que nous commençons à exploiter dans l'Annexe D est la considération des aspects environnementaux dans la planification de la production, le but étant d'inscrire

l'approche dans un cadre de chaîne logistique durable. En effet, de nos jours, les entreprises sont amenées à respecter des contraintes environnementales imposées par des lois nationales ou internationales visant à réduire les niveaux de pollution, le critère économique n'étant plus le seul aspect à considérer. Face à ce constat, nous proposons des modèles mathématiques pour les problèmes intégrés mono-niveau et multi-niveaux de dimensionnement de lots et d'ordonnement, en incluant des contraintes d'émissions de carbone, comme le font Absi *et al.* [3] pour des problèmes de dimensionnement de lots mono-produit et multi-sources sans contraintes de capacité. Pour résoudre le problème intégré, des modifications mineures de l'approche sont nécessaires au niveau de l'algorithme de programmation dynamique de Wagelmans *et al.* [230] et de l'heuristique de lissage.

Annexe A

Fonction objectif Lagrangienne du problème à plusieurs niveaux

Nous décrivons ici comment exprimer la fonction objectif relâchée uniquement en fonction des variables de décisions et des multiplicateurs Lagrangiens, de façon à pouvoir utiliser l'algorithme de Wagelmans *et al.* [230] pour le calcul des tailles de lot. Avant toute transformation, la fonction objectif Lagrangienne est exprimée de la façon suivante.

$$\begin{aligned} F(X, Y, \beta, \gamma) = & \min \sum_{i=1}^N \sum_{l=1}^T (c_i^s Y_{il} + c_i^p X_{il} + e_i E_{il}) \\ & + \sum_{c \in \mathcal{C}(y)} \beta_c \left[r(o_c^f) + \sum_{o \in c} (p_o^u X_{i(o)l(o)} + s_o^t Y_{i(o)l(o)}) - \sum_{l=1}^{l(o_c^l)} c_l \right] \\ & + \sum_{i=1}^N \sum_{l=1}^T \gamma_{il} \left(\sum_{j \in \mathcal{DS}(i)} g_{ij} E_{j,l+L_j} - E_{il} \right) \end{aligned} \quad (\text{A.1})$$

Nous pouvons regrouper les variables d'échelon stock E_{il} pour obtenir l'expression suivante.

$$\begin{aligned}
F(X, Y, \beta, \gamma) = & \min \sum_{i=1}^N \sum_{l=1}^T [c_i^s Y_{il} + c_i^p X_{il} + (e_i - \gamma_{il}) E_{il}] \\
& + \sum_{c \in \mathcal{C}(y)} \beta_c \left[r(o_c^f) + \sum_{o \in c} (p_o^u X_{i(o)l(o)} + s_o^t Y_{i(o)l(o)}) - \sum_{l=1}^{l(o_c^l)} c_l \right] \\
& + \sum_{i=1}^N \sum_{l=1}^T \gamma_{il} \sum_{j \in \mathcal{DS}(i)} g_{ij} E_{j,l+L_j}
\end{aligned} \tag{A.2}$$

Comme $E_{i1} = 0 \forall i \in \{1, \dots, N\}$, alors

$$E_{il} = \sum_{k=1}^l X_{ik} - \sum_{k=1}^l D_{ik} \quad \forall i \in \{1, \dots, N\} \tag{A.3}$$

En remplaçant E_{il} par l'expression précédente, nous obtenons la fonction suivante :

$$\begin{aligned}
F(X, Y, \beta, \gamma) = & \min \sum_{i=1}^N \sum_{l=1}^T \left[c_i^s Y_{il} + c_i^p X_{il} + (e_i - \gamma_{il}) \sum_{k=1}^l X_{ik} \right] \\
& + \sum_{c \in \mathcal{C}(y)} \beta_c \left[r(o_c^f) + \sum_{o \in c} (p_o^u X_{i(o)l(o)} + s_o^t Y_{i(o)l(o)}) - \sum_{l=1}^{l(o_c^l)} c_l \right] \\
& + \sum_{i=1}^N \sum_{l=1}^T \gamma_{il} \sum_{j \in \mathcal{DS}(i)} g_{ij} E_{j,l+L_j} - \sum_{i=1}^N \sum_{l=1}^T (e_i - \gamma_{il}) \sum_{k=1}^l D_{ik}
\end{aligned} \tag{A.4}$$

Nous pouvons factoriser le premier terme de la fonction précédente en termes de X comme suit :

$$\begin{aligned}
F(X, Y, \beta, \gamma) = & \min \sum_{i=1}^N \sum_{l=1}^T \left\{ c_i^s Y_{il} + \left[c_i^p + \sum_{k=l}^T (e_i - \gamma_{ik}) \right] X_{il} \right\} \\
& + \sum_{c \in \mathcal{C}(y)} \beta_c \left[r(o_c^f) + \sum_{o \in c} (p_o^u X_{i(o)l(o)} + s_o^t Y_{i(o)l(o)}) - \sum_{l=1}^{l(o_c^l)} c_l \right] \\
& + \sum_{i=1}^N \sum_{l=1}^T \gamma_{il} \sum_{j \in \mathcal{DS}(i)} g_{ij} E_{j,l+L_j} - \sum_{i=1}^N \sum_{l=1}^T (e_i - \gamma_{il}) \sum_{k=1}^l D_{ik}
\end{aligned} \tag{A.5}$$

En factorisant les autres termes en fonction de X et Y , nous obtenons l'expression suivante :

$$\begin{aligned}
F(X, Y, \beta, \gamma) = & \min \sum_{i=1}^N \sum_{l=1}^T \left\{ \left(c_i^s + \sum_{c \in \mathcal{C}(y)} \beta_c + \sum_{o \in c} s_o^t \right) Y_{il} \right. \\
& + \left. \left[c_i^p + \sum_{k=l}^T (e_i - \gamma_{ik}) + \sum_{c \in \mathcal{C}(y)} \beta_c \sum_{o \in c} p_o^u \right] X_{il} \right\} \\
& + \sum_{c \in \mathcal{C}(y)} \beta_c \left[r(o_c^f) - \sum_{l=1}^{l(o_c^l)} c_l \right] - \sum_{i=1}^N \sum_{l=1}^T (e_i - \gamma_{il}) \sum_{k=1}^l D_{ik} \\
& + \sum_{i=1}^N \sum_{l=1}^T \gamma_{il} \sum_{j \in \mathcal{DS}(i)} g_{ij} E_{j,l+L_j}
\end{aligned} \tag{A.6}$$

En remplaçant la variable d'échelon stock $E_{j,l+L_j}$ des successeurs $\mathcal{DS}(i)$ avec (A.3), nous obtenons l'expression suivante :

$$\begin{aligned}
F(X, Y, \beta, \gamma) = \min \sum_{i=1}^N \sum_{l=1}^T & \left\{ \left(c_i^s + \sum_{c \in \mathcal{C}(y)} \beta_c + \sum_{o \in c} s_o^t \right) Y_{il} \right. \\
& + \left[c_i^p + \sum_{k=l}^T (e_i - \gamma_{ik}) + \sum_{c \in \mathcal{C}(y)} \beta_c \sum_{o \in c} p_o^u \right] X_{il} \left. \right\} \\
& + \sum_{c \in \mathcal{C}(y)} \beta_c \left[r(o_c^f) - \sum_{l=1}^{l(o_c^l)} c_l \right] - \sum_{i=1}^N \sum_{l=1}^T (e_i - \gamma_{il}) \sum_{k=1}^l D_{ik} \\
& + \sum_{i=1}^N \sum_{l=1}^T \gamma_{il} \sum_{j \in \mathcal{DS}(i)} g_{ij} \left(\sum_{k=1+L_j}^{l+L_j} X_{jk} - \sum_{k=1+L_j}^{l+L_j} D_{jk} \right)
\end{aligned} \tag{A.7}$$

Nous pouvons séparer le nouveau terme pour obtenir l'expression suivante :

$$\begin{aligned}
F(X, Y, \beta, \gamma) = \min \sum_{i=1}^N \sum_{l=1}^T & \left\{ \left(c_i^s + \sum_{c \in \mathcal{C}(y)} \beta_c + \sum_{o \in c} s_o^t \right) Y_{il} \right. \\
& + \left[c_i^p + \sum_{k=l}^T (e_i - \gamma_{ik}) + \sum_{c \in \mathcal{C}(y)} \beta_c \sum_{o \in c} p_o^u \right] X_{il} \left. \right\} \\
& + \sum_{c \in \mathcal{C}(y)} \beta_c \left[r(o_c^f) - \sum_{l=1}^{l(o_c^l)} c_l \right] - \sum_{i=1}^N \sum_{l=1}^T \sum_{k=1}^T (e_i - \gamma_{ik}) D_{il} \\
& + \sum_{i=1}^N \sum_{l=1}^T \sum_{j \in \mathcal{DS}(i)} \sum_{k=1+L_j}^{l+L_j} \gamma_{il} g_{ij} X_{jk} \\
& + \sum_{i=1}^N \sum_{l=1}^T \sum_{j \in \mathcal{DS}(i)} \sum_{k=1+L_j}^{l+L_j} \gamma_{il} g_{ij} D_{jk}
\end{aligned} \tag{A.8}$$

Nous pouvons réécrire les deux nouveaux termes pour garder les mêmes indices dans toute la fonction comme suit :

$$\begin{aligned}
F(X, Y, \beta, \gamma) = \min \sum_{i=1}^N \sum_{l=1}^T & \left\{ \left(c_i^s + \sum_{c \in \mathcal{C}(y)} \beta_c + \sum_{o \in c} s_o^t \right) Y_{il} \right. \\
& + \left[c_i^p + \sum_{k=l}^T (e_i - \gamma_{ik}) + \sum_{c \in \mathcal{C}(y)} \beta_c \sum_{o \in c} p_o^u \right] X_{il} \left. \right\} \\
& + \sum_{c \in \mathcal{C}(y)} \beta_c \left[r(o_c^f) - \sum_{l=1}^{l(o_c^l)} c_l \right] - \sum_{i=1}^N \sum_{l=1}^T \sum_{k=1}^T (e_i - \gamma_{ik}) D_{il} \\
& + \sum_{i=1}^N \sum_{l=1}^T \sum_{j \in \mathcal{DS}(i)} \sum_{k=l}^T \gamma_{il} g_{ij} X_{j,l+L_j} \\
& + \sum_{i=1}^N \sum_{l=1}^T \sum_{j \in \mathcal{DS}(i)} \sum_{k=l}^T \gamma_{il} g_{ij} D_{j,l+L_j}
\end{aligned} \tag{4.18}$$

Nous observons que des tailles de lots $X_{j,l+L_j}$ associées aux successeurs $\mathcal{DS}(i)$ du produit i sont présentes dans la fonction objectif. Afin de pouvoir utiliser l'algorithme de programmation dynamique de Wagelmans et al. [230], nous devons réécrire la fonction pour obtenir des tailles de lots associées à un seul niveau de la nomenclature. Nous pouvons donc finalement exprimer les tailles de lots en fonction des liens avec les composants pour obtenir la fonction objectif Lagrangienne suivante, présentée dans la Section 4.3.1.1 Chapitre 4 :

$$\begin{aligned}
F(X, Y, \beta, \gamma) = \min \sum_{i=1}^N \sum_{l=1}^T & \left\{ \left(c_i^s + \sum_{c \in \mathcal{C}(y)} \beta_c + \sum_{o \in c} s_o^t \right) Y_{il} \right. \\
& + \left[c_i^p + \sum_{k=l}^T (e_i - \gamma_{ik}) + \sum_{c \in \mathcal{C}(y)} \beta_c \sum_{o \in c} p_o^u + \sum_{j \in \mathcal{DP}(i)} \sum_{k=l-L_j}^{T-L_j} \gamma_{jk} g_{ji} \right] X_{il} \left. \right\} \\
& + \sum_{c \in \mathcal{C}(y)} \beta_c \left[r(o_c^f) - \sum_{l=1}^{l(o_c^l)} c_l \right] - \sum_{i=1}^N \sum_{l=1}^T \sum_{k=1}^T (e_i - \gamma_{ik}) D_{il} \\
& + \sum_{i=1}^N \sum_{l=1}^T \sum_{j \in \mathcal{DS}(i)} \sum_{k=l}^T \gamma_{il} g_{ij} D_{j,l+L_j}
\end{aligned} \tag{4.20}$$

Annexe B

Expérimentations avec l'algorithme de lissage descendant

Dans la Section 4.4 du Chapitre 4, nous avons présenté les résultats numériques de nombreuses expérimentations réalisées avec plusieurs nomenclatures multi-niveaux. Parmi les trois algorithmes de lissage étudiés pour la validation des contraintes de nomenclature, l'algorithme descendant (suivant la logique MRP) s'est avéré inefficace pour la plupart des instances. C'est pourquoi nous avons présenté uniquement les résultats associés aux algorithmes glouton et ascendant, excepté pour la nomenclature 6x6-1, où nous avons voulu montrer le comportement de l'algorithme descendant. Néanmoins, nous présentons dans ce qui suit les résultats obtenus avec cet algorithme pour toutes les autres nomenclatures.

B.1 Nomenclature 6x6-2

Les résultats correspondants sont présentés dans le Tableau B.1. Comme pour la nomenclature 6x6-1, l'approche n'arrive pas, dans beaucoup de cas, à déterminer de solutions réalisables à part la solution initiale. Uniquement SP4 propose des solutions réalisables différentes du plan initial. C'est le cas des trois dernières instances avec $T = 5$.

Nous constatons à nouveau que les temps de calcul avec SP2 et SP4 (et aussi SP5) sont considérablement réduits par rapport à l'algorithme glouton et que ceux avec SP1 et SP3 sont fortement augmentés. Une fois de plus, la raison est la facilité pour construire une solution respectant la nomenclature sans respecter la capacité, en suivant la logique MRP. Néanmoins, il est ensuite très compliqué de réparer les contraintes de capacité, tout en respectant la nomenclature.

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)
1	12268	3.23	7.22	12268	3.23	1.06	12268	3.23	7.61	12268	3.23	7.46	12268	3.23	1.28
2	12268	3.23	7.69	12268	3.23	1.57	12268	3.23	8.05	12268	3.23	7.36	12268	3.23	1.52
3	12268	3.23	6.80	12268	3.23	1.37	12268	3.23	7.15	11979	0.80	6.99	12268	3.23	1.42
4	12268	3.23	6.86	12268	3.23	1.43	12268	3.23	6.99	12004	1.01	7.22	12268	3.23	1.42
5	12268	3.23	7.28	12268	3.23	1.37	12268	3.23	7.07	12004	1.01	7.30	12268	3.23	1.46
6	29708	1.16	22.53	29708	1.16	3.02	29708	1.16	49.19	29708	1.16	1.89	29708	1.16	3.92
7	29708	1.16	24.74	29708	1.16	2.2	29708	1.16	35.29	29708	1.16	1.69	29708	1.16	2.89
8	29708	1.16	19.75	29708	1.16	2.1	29708	1.16	32.79	29708	1.16	1.55	29708	1.16	2.79
9	29708	1.36	16.79	29708	1.36	1.95	29708	1.36	26.64	29708	1.36	1.54	29708	1.36	2.62
10	29708	1.48	20.5	29708	1.48	2.33	29708	1.48	33.39	29708	1.48	1.72	29708	1.48	3.27
11	-	-	16.27	-	-	7.92	-	-	28.93	-	-	18.24	-	-	21.12
12	67024	0.72	13.05	67024	0.72	8.00	67024	0.72	97.93	67024	0.72	5.8	67024	0.72	17.75
13	67024	0.73	14.3	67024	0.73	6.58	67024	0.73	59.62	67024	0.73	9.70	67024	0.73	15.99
14	67024	0.80	12.28	67024	0.80	3.33	67024	0.80	41.96	67024	0.80	6.75	67024	0.80	6.43
15	67024	0.85	18.33	67024	0.85	2.78	67024	0.85	33.55	67024	0.85	2.85	67024	0.85	9.12

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A												
1	12268	3.23	0.00	12268	3.23	0.00	12268	3.23	0.00	12268	3.23	0.00	12268	3.23	0.00
2	12268	3.23	0.00	12268	3.23	0.00	12268	3.23	0.00	12268	3.23	0.00	12268	3.23	0.00
3	12268	3.23	0.00	12268	3.23	0.00	12268	3.23	0.00	11979	0.80	0.00	12268	3.23	0.00
4	12268	3.23	0.00	12268	3.23	0.00	12268	3.23	0.00	12004	1.01	0.00	12268	3.23	0.00
5	12268	3.23	0.00	12268	3.23	0.00	12268	3.23	0.00	12004	1.01	0.00	12268	3.23	0.00
6	29708	1.15	0.00	29708	1.15	0.00	29708	1.15	0.00	29708	1.15	0.00	29708	1.15	0.00
7	29708	1.15	0.00	29708	1.15	0.00	29708	1.15	0.00	29708	1.15	0.00	29708	1.15	0.00
8	29708	1.29	0.00	29708	1.29	0.00	29708	1.29	0.00	29708	1.29	0.00	29708	1.29	0.00
9	29708	1.41	0.00	29708	1.41	0.00	29708	1.41	0.00	29708	1.41	0.00	29708	1.41	0.00
10	29708	1.52	0.00	29708	1.52	0.00	29708	1.52	0.00	29708	1.52	0.00	29708	1.52	0.00
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	67024	0.72	0.00	67024	0.72	0.00	67024	0.72	0.00	67024	0.72	0.00	67024	0.72	0.00
13	67024	-0.07	0.00												
14	67024	0.70	0.00	67024	0.70	0.00	67024	0.70	0.00	67024	0.70	0.00	67024	0.70	0.00
15	67024	0.73	0.00	67024	0.73	0.00	67024	0.73	0.00	67024	0.73	0.00	67024	0.73	0.00

TABLEAU B.1 – Résultats pour la nomenclature 6x6-2 avec l'algorithme descendant

Aucune amélioration de la borne supérieure n'est possible en résolvant le problème intégré.

B.2 Nomenclature 6x6-3

Les résultats sont présentés dans le Tableau B.2. Cette fois-ci, des solutions réalisables de meilleure qualité que les plans initiaux sont obtenues pour tous les groupes d'instances ($T = 5$, $T = 10$ et $T = 20$). Néanmoins, les solutions sont d'une qualité très nettement inférieure à celle des solutions déterminées avec l'algorithme glouton, les écarts moyens étant de 3.87%, 3.43%, 4.29%, 3.18% et 3.97% avec SP1, SP2, SP3, SP4 et SP5 respectivement. Une nouvelle fois, les temps de calcul associés à SP2 et SP4 sont significativement réduits (49.74% et 73.24% respectivement) et ceux associés à SP1 et SP3 augmentent fortement (84.46% et

81.82% respectivement), montrant ainsi la difficulté de réparer les contraintes de nomenclature, en respectant la capacité, sous une logique descendante.

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)
1	6812	5.83	3.83	6510	1.13	0.87	6636	3.09	5.67	6510	1.13	0.84	6653	3.36	0.94
2	6564	2.80	2.37	6539	2.41	0.97	6564	2.80	3.53	6539	2.41	0.33	6564	2.80	0.97
3	6584	3.67	1.35	6594	3.83	0.78	6629	4.38	1.71	6540	2.98	0.28	6585	3.68	0.75
4	6445	1.74	1.01	6507	2.72	0.71	6445	1.74	1.09	6448	1.78	0.30	6507	2.72	0.74
5	6397	0.98	3.52	6489	2.43	1.26	6397	0.98	3.70	6489	2.43	0.28	6493	2.49	1.21
6	14315	3.05	95.70	14145	1.83	17.60	14512	4.47	150.22	14157	1.91	1.82	14403	3.69	21.25
7	14290	3.55	71.96	14065	1.92	11.72	14512	5.16	209.61	14065	1.92	1.19	14356	4.03	14.77
8	14109	2.75	61.75	14152	3.07	7.31	14512	5.69	38.40	14152	3.07	1.14	14185	3.31	7.15
9	14340	4.94	6.93	14195	3.88	7.23	14337	4.92	8.96	14195	3.88	1.05	14116	3.30	7.69
10	14239	4.58	6.77	14239	4.58	6.70	14436	6.03	10.41	14239	4.58	0.95	14388	5.68	6.52
11	30144	4.02	11.74	30144	4.02	13.42	30144	4.02	104.24	30144	4.02	11.17	30144	4.02	10.57
12	30144	4.81	53.45	30072	4.56	12.05	30144	4.81	59.05	29961	4.18	11.19	30144	4.81	7.07
13	30038	5.09	64.42	29990	4.93	9.02	30038	5.09	72.75	29933	4.73	14.68	30031	5.07	8.11
14	30016	5.16	20.60	29979	5.03	4.47	30076	5.37	35.66	29759	4.26	5.24	30060	5.31	8.18
15	29935	5.00	49.11	29985	5.18	5.50	30144	5.74	92.76	29776	4.44	4.21	30029	5.33	4.85

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A												
1	6812	7.53	0.00	6510	2.76	0.00	6571	3.73	0.98	6434	1.56	1.17	6335*	0.00	4.78
2	6441	1.67	1.87	6409	1.17	1.99	6564	3.61	0.00	6431	1.52	1.65	6410	1.18	2.35
3	6577	3.82	0.11	6509	2.75	1.29	6385	0.79	3.68	6436	1.59	1.59	6385	0.79	3.04
4	6397	0.98	0.74	6428	1.47	1.21	6445	1.74	0.00	6397	0.98	0.79	6397	0.98	1.69
5	6397	0.98	0.00	6397	0.98	1.42	6397	0.98	0.00	6431	1.52	0.89	6397	0.98	1.48
6	14315	4.60	0.00	14083	2.91	0.44	14407	5.28	0.72	14083	2.91	0.52	14122	3.19	1.95
7	14290	5.19	0.00	13999	3.05	0.47	14152	4.17	2.48	14037	3.33	0.20	14279	5.11	0.54
8	14109	4.14	0.00	14013	3.43	0.98	14394	6.24	0.81	14016	3.45	0.96	14185	4.70	0.00
9	14038	3.63	2.11	14020	3.50	1.23	14071	3.88	1.86	14035	3.61	1.13	13698	1.12	2.96
10	13776	1.70	3.25	13698	1.12	3.80	13645	0.73	5.48	13645	0.73	4.17	13754	1.54	4.41
11	30144	4.95	0.00	30113	4.84	0.10	30144	4.95	0.00	30144	4.95	0.00	30144	4.95	0.00
12	30144	5.90	0.00	29901	5.04	0.57	30144	5.90	0.00	29702	4.35	0.86	29821	4.76	1.07
13	30038	5.87	0.00	29864	5.25	0.42	30038	5.87	0.00	29763	4.90	0.57	29937	5.51	0.31
14	30016	5.82	0.00	29171	2.84	2.70	30076	6.03	0.00	29759	4.91	0.00	29873	5.32	0.62
15	29935	5.53	0.00	29614	4.40	1.24	30144	6.27	0.00	29763	4.93	0.04	29863	5.28	0.55

TABLEAU B.2 – Résultats pour la nomenclature 6x6-3 avec l’algorithme descendant

Plusieurs améliorations de la borne supérieure sont obtenues en résolvant le problème intégré et, hormis la stratégie SP1, les écarts moyens avec la solution du solveur sont réduits. La stratégie la plus performante est SP4, avec $\bar{G} = 3.02\%$, mais SP5 devient compétitive avec un écart moyen de 3.03% , grâce à une amélioration moyenne de UB de 1.72% (la plus élevée).

B.3 Nomenclature 6x6-4

Les résultats sont présentés dans le Tableau B.3. Des solutions différentes du plan initial sont proposées (tout comme pour la nomenclature 6x6-3), mais la qualité est significativement

inférieure à celle obtenue avec l'algorithme glouton. Une fois de plus, le temps de calcul associé aux stratégies SP2, SP4 et SP5 est réduit (entre 28.75% et 35.52%) par rapport à l'algorithme glouton, mais il augmente fortement (entre 420.69% et 433.95%) avec SP1 et SP3. La stratégie qui présente la meilleure performance est SP4, avec $\bar{G} = 3.71\%$ (représentant une réduction entre 29.60% et 38.27% de l'écart par rapport aux autres stratégies) et $\bar{t} = 2.49$ secondes. La stratégie la moins performante en qualité de solution est SP5, avec $\bar{G} = 6.01\%$.

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)
1	3646	2.27	1.68	3632	1.88	1.13	3731	4.66	2.39	3616	1.43	0.36	3908	9.62	0.86
2	3594	1.41	1.57	3584	1.13	0.90	3639	2.68	5.38	3608	1.81	0.44	3678	3.78	0.99
3	3563	1.02	1.52	3570	1.22	0.82	3563	1.02	2.42	3598	2.01	0.47	3591	1.81	0.95
4	3563	1.14	1.52	3563	1.14	0.80	3563	1.14	2.86	3621	2.78	0.37	3563	1.14	0.87
5	3563	1.14	1.58	3604	2.30	0.84	3563	1.14	2.78	3569	1.31	0.44	3563	1.14	0.89
6	8220	9.91	1.50	8220	9.91	1.42	8220	9.91	9.63	7905	5.70	1.44	8220	9.91	1.88
7	8220	10.26	1.79	8220	10.26	1.31	8220	10.26	3.25	7897	5.93	1.16	8220	10.26	1.50
8	8220	10.39	3.59	8220	10.39	2.17	8220	10.39	17.98	7796	4.70	1.25	8220	10.39	2.64
9	8220	11.07	11.71	8220	11.07	1.79	8220	11.07	16.69	7738	4.55	1.40	8220	11.07	2.17
10	7826	6.37	7.21	7649	3.97	1.54	8220	11.73	12.72	7721	4.95	0.98	7960	8.20	1.86
11	17680	9.76	63.45	17680	9.76	5.54	17680	9.76	93.37	16839	4.54	6.83	17680	9.76	11.37
12	16812	4.66	61.01	16638	3.57	5.43	17028	6.00	74.35	16669	3.77	5.93	16639	3.58	7.85
13	16576	3.23	58.29	16611	3.44	5.60	16576	3.23	57.61	16620	3.50	5.66	16611	3.44	8.69
14	16561	3.13	41.27	16561	3.13	5.56	16581	3.26	52.93	16712	4.07	5.57	16589	3.31	5.46
15	16561	3.24	36.82	16561	3.24	4.86	16561	3.24	37.24	16780	4.60	5.09	16475	2.70	4.77

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A	UB	G	A	UB	G	A	UB	G	A	UB	G	A
1	3646	3.49	0.00	3570	1.33	1.71	3563	1.14	4.50	3570	1.33	1.27	3571	1.36	8.62
2	3594	2.02	0.00	3578	1.56	0.17	3563	1.14	2.09	3570	1.33	1.05	3592	1.96	2.34
3	3563	1.14	0.00	3559	1.02	0.31	3563	1.14	0.00	3549	0.74	1.36	3563	1.14	0.78
4	3563	1.14	0.00	3563	1.14	0.00	3563	1.14	0.00	3579	1.59	1.16	3563	1.14	0.00
5	3531	0.23	0.90	3531	0.23	2.03	3563	1.14	0.00	3553	0.85	0.45	3549	0.74	0.39
6	8220	10.53	0.00	8043	8.15	2.15	8220	10.53	0.00	7740	4.07	2.09	8220	10.53	0.00
7	7861	5.84	4.37	8018	7.96	2.46	7861	5.84	4.37	7725	4.01	2.18	8008	7.82	2.58
8	8055	8.75	2.01	7677	3.65	6.61	8220	10.98	0.00	7648	3.25	1.90	7919	6.91	3.66
9	7664	4.17	6.76	7631	3.72	7.17	7680	4.39	6.57	7563	2.80	2.26	7586	3.11	7.71
10	7654	4.04	2.20	7606	3.38	0.56	7802	6.05	5.09	7536	2.43	2.40	7606	3.38	4.45
11	17680	9.55	0.00	16641	3.11	5.88	17680	9.55	0.00	16620	2.98	1.30	16790	4.03	5.03
12	16812	4.21	0.00	16638	3.13	0.00	17028	5.55	0.00	16620	3.02	0.29	16639	3.14	0.00
13	16576	2.91	0.00	16581	2.94	0.18	16576	2.91	0.00	16620	3.18	0.00	16543	2.71	0.41
14	16529	2.69	0.19	16441	2.14	0.72	16581	3.01	0.00	16641	3.39	0.42	16425	2.04	0.99
15	16404	1.75	0.95	16404	1.75	0.95	16404	1.75	0.95	16612	3.04	1.00	16420	1.85	0.33

TABLEAU B.3 – Résultats pour la nomenclature 6x6-4 avec l'algorithme descendant

Concernant le problème intégré, d'importantes améliorations sont obtenues avec toutes les stratégies, diminuant significativement les écarts moyens sur la valeur des solutions trouvées par le solveur. La stratégie qui génère la plus grande amélioration moyenne est SP5, avec $\bar{A} = 2.49\%$ et un écart moyen de 3.46%. Cependant, la stratégie la plus performante reste SP4, avec $\bar{G} = 2.54\%$, grâce à une amélioration moyenne de UB de 1.28%.

B.4 Nomenclature 10x10-1

Les résultats sont présentés dans le Tableau B.4. Nous constatons que les solutions sont exactement les mêmes que celles obtenues avec l’algorithme glouton, celles-ci correspondant au plan initial. Cependant, l’algorithme descendant permet de réduire significativement les temps de calcul (entre 69.53% et 81.76% de réduction en moyenne), excepté pour SP1, où le temps augmente de 163.45% en moyenne (comportement aussi observé avec les nomenclatures à 6 produits). La procédure la plus rapide est SP3, avec $\bar{t} = 11.43$ secondes, et la plus lente est SP5, avec $\bar{t} = 40.44$ secondes. Les temps sont élevés pour les instances avec 20 périodes, car plus de mouvements sont nécessaires pour lisser la production.

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)
1	12910*	0.00	3.17	12910*	0.00	1.90	12910*	0.00	2.31	12910*	0.00	1.92	12910*	0.00	2.02
2	12910*	0.00	1.84	12910*	0.00	1.42	12910*	0.00	1.75	12910*	0.00	1.45	12910*	0.00	1.37
3	12910*	0.00	1.52	12910*	0.00	1.34	12910*	0.00	1.54	12910*	0.00	1.31	12910*	0.00	1.32
4	12910*	0.00	1.49	12910*	0.00	1.33	12910*	0.00	1.51	12910*	0.00	1.33	12910*	0.00	1.32
5	38588	0.24	11.28	38588	0.24	4.53	38588	0.24	5.46	38588	0.24	2.42	38588	0.24	4.17
6	38588	0.25	2.54	38588	0.25	4.77	38588	0.25	8.71	38588	0.25	3.23	38588	0.25	6.79
7	38588	0.29	18.54	38588	0.29	2.71	38588	0.29	6.48	38588	0.29	2.20	38588	0.29	2.52
8	38588	0.34	176.23	38588	0.34	10.25	38588	0.34	78.28	38588	0.34	6.94	38588	0.34	11.49
9	89368	0.20	6.96	89368	0.20	144.94	89368	0.20	7.77	89368	0.20	62.81	89368	0.20	179.40
10	89368	0.22	5.93	89368	0.22	93.19	89368	0.22	5.58	89368	0.22	45.63	89368	0.22	115.15
11	89368	0.30	16.20	89368	0.30	88.91	89368	0.30	11.13	89368	0.30	37.61	89368	0.30	86.69
12	89368	0.28	6.85	89368	0.28	62.23	89368	0.28	6.58	89368	0.28	30.89	89368	0.28	72.98

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A												
1	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00
2	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00
3	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00	12910*	0.00	0.00
4	12910	-0.14	0.00												
5	38588	-	0.00	38588	-	0.00	38588	-	0.00	38588	-	0.00	38588	-	0.00
6	38588	-	0.00	38588	-	0.00	38588	-	0.00	38588	-	0.00	38588	-	0.00
7	38588	-	0.00	38588	-	0.00	38588	-	0.00	38588	-	0.00	38588	-	0.00
8	38588	-	0.00	38588	-	0.00	38588	-	0.00	38588	-	0.00	38588	-	0.00
9	89368	-	0.00	89368	-	0.00	89368	-	0.00	89368	-	0.00	89368	-	0.00
10	89368	-	0.00	89368	-	0.00	89368	-	0.00	89368	-	0.00	89368	-	0.00
11	89368	-	0.00	89368	-	0.00	89368	-	0.00	89368	-	0.00	89368	-	0.00
12	89368	-	0.00	89368	-	0.00	89368	-	0.00	89368	-	0.00	89368	-	0.00

TABLEAU B.4 – Résultats pour la nomenclature 10x10-1 avec l’algorithme descendant

Une nouvelle fois, aucune amélioration de la borne supérieure n’est réalisée en résolvant le problème intégré, mais la solution associée à la quatrième instance est meilleure que celle obtenue par le solveur.

B.5 Nomenclature 10x10-2

Les résultats sont présentés dans le Tableau B.5. Seule SP4 conduit à des solutions différentes du plan initial, et ceci seulement pour les deux dernières instances avec $T = 5$. C'est donc SP4 la stratégie présentant le meilleur écart moyen par rapport à la solution du solveur commercial, avec $\bar{G} = 5.74\%$. Dans les autres cas, l'écart est de 6.20%. Les écarts moyens augmentent entre 15.03% et 37.17% par rapport à ceux de l'algorithme glouton. En ce qui concerne le temps de calcul, SP4 est également la stratégie la plus performante avec $\bar{t} = 4.53$ secondes. De manière similaire à ce que nous avons observé pour la nomenclature 10x10-1, les temps moyens de calcul avec l'algorithme descendant diminuent avec SP2, SP3, SP4 et SP5 (entre 11.92% et 31.43%) et augmentent fortement avec SP1 (127.40%).

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)	UB	G	t(s)
1	13476	7.34	3.02	13476	7.34	2.69	13476	7.34	4.15	13476	7.34	2.06	13476	7.34	2.77
2	13476	8.44	17.72	13476	8.44	3.64	13476	8.44	17.99	13476	8.44	2.64	13476	8.44	3.61
3	13476	8.44	13.87	13476	8.44	3.24	13476	8.44	13.72	13476	8.44	2.48	13476	8.44	3.09
4	13476	8.84	10.61	13476	8.84	3.06	13476	8.84	10.35	13021	5.17	2.23	13476	8.84	2.96
5	13476	8.84	10.04	13476	8.84	2.75	13476	8.84	5.96	13081	5.65	2.06	13476	8.84	2.64
6	32060	4.87	32.97	32060	4.87	11.31	32060	4.87	24.28	32060	4.87	7.25	32060	4.87	17.27
7	32060	5.09	6.52	32060	5.09	2.76	32060	5.09	4.05	32060	5.09	2.09	32060	5.09	3.00
8	32060	4.96	9.59	32060	4.96	2.95	32060	4.96	4.11	32060	4.96	2.17	32060	4.96	3.19
9	32060	5.71	72.90	32060	5.71	6.26	32060	5.71	30.64	32060	5.71	5.79	32060	5.71	10.35
10	32060	5.96	57.01	32060	5.96	3.91	32060	5.96	23.89	32060	5.96	3.45	32060	5.96	5.26
11	69112	4.42	11.93	69112	4.42	12.71	69112	4.42	3.51	69112	4.42	7.21	69112	4.42	13.86
12	69112	4.68	8.49	69112	4.68	6.16	69112	4.68	3.32	69112	4.68	5.12	69112	4.68	7.26
13	69112	4.76	12.18	69112	4.76	12.47	69112	4.76	3.38	69112	4.76	7.44	69112	4.76	17.68
14	69112	5.21	5.70	69112	5.21	13.90	69112	5.21	3.34	69112	5.21	7.77	69112	5.21	17.17
15	69112	5.41	11.30	69112	5.41	13.22	69112	5.41	3.55	69112	5.41	8.19	69112	5.41	19.17
Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A	UB	G	A	UB	G	A	UB	G	A	UB	G	A
1	13476	7.31	0.00	13476	7.31	0.00	13476	7.31	0.00	13476	7.31	0.00	13476	7.31	0.00
2	13476	8.84	0.00	13476	8.84	0.00	13476	8.84	0.00	13476	8.84	0.00	13476	8.84	0.00
3	13476	8.18	0.00	13476	8.18	0.00	13476	8.18	0.00	13476	8.18	0.00	13476	8.18	0.00
4	13476	8.84	0.00	13476	8.84	0.00	13476	8.84	0.00	13021	5.17	0.00	13476	8.84	0.00
5	13476	10.03	0.00	13476	10.03	0.00	13476	10.03	0.00	13081	6.80	0.00	13476	10.03	0.00
6	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00
7	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00
8	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00
9	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00
10	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00	32060	-	0.00
11	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00
12	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00
13	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00
14	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00
15	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00	69112	-	0.00

TABLEAU B.5 – Résultats pour la nomenclature 10x10-2 avec l'algorithme descendant

En résolvant le problème intégré, aucune amélioration de la borne supérieure n'est réalisée, mais l'approche reste avantageuse, car le solveur ne résout pas la plupart des instances.

B.6 Nomenclature 10x10-3

Les résultats sont présentés dans le Tableau B.6. Tout comme l'algorithme glouton, l'algorithme descendant ne permet pas d'obtenir des solutions réalisables différentes du plan initial pour la plupart des instances. Une nouvelle fois, ce sont les trois dernières instances avec $T = 5$ qui peuvent être améliorées par rapport au plan initial. Néanmoins, l'algorithme descendant ne permet pas de les résoudre de façon optimale. La qualité de la solution est globalement détériorée par rapport aux résultats associés à l'algorithme glouton, avec une augmentation de l'écart moyen \bar{G} entre 11.35% et 31.87%. Les temps de calcul sont aussi plus importants, sauf pour SP4. L'augmentation du temps est surtout significative avec SP1, où l'on passe de 4.16 secondes à 14.28 secondes. La stratégie la plus performante est SP4, avec $\bar{G} = 10.47\%$ et $\bar{t} = 4.41$ secondes.

Aucune amélioration n'est réalisée en résolvant le problème intégré.

Problème à séquence fixée															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	t(s)												
1	9344	14.02	6.95	9344	14.02	2.84	9344	14.02	3.53	9344	14.02	0.86	9344	14.02	2.89
2	9344	14.26	8.49	9344	14.26	3.09	9344	14.26	3.66	9344	14.26	0.83	9344	14.26	3.05
3	8853	9.57	17.70	8763	8.45	3.20	8924	10.45	6.45	8707	7.76	0.87	8798	8.89	3.10
4	8908	10.25	16.24	8763	8.45	3.94	8962	10.92	6.60	8707	7.76	0.95	8962	10.92	3.88
5	9344	15.64	21.51	9344	15.64	3.93	9344	15.64	7.48	8707	7.76	0.92	8803	8.95	3.91
6	20412	9.98	14.97	20412	9.98	3.39	20412	9.98	10.13	20412	9.98	4.68	20412	9.98	4.16
7	20412	10.78	18.93	20412	10.78	3.63	20412	10.78	12.19	20412	10.78	4.71	20412	10.78	4.23
8	20412	10.39	18.99	20412	10.39	3.70	20412	10.39	14.40	20412	10.39	5.04	20412	10.39	4.62
9	20412	11.15	17.13	20412	11.15	7.17	20412	11.15	14.28	20412	11.15	4.51	20412	11.15	7.79
10	20412	11.53	14.01	20412	11.53	3.11	20412	11.53	6.52	20412	11.53	4.63	20412	11.53	3.46
11	44088	10.01	12.12	44088	10.01	21.31	44088	10.01	4.63	44088	10.01	12.76	44088	10.01	28.39
12	44088	10.21	11.02	44088	10.21	13.06	44088	10.21	4.41	44088	10.21	8.18	44088	10.21	12.83
13	44088	10.35	10.72	44088	10.35	7.61	44088	10.35	4.25	44088	10.35	6.19	44088	10.35	9.80
14	44088	10.72	13.82	44088	10.72	6.87	44088	10.72	4.40	44088	10.72	5.80	44088	10.72	8.03
15	44088	10.39	11.61	44088	10.39	5.83	44088	10.39	4.41	44088	10.39	5.23	44088	10.39	6.20

Problème intégré															
Instance	SP1			SP2			SP3			SP4			SP5		
	UB	G	A												
1	9344	15.64	0.00	9344	15.64	0.00	9344	15.64	0.00	8656	7.13	7.36	9344	15.64	0.00
2	9344	15.64	0.00	9344	15.64	0.00	9344	15.64	0.00	9344	15.64	0.00	9344	15.64	0.00
3	8853	9.57	0.00	8763	8.45	0.00	8924	10.45	0.00	8707	7.76	0.00	8798	8.89	0.00
4	8908	10.25	0.00	8763	8.45	0.00	8962	10.92	0.00	8707	7.76	0.00	8962	10.92	0.00
5	9344	15.64	0.00	9344	15.64	0.00	9344	15.64	0.00	8707	7.76	0.00	8803	8.95	0.00
6	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00
7	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00
8	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00
9	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00
10	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00	20412	-	0.00
11	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00
12	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00
13	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00
14	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00
15	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00	44088	-	0.00

TABLEAU B.6 – Résultats pour la nomenclature 10x10-3 avec l'algorithme descendant

Annexe C

Intégration de la planification de la distribution

En management de la chaîne logistique, la gestion de la production est une étape fondamentale car elle permet de fabriquer les produits nécessaires pour satisfaire les demandes dans les délais au moindre coût possible. Néanmoins, comme nous l'avons discuté dans le Chapitre 1, la coordination de cette étape avec celles d'approvisionnement, de distribution et des ventes est primordiale pour garantir une prise de décisions cohérentes, évitant les effets perturbateurs sur les flux logistiques comme l'« *effet coup de fouet* »[145].

Nous nous intéressons à l'intégration des flux de distribution dans une chaîne logistique avec plusieurs sites de production, chacun fabriquant certains composants d'une nomenclature de produits à plusieurs niveaux. La satisfaction des demandes internes de chaque site de production peut aussi être associée à des flux d'approvisionnement. Nous considérons le scénario où des demandes externes existent pour chaque élément de la nomenclature. Ce problème est une généralisation du problème étudié dans le Chapitre 4. Il s'agit du problème intégré de dimensionnement de lots, d'ordonnancement et de distribution dans une chaîne logistique multi-sources et multi-sites avec des ateliers de production multi-produits et multi-ressources de type job-shop.

Un exemple d'un système avec plusieurs sources et sites de production est illustré dans la Figure C.1. Les sites 7, 8 et 9 fabriquent des produits pour les sites 2, 3, 4, 5 et 6, et ceux-ci produisent pour le site 1. Une nomenclature associée à ce réseau multi-sources et multi-sites est présentée dans la Figure C.2.

Les hypothèses du problème sont les suivantes :

- Un produit peut être fabriqué sur différents sites de production, c'est-à-dire que les sites sont en concurrence,

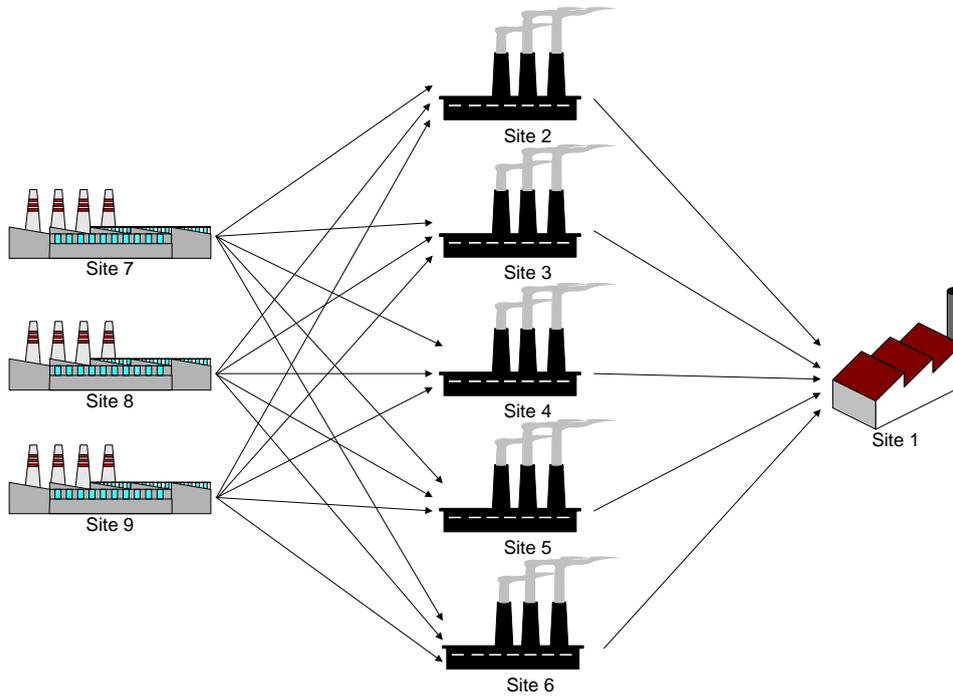


FIGURE C.1 – Exemple de réseau multi-sources et multi-sites

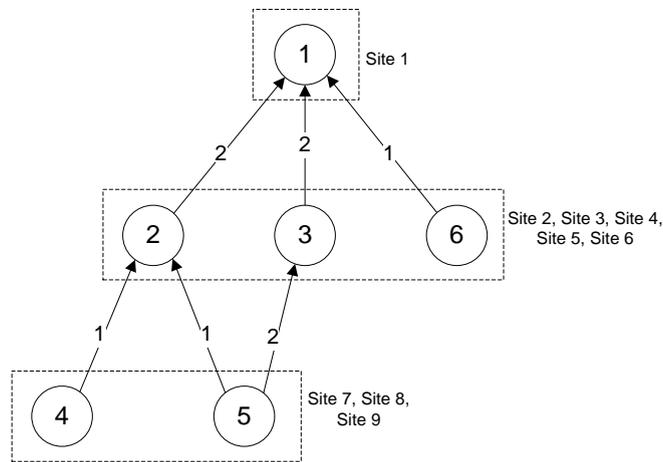


FIGURE C.2 – Exemple de nomenclature de produits pour un chaîne logistique multi-sources et multi-sites

- La capacité de production varie par période et par site de production,
- Les durées opératoires et les temps de setup dépendent du site de production,

- Les coûts de production, les coûts de setup et les coûts de stockage dépendent du site de production,
- Les coûts de transport dépendent du site de production et du site de livraison,
- Les délais de livraison sont identiques entre tous les sites.
- Les inventaires initiaux de tous les produits dans tous les sites sont égaux à zéro,
- La rupture de stock n'est pas autorisée,
- La préemption des opérations n'est pas autorisée,

Comme les délais de livraison sont identiques, ils peuvent être comptabilisés dans les délais d'obtention des produits, comme dans le problème multi-niveaux, mono-source et mono-site résolu dans le Chapitre 4.

Le problème intégré peut être décomposé en :

- Un problème de planification globale de la production et de la distribution, décidant les quantités de production par produit, par période et par site et les flux de distribution ou d'approvisionnement entre sites,
- Plusieurs problèmes d'ordonnancement de la production (un par site), affectant les opérations aux ressources et décidant de leurs dates de début et de fin.

Comme les problèmes d'ordonnancement diffèrent entre les sites de production, un graphe disjonctif différent par site peut être utilisé pour modéliser chaque problème d'ordonnancement.

Dans ce qui suit, nous explicitons la fonction objectif et les contraintes du problème qui changent par rapport au modèle mathématique du Chapitre 4. Nous proposons deux types de formulations. La première est basée sur les flux de matières par produit et la deuxième suit les flux de matières par site de production.

Fonction objectif :

La différence entre la nouvelle fonction objectif et celle du problème multi-niveaux, mono-source et mono-site est que maintenant (i) les tailles des lots sont désagrégées, soit par rapport aux flux entre sites, soit par rapport aux flux par produit, (ii) les coûts de setup, de production et de stockage dépendent du site de production et (iii) les coûts de transport sont considérés.

En suivant les flux de matières par produit, la fonction objectif est la suivante :

$$\min \sum_{s=1}^S \sum_{i=1}^N \sum_{l=1}^T \left(c_{i,s}^{set} Y_{i,l,s} + \sum_{s' \in \mathcal{SPS}(i)} (c_{i,s}^p + c_{i,s,s'}^t) X_{i,l,s,s'} + c_{i,s}^{inv} I_{i,l,s} \right) \quad (C.1)$$

où $X_{i,l,s,s'}$ est la taille de lot du produit i fabriqué à la période l dans le site s et destiné au site s' , $Y_{i,l,s}$ est la variable binaire de setup du produit i à la période l au site s , $I_{i,l,s}$ est le niveau de stock du produit i à la période l au site s , $c_{i,s}^{set}$ est le coût de setup du produit i au site s , $c_{i,s}^p$ est le coût unitaire de production du produit i au site s , $c_{i,s,s'}^t$ est le coût unitaire de transport du produit i depuis le site s jusqu'au site s' , $c_{i,s}^{inv}$ est le coût unitaire de stockage du produit i au site s et $\mathcal{SPS}(i)$ est l'ensemble des sites de production destinataires du produit i .

En considérant les flux de matières par site de production, la fonction objectif peut être modélisée de la manière suivante :

$$\min \sum_{s=1}^S \sum_{i=1}^N \sum_{l=1}^T \left(c_{i,s}^{set} Y_{i,l,s} + \sum_{s_p \in \mathcal{SPP}(i)} \sum_{s_s \in \mathcal{SPS}(i)} (c_{i,s}^p + c_{i,s,s_s}^t) X_{i,l,s_p,s_s} + c_{i,s}^{inv} I_{i,l,s} \right) \quad (\text{C.2})$$

où $\mathcal{SPP}(i)$ est l'ensemble des sites de production pouvant livrer des composants au produit i (sites source ou sites de provenance). Le site de provenance s_p est un nouvel indice à considérer dans la taille de lot.

Contraintes d'équilibre des stocks :

À travers les flux de matières par produit, les contraintes d'équilibre des stocks peuvent être définies de la façon suivante :

$$\sum_{s=1}^S I_{i,l,s} = \sum_{s=1}^S I_{i,l-1,s} + \sum_{s=1}^S \sum_{s' \in \mathcal{SPS}(i)} X_{i,l,s,s'} - d_{i,l} - \sum_{s' \in \mathcal{SPS}(i)} \sum_{j \in \mathcal{DS}(i)} g_{ij} \sum_{s'' \in \mathcal{SPS}(j)} X_{j,l+L_j,s',s''} \quad (\text{C.3})$$

$$\forall i \in \{1, \dots, N\}, \forall l \in \{1, \dots, T\}$$

où $\mathcal{DS}(i)$ est l'ensemble des successeurs directs du produit i dans la nomenclature, g_{ij} est le nombre d'unités de i nécessaires pour fabriquer une unité du produit j , $d_{i,l}$ est la demande externe du produit i à la période l et L_j est le délai de disponibilité (délai d'obtention plus délai de livraison) du produit j .

En raisonnant par site de production, nous devons ajouter le site de provenance comme nouvel indice pour obtenir les contraintes suivantes :

$$\begin{aligned}
I_{i,l,s} &= I_{i,l-1,s} + \sum_{s_p \in SPP(i)} \sum_{s_s \in SPs(i)} X_{i,l,s_p,s,s_s} - d_{i,l,s} - \sum_{s'_s \in SPs(j)} \sum_{s_s \in SPs(i)} \sum_{j \in DS(i)} g_{ij} X_{j,l+L_j,s,s_s,s'_s} \\
&\forall i \in \{1, \dots, N\}, \forall l \in \{1, \dots, T\}, \forall s \in \{1, \dots, S\}
\end{aligned} \tag{C.4}$$

Non seulement les tailles des lots sont désagrégées par site source s_p , mais aussi la demande externe est désagrégée par site de production s . Une partie de la demande totale du produit i à la période l est affectée à chaque site de production s .

Contraintes de capacité détaillées pour une séquence fixée :

Nous utilisons les contraintes de capacité des modèles avec séquence fixée présentés dans les chapitres 3 et 4, mais nous ajoutons un nouvel indice associé au site de production. En effet, comme chaque site est indépendant, des contraintes indépendantes par site sont à considérer.

Les contraintes de capacité détaillées pour une séquence fixée correspondant aux flux de matières par produit sont les suivantes :

$$r(o_c^f) + \sum_{o \in c} \sum_{s_s \in SPs(i(o))} (p_o^u X_{i(o),l(o),s,s_s} + s_o^t Y_{i(o),l(o),s}) \leq \sum_{l=1}^{l(o_c^f)} c_{l,s} \quad \forall c \in \mathcal{C}(y, s) \tag{C.5}$$

où $r(o_c^f)$ est la date de disponibilité de la première opération o_c^f du chemin c ($r(o)$ est calculée avec (3.13)-(3.16), comme expliqué dans la Section 3.2.3 du Chapitre 3), o_c^l est la dernière opération du chemin c , $c_{l,s}$ est la capacité de production à la période l au site de production s et $\mathcal{C}(y, s)$ est l'ensemble des chemins du graphe conjonctif associé à la séquence fixée y et au site de production s .

En raisonnant par site de production, les contraintes de capacité détaillées sont modélisées avec l'expression suivante :

$$r(o_c^f) + \sum_{o \in c} \sum_{s_p \in SPP(i(o))} \sum_{s_s \in SPs(i(o))} (p_o^u X_{i(o),l(o),s_p,s,s_s} + s_o^t Y_{i(o),l(o),s}) \leq \sum_{l=1}^{l(o_c^f)} c_{l,s} \quad \forall c \in \mathcal{C}(y, s) \tag{C.6}$$

Contraintes de production :

Les contraintes liant les tailles des lots X_{i,l,s,s_s} et les variables binaires de setup $Y_{i,l,s}$, en suivant les flux de matières par produit, sont modélisées avec l'expression suivante :

$$\sum_{s_s \in SP_S(i)} X_{i,l,s,s_s} \leq \left(\sum_{k=l}^T D_{i,k} \right) Y_{i,l,s} \quad \forall i \in \{1, \dots, N\}, l \in \{1, \dots, T\}, s \in \{1, \dots, S\}, s_s \in \{1, \dots, S\} \quad (\text{C.7})$$

où $D_{i,l}$ est la demande totale (somme des demandes externe et interne) du produit i à la période l , calculée récursivement avec l'équation suivante :

$$D_{i,l} = d_{i,l} + \sum_{j \in DS(i)} g_{ij} D_{i,l+L_j} \quad \forall i \in \{1, \dots, N\}, l \in \{1, \dots, T\} \quad (\text{C.8})$$

En suivant les flux de matières par site de production, les contraintes de production sont modélisées comme suit :

$$\sum_{s_p \in SPP(i)} \sum_{s_s \in SP_S(i)} X_{i,l,s_p,s,s_s} \leq \left(\sum_{k=l}^T D_{i,k,s} \right) Y_{i,l,s} \quad \forall i \in \{1, \dots, N\}, l \in \{1, \dots, T\}, \quad (\text{C.9})$$

$$s_p \in \{1, \dots, S\}, s \in \{1, \dots, S\}, s_s \in \{1, \dots, S\}$$

La demande totale $D_{i,l,s}$ du produit i à la période l au site de production s est calculée de manière récursive avec l'équation suivante :

$$D_{i,l,s} = d_{i,l,s} + \sum_{s_s \in SP_S(i)} \sum_{j \in DS(i)} g_{ij} D_{i,l+L_j,s_s} \quad (\text{C.10})$$

$$\forall i \in \{1, \dots, N\}, l \in \{1, \dots, T\}, s \in \{1, \dots, S\}, s_s \in \{1, \dots, S\}$$

Les demandes totales calculées avec (C.8) et (C.10) représentent aussi les demandes d'échelon dans des contraintes d'équilibre des échelons stock. Nous pouvons en effet modéliser le problème en utilisant les notations d'échelon stock, comme dans les sections 4.2.3 et 4.2.4 du Chapitre 4. L'idée est une fois de plus de diminuer le nombre de variables de décision du modèle pour pouvoir utiliser la propriété de Wagner et Within [231] à travers l'algorithme de Wagelmans *et al.* [230] qui permet de résoudre un ensemble de problèmes de dimensionnement de

lots mono-produit sans contraintes de capacité. Suivant l'approche intégrée présentée dans les chapitres 3 et 4, une heuristique Lagrangienne permettrait de résoudre le problème de dimensionnement de lots, d'ordonnancement et de distribution avec séquence fixée. Puis, une méthode d'amélioration de la séquence permettrait de faire évoluer l'ordonnancement dans tous les sites de production.

Dans ce qui suit, nous proposons des modèles mathématiques intégrés et avec séquence fixée, en utilisant la notation classique et la notation d'échelon stock.

Modèle intégré avec notation classique et flux de matières par produit :

$$\min \sum_{s=1}^S \sum_{i=1}^N \sum_{l=1}^T \left(c_{i,s}^{set} Y_{i,l,s} + \sum_{s_s \in \mathcal{SPS}(i)} (c_{i,s}^p + c_{i,s,s_s}^t) X_{i,l,s,s_s} + c_{i,s}^{inv} I_{i,l,s} \right) \quad (\text{C.1})$$

s.c.

$$\begin{aligned} \sum_{s=1}^S I_{i,l,s} = & \sum_{s=1}^S I_{i,l-1,s} + \sum_{s=1}^S \sum_{s_s \in \mathcal{SPS}(i)} X_{i,l,s,s_s} - d_{i,l} \\ & - \sum_{s_s \in \mathcal{SPS}(i)} \sum_{j \in \mathcal{DS}(i)} g_{ij} \sum_{s' \in \mathcal{SPS}(j)} X_{j,l+L_j,s_s,s'} \quad \forall i, l \end{aligned} \quad (\text{C.29})$$

$$t(o') \geq t(o) + p_o^u X_{i(o),l(o),s(o),s_s(o)} + s_o^t Y_{i(o),l(o),s(o)} \quad \forall (o, o') \in \mathcal{A} \quad (\text{C.11})$$

$$\begin{cases} t(o) \geq t(o') + p_{o'}^u X_{i(o'),l(o'),s(o'),s_s(o')} + s_{o'}^t Y_{i(o'),l(o'),s(o')} \\ \text{ou} \\ t(o') \geq t(o) + p_o^u X_{i(o),l(o),s(o),s_s(o)} + s_o^t Y_{i(o),l(o),s(o)} \end{cases} \quad \forall (o, o') \in \mathcal{E} \quad (\text{C.12})$$

$$t_o + p_o^u X_{i(o),l(o),s(o),s_s(o)} + s_o Y_{i(o),l(o),s(o)} \leq \sum_{l=1}^{l(o)} c_{l,s} \quad \forall o \in \mathcal{L} \quad (\text{C.13})$$

$$t_o + p_o^u X_{i(o),l(o),s(o),s_s(o)} + s_o Y_{i(o),l(o),s(o)} \geq \sum_{l=1}^{l(o)-1} c_{l,s} \quad \forall o \in \mathcal{L} \quad (\text{C.14})$$

$$t_o \geq \sum_{l=1}^{l(o)-L_{i(o)}} c_{l,s} \quad \forall o \in \mathcal{F} \text{ et } L_{i(o)} > 0 \quad (\text{C.15})$$

$$\sum_{s_s \in \mathcal{SPS}(i)} X_{i,l,s,s_s} \leq \left(\sum_{k=l}^T D_{i,k} \right) Y_{i,l,s} \quad \forall i, l, s \quad (\text{C.7})$$

$$t_o \geq 0 \quad \forall o \quad (\text{3.12})$$

$$X_{i,l,s,s_s}, I_{i,l,s} \geq 0 \quad \forall i, l, s, s_s \quad (\text{C.16})$$

$$Y_{i,l,s} \in \{0, 1\} \quad \forall i, l, s \quad (\text{C.17})$$

$$X_{i,l,s,s_s} = 0 \quad \forall i; \forall l < M_i + L_i \quad (\text{C.18})$$

La fonction objectif (C.1) minimise l'ensemble des coûts (setup, production, transport et stockage). Les contraintes (C.29) garantissent l'équilibre des stocks. Les contraintes (C.11) per-

mettent de respecter les contraintes de précédence des gammes opératoires. Les contraintes (C.12) correspondent aux contraintes de précédence entre les opérations traitées sur la même ressource. Les contraintes de capacité (C.13), (C.14) et (C.15) limitent les dates de début des opérations, comme expliqué dans la Section 3.2 du Chapitre 3 avec les équations (3.6), (3.7) et (3.8). Les contraintes de production (C.7) lient les tailles des lots aux variables de setup. Les contraintes (3.12) forcent les dates de début des opérations à être strictement positives. Les contraintes (C.19) forcent les tailles des lots et les inventaires à être strictement positives. Les contraintes (C.17) forcent les variables de setup à prendre des valeurs binaires et, les contraintes (C.18) empêchent la fabrication des produits avant leur approvisionnement en composants, comme expliqué dans la Section 4.2 du Chapitre 4.

Modèle avec séquence fixée, notation classique et flux de matières par produit :

$$\min \sum_{s=1}^S \sum_{i=1}^N \sum_{l=1}^T \left(c_{i,s}^{set} Y_{i,l,s} + \sum_{s_s \in SP_S(i)} (c_{i,s}^p + c_{i,s,s_s}^t) X_{i,l,s,s_s} + c_{i,s}^{inv} I_{i,l,s} \right) \quad (\text{C.1})$$

s.c.

$$\begin{aligned} \sum_{s=1}^S I_{i,l,s} &= \sum_{s=1}^S I_{i,l-1,s} + \sum_{s=1}^S \sum_{s_s \in SP_S(i)} X_{i,l,s,s_s} - d_{i,l} \\ &\quad - \sum_{s_s \in SP_S(i)} \sum_{j \in DS(i)} g_{ij} \sum_{s' \in SP_S(j)} X_{j,l+L_j,s_s,s'} \quad \forall i, l \end{aligned} \quad (\text{C.29})$$

$$r(o_c^f) + \sum_{o \in c} \sum_{s_s \in SP_S(i(o))} (p_o^u X_{i(o),l(o),s,s_s} + s_o^t Y_{i(o),l(o),s}) \leq \sum_{l=1}^{l(o_c^l)} c_{l,s} \quad \forall c \in \mathcal{C}(y, s) \quad (\text{C.5})$$

$$\sum_{s_s \in SP_S(i)} X_{i,l,s,s_s} \leq \left(\sum_{k=l}^T D_{i,k} \right) Y_{i,l,s} \quad \forall i, l, s \quad (\text{C.7})$$

$$X_{i,l,s,s_s}, I_{i,l,s} \geq 0 \quad \forall i, l, s, s_s \quad (\text{C.19})$$

$$Y_{i,l,s} \in \{0, 1\} \quad \forall i, l, s \quad (\text{C.17})$$

La fonction objectif (C.1), les contraintes d'équilibre des stocks (C.29), les contraintes de production (C.7), les contraintes de non-négativité (C.19) et les contraintes (C.17) imposant des valeurs binaires aux variables de setup restent identiques à celles du modèle intégré. Néanmoins, les contraintes de précédence (C.11) et (C.12), les contraintes de capacité (C.13)-(C.15) et les contraintes de non-négativité des dates de début des opérations (3.12) sont remplacées par

les contraintes de capacité détaillées (C.5) associées à la séquence y fixée de chaque site de production s . Les valeurs des dates de disponibilité $r(o)$ sont calculées avec les équations (3.13)-(3.16), comme expliqué dans la Section 3.2.3 du Chapitre 3.

Modèle intégré avec notation classique et flux de matières par site de production :

$$\min \sum_{s=1}^S \sum_{i=1}^N \sum_{l=1}^T \left(c_{i,s}^{set} Y_{i,l,s} + \sum_{s_p \in SPP(i)} \sum_{s_s \in SPS(i)} (c_{i,s}^p + c_{i,s,s_s}^t) X_{i,l,s_p,s,s_s} + c_{i,s}^{inv} I_{i,l,s} \right) \quad (C.2)$$

s.c.

$$\begin{aligned} I_{i,l,s} = & I_{i,l-1,s} + \sum_{s_p \in SPP(i)} \sum_{s_s \in SPS(i)} X_{i,l,s_p,s,s_s} - d_{i,l,s} \\ & - \sum_{s'_s \in SPS(j)} \sum_{s_s \in SPS(i)} \sum_{j \in DS(i)} g_{ij} \sum_{s' \in SPS(j)} X_{j,l+L_j,s,s_s,s'} \end{aligned} \quad \forall i, l, s \quad (C.4)$$

$$t(o') \geq t(o) + p_o^u X_{i(o),l(o),s_p(o),s(o),s_s(o)} + s_o^t Y_{i(o),l(o),s(o)} \quad \forall (o, o') \in \mathcal{A} \quad (C.20)$$

$$\begin{cases} t(o) \geq t(o') + p_{o'}^u X_{i(o'),l(o'),s_p(o'),s(o'),s_s(o')} + s_{o'}^t Y_{i(o'),l(o'),s(o')} \\ \text{ou} \\ t(o') \geq t(o) + p_o^u X_{i(o),l(o),s_p(o),s(o),s_s(o)} + s_o^t Y_{i(o),l(o),s(o)} \end{cases} \quad \forall (o, o') \in \mathcal{E} \quad (C.21)$$

$$t_o + p_o^u X_{i(o),l(o),s_p(o),s(o),s_s(o)} + s_o Y_{i(o),l(o),s(o)} \leq \sum_{l=1}^{l(o)} c_{l,s} \quad \forall o \in \mathcal{L} \quad (C.22)$$

$$t_o + p_o^u X_{i(o),l(o),s_p(o),s(o),s_s(o)} + s_o Y_{i(o),l(o),s(o)} \geq \sum_{l=1}^{l(o)-1} c_{l,s} \quad \forall o \in \mathcal{L} \quad (C.23)$$

$$t_o \geq \sum_{l=1}^{l(o)-L_{i(o)}} c_{l,s} \quad \forall o \in \mathcal{F} \text{ et } L_{i(o)} > 0 \quad (C.24)$$

$$\sum_{s_p \in SPP(i)} \sum_{s_s \in SPS(i)} X_{i,l,s_p,s,s_s} \leq \left(\sum_{k=l}^T D_{i,k,s} \right) Y_{i,l,s} \quad \forall i, l, s \quad (C.9)$$

$$t_o \geq 0 \quad \forall o \quad (3.12)$$

$$X_{i,l,s_p,s,s_s}, I_{i,l,s} \geq 0 \quad \forall i, l, s_p, s, s_s \quad (C.25)$$

$$Y_{i,l,s} \in \{0, 1\} \quad \forall i, l, s \quad (C.17)$$

$$X_{i,l,s_p,s,s_s} = 0 \quad \forall i; \forall l < M_i + L_i \quad (C.26)$$

La fonction objectif (C.2) minimise la somme des coûts de setup, de production, de transport et de stockage. Les contraintes (C.4) modélisent l'équilibre des stocks. Les contraintes (C.20) et (C.21) correspondent aux contraintes de précedence entre les opérations. Les contraintes

(C.22)-(C.24) garantissent le respect de la capacité de production. Les contraintes (C.9) lient les deux types de variable de décision. Les contraintes (3.12) permettent de garantir la non-négativité des dates de début des opérations. Les contraintes (C.35) forcent les tailles des lots et les inventaires à être strictement positifs. Les contraintes (C.17) garantissent la nature binaire des variables de setup et, les contraintes (C.26) empêchent la fabrication d'un produit tant que son approvisionnement en composants n'est pas garanti.

Modèle avec séquence fixée, notation classique et flux de matières par site de production :

$$\min \sum_{s=1}^S \sum_{i=1}^N \sum_{l=1}^T \left(c_{i,s}^{set} Y_{i,l,s} + \sum_{s_p \in SPP(i)} \sum_{s_s \in SPS(i)} (c_{i,s}^p + c_{i,s,s_s}^t) X_{i,l,s_p,s,s_s} + c_{i,s}^{inv} I_{i,l,s} \right) \quad (C.2)$$

s.c.

$$\begin{aligned} I_{i,l,s} = & I_{i,l-1,s} + \sum_{s_p \in SPP(i)} \sum_{s_s \in SPS(i)} X_{i,l,s_p,s,s_s} - d_{i,l,s} \\ & - \sum_{s'_s \in SPS(j)} \sum_{s_s \in SPS(i)} \sum_{j \in DS(i)} g_{ij} \sum_{s' \in SPS(j)} X_{j,l+L_j,s,s,s'} \end{aligned} \quad \forall i, l, s \quad (C.4)$$

$$r(o_c^f) + \sum_{o \in c} \sum_{s_p \in SPP(i(o))} \sum_{s_s \in SPS(i(o))} (p_o^u X_{i(o),l(o),s_p,s,s_s} + s_o^t Y_{i(o),l(o),s}) \leq \sum_{l=1}^{l(o_c)} c_{l,s} \quad \forall c \in \mathcal{C}(y, s) \quad (C.6)$$

$$\sum_{s_p \in SPP(i)} \sum_{s_s \in SPS(i)} X_{i,l,s_p,s,s_s} \leq \left(\sum_{k=l}^T D_{i,k,s} \right) Y_{i,l,s} \quad \forall i, l, s \quad (C.9)$$

$$X_{i,l,s_p,s,s_s}, I_{i,l,s} \geq 0 \quad \forall i, l, s_p, s, s_s \quad (C.27)$$

$$Y_{i,l,s} \in \{0, 1\} \quad \forall i, l, s \quad (C.17)$$

La fonction objectif (C.2), les contraintes d'équilibre des stocks (C.4), les contraintes de production (C.9), les contraintes de non-négativité des tailles des lots et des inventaires (C.35) et les contraintes binaires des variables de setup (C.17) sont les mêmes que celles du modèles

intégré. En revanche, les contraintes de précédence, de capacité et de non-négativité des dates de début des opérations du modèle intégré sont remplacées par les contraintes de capacité détaillées (C.6) associées aux couples (y, s) .

Modèle intégré avec échelon stock et flux de matières par produit :

$$\min \sum_{s=1}^S \sum_{i=1}^N \sum_{l=1}^T \left(c_{i,s}^{set} Y_{i,l,s} + \sum_{s_s \in \mathcal{SPS}(i)} (c_{i,s}^p + c_{i,s,s_s}^t) X_{i,l,s,s_s} + e_i E_{i,l} \right) \quad (\text{C.28})$$

s.c.

$$E_{i,l} = E_{i,l-1} + \sum_{s=1}^S \sum_{s_s \in \mathcal{SPS}(i)} X_{i,l,s,s_s} - D_{i,l} \quad \forall i, l \quad (\text{C.29})$$

$$\sum_{j \in \mathcal{DS}(i)} g_{ij} E_{j,l+L_j} - E_{i,l} \leq 0 \quad \forall i, l \quad (\text{C.30})$$

$$t(o') \geq t(o) + p_o^u X_{i(o),l(o),s(o),s_s(o)} + s_o^t Y_{i(o),l(o),s(o)} \quad \forall (o, o') \in \mathcal{A} \quad (\text{C.11})$$

$$\begin{cases} t(o) \geq t(o') + p_{o'}^u X_{i(o'),l(o'),s(o'),s_s(o')} + s_{o'}^t Y_{i(o'),l(o'),s(o')} \\ \text{ou} \\ t(o') \geq t(o) + p_o^u X_{i(o),l(o),s(o),s_s(o)} + s_o^t Y_{i(o),l(o),s(o)} \end{cases} \quad \forall (o, o') \in \mathcal{E} \quad (\text{C.12})$$

$$t_o + p_o^u X_{i(o),l(o),s(o),s_s(o)} + s_o Y_{i(o),l(o),s(o)} \leq \sum_{l=1}^{l(o)} c_{l,s} \quad \forall o \in \mathcal{L} \quad (\text{C.13})$$

$$t_o + p_o^u X_{i(o),l(o),s(o),s_s(o)} + s_o Y_{i(o),l(o),s(o)} \geq \sum_{l=1}^{l(o)-1} c_{l,s} \quad \forall o \in \mathcal{L} \quad (\text{C.14})$$

$$t_o \geq \sum_{l=1}^{l(o)-L_{i(o)}} c_{l,s} \quad \forall o \in \mathcal{F} \text{ et } L_{i(o)} > 0 \quad (\text{C.15})$$

$$\sum_{s_s \in \mathcal{SPS}(i)} X_{i,l,s,s_s} \leq \left(\sum_{k=l}^T D_{i,k} \right) Y_{i,l,s} \quad \forall i, l, s \quad (\text{C.7})$$

$$t_o \geq 0 \quad \forall o \quad (\text{3.12})$$

$$X_{i,l,s,s_s}, E_{i,l} \geq 0 \quad \forall i, l, s, s_s \quad (\text{C.31})$$

$$Y_{i,l,s} \in \{0, 1\} \quad \forall i, l, s \quad (\text{C.17})$$

$$X_{i,l,s,s_s} = 0 \quad \forall i; \forall l < M_i + L_i \quad (\text{C.18})$$

La fonction objectif (C.28) minimise la somme des coûts de setup, de production, de transport et d'échelon stock. Pour ne pas introduire un nouvel indice dans les variables d'échelon

stock et maintenir un raisonnement uniquement par produit, le coût unitaire d'échelon stock est supposé être constant indépendamment du site de production. Ceci fait que l'on comptabilise l'échelon stock $E_{i,l}$ du produit i à la période l comme la somme des échelons stock du job (i, l) pour tous les sites de production. Les contraintes (C.29) garantissent l'équilibre des échelons stock sur l'ensemble des sites de production. Les contraintes (C.30) correspondent aux contraintes de nomenclature. Les contraintes (C.11)-(C.12) correspondent aux contraintes de précedence des opérations. Les contraintes (C.13)-(C.15) correspondent aux contraintes de capacité. Les contraintes de production (C.7) lient les tailles des lots aux variables de setup. Les contraintes (3.12) garantissent que les dates de début des opérations ne soient pas négatives. Les contraintes (C.31) assurent la non-négativité des tailles des lots et des échelons stock. Les contraintes (C.17) forcent les variables de setup à prendre des valeurs binaires et, les contraintes (C.18) empêchent la fabrication d'un produit avant qu'il ne soit approvisionné en composants.

Modèle avec séquence fixée, échelon stock et flux de matières par produit :

$$\min \sum_{s=1}^S \sum_{i=1}^N \sum_{l=1}^T \left(c_{i,s}^{set} Y_{i,l,s} + \sum_{s_s \in SP\mathcal{S}(i)} (c_{i,s}^p + c_{i,s,s_s}^t) X_{i,l,s,s_s} + e_i E_{i,l} \right) \quad (\text{C.28})$$

s.c.

$$E_{i,l} = E_{i,l-1} + \sum_{s=1}^S \sum_{s_s \in SP\mathcal{S}(i)} X_{i,l,s,s_s} - D_{i,l} \quad \forall i, l \quad (\text{C.29})$$

$$\sum_{j \in \mathcal{DS}(i)} g_{ij} E_{j,l+L_j} - E_{i,l} \leq 0 \quad \forall i, l \quad (\text{C.30})$$

$$r(o_c^f) + \sum_{o \in c} \sum_{s_s \in SP\mathcal{S}(i(o))} (p_o^u X_{i(o),l(o),s,s_s} + s_o^t Y_{i(o),l(o),s}) \leq \sum_{l=1}^{l(o_c^l)} c_{l,s} \quad \forall c \in \mathcal{C}(y, s) \quad (\text{C.5})$$

$$\sum_{s_s \in SP\mathcal{S}(i)} X_{i,l,s,s_s} \leq \left(\sum_{k=l}^T D_{i,k} \right) Y_{i,l,s} \quad \forall i, l, s \quad (\text{C.7})$$

$$X_{i,l,s,s_s}, E_{i,l} \geq 0 \quad \forall i, l, s, s_s \quad (\text{C.31})$$

$$Y_{i,l,s} \in \{0, 1\} \quad \forall i, l, s \quad (\text{C.17})$$

La fonction objectif (C.28), les contraintes d'équilibre des échelons stock (C.29), les contraintes de nomenclature (C.30), les contraintes de production (C.7), les contraintes de non-négativité et les contraintes des variables binaires (C.17) sont identiques à celles du modèle intégré. La

différence réside dans les contraintes de capacité détaillée pour une séquence fixée (C.5), qui remplacent les contraintes de précédence, de capacité et de non-négativité des dates de début des opérations du modèle intégré.

Modèle intégré avec échelon stock et flux de matières par site de production :

$$\min \sum_{s=1}^S \sum_{i=1}^N \sum_{l=1}^T \left(c_{i,s}^{set} Y_{i,l,s} + \sum_{s_p \in \mathcal{SPP}(i)} \sum_{s_s \in \mathcal{SPS}(i)} (c_{i,s}^p + c_{i,s,s_s}^t) X_{i,l,s_p,s,s_s} + e_{i,s} E_{i,l,s} \right) \quad (\text{C.32})$$

s.c.

$$E_{i,l,s} = E_{i,l-1,s} + \sum_{s_p \in \mathcal{SPP}(i)} \sum_{s_s \in \mathcal{SPS}(i)} X_{i,l,s_p,s,s_s} - D_{i,l,s} \quad \forall i, l, s \quad (\text{C.33})$$

$$\sum_{j \in \mathcal{DS}(i)} g_{ij} E_{j,l+L_j,s} - E_{i,l,s} \leq 0 \quad \forall i, l, s \quad (\text{C.34})$$

$$t(o') \geq t(o) + p_o^u X_{i(o),l(o),s_p(o),s(o),s_s(o)} + s_o^t Y_{i(o),l(o),s(o)} \quad \forall (o, o') \in \mathcal{A} \quad (\text{C.20})$$

$$\begin{cases} t(o) \geq t(o') + p_{o'}^u X_{i(o'),l(o'),s_p(o'),s(o'),s_s(o')} + s_{o'}^t Y_{i(o'),l(o'),s(o')} \\ \text{ou} \\ t(o') \geq t(o) + p_o^u X_{i(o),l(o),s_p(o),s(o),s_s(o)} + s_o^t Y_{i(o),l(o),s(o)} \end{cases} \quad \forall (o, o') \in \mathcal{E} \quad (\text{C.21})$$

$$t_o + p_o^u X_{i(o),l(o),s_p(o),s(o),s_s(o)} + s_o Y_{i(o),l(o),s(o)} \leq \sum_{l=1}^{l(o)} c_{l,s} \quad \forall o \in \mathcal{L} \quad (\text{C.22})$$

$$t_o + p_o^u X_{i(o),l(o),s_p(o),s(o),s_s(o)} + s_o Y_{i(o),l(o),s(o)} \geq \sum_{l=1}^{l(o)-1} c_{l,s} \quad \forall o \in \mathcal{L} \quad (\text{C.23})$$

$$t_o \geq \sum_{l=1}^{l(o)-L_i(o)} c_{l,s} \quad \forall o \in \mathcal{F} \text{ et } L_i(o) > 0 \quad (\text{C.24})$$

$$\sum_{s_p \in \mathcal{SPP}(i)} \sum_{s_s \in \mathcal{SPS}(i)} X_{i,l,s_p,s,s_s} \leq \left(\sum_{k=l}^T D_{i,k,s} \right) Y_{i,l,s} \quad \forall i, l, s \quad (\text{C.9})$$

$$t_o \geq 0 \quad \forall o \quad (\text{3.12})$$

$$X_{i,l,s_p,s,s_s}, E_{i,l,s} \geq 0 \quad \forall i, l, s_p, s, s_s \quad (\text{C.35})$$

$$Y_{i,l,s} \in \{0, 1\} \quad \forall i, l, s \quad (\text{C.17})$$

$$X_{i,l,s_p,s,s_s} = 0 \quad \forall i; \forall l < M_i + L_i \quad (\text{C.26})$$

La différence par rapport au modèle intégré suivant les flux de matières par produits est que nous ajoutons un nouvel indice aux variables de tailles des lots pour tenir compte du site de provenance. Nous ajoutons également un indice aux variables d'échelon stock pour comp-

tabiliser l'échelon stock dans chaque site de production. Nous pouvons donc considérer des coûts unitaires d'échelon stock dépendant du site de production. La nouvelle fonction objectif est modélisée par (C.32). Les contraintes d'équilibre d'échelons stock (C.33) sont décomposées par site de production. Les contraintes (C.36) correspondent aux contraintes de nomenclature. Les contraintes (C.20) et (C.21) modélisent les contraintes de précédence des opérations. Les contraintes (C.22)-(C.22) correspondent aux contraintes de capacité. Les contraintes de production sont modélisées avec (C.9). Les contraintes de non-négativité des dates de début des opérations sont modélisées avec (3.12). Les contraintes (C.35) forcent les tailles de lots et les variables de setup à être positives. La nature binaire des variables de setup est garantie avec (C.17) et, les contraintes (C.26) permettent de lancer la production d'un produit lorsque son approvisionnement en composants est garanti.

Modèle avec séquence fixée, échelon stock et flux de matières par site de production :

$$\min \sum_{s=1}^S \sum_{i=1}^N \sum_{l=1}^T \left(c_{i,s}^{set} Y_{i,l,s} + \sum_{s_p \in SPP(i)} \sum_{s_s \in SPs(i)} (c_{i,s}^p + c_{i,s,s_s}^t) X_{i,l,s_p,s,s_s} + e_{i,s} E_{i,l,s} \right) \quad (\text{C.32})$$

s.c.

$$E_{i,l,s} = E_{i,l-1,s} + \sum_{s_p \in SPP(i)} \sum_{s_s \in SPs(i)} X_{i,l,s_p,s,s_s} - D_{i,l,s} \quad \forall i, l, s \quad (\text{C.33})$$

$$\sum_{j \in DS(i)} g_{ij} E_{j,l+L_j,s} - E_{i,l,s} \leq 0 \quad \forall i, l, s \quad (\text{C.36})$$

$$r(o_c^f) + \sum_{o \in c} \sum_{s_p \in SPP(i(o))} \sum_{s_s \in SPs(i(o))} (p_o^u X_{i(o),l(o),s_p,s,s_s} + s_o^t Y_{i(o),l(o),s}) \leq \sum_{l=1}^{l(o_c^l)} c_{l,s} \quad \forall c \in \mathcal{C}(y, s) \quad (\text{C.6})$$

$$\sum_{s_p \in SPP(i)} \sum_{s_s \in SPs(i)} X_{i,l,s_p,s,s_s} \leq \left(\sum_{k=l}^T D_{i,k,s} \right) Y_{i,l,s} \quad \forall i, l, s \quad (\text{C.9})$$

$$X_{i,l,s_p,s,s_s}, I_{i,l,s} \geq 0 \quad \forall i, l, s_p, s, s_s \quad (\text{C.35})$$

$$Y_{i,l,s} \in \{0, 1\} \quad \forall i, l, s \quad (\text{C.17})$$

Comme dans les modèles avec séquence fixée présentés dans cette annexe, la différence par rapport au modèle intégré est que les contraintes de capacité détaillées pour une séquence fixée (C.6) remplacent les contraintes de précédence (C.20) et (C.21), de capacité (C.22)-(C.24) et de non-négativité des dates de début des opérations du modèle intégré (3.12).

Une autre variante à étudier dans des travaux futurs est la prise en compte des délais de livraison différents selon les sites de production.

Annexe D

Prise en compte de l'aspect environnemental

De nos jours, l'aspect économique n'est pas le seul facteur à considérer pour garantir le succès d'une chaîne logistique. Les aspects environnementaux et sociaux sont des facteurs qui peuvent permettre à une entreprise de se différencier et donc d'avoir un avantage sur un marché par rapport à certains de ses concurrents. La prise en compte de ces trois facteurs dans le management d'une chaîne logistique constitue ce que l'on appelle « la chaîne logistique durable ». En effet, pour permettre la durabilité d'une chaîne logistique à long terme, des contraintes économiques, environnementales et sociales doivent être considérées dans le processus de décisions.

Nous nous intéressons à l'intégration des aspects économique et environnemental, l'idée étant d'inclure des contraintes environnementales dans le problème de planification et d'ordonnement de la production. L'aspect environnemental est souvent considéré dans la littérature en quantifiant les émissions de carbone. Les émissions de carbone sont en effet un des principaux éléments de pollution générés par les industries. Pour cette raison, la législation internationale a établi des lois limitant et taxant les émissions de carbone générées par les activités de production et de distribution [184].

Très peu de travaux intégrant des contraintes d'émissions de carbones dans des problèmes de planification de la production ou de la distribution ont été étudiés dans la littérature. À notre connaissance, seuls Benjaafar *et al.* [29] et Absi *et al.* [3] ont étudié le problème de dimensionnement de lots avec contraintes d'émissions de carbone. D'autres travaux intègrent l'impact environnemental sous forme de coûts dans la fonction objectif [13].

Nous nous basons sur le travail d'Absi *et al.* [3], où un problème de dimensionnement de lots mono-produit et multi-sources sans contraintes de capacité et avec contraintes d'émissions

de carbone est étudié. Les auteurs modélisent le problème avec différents modes de production et de transport possibles, un mode correspondant à la combinaison d'une source (site de production) et d'un moyen de transport. Les auteurs proposent les contraintes d'émissions de carbone suivantes, où ω_l^m est la quantité d'émissions de carbone générées par la production et distribution d'une unité de produit à la période l en utilisant le mode m , Ω_l^{max} est le nombre maximum d'émissions de carbone allouées à la période l , x_l^m est le nombre d'unités produites et distribuées à la période l en utilisant le mode m et J_l est la quantité d'émissions de carbone disponibles à la fin de la période l .

- **Contraintes périodiques.** La quantité d'émissions de carbone qui n'est pas utilisée à une période donnée est perdue.

$$\sum_{m=1}^M (\omega_l^m - \Omega_l^{max}) X'_{lm} \leq 0 \quad \forall l = 1, \dots, T \quad (D.1)$$

- **Contraintes cumulatives.** La quantité d'émissions de carbone qui n'est pas utilisée à une période donnée peut être utilisée à une période future sans dépasser la capacité cumulée.

$$J_l = J_{l-1} - \sum_{m=1}^M (\omega_l^m - \Omega_l^{max}) X'_{lm} \leq 0 \quad \forall l = 1, \dots, T \quad (D.2)$$

- **Contraintes globales.** Les émissions de carbone totales (sur tout l'horizon de planification) ne doivent pas dépasser l'impact maximum alloué.

$$\sum_{l=1}^T \sum_{m=1}^M (\omega_l^m - \Omega_l^{max}) X'_{lm} \leq 0 \quad (D.3)$$

- **Contraintes glissantes.** La quantité d'émissions de carbone qui n'est pas utilisée à une période l peut être utilisée à une période future entre $l + 1$ et R .

$$\sum_{k=l-R+1}^l \sum_{m=1}^M (\omega_k^m - \Omega_k^{max}) X'_{km} \leq 0 \quad \forall l = R, \dots, T \quad (D.4)$$

Les auteurs démontrent que le problème de dimensionnement de lots avec contraintes d'émissions de carbone périodiques peut être résolu en temps polynomial. Les problèmes associés à l'intégration des trois autres types de contraintes d'émissions de carbonnes sont \mathcal{NP} -difficiles. Les auteurs montrent aussi que l'algorithme de programmation dynamique de Wagelmans *et al.* [230] peut être adapté pour résoudre le problème avec contraintes d'émissions de carbone

périodiques en un temps $O(TM^2 \log M + T^2)$, où M est le nombre de modes et T est le nombre de périodes de l'horizon de planification.

Notre idée, dans un premier temps, est d'intégrer les contraintes d'émissions de carbone périodiques dans notre approche. D'une part, nous pouvons continuer à utiliser l'algorithme de programmation dynamique de Wagelmans *et al.* [230] (avec des modifications mineures) pour le calcul d'un plan de production avec contraintes de capacité et de nomenclature (dans le cas multi-niveaux) relâchées. D'autre part, aucune procédure supplémentaire n'est nécessaire pour satisfaire les nouvelles contraintes. Nous devons juste restreindre les quantités à déplacer dans l'heuristique de lissage, afin que la réparation des contraintes de capacité ou de nomenclature n'entraîne pas la violation des contraintes d'émissions de carbone. Au niveau de la modélisation, plusieurs modifications sont nécessaires.

Nous présentons, dans ce qui suit, les modèles mathématiques intégrés et avec séquence fixée des problèmes mono-niveau et multi-niveaux de dimensionnement de lots et d'ordonnement avec contraintes d'émissions de carbone, en utilisant la notation classique et la notation d'échelon stock.

Modèle intégré avec notation classique pour le problème mono-niveau :

$$\min \sum_{i=1}^N \sum_{l=1}^T \sum_{m=1}^M \left(c_{im}^p X'_{ilm} + c_i^{inv} I_{il} + c_{im}^s Y'_{ilm} \right) \quad (\text{D.5})$$

s.c.

$$I_{il} = I_{i,l-1} + \sum_{m=1}^M X'_{ilm} - D_{il} \quad \forall i, l, m \quad (\text{D.6})$$

$$t(o') \geq t(o) + p_o^u \sum_{m=1}^M X'_{i(o),l(o),m} + s_o^t Y'_{i(o),l(o),m} \quad \forall (o, o') \in \mathcal{A} \quad (\text{D.7})$$

$$\left\{ \begin{array}{l} t(o) \geq t(o') + p_{o'}^u \sum_{m=1}^M X'_{i(o'),l(o'),m} + s_{o'}^t Y'_{i(o'),l(o'),m} \\ \text{ou} \\ t(o') \geq t(o) + p_o^u \sum_{m=1}^M X'_{i(o),l(o),m} + s_o^t Y'_{i(o),l(o),m} \end{array} \right. \quad \forall (o, o') \in \mathcal{E} \quad (\text{D.8})$$

$$t_o + p_o^u \sum_{m=1}^M X'_{i(o)l(o),m} + s_o Y'_{i(o)l(o),m} \leq \sum_{l=1}^{l(o)} c_l \quad \forall o \in \mathcal{L} \quad (\text{D.9})$$

$$t_o + p_o^u \sum_{m=1}^M X'_{i(o)l(o),m} + s_o Y'_{i(o)l(o),m} \geq \sum_{l=1}^{l(o)-1} c_l \quad \forall o \in \mathcal{L} \quad (\text{D.10})$$

$$t_o \geq \sum_{l=1}^{l(o)-L_{i(o)}} c_l \quad \forall o \in \mathcal{F}, L_{i(o)} > 0 \quad (3.8)$$

$$\sum_{m=1}^M (\omega_{il}^m - \Omega_{il}^{max}) X'_{ilm} \leq 0 \quad \forall i, l \quad (\text{D.1})$$

$$X'_{ilm} \leq \left(\sum_{k=l}^T D_{ik} \right) Y'_{ilm} \quad \forall i, l, m \quad (\text{D.11})$$

$$X'_{ilm}, I_{il} \geq 0 \quad \forall i, l, m \quad (\text{D.12})$$

$$Y'_{ilm} \in \{0, 1\} \quad \forall i, l, m \quad (\text{D.13})$$

$$t_o \geq 0 \quad \forall o \quad (3.12)$$

Les tailles de lots, modélisées par X_{il} dans les chapitres 3 et 4, sont désagrégées par mode et deviennent X'_{ilm} . De même, les variables binaires de setup, initialement modélisées par Y_{il} ,

incluent le mode comme nouvel indice et deviennent Y'_{ilm} . La fonction objectif (D.5) minimise la somme des coûts de production, de stockage et de setup. Les contraintes (D.6) modélisent l'équilibre des stocks. Les contraintes (D.7) représentent les contraintes de précédence entre les opérations d'une même gamme opératoire. Les contraintes (D.8) correspondent aux contraintes de précédence entre les opérations traitées sur la même ressource. Les contraintes (D.9), (D.10) et (3.8) définissent la capacité de production. Les contraintes (D.1) correspondent aux contraintes d'émissions de carbone périodiques. Les contraintes (D.11) lient les tailles de lots aux décisions de setup. Les contraintes (D.12) indiquent la non-négativité des tailles de lots et des inventaires. Les contraintes (D.13) forcent les variables de setup à prendre des valeurs binaires, et les contraintes (3.12) assurent la non-négativité des dates de début des opérations.

Modèle avec séquence fixée et notation classique pour le problème mono-niveau :

$$\min \sum_{i=1}^N \sum_{l=1}^T \sum_{m=1}^M \left(c_{im}^p X'_{ilm} + c_i^{inv} I_{il} + c_{im}^s Y'_{ilm} \right) \quad (\text{D.5})$$

s.c.

$$I_{il} = I_{i,l-1} + \sum_{m=1}^M X'_{ilm} - D_{il} \quad \forall i, l, m \quad (\text{D.6})$$

$$r(o_c^f) + \sum_{o \in c} \left(p_o^u \sum_{m=1}^M X'_{i(o),l(o),m} + s_o^t Y'_{i(o),l(o),m} \right) \leq \sum_{l=1}^{l(o_c^l)} c_l \quad \forall c \in \mathcal{C}(y) \quad (\text{D.14})$$

$$\sum_{m=1}^M (\omega_{il}^m - \Omega_{il}^{max}) X'_{ilm} \leq 0 \quad \forall i, l \quad (\text{D.1})$$

$$X'_{ilm} \leq \left(\sum_{k=l}^T D_{ik} \right) Y'_{ilm} \quad \forall i, l, m \quad (\text{D.11})$$

$$X'_{ilm}, I_{il} \geq 0 \quad \forall i, l, m \quad (\text{D.12})$$

$$Y'_{ilm} \in \{0, 1\} \quad \forall i, l, m \quad (\text{D.13})$$

Nous retrouvons la fonction objectif (D.5) minimisant la somme de tous les coûts. Les contraintes d'équilibre des stocks sont à nouveau modélisées avec (D.6). Les contraintes de précédence et de capacité sont remplacées par les contraintes de capacité détaillées (D.14) pour une séquence fixée y . Les contraintes d'émissions de carbone périodiques (D.1), les contraintes de production (D.11) lient les variables de décision, les contraintes de non-négativité (D.12) et

les contraintes (D.13) imposant des valeurs binaires aux variables de setup restent identiques à celles du modèle intégré.

Modèle intégré avec notation classique pour le problème multi-niveaux :

$$\min \sum_{i=1}^N \sum_{l=1}^T \sum_{m=1}^M \left(c_{im}^p X'_{ilm} + c_i^{inv} I_{il} + c_{im}^s Y'_{ilm} \right) \quad (\text{D.5})$$

s.c.

$$I_{il} = I_{i,l-1} + \sum_{m=1}^M X'_{ilm} - d_{il} - \sum_{m=1}^M \sum_{j \in \mathcal{DS}(i)} g_{ij} X'_{j,l+L_j,m} \quad \forall i, l, m \quad (\text{D.15})$$

$$t(o') \geq t(o) + p_o^u \sum_{m=1}^M X'_{i(o),l(o),m} + s_o^t Y'_{i(o),l(o),m} \quad \forall (o, o') \in \mathcal{A} \quad (\text{D.7})$$

$$\left\{ \begin{array}{l} t(o) \geq t(o') + p_{o'}^u \sum_{m=1}^M X'_{i(o'),l(o'),m} + s_{o'}^t Y'_{i(o'),l(o'),m} \\ \text{ou} \\ t(o') \geq t(o) + p_o^u \sum_{m=1}^M X'_{i(o),l(o),m} + s_o^t Y'_{i(o),l(o),m} \end{array} \right. \quad \forall (o, o') \in \mathcal{E} \quad (\text{D.8})$$

$$t_o + p_o^u \sum_{m=1}^M X'_{i(o)l(o),m} + s_o Y'_{i(o)l(o),m} \leq \sum_{l=1}^{l(o)} c_l \quad \forall o \in \mathcal{L} \quad (\text{D.9})$$

$$t_o + p_o^u \sum_{m=1}^M X'_{i(o)l(o),m} + s_o Y'_{i(o)l(o),m} \geq \sum_{l=1}^{l(o)-1} c_l \quad \forall o \in \mathcal{L} \quad (\text{D.10})$$

$$t_o \geq \sum_{l=1}^{l(o)-L_{i(o)}} c_l \quad \forall o \in \mathcal{F}, L_{i(o)} > 0 \quad (\text{3.8})$$

$$\sum_{m=1}^M (\omega_{il}^m - \Omega_{il}^{max}) X'_{ilm} \leq 0 \quad \forall i, l \quad (\text{D.1})$$

$$X'_{ilm} \leq \left(\sum_{k=l}^T D_{ik} \right) Y'_{ilm} \quad \forall i, l, m \quad (\text{D.11})$$

$$X'_{ilm}, I_{il} \geq 0 \quad \forall i, l, m \quad (\text{D.12})$$

$$Y'_{ilm} \in \{0, 1\} \quad \forall i, l, m \quad (\text{D.13})$$

$$t_o \geq 0 \quad \forall o \quad (\text{3.12})$$

$$X'_{ilm} = 0 \quad \forall i; \forall l < M_i + L_i; \forall m \quad (\text{D.16})$$

La fonction objectif (D.5) est la même que pour le problème mono-niveau. La contrainte (D.15) modélise l'équilibre des stocks, en tenant compte de tailles de lots des successeurs dans la nomenclature. Les contraintes (D.7)-(3.12) modélisant les contraintes de précédence, de capacité, d'émissions de carbone périodiques, de production, de non-négativité et de setup restent identiques à celles du problème mono-niveau. Enfin, les contraintes (D.16) empêchent la fabrication d'un produit avant que son approvisionnement en composants ne soit assuré.

Modèle intégré avec échelon stock pour le problème multi-niveaux :

$$\min \sum_{i=1}^N \sum_{l=1}^T \sum_{m=1}^M \left(c_{im}^p X'_{ilm} + e_i E_{il} + c_{im}^s Y'_{ilm} \right) \quad (\text{D.17})$$

s.c.

$$E_{il} = E_{i,l-1} + \sum_{m=1}^M X'_{ilm} - D_{il} \quad \forall i, l, m \quad (\text{D.18})$$

$$\sum_{j \in \mathcal{DS}(i)} g_{ij} E_{j,l+L_j} - E_{il} \leq 0 \quad \forall i, l \quad (4.16)$$

$$t(o') \geq t(o) + p_o^u \sum_{m=1}^M X'_{i(o),l(o),m} + s_o^t Y'_{i(o),l(o),m} \quad \forall (o, o') \in \mathcal{A} \quad (\text{D.7})$$

$$\left\{ \begin{array}{l} t(o) \geq t(o') + p_{o'}^u \sum_{m=1}^M X'_{i(o'),l(o'),m} + s_{o'}^t Y'_{i(o'),l(o'),m} \\ \text{ou} \\ t(o') \geq t(o) + p_o^u \sum_{m=1}^M X'_{i(o),l(o),m} + s_o^t Y'_{i(o),l(o),m} \end{array} \right. \quad \forall (o, o') \in \mathcal{E} \quad (\text{D.8})$$

$$t_o + p_o^u \sum_{m=1}^M X'_{i(o)l(o),m} + s_o Y'_{i(o)l(o),m} \leq \sum_{l=1}^{l(o)} c_l \quad \forall o \in \mathcal{L} \quad (\text{D.9})$$

$$t_o + p_o^u \sum_{m=1}^M X'_{i(o)l(o),m} + s_o Y'_{i(o)l(o),m} \geq \sum_{l=1}^{l(o)-1} c_l \quad \forall o \in \mathcal{L} \quad (\text{D.10})$$

$$t_o \geq \sum_{l=1}^{l(o)-L_{i(o)}} c_l \quad \forall o \in \mathcal{F}, L_{i(o)} > 0 \quad (3.8)$$

$$\sum_{m=1}^M (\omega_{il}^m - \Omega_{il}^{max}) X'_{ilm} \leq 0 \quad \forall i, l \quad (\text{D.1})$$

$$X'_{ilm} \leq \left(\sum_{k=l}^T D_{ik} \right) Y'_{ilm} \quad \forall i, l, m \quad (\text{D.11})$$

$$X'_{ilm}, E_{il} \geq 0 \quad \forall i, l, m \quad (\text{D.19})$$

$$Y'_{ilm} \in \{0, 1\} \quad \forall i, l, m \quad (\text{D.13})$$

$$t_o \geq 0 \quad \forall o \quad (3.12)$$

$$X'_{ilm} = 0 \quad \forall i; \forall l < M_i + L_i; \forall m \quad (\text{D.16})$$

Dans la nouvelle fonction objectif (D.17), la somme des coûts de stockage est remplacée par la somme des coûts d'échelon stock. Les contraintes (D.18) modélisent l'équilibre des échelons stocks. Les contraintes (4.16) correspondent aux contraintes de nomenclature. Les contraintes restantes ((D.7)-(D.16)) sont similaires à celles du modèle avec notation classique.

Modèle avec séquence fixée et échelon stock pour le problème multi-niveaux :

$$\min \sum_{i=1}^N \sum_{l=1}^T \sum_{m=1}^M \left(c_{im}^p X'_{ilm} + e_i E_{il} + c_{im}^s Y'_{ilm} \right) \quad (\text{D.17})$$

s.c.

$$E_{il} = E_{i,l-1} + \sum_{m=1}^M X'_{ilm} - D_{il} \quad \forall i, l, m \quad (\text{D.18})$$

$$\sum_{j \in \mathcal{DS}(i)} g_{ij} E_{j,l+L_j} - E_{il} \leq 0 \quad \forall i, l \quad (4.16)$$

$$r(o_c^f) + \sum_{o \in c} \left(p_o^u \sum_{m=1}^M X'_{i(o),l(o),m} + s_o^t Y'_{i(o),l(o),m} \right) \leq \sum_{l=1}^{l(o_c)} c_l \quad \forall c \in \mathcal{C}(y) \quad (\text{D.14})$$

$$\sum_{m=1}^M (\omega_{il}^m - \Omega_{il}^{max}) X'_{ilm} \leq 0 \quad \forall i, l \quad (\text{D.1})$$

$$X'_{ilm} \leq \left(\sum_{k=l}^T D_{ik} \right) Y'_{ilm} \quad \forall i, l, m \quad (\text{D.11})$$

$$X'_{ilm}, E_{il} \geq 0 \quad \forall i, l, m \quad (\text{D.19})$$

$$Y'_{ilm} \in \{0, 1\} \quad \forall i, l, m \quad (\text{D.13})$$

La différence par rapport au modèle du problème intégré avec échelon stock est la substitution des contraintes (D.7)-(D.10), (3.8) et (3.12) par les contraintes de capacité détaillées (D.14). Les autres contraintes et la fonction objectif restent identiques.

Bibliographie

- [1] AARDAL, K., AND LARSSON, T. A Benders decomposition based heuristic for the hierarchical production planning problem. *European Journal of Operational Research* 45, 1 (1990), 4–14.
 - [2] AARTS, E. H. L., REIJNHOUT, M. F., STEHOUWER, H. P., AND WESSELS, J. A novel decomposition approach for on-line lot-sizing. *European Journal of Operational Research* 122, 2 (2000), 339–353.
 - [3] ABSI, N., DAUZÈRE-PÉRÈS, S., KEDAD-SIDHOUM, S., PENZ, B., AND RAPINE, C. Lot Sizing with Carbon Emission Constraints. *European Journal of Operational Research* (Dec. 2012).
 - [4] ABSI, N., AND KEDAD-SIDHOUM, S. The multi-item capacitated lot-sizing problem with setup times and shortage costs. *European Journal of Operational Research* 185, 3 (Mar. 2008), 1351–1374.
 - [5] ABSI, N., KEDAD-SIDHOUM, S., AND DAUZÈRE-PÉRÈS, S. Uncapacitated lot-sizing problem with production time windows, early productions, backlogs and lost sales. *International Journal of Production Research* 49, 9 (May 2011), 2551–2566.
 - [6] ADAMS, J., BALAS, E., AND ZAWACK, D. The Shifting Bottleneck Procedure for Job Shop Scheduling. *Management Science* 34, 3 (1988), 391–401.
 - [7] AFENTAKIS, P., AND GAVISH, B. Optimal Lot-Sizing Algorithms for Complex Product Structures. *Operations Research* 34, 2 (Mar. 1986), 237–249.
 - [8] AFENTAKIS, P., GAVISH, B., AND KARMARKAR, U. S. Computationally Efficient Optimal Solutions to the Lot-Sizing Problem in Multistage Assembly Systems. *Management Science* 30, 2 (1984), 222–240.
 - [9] AGGARWAL, A., AND PARK, J. K. Improved Algorithms for Economic Lot Size Problems. *Operations Research* 41, 3 (1993), 549–571.
-

- [10] AGGOUNE, R. Minimizing the makespan for the flow shop scheduling problem with availability constraints. *European Journal of Operational Research* 153, 3 (Mar. 2004), 534–543.
- [11] AGGOUNE, R., DAUZÈRE-PÉRÈS, S., WOLOSEWICZ, C., AND GÓMEZ URRUTIA, E. D. A solution procedure for an integrated lot-sizing and scheduling problem. In *ROA-DEF2011* (2011).
- [12] AGGOUNE, R., AND PORTMANN, M.-C. Flow shop scheduling problem with limited machine availability : A heuristic approach. *International Journal of Production Economics* 99, 1-2 (Jan. 2006), 4–15.
- [13] AISSAOUI, N., HAOUARI, M., AND HASSIN, E. Supplier selection and order lot sizing modeling : A review. *Computers & Operations Research* 34, 12 (2007), 3516–3540. Operations Research and Outsourcing.
- [14] AKARTUNALI, K., AND MILLER, A. J. A heuristic approach for big bucket multi-level production planning problems. *European Journal of Operational Research* 193, 2 (Mar. 2009), 396–411.
- [15] ALLAHVERDI, A., NG, C. T., CHENG, T. C. E., AND KOVALYOV, M. Y. A survey of scheduling problems with setup times or costs. *European Journal of Operational Research* 187, 3 (June 2008), 985–1032.
- [16] ALMADA-LOBO, B., AND JAMES, R. J. Neighbourhood search meta-heuristics for capacitated lot-sizing with sequence-dependent setups. *International Journal of Production Research* 48, 3 (Feb. 2010), 861–878.
- [17] ALMADA-LOBO, B., KLABJAN, D., CARRAVILLA, M. A., AND OLIVEIRA, J. F. Single machine multi-product capacitated lot sizing with sequence-dependent setups. *International Journal of Production Research* 45, 20 (Oct. 2007), 4873–4894.
- [18] ALMADA-LOBO, B., KLABJAN, D., CARRAVILLA, M. A., AND OLIVEIRA, J. F. Multiple machine continuous setup lotsizing with sequence-dependent setups. *Computational Optimization and Applications* 47, 3 (Jan. 2010), 529–552.
- [19] ALMADA-LOBO, B., OLIVEIRA, J. F., AND ANTÓNIA CARRAVILLA, M. A note on “the capacitated lot-sizing and scheduling problem with sequence-dependent setup costs and setup times. *Computers & Operations Research* 35, 4 (Apr. 2008), 1374–1376.
- [20] ALMADA-LOBO, B., OLIVEIRA, J. F., AND CARRAVILLA, M. A. Production planning and scheduling in the glass container industry : A VNS approach. *International Journal of Production Economics* 114, 1 (July 2008), 363–375.

- [21] ALMEDER, C. A hybrid optimization approach for multi-level capacitated lot-sizing problems. *European Journal of Operational Research* 200, 2 (Jan. 2010), 599–606.
- [22] AMORIM, P., PINTO-VARELA, T., ALMADA-LOBO, B., AND BARBÓSA-PÓVOA, A. Comparing models for lot-sizing and scheduling of single-stage continuous processes : Operations research and process systems engineering approaches. *Computers & Chemical Engineering* 52 (May 2013), 177–192.
- [23] APPLGATE, D. L., VAŠEK CHVÁTAL, R. E. B., AND COOK, W. J. *The Traveling Salesman Problem : A Computational Study*. Princeton University Press, Princeton, New Jersey, 2006.
- [24] ARAUJO, S. A., ARENALES, M. N., AND CLARK, A. R. Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups. *Journal of Heuristics* 13, 4 (Apr. 2007), 337–358.
- [25] ARTIGUES, C., AND FEILLET, D. A branch and bound method for the job-shop problem with sequence-dependent setup times. *Annals of Operations Research* 159, 1 (Dec. 2007), 135–159.
- [26] BARNHART, C., JOHNSON, E. L., NEMHAUSER, G. L., SAVELSBERGH, M. W. P., AND VANCE, P. H. Branch-and-Price : Column Generation for Solving Huge Integer Programs. *Operations Research* 46, 3 (1998), 316–329.
- [27] BELLMAN, R. *Dynamic Programming*, 1st ed. Princeton University Press, New Jersey, 1957.
- [28] BENDERS, J. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* 4, 1 (1962), 238–252.
- [29] BENJAAFAR, S., LI, Y., AND DASKIN, M. Carbon footprint and the management of supply chains : insights from simple models. 2010.
- [30] BERRETTA, R., FRANÇA, P. M., AND ARMENTANO, V. A. Metaheuristic Approaches for the Multilevel Resource-Constrained Lot-Sizing Problem with Setup and Lead Times. *Asia-Pacific Journal of Operational Research* 22, 2 (2005), 261–286.
- [31] BITRAN, G. R., AND YANASSE, H. H. Computational complexity of the capacitated lot size problem. *Management Science* 28, 10 (1982), 1174–1187.
- [32] BOCTOR, F. F., AND POULIN, P. Heuristics for the N-product, M-stage, economic lot sizing and scheduling problem with dynamic demand. *International Journal of Production Research* 43, 13 (July 2005), 2809–2828.
- [33] BRAHIMI, N., DAUZÈRE-PÉRÈS, S., NAJID, N. M., AND NORDLI, A. Single item lot sizing problems. *European Journal of Operational Research* 168 (Jan. 2006), 1–16.

- [34] BRÜGGEMANN, W., AND JAHNKE, H. The discrete lot-sizing and scheduling problem : Complexity and modification for batch availability. *European Journal of Operational Research* 124, 3 (2000), 511–528.
- [35] BRUSCO, M. J., AND STAHL, S. *Branch-and-Bound Applications in Combinatorial Data Analysis*, 1st ed. Springer Science+Business Media, Inc., New York, 2005.
- [36] BUSCHKÜHL, L., SAHLING, F., HELBER, S., AND TEMPELMEIER, H. Dynamic capacitated lot-sizing problems : a classification and review of solution approaches. *OR Spectrum* 32, 2 (Oct. 2010), 231–261.
- [37] CHEN, H., CHU, C., AND PROTH, J. M. A more efficient Lagrangian relaxation approach to job-shop scheduling problems. In *IEEE International Conference on Robotics and Automation* (1995), pp. 496–501.
- [38] CLARK, A. J., AND SCARF, H. Optimal policies for a multi-echelon inventory problem. *Management Science* 6, 4 (1960), 475–490.
- [39] CLARK, A. R., AND ARMENTANO, V. A. Echelon stock formulation for multi-stage lot-sizing with component lead times. *International Journal of Systems Science* 24, 9 (1993), 1759–1775.
- [40] CLARK, A. R., AND ARMENTANO, V. A. A Heuristic for a Resource-capacitated Multi-stage Lot-sizing Problem with Lead Times. *Journal of the Operational Research Society* 46 (1995), 1208–1222.
- [41] CONEJO, A. J., CASTILLO, E., MÍNGUEZ, R., AND GARCÍA-BERTRAND, R. *Decomposition Techniques in Mathematical Programming*, 1st ed. Springer-Verlag, Berlin, Heidelberg, 2006.
- [42] CONSTANTINO, M. A polyhedral approach to a production planning problem. *Annals of Operations Research* 96, 1-4 (2000), 75–95.
- [43] COYLE, J. J., LANGLEY, C. J., GIBSON, B. J., NOVACK, R. A., AND BARDI, E. J. *Supply Chain Management : A Logistics Perspective*, 8th ed. Cengage Learning, Mason, 2009.
- [44] CRAUWELS, H. A. J., POTTS, C. N., AND VAN WASSENHOVE, L. N. Local search heuristics for single machine scheduling with batch set-up times to minimize total weighted completion time. *Annals of Operations Research* 70 (1997), 261–279.
- [45] DANTZIG, G. B., AND WOLFE, P. Decomposition Principle for Linear Programs. *Operations Research* 8, 1 (1960), 101–111.

- [46] DAUZÈRE-PÉRÈS, S., AND LASSERRE, J.-B. A modified shifting bottleneck procedure for job-shop scheduling. *International Journal of Production Research* 31, 4 (1993), 923–932.
- [47] DAUZÈRE-PÉRÈS, S., AND LASSERRE, J.-B. Integration of lotsizing and scheduling decisions in a job-shop. *European Journal of Operational Research* 75 (1994), 413–426.
- [48] DAUZÈRE-PÉRÈS, S., AND LASSERRE, J.-B. On the importance of sequencing decisions in production planning and scheduling. *International Transactions in Operational Research* 9 (2002), 779–793.
- [49] DAUZÈRE-PÉRÈS, S., AND PAULLI, J. An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search. *Annals of Operations Research* 70, 0 (1997), 281–306.
- [50] DE MATTA, R., AND GUIGNARD, M. The performance of rolling production schedules in a process industry. *IIE Transactions* 27 (1995), 564–573.
- [51] DE SANTA-EULALIA, L. A., D’AMOURS, S., FRAYRET, J.-M., MENEGUSSO, C. C., AND CAMBIAGHI AZEVEDO, R. Advanced Supply Chain Planning Systems (APS) Today and Tomorrow. In *Supply Chain Management - Pathways for Research and Practice*, D. Onkal, Ed. InTech, 2011, pp. 171–200.
- [52] DEGRAEVE, Z., AND JANS, R. A New Dantzig-Wolfe Reformulation and Branch-and-Price Algorithm for the Capacitated Lot-Sizing Problem with Setup Times. *Operations Research* 55, 5 (2007), 909–920.
- [53] DELLA CROCE, F., GHIRARDI, M., AND TADEI, R. An improved branch-and-bound algorithm for the two machine total completion time flow shop problem. *European Journal of Operational Research* 139, 2 (2002), 293–301.
- [54] DESAULNIERS, G., DESROSIERS, J., AND SOLOMON, M. M. *Column Generation*, 1st ed. Springer Science+Business Media, Inc., New York, 2010.
- [55] DIABY, M., BAHL, H. C., KARWAN, M. H., AND ZIONTS, S. A Lagrangean relaxation approach for very-large-scale capacitated lot-sizing. *Management Science* 38, 9 (1992), 1329–1340.
- [56] DIABY, M., BAHL, H. C., KARWAN, M. H., AND ZIONTS, S. Capacitated lot-sizing and scheduling by Lagrangean relaxation. *European Journal of Operational Research* 59, 3 (1992), 448–458.
- [57] DITTMANN, J. P. *Supply Chain Transformation : Building and Executing an Integrated Supply Chain Strategy*, 1st ed. McGraw-Hill, 2012.

- [58] DORIGO, M., AND STÜTZLE, T. *Ant Colony Optimization*, 1st ed. Massachusetts Institute of Technology, 2004.
- [59] DRÉO, J., SIARRY, P., PÉTROVSKY, A., AND TAILLARD, E. *Metaheuristics for Hard Optimization : Methods and Case Studies*, 1st ed. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [60] DREXL, A., AND HAASE, K. Proportional lotsizing and scheduling. *International Journal of Production Economics* 40 (1995), 73–87.
- [61] DREXL, A., AND KIMMS, A. Lot sizing and scheduling - Survey and extensions. *European Journal of Operational Research* 99 (1997), 221–235.
- [62] DU, D.-Z., AND KO, K.-I. *Theory of Computational Complexity*. John Wiley & Sons, Inc., 2000.
- [63] DUECK, G., AND SCHEUER, T. Threshold accepting : A general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics* 90, 1 (1990), 161–175.
- [64] FANDEL, G., AND STAMMEN-HEGENE, C. Simultaneous lot sizing and scheduling for multi-product multi-level production. *International Journal of Production Economics* 104, 2 (Dec. 2006), 308–316.
- [65] FEDERGRUEN, A., AND TZUR, M. A 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 (1991), 909–925.
- [66] FEO, T. A., AND RESENDE, M. G. Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization* 6 (1995), 109–134.
- [67] FERREIRA, D., CLARK, A. R., ALMADA-LOBO, B., AND MORABITO, R. Single-stage formulations for synchronised two-stage lot sizing and scheduling in soft drink production. *International Journal of Production Economics* 136, 2 (Apr. 2012), 255–265.
- [68] FERREIRA, D., MORABITO, R., AND RANGEL, S. Solution approaches for the soft drink integrated production lot sizing and scheduling problem. *European Journal of Operational Research* 196, 2 (July 2009), 697–706.
- [69] FISHER, M. L. The Lagrangian Relaxation Method for Solving Integer Programming Problems. *Management Science* 27, 1 (Jan. 1981), 1–18.
- [70] FLEISCHMANN, B. The discrete lot-sizing and scheduling problem. *European Journal of Operational Research* 44, 3 (1990), 337–348.

- [71] FLEISCHMANN, B., AND MEYR, H. The general lotsizing and scheduling problem. *OR Spectrum* 19 (1997), 11–21.
- [72] FLEISCHMANN, B., AND MEYR, H. Advanced planning. In *Supply Chain Management and Advanced Planning*, H. Stadtler and C. Kilger, Eds., 4th ed. Springer Science+Business Media, Inc., Berlin, Heidelberg, New York, 2008, pp. 81–109.
- [73] FLORIAN, M., LENSTRA, J. K., AND RINNOOY KAN, A. H. G. Deterministic Production Planning : Algorithms and Complexity. *Management Science* 26, 7 (1980), 669–679.
- [74] FRANÇA, P. M., ARMENTANO, V. A., BERRETTA, R. E., AND CLARK, A. R. A Heuristic Method for Lot-Sizing in Multi-Stage Systems. *Computers & Operations Research* 24, 9 (1997), 861–874.
- [75] GELDERS, L. F., MAES, J., AND VAN WASSENHOVE, L. N. A branch and bound algorithm for the multi item single level capacitated dynamic lotsizing problem. In *Multi-Stage Production Planning and Inventory Control*, S. Axsäter, C. Schneeweiss, and E. Silver, Eds., vol. 266 of *Lecture Notes in Economics and Mathematical Systems*. Springer Berlin Heidelberg, 1986, pp. 92–108.
- [76] GÉLINAS, S., AND SOUMIS, F. Dantzig-Wolfe Decomposition for Job Shop Scheduling. In *Column Generation*, G. Desaulniers, J. Desrosiers, and M. M. Solomon, Eds. Springer, 2005, pp. 271–302.
- [77] GENIN, P., LAMOURI, S., AND THOMAS, A. Planification avancée : APS. *Techniques de l'ingénieur Modes de pilotage des flux logistiques base documentaire : TIB121DUO.*, ref. article : ag5120 (2005), 1–12.
- [78] GICQUEL, C., MIÈGEVILLE, N., MINOUX, M., AND DALLERY, Y. Discrete lot sizing and scheduling using product decomposition into attributes. *Computers & Operations Research* 36, 9 (Sept. 2009), 2690–2698.
- [79] GICQUEL, C., MINOUX, M., AND DALLERY, Y. On the discrete lot-sizing and scheduling problem with sequence-dependent changeover times. *Operations Research Letters* 37, 1 (Jan. 2009), 32–36.
- [80] GICQUEL, C., WOLSEY, L. A., AND MINOUX, M. On discrete lot-sizing and scheduling on identical parallel machines. *Optimization Letters* 6, 3 (Feb. 2011), 545–557.
- [81] GLOVER, F. W. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research* 13, 5 (1986), 533–549.
- [82] GLOVER, F. W., AND LAGUNA, M. *Tabu search*. Kluwer Academic Publishers, Massachusetts, 1997.

- [83] GÓMEZ RAVETTI, M., MATEUS, G. R., ROCHA, P. L., AND PARDALOS, P. M. A scheduling problem with unrelated parallel machines and sequence dependent setups. *International Journal of Operational Research* 2, 4 (2007), 380–399.
- [84] GÓMEZ URRUTIA, E. D., AGGOUNE, R., AND DAUZÈRE-PÉRÈS, S. Improvements of an Integrated Approach for Lot Sizing and Detailed Scheduling. In *Proceedings of the 2nd International Workshop on Lot Sizing* (Istanbul, 2011).
- [85] GÓMEZ URRUTIA, E. D., AGGOUNE, R., AND DAUZÈRE-PÉRÈS, S. An Integrated Approach for Solving Multi-Level Lot-Sizing and Scheduling Problems with Detailed Capacity Constraints. In *Proceedings of the 3rd International Workshop on Lot Sizing* (Rotterdam, 2012), pp. 1–4.
- [86] GÓMEZ URRUTIA, E. D., AGGOUNE, R., AND DAUZÈRE-PÉRÈS, S. New integrated approach for solving multi-level lot sizing and scheduling problems. In *9th International Conference of Modeling, Optimization and Simulation* (Bordeaux, 2012).
- [87] GÓMEZ URRUTIA, E. D., AGGOUNE, R., AND DAUZÈRE-PÉRÈS, S. Nouvelle méthode intégrée pour résoudre le problème multi-niveaux de planification et d’ordonnancement. In *ROADEF2012* (Angers, 2012).
- [88] GÓMEZ URRUTIA, E. D., AGGOUNE, R., AND DAUZÈRE-PÉRÈS, S. Analysis and Improvements of a Solving Procedure for the Integration of Lot-Sizing and Scheduling Decisions. In *Proceedings of the 5th International Conference on Industrial Engineering and Systems Management* (Rabat, 2013), no. October, p. 8.
- [89] GÓMEZ URRUTIA, E. D., AGGOUNE, R., AND DAUZÈRE-PÉRÈS, S. Résolution d’un problème de planification et d’ordonnancement multi-niveaux dans un job-shop. In *ROADEF2013* (Troyes, 2013).
- [90] GÓMEZ URRUTIA, E. D., AGGOUNE, R., AND DAUZÈRE-PÉRÈS, S. Smoothing Procedures for an Integrated Approach for Multi Level Lot Sizing and Detailed Scheduling. In *Proceedings of the 4th International Workshop on Lot Sizing* (Brussels, 2013), p. 4.
- [91] GÓMEZ URRUTIA, E. D., AGGOUNE, R., AND DAUZÈRE-PÉRÈS, S. Solving the integrated lot-sizing and job-shop scheduling problem. *International Journal of Production Research* (Mar. 2014).
- [92] GÓMEZ URRUTIA, E. D., AGGOUNE, R., DAUZÈRE-PÉRÈS, S., AND WOLOSEWICZ, C. An Integrated Approach for Lot Sizing and Detailed Scheduling in Job-Shops. In *Proceedings of the 4th International Conference on Information Systems, Logistics and Supply Chain* (Quebec, 2012).

- [93] GOPALAKRISHNAN, M., DING, K., BOURJOLLY, J.-M., AND MOHAN, S. A Tabu-Search Heuristic for the Capacitated Lot-Sizing Problem with Set-up Carryover. *Management Science* 47, 6 (2001), 851–863.
- [94] GOREN, H. G., TUNALI, S., AND JANS, R. A hybrid approach for the capacitated lot sizing problem with setup carryover. *International Journal of Production Research* 50, 6 (2012), 1582–1597.
- [95] GROSSMANN, I. E., VAN DEN HEEVER, S. A., AND HARJUNKOSKI, I. Discrete Optimization Methods and their Role in the Integration of Planning and Scheduling. *AIChE Symposium Series* 326 (2002), 150–168.
- [96] GUAN, Y., AHMED, S., NEMHAUSER, G. L., AND MILLER, A. J. A branch-and-cut algorithm for the stochastic uncapacitated lot-sizing problem. *Mathematical Programming* 105, 1 (2006), 55–84.
- [97] GUIGNARD, M., AND KIM, S. Lagrangean Decomposition : A Model Yielding Stronger Lagrangean Bounds. *Mathematical Programming* 39 (1987), 215–228.
- [98] GUNER GOREN, H., TUNALI, S., AND JANS, R. A review of applications of genetic algorithms in lot sizing. *Journal of Intelligent Manufacturing* 21, 4 (Dec. 2008), 575–590.
- [99] GUPTA, D., AND MAGNUSSON, T. The capacitated lot-sizing and scheduling problem with sequence-dependent setup costs and setup times. *Computers & Operations Research* 32, 4 (Apr. 2005), 727–747.
- [100] HAASE, K. Capacitated lot-sizing with sequence dependent setup costs. *OR Spectrum* 18, 1 (1996), 51–59.
- [101] HAASE, K. Capacitated Lot-Sizing with Linked Production Quantities of Adjacent Periods 1 Introduction. In *Beyond Manufacturing Resource Planning (MRP II)*. Springer Berlin Heidelberg, 1998, ch. 3, pp. 127–146.
- [102] HAASE, K., AND KIMMS, A. Lot sizing and scheduling with sequence-dependent setup costs and times and efficient rescheduling opportunities. *International Journal of Production Economics* 66 (2000), 159–169.
- [103] HAKSÖZ, C., AND PINEDO, M. Economic lot scheduling with resources in parallel. *International Journal of Production Research* 49, 9 (May 2011), 2625–2641.
- [104] HANSEN, P., AND MLADENović, N. Variable neighborhood search : Principles and applications. *European Journal of Operational Research* 130, 3 (2001), 449–467.
- [105] HANSEN, P., MLADENović, N., AND PEREZ-BRITO, D. Variable Neighborhood Decomposition Search. *Journal of Heuristics* 7, 4 (2001), 335–350.

- [106] HARIRI, A. M. A., AND POTTS, C. N. Single machine scheduling with batch set-up times to minimize maximum lateness. *Annals of Operations Research* 70 (1997), 75–92.
- [107] HARRIS, F. W. How Many Parts to Make at Once. *Operations Research* 38, 6 (Dec. 1990), 947–950.
- [108] HART, P. E., NILSSON, N. J., AND RAPHAEL, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics* 4, 2 (1968), 100–107.
- [109] HAUPT, R. A survey of priority rule-based scheduling. *OR Spectrum* 11, 1 (1989), 3–16.
- [110] HELBER, S., AND SAHLING, F. A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics* 123, 2 (Feb. 2010), 247–256.
- [111] HELBER, S., SAHLING, F., AND SCHIMMELPFENG, K. Dynamic capacitated lot sizing with random demand and dynamic safety stocks. *OR Spectrum* (Feb. 2012).
- [112] HINDI, K. S. Solving the single-item, capacitated dynamic lot-sizing problem with startup and reservation costs by tabu search. *Computers & Industrial Engineering* 28, 4 (1995), 701–707.
- [113] HINDI, K. S. Solving the CLSP by a Tabu Search Heuristic. *Journal of the Operational Research Society* 47, 1 (1996), 151–161.
- [114] HOITOMT, D. J., LUH, P. B., AND PATTIPATI, K. R. A Practical Approach to Job-Shop Scheduling Problem. *IEEE Transactions on Robotics and Automation* 9, 1 (1993), 1–13.
- [115] HSU, W.-L. On the General Feasibility Test of Scheduling Lot Sizes for Several Products on One Machine. *Management Science* 29, 1 (1983), 93–105.
- [116] HUNG, Y.-F., AND CHIEN, K.-L. A Multi-Class Multi-Level Capacitated Lot Sizing Model. *Journal of the Operational Research Society* 51, 11 (2000), 1309–1318.
- [117] IRANPOOR, M., FATEMI GHOMI, S. M. T., AND MOHAMADINIA, A. Earliness tardiness production planning and scheduling in flexible flowshop systems under finite planning horizon. *Applied Mathematics and Computation* 184, 2 (Jan. 2007), 950–964.
- [118] JACOBS, F. R., BERRY, W. L., WHYBARK, D. C., AND VOLLMANN, T. E. *Manufacturing Planning and Control for Supply Chain Management : APICS/CPIM Certification Edition*. McGraw-Hill, 2011.
- [119] JAIN, A. S., AND MEERAN, S. Deterministic job-shop scheduling : Past, present and future. *European Journal of Operational Research* 113, 2 (Mar. 1999), 390–434.

- [120] JAMES, R. J. W., AND ALMADA-LOBO, B. Single and parallel machine capacitated lotsizing and scheduling : New iterative MIP-based neighborhood search heuristics. *Computers & Operations Research* 38, 12 (Dec. 2011), 1816–1825.
- [121] JANS, R., AND DEGRAEVE, Z. An industrial extension of the discrete lot-sizing and scheduling problem. *IIE Transactions* 36, 1 (2004), 47–58.
- [122] JORDAN, C., AND DREXL, A. Discrete Lotsizing and Scheduling by Batch Sequencing. *Management Science* 44, 5 (1998), 698–713.
- [123] JORDAN, C., AND KOPPELMANN, J. Multi-level lotsizing and scheduling by batch sequencing. *The Journal of the Operational Research Society* 49, 11 (1998), 1212–1218.
- [124] KACEM, I. Approximation algorithms for the makespan minimization with positive tails on a single machine with a fixed non-availability interval. *Journal of Combinatorial Optimization* 17, 2 (Sept. 2007), 117–133.
- [125] KACEM, I., CHU, C., AND SOUISSI, A. Single-machine scheduling with an availability constraint to minimize the weighted sum of the completion times. *Computers & Operations Research* 35, 3 (Mar. 2008), 827–844.
- [126] KACEM, I., AND KELLERER, H. Fast approximation algorithms to minimize a special weighted flow-time criterion on a single machine with a non-availability interval and release dates. *Journal of Scheduling* 14, 3 (Dec. 2011), 257–265.
- [127] KACZMARCZYK, W. Proportional lot-sizing and scheduling problem with identical parallel machines. *International Journal of Production Research* 49, 9 (2011), 2605–2623.
- [128] KANNEGIESSER, M. Value Chain Management. In *Value Chain Management in the Chemical Industry : Global Value Chain Planning of Commodities*, 1st ed. Physica-Verlag HD, Heidelberg, 2008, pp. 11–61.
- [129] KARIMI, B., FATEMI GHOMI, S. M. T., AND WILSON, J. M. The capacitated lot sizing problem : a review of models and algorithms. *Omega* 31, 5 (Oct. 2003), 365–378.
- [130] KARMARKAR, U. S., KEKRE, S., AND KEKRE, S. The Dynamic Lot-Sizing Problem with Startup and Reservation Costs. *Operations Research* 35, 3 (1987), 389–398.
- [131] KARMARKAR, U. S., AND SCHRAGE, L. The Deterministic Dynamic Product Cycling Problem. *Operations Research* 33, 2 (1985), 326–345.
- [132] KIMMS, A. Competitive methods for multi-level lot sizing and scheduling : tabu search and randomized regrets. *International Journal of Production Economics* 34, 8 (1996), 2279–2298.

- [133] KIMMS, A. Multi-level, single-machine lot sizing and scheduling (with initial inventory). *European Journal of Operational Research* 89, 1-2 (Feb. 1996), 86–99.
- [134] KIMMS, A. A genetic algorithm for multi-level, multi-machine lot sizing and scheduling. *Computers & Operations Research* 26, 8 (July 1999), 829–848.
- [135] KIMMS, A., AND DREXL, A. Proportional lot sizing and scheduling : Some extensions. *Networks* 32, 2 (Sept. 1998), 85–101.
- [136] KIMMS, A., AND DREXL, A. Some Insights into Proportional Lot Sizing and Scheduling. *The Journal of the Operational Research Society* 49, 11 (1998), 1196–1205.
- [137] KIRKPATRICK, S., GELATT, C. D., AND VECCHI, M. P. Optimization by Simulated Annealing. *Science* 220, 4598 (1983), 671–680.
- [138] KOLISCH, R. *Make-to-Order Assembly Management*, 1st ed. Springer, 2001.
- [139] KORF, R. E. Depth-First Iterative-Deeping : An Optimal Admissible Tree Search. *Artificial Intelligence* 27, 1 (1985), 97–109.
- [140] KOVÁCS, A., BROWN, K. N., AND ARMAGAN TARIM, S. An efficient MIP model for the capacitated lot-sizing and scheduling problem with sequence-dependent setups. *International Journal of Production Economics* 118, 1 (Mar. 2009), 282–291.
- [141] LAMOURE, S., AND THOMAS, A. Flux poussés : MRP et DRP. *Techniques de l'ingénieur Modes de pilotage des flux logistiques base docum*, ref. article : ag5110 (2000), 1–13.
- [142] LAND, A. H., AND DOIG, A. G. An Automatic Method of Solving Discrete Programming Problems. *Econometrica* 28, 3 (1960), 497–520.
- [143] LANG, J. C., AND SHEN, Z.-J. M. Fix-and-optimize heuristics for capacitated lot-sizing with sequence-dependent setups and substitutions. *European Journal of Operational Research* 214, 3 (Nov. 2011), 595–605.
- [144] LASSERRE, J.-B. An Integrated Model for Job-Shop Planning and Scheduling. *Management Science* 38, 8 (1992), 1201–1211.
- [145] LEE, H. L., PADMANABHAN, V., AND WHANG, S. The Bullwhip Effect In Supply Chains. *Sloan Management Review* 38, 3 (1997), 93–102.
- [146] LEMARÉCHAL, C. Lagrangian Relaxation. In *Computational Combinatorial Optimization*, M. Jünger and D. Naddef, Eds. Springer-Verlag, Berlin, 2001, ch. 2241, pp. 112–156.
- [147] LENSTRA, J. K., RINNOOY KAN, A. H. G., AND BRUCKER, P. Complexity of Machine Scheduling Problems. *Annals of Discrete Mathematics* 1 (1977), 343–362.
- [148] LEON, A. *Enterprise Resource Planning*, 2nd ed. McGraw-Hill, New Delhi, 2008.

- [149] LEW, A., AND MAUCH, H. *Dynamic Programming : A Computational Tool*. Springer-Verlag, Berlin, Heidelberg, 2007.
- [150] LI, Z., AND IERAPETRITOU, M. G. Integrated production planning and scheduling using a decomposition framework. *Chemical Engineering Science* 64, 16 (Aug. 2009), 3585–3597.
- [151] LI, Z., AND IERAPETRITOU, M. G. Production planning and scheduling integration through augmented Lagrangian optimization. *Computers & Chemical Engineering* 34, 6 (June 2010), 996–1006.
- [152] LI, Z., AND IERAPETRITOU, M. G. Rolling horizon based planning and scheduling integration with production capacity consideration. *Chemical Engineering Science* 65, 22 (Nov. 2010), 5887–5900.
- [153] LIAW, C.-F. An iterative improvement approach for the nonpreemptive open shop scheduling problem. *European Journal of Operational Research* 111, 3 (1998), 509–517.
- [154] MAES, J., MCCLAIN, J. O., AND VAN WASSENHOVE, L. N. Multilevel capacitated lotsizing complexity and LP-based heuristics. *European Journal of Operational Research* 53 (1991), 131–148.
- [155] MARAVELIAS, C. T., AND SUNG, C. Integration of production planning and scheduling : Overview, challenges and opportunities. *Computers & Chemical Engineering* 33, 12 (Dec. 2009), 1919–1930.
- [156] MARINELLI, F., NENNI, M. E., AND SFORZA, A. Capacitated lot sizing and scheduling with parallel machines and shared buffers : A case study in a packaging company. *Annals of Operations Research* 150, 1 (Jan. 2007), 177–192.
- [157] MAROS, I. *Computational Techniques of the Simplex Method*, 1st ed. Kluwer Academic Publishers, 2003.
- [158] MATEUS, G. R., GÓMEZ RAVETTI, M., DE SOUZA, M. C., AND VALERIANO, T. M. Capacitated lot sizing and sequence dependent setup scheduling : an iterative approach for integration. *Journal of Scheduling* 13, 3 (Dec. 2010), 245–259.
- [159] MATI, Y., DAUZÈRE-PÉRÈS, S., AND LAHLOU, C. A general approach for optimizing regular criteria in the job-shop scheduling problem. *European Journal of Operational Research* 212, 1 (July 2011), 33–42.
- [160] MEHRA, A., MINIS, I., AND PROTH, J. M. Hierarchical production planning for complex manufacturing systems. *Advances in Engineering Software* 26 (1996), 209–218.

- [161] MENEZES, A. A., CLARK, A. R., AND ALMADA-LOBO, B. Capacitated lot-sizing and scheduling with sequence-dependent, period-overlapping and non-triangular setups. *Journal of Scheduling* 14, 2 (Sept. 2010), 209–219.
- [162] MEYR, H. Simultaneous lotsizing and scheduling by combining local search with dual reoptimization. *European Journal of Operational Research* 120, 2 (Jan. 2000), 311–326.
- [163] MEYR, H. Simultaneous lotsizing and scheduling on parallel machines. *European Journal of Operational Research* 139, 2 (June 2002), 277–292.
- [164] MILLAR, H. H., AND YANG, M. Lagrangian heuristics for the capacitated multi-item lot-sizing problem with backordering. *International Journal of Production Economics* 34, 1 (1994), 1–15.
- [165] MIN, H., AND ZHOU, G. Supply chain modeling : past, present and future. *Computers & Industrial Engineering* 43, 1-2 (July 2002), 231–249.
- [166] MLADENOVIC, N., AND HANSEN, P. Variable neighborhood search. *Computers & Operations Research* 24, 11 (1997), 1097–1100.
- [167] MOHAMMADI, M. Integrating lotsizing, loading, and scheduling decisions in flexible flow shops. *International Journal of Advanced Manufacturing Technology* 50, 9-12 (Mar. 2010), 1165–1174.
- [168] MOHAMMADI, M., FATEMI GHOMI, S. M. T., KARIMI, B., AND TORABI, S. A. Rolling-horizon and fix-and-relax heuristics for the multi-product multi-level capacitated lotsizing problem with sequence-dependent setups. *Journal of Intelligent Manufacturing* 21, 4 (Dec. 2008), 501–510.
- [169] MOHAMMADI, M., FATEMI GHOMI, S. M. T., KARIMI, B., AND TORABI, S. A. MIP-based heuristics for lotsizing in capacitated pure flow shop with sequence-dependent setups. *International Journal of Production Research* 48, 10 (May 2010), 2957–2973.
- [170] MOHAN, S., GOPALAKRISHNAN, M., MARATHE, R., AND RAJAN, A. A note on modelling the capacitated lot-sizing problem with set-up carryover and set-up splitting. *International Journal of Production Research* 50, 19 (2012), 5538–5543.
- [171] MURTHY, P. R. *Production and Operations Management*, 2nd ed. New Age International Pvt Ltd Publishers, 2009.
- [172] NERI, F., COTTA, C., AND MOSCATO, P. *Handbook of Memetic Algorithms*. Springer, 2012.
- [173] NG, C. T., WANG, J.-B., CHENG, T. C. E., AND LIU, L. L. A branch-and-bound algorithm for solving a two-machine flow shop problem with deteriorating jobs. *Computers & Operations Research* 37, 1 (Jan. 2010), 83–90.

- [174] OUENNICHE, J., BOCTOR, F. F., AND MARTEL, A. The impact of sequencing decisions on multi-item lot sizing and scheduling in flow shops. *International Journal of Production Research* 37, 10 (1999), 2253–2270.
- [175] OUERFELLI, H., DAMMAK, A., AND KALLEL CHTOUROU, E. K. Benders decomposition method for an integrated problem of production Lot-sizing and Scheduling. In *International Conference on Computers & Industrial Engineering* (Troyes, 2009), IEEE, pp. 323–327.
- [176] ÖZDAMAR, L., AND BARBAROSOGLU, G. An integrated Lagrangean relaxation-simulated annealing approach to the multi-level multi-item capacitated lot sizing problem. *International Journal of Production Economics* 68, 3 (Dec. 2000), 319–331.
- [177] ÖZDAMAR, L., AND BOZYEL, M. A. The capacitated lot sizing problem with overtime decisions and setup times. *IIE Transactions* 32, 11 (2000), 1043–1057.
- [178] PATIL, M., AKRANI, G., AND KATYANI, M. Types of Production System - Intermittent and Continuous. Kalyan City Life. <http://kalyan-city.blogspot.com/2012/02/types-of-production-system-intermittent.html>, 2012. [Online; accessed 20-January-2014].
- [179] PEREIRA LOPES, M. J., AND DE CARVALHO, J. V. A branch-and-price algorithm for scheduling parallel machines with sequence dependent setup times. *European Journal of Operational Research* 176, 3 (Feb. 2007), 1508–1527.
- [180] PEZZELLA, F., MORGANTI, G., AND CIASCETTI, G. A genetic algorithm for the Flexible Job-shop Scheduling Problem. *Computers & Operations Research* 35, 10 (2008), 3202–3212.
- [181] PINEDO, M. L. *Planning and scheduling in manufacturing and services*, 2nd ed. Springer Science+Business Media, Inc., New York, 2009.
- [182] PINEDO, M. L. *Scheduling - Theory, Algorithms, and Systems*, 4th ed. Springer Science+Business Media, Inc., New York, 2012.
- [183] PISTIKOPOULOS, E. N., GEORGIADIS, M. C., AND DUA, V. *Multi-parametric programming : theory, algorithms and applications*. Wiley-VCH, 2007.
- [184] PLEVIN, R. J., O’HARE, M., JONES, A. D., TORN, M. S., AND GIBBS, H. K. Greenhouse gas emissions from biofuels’ indirect land use change are uncertain but may be much greater than previously estimated. *Environmental science & technology* 44, 21 (Nov. 2010), 8015–8021.
- [185] POCHE, Y., AND WOLSEY, L. A. *Production Planning by Mixed Integer Programming*. Springer Science+Business Media, Inc., New York, 2006.

- [186] POTTS, C. N., AND KOVALYOV, M. Y. Scheduling with batching : A review. *European Journal of Operational Research* 120, 2 (Jan. 2000), 228–249.
- [187] PRASAD, S., AND LIAO, Y. Multi-Product Lot Scheduling with Backordering and Shelf-Life Constraints. *Omega* (June 2012).
- [188] PTAK, C. A., AND SMITH, C. *Orlicky's Material Requirements Planning*, 3rd ed. McGraw-Hill, 2011.
- [189] QUADT, D., AND KUHN, H. Capacitated lot-sizing with extensions : a review. *4OR* 6 (Oct. 2008), 61–83.
- [190] RÍOS-MERCADO, R. Z., AND BARD, J. F. Computational experience with a branch-and-cut algorithm for flowshop scheduling with setups. *Computers & Operations Research* 25, 5 (1998), 351–366.
- [191] ROCHA DE PAULA, M., GÓMEZ RAVETTI, M., MATEUS, G. R., AND PARDALOS, P. M. Solving parallel machines scheduling problems with sequence-dependent setup times using variable neighbourhood search. *IMA Journal of Management Mathematics* 18, 2 (Mar. 2007), 101–115.
- [192] ROSSI, F., VAN BEEK, P., AND WALSH, T. *Handbook of Constraint Programming*, 1st ed., vol. 2 of *Foundations of Artificial Intelligence*. Elsevier, Amsterdam, 2006.
- [193] ROUX, W., DAUZÈRE-PÉRÈS, S., AND LASSERRE, J.-B. Planning and scheduling in a multi-site environment. *Production Planning & Control* 10, 1 (1999), 19–28.
- [194] ROY, B., AND SUSSMANN, B. *Les problèmes d'ordonnancement avec contraintes disjonctives*. SEMA, Note D.S. No. 9 bis, Montrouge, 1964.
- [195] RUSZCZYŃSKI, A. On convergence of an Augmented Lagrangian Decomposition Method for Sparse Convex Optimization. *Mathematics of Operations Research* 20, 3 (1995), 634–656.
- [196] SADFI, C., PENZ, B., RAPINE, C., BŁAŻEWICZ, J., AND FORMANOWICZ, P. An improved approximation algorithm for the single machine total completion time scheduling problem with availability constraints. *European Journal of Operational Research* 161, 1 (Feb. 2005), 3–10.
- [197] SAHLING, F., BUSCHKÜHL, L., TEMPELMEIER, H., AND HELBER, S. Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. *Computers & Operations Research* 36, 9 (Sept. 2009), 2546–2553.
- [198] SALOMON, M., KROON, L. G., KUIK, R., AND VAN WASSENHOVE, L. N. Some Extensions of the Discrete Lotsizing and Scheduling Problem. *Management Science* 37, 7 (1991), 801–812.

- [199] SALOMON, M., SOLOMON, M. M., VAN WASSENHOVE, L. N., DUMAS, Y., AND DAUZÈRE-PÉRÈS, S. Solving the discrete lotsizing and scheduling problem with sequence dependent set-up costs and set-up times using the Travelling Salesman Problem with time windows. *European Journal of Operational Research* 100 (1997), 494–513.
- [200] SANDBOTHE, R. A. A User Interactive Heuristic Procedure for Solving the Multiple Product Cycling Problem. *Computers & Operations Research* 23, 9 (1996), 897–907.
- [201] SEEANNER, F., ALMADA-LOBO, B., AND MEYR, H. Combining the principles of variable neighborhood decomposition search and the fix&optimize heuristic to solve multi-level lot-sizing and scheduling problems. *Computers & Operations Research* 40, 1 (Jan. 2013), 303–317.
- [202] SEEANNER, F., AND MEYR, H. Multi-stage simultaneous lot-sizing and scheduling for flow line production. *OR Spectrum* 35, 1 (2013), 33–73.
- [203] SHAH, N. K., AND IERAPETRITOU, M. G. Integrated production planning and scheduling optimization of multisite, multiproduct process industry. *Computers & Chemical Engineering* 37 (Feb. 2012), 214–226.
- [204] SHEIKH, K. *Manufacturing Resource Planning (MRP II) : With an introduction to ERP, SCM, and CRM*, 1st ed. McGraw-Hill, 2003.
- [205] SHIM, I.-S., KIM, H.-C., DOH, H.-H., AND LEE, D.-H. A two-stage heuristic for single machine capacitated lot-sizing and scheduling with sequence-dependent setup costs. *Computers & Industrial Engineering* 61, 4 (Nov. 2011), 920–929.
- [206] SILVER, E., AND MEAL, H. 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 (1973), 64–74.
- [207] STADTLER, H. Mixed integer programming model formulations for dynamic multi-item multi-level capacitated lotsizing. *European Journal of Operational Research* 94, 3 (Nov. 1996), 561–581.
- [208] STADTLER, H. Multi-level single machine lot-sizing and scheduling with zero lead times. *European Journal of Operational Research* 209, 3 (Mar. 2011), 241–252.
- [209] STADTLER, H., FLEISCHMANN, B., GRUNOW, M., MEYR, H., AND SÜRIE, C. Advanced Planning in Supply Chains : Illustrating the Concepts Using an SAP APO Case Study. In *Management for Professionals*. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2012, ch. 2, pp. 21–34.

- [210] STADTLER, H., AND KILGER, C. *Supply Chain Management and Advanced Planning : Concepts, Models, Software and Case Studies*, 4th ed. Springer Science+Business Media, Inc., Berlin, Heidelberg, New York, 2008.
- [211] SUERIE, C. Modeling of period overlapping setup times. *European Journal of Operational Research* 174, 2 (Oct. 2006), 874–886.
- [212] TALBI, E.-G. *Metaheuristics From Design to Implementation*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2009.
- [213] TEMPELMEIER, H. A column generation heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint. *Omega* 39, 6 (Dec. 2011), 627–633.
- [214] TEMPELMEIER, H., AND BUSCHKÜHL, L. Dynamic multi-machine lotsizing and sequencing with simultaneous scheduling of a common setup resource. *International Journal of Production Economics* 113, 1 (May 2008), 401–412.
- [215] TEMPELMEIER, H., AND DERSTROFF, M. A Lagrangean-Based Heuristic for Dynamic Multilevel Multiitem Constrained Lotsizing with Setup Times. *Management Science* 42, 5 (1996), 738–757.
- [216] TEMPELMEIER, H., AND HELBER, S. A heuristic for dynamic multi-item multi-level capacitated lotsizing for general product structures. *European Journal of Operational Research* 75, 2 (1994), 296–311.
- [217] TEMPELMEIER, H., AND HERPERS, S. ABC β -a heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint. *International Journal of Production Research* 48, 17 (2010), 5181–5193.
- [218] THIZY, J. M. Analysis of Lagrangian Decomposition for the Multi-Item Capacitated Lot-Sizing Problem. *Information Systems & Operational Research* 29, 4 (1991), 271–283.
- [219] THIZY, J. M., AND VAN WASSENHOVE, L. N. Lagrangean Relaxation for the Multi-Item Capacitated Lot-Sizing Problem : A Heuristic Implementation. *IIE Transactions* 17, 4 (1985), 308–313.
- [220] TIACCI, L., AND SAETTA, S. Demand forecasting, lot sizing and scheduling on a rolling horizon basis. *International Journal of Production Economics*, 1997 (Feb. 2012), 1–12.
- [221] TOLEDO, F. M. B., AND ARMENTANO, V. A. A Lagrangian-based heuristic for the capacitated lot-sizing problem in parallel machines. *European Journal of Operational Research* 175, 2 (Dec. 2006), 1070–1083.
- [222] TOSO, E. A. V., MORABITO, R., AND CLARK, A. R. Lot sizing and sequencing optimisation at an animal-feed plant. *Computers & Industrial Engineering* 57, 3 (Oct. 2009), 813–821.

- [223] TRIGEIRO, W. W., THOMAS, L. J., AND MCCLAIN, J. O. Capacitated Lot Sizing with Setup Times. *Management Science* 35, 3 (Mar. 1989), 353–366.
- [224] VAESSENS, R. J. M., AARTS, E. H. L., AND LENSTRA, J. K. Job Shop Scheduling by Local Search. *INFORMS Journal on Computing* 8, 3 (1996), 302–317.
- [225] VAN HOESEL, S., KUIK, R., SALOMON, M., AND VAN WASSENHOVE, L. N. The single-item discrete lotsizing and scheduling problem : optimization by linear and dynamic programming. *Discrete Applied Mathematics* 48, 3 (Feb. 1994), 289–303.
- [226] VAN LAARHOVEN, P. J. M., AARTS, E. H. L., AND LENSTRA, J. K. Job Shop Scheduling by Simulated Annealing. *Operations Research* 40, 1 (1992), 113–125.
- [227] VANDERBECK, F. Lot-Sizing with Start-Up Times. *Management Science* 44, 10 (Oct. 1998), 1409–1425.
- [228] ČERNÝ, V. Thermodynamical Approach to the Traveling Salesman Problem : An Efficient Simulation Algorithm. *Journal of Optimization Theory and Applications* 45, 1 (1985), 41–51.
- [229] VENKATESWARAN, J., AND SON, Y.-J. Hybrid system dynamic-discrete event simulation-based architecture for hierarchical production planning. *International Journal of Production Research* 43, 20 (Oct. 2005), 4397–4429.
- [230] WAGELMANS, A., VAN HOESEL, S., AND KOLEN, A. Economic Lot Sizing : An $O(n \log n)$ Algorithm That Runs in Linear Time in the Wagner-Whitin Case. *Operations Research* 40, 1 (1992), S145–S156.
- [231] WAGNER, H. M., AND WHITIN, T. M. Dynamic version of the economic lot size model. *Management Science* 5, 1 (1958), 89–96.
- [232] WALIGÓRA, G. Tabu search for discrete-continuous scheduling problems with heuristic continuous resource allocation. *European Journal of Operational Research* 193, 3 (Mar. 2009), 849–856.
- [233] WOLOSEWICZ, C. *Approche intégrée en planification et ordonnancement de la production*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne, 2008.
- [234] WOLOSEWICZ, C., DAUZÈRE-PÉRÈS, S., AND AGGOUNE, R. A new approach for solving integrated planning and scheduling problem. In *INCOM2006* (2006).
- [235] WOLOSEWICZ, C., DAUZÈRE-PÉRÈS, S., AND AGGOUNE, R. A solving procedure for a general integrated lot sizing and fixed scheduling problem. Submitted to *European Journal of Operational Research* (2012).
- [236] WOLSEY, L. A. *Integer Programming*, 1st ed. John Wiley & Sons, Inc., 1998.

- [237] WOLSEY, L. A. Solving Multi-Item Lot-Sizing Problems with an MIP Solver Using Classification and Reformulation. *Management Science* 48, 12 (2002), 1587–1602.
- [238] WU, T., AND SHI, L. Mathematical models for capacitated multi-level production planning problems with linked lot sizes. *International Journal of Production Research* 49, 20 (2011), 6227–6247.
- [239] WU, T., SHI, L., GEUNES, J., AND AKARTUNALI, K. An optimization framework for solving capacitated multi-level lot-sizing problems with backlogging. *European Journal of Operational Research* 214, 2 (Oct. 2011), 428–441.
- [240] XIAO, Y., KAKU, I., ZHAO, Q., AND ZHANG, R. A reduced variable neighborhood search algorithm for uncapacitated multilevel lot-sizing problems. *European Journal of Operational Research* 214, 2 (Oct. 2011), 223–231.
- [241] XPRESS MP. *Xpress MP software development, Dash Optimization*. Englewoods Cliffs, New Jersey, Release 2006.
- [242] ZHANG, M. Two-stage minimax regret robust uncapacitated lot-sizing problems with demand uncertainty. *Operations Research Letters* 39, 5 (Sept. 2011), 342–345.
- [243] ZHANG, X.-D., AND YAN, H.-S. Integrated optimization of production planning and scheduling for a kind of job-shop. *International Journal of Advanced Manufacturing Technology* 26, 7-8 (Jan. 2005), 876–886.
- [244] ZHU, X., AND WILHELM, W. E. Scheduling and lot sizing with sequence-dependent setup : A literature review. *IIE Transactions* 38, 11 (Nov. 2006), 987–1007.

**École Nationale Supérieure des Mines
de Saint-Étienne**

NNT :

Edwin David GOMEZ URRUTIA

INTEGRATED OPTIMIZATION OF PLANNING AND SCHEDULING DECISIONS IN A SUPPLY CHAIN

Specialty : [Industrial Engineering](#)

Keywords : Integrated approach, supply chain, production planning, scheduling, optimization.

Abstract :

In this thesis, we study the optimization of flow planning and scheduling, within a strategy to integrate decisions for supply chain planning at tactical level, taking into account operational constraints. The goal of this work is to address the need for consistency between decisions arising from production planning and scheduling. These decisions are often taken in a sequential order, leading most of the time to unfeasible production plans. We propose an integrated approach to solve single-level and multi-level problems in multi-item multi-resource systems configured as job-shops.

Both capacitated production planning and scheduling problems, in complex manufacturing systems, are \mathcal{NP} -hard. Therefore, integrating constraints of both problems generates a new problem which is even more difficult to solve. We propose a decomposition of the integrated problem into a set of several sub-problems with fixed sequence, solved by Lagrangian Relaxation. The sequence improvement is guided by a Tabu Search. The efficiency of the integrated approach comparing to a standard solver is proved in terms of solution quality and computational effort. In case of multi-level problems, we propose a new mathematical model based on the concept of echelon stock, as well as new algorithms and smoothing strategies to build production plans respecting detailed capacity and bill-of-materials constraints.

**École Nationale Supérieure des Mines
de Saint-Étienne**

NNT :

Edwin David GOMEZ URRUTIA

OPTIMISATION INTÉGRÉE DES DÉCISIONS EN PLANIFICATION ET ORDONNANCEMENT DANS UNE CHAÎNE LOGISTIQUE

Spécialité : **Génie Industriel**

Mots Clefs : approche intégrée, chaîne logistique, planification, ordonnancement, optimisation.

Résumé :

Dans cette thèse, nous étudions l'optimisation des problèmes de planification et d'ordonnancement des flux, dans une stratégie d'intégration des décisions, pour planifier la chaîne logistique au niveau tactique avec prise en compte de contraintes opérationnelles. Le but de ce travail est de répondre au besoin de cohérence entre les décisions de planification et d'ordonnancement, qui sont souvent prises de manière séquentielle ne garantissant pas la faisabilité des plans de production. Nous proposons une approche intégrée pour résoudre des problèmes mono-niveau et multi-niveaux, dans des systèmes multi-produits et multi-ressources dans des ateliers de type job-shop.

Les problèmes de planification avec contraintes de capacité et les problèmes d'ordonnancement dans des systèmes complexes sont des problèmes \mathcal{NP} -difficiles. Intégrer les contraintes propres aux deux problèmes engendre un nouveau problème qui est d'autant plus complexe. Nous proposons une décomposition du problème intégré en un ensemble de sous-problèmes de planification avec séquence fixée, résolus par relaxation Lagrangienne. L'amélioration de la séquence est guidée par une recherche taboue. L'efficacité de l'approche intégrée, par rapport à un solveur commercial, a été prouvée en termes de qualité des solutions et d'effort de calcul. Pour les problèmes multi-niveaux, nous proposons une nouvelle formulation basée sur la notion d'échelon stock, ainsi que de nouveaux algorithmes et stratégies de lissage de la production, pour construire des plans de production respectant les contraintes de capacité détaillées et de nomenclature.