

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE



UNIVERSITE BATNA 2  
DÉPARTEMENT DE GÉNIE INDUSTRIEL

*Laboratoire d'Automatique et Productique*

## THÈSE

Présentée en vue d'obtenir le grade de

## DOCTORAT EN SCIENCES

Par

AHMED SENOUSSE

Thème

OPTIMISATION INTÉGRÉE DES FONCTIONS  
PRODUCTION, STOCKAGE ET DISTRIBUTION DANS  
UNE CHAÎNE LOGISTIQUE

Pr.	LEILA HAYET MOUSS	Université Batna 2	(Présidente)
Pr.	BERNARD PENZ	Université Grenoble Alpes	(Examineur)
Pr.	STÉPHANE DAUZÈRE-PÉRÈS	Ecole des Mines de Saint-Etienne	(Examineur)
Pr.	LYES BENYOUCEF	Université d'Aix-Marseille	(Examineur)
Pr.	ABDELKAMEL TARI	Université de Béjaïa	(Examineur)
Pr.	NADIA KINZA MOUSS	Université Batna 2	(Directrice de thèse)
Dr.	NADJIB BRAHIMI	Ecole des Mines de Nantes	(Invité)

2015/2016



## **D**édicaces

*A la mémoire de mon cher père, pour les sacrifices qu'il a consenti pour mon instruction et mon bien être. J'aurais tant aimé qu'il soit présent pour partager avec moi cet événement... Allah Yarhamou.*

*A ma très chère mère pour sa tendresse, sa patience et sa générosité et qui n'a cessé de prier pour moi.*

*A ma femme qui n'a ménagé aucun effort à mon égard et qui a su me remonter le moral dans les moments les plus difficiles.*

*A ceux qui ont donné le goût à ma vie, mes trois enfants : Ajoued, Achraf et Atyab.*

*A mon frère et mes sœurs.*

*A toute ma famille.*

*A tout mes amis.*

*A tous ceux que ma réussite leur tient à cœur.*



# REMERCIEMENTS

**J**E voudrais remercier tout d'abord le bon Dieu, qui m'a donné la patience l'ambition, le défi, la santé et le courage pour terminer cette thèse.

J'exprime mes plus profonds remerciements à Madame le Professeur Nadia Kinza Mouss pour m'avoir encadré et apporté son soutien tout au long de cette thèse, pour ses conseils, pour le temps qu'elle m'a consacré, ainsi pour sa confiance et sa patience. J'apprécie ses qualités humaines d'écoute et de compréhension tout au long de ce travail.

Je tiens à remercier Madame le Professeur Leila Hayet Mouss la directrice du Laboratoire d'Automatique et Productique (LAP) pour sa rigueur, son dynamisme et pour l'honneur qu'elle m'a fait en acceptant d'être présidente de mon jury de thèse. Je tiens à l'assurer de ma profonde reconnaissance pour l'intérêt qu'elle porte à ce travail.

Je tiens également à remercier le Professeur Bernard Penz pour m'avoir accueilli plusieurs fois au sein du laboratoire G-SCOP de Grenoble, pour ses idées qui m'ont beaucoup aidé pour me faire avancer dans cette thèse, et pour avoir participé à mon jury de thèse.

Je remercie le Professeur Stéphane Dauzère-Pérès pour son accueil chaleureux lors de ma visite au Centre Microélectronique de Provence (CMP) de Gardanne. Son contact a été très enrichissant tant au niveau humain qu'au niveau scientifique.

Un grand merci au Professeur Lyes Ben Youcef pour son intérêt permanent à mon égard, pour son soutien sur le plan humain et pour l'honneur qu'il m'a fait de siéger parmi les membres du jury.

Je souhaiterais exprimer ma gratitude au Professeur Abdelkamel Tari pour l'intérêt qu'il a porté à ce travail, d'avoir accepté de faire partie de mon jury et d'avoir apporté des remarques pertinentes.

Ce travail n'aurait pu être mené à bien sans l'aide du Docteur Nadjib Brahimi, auquel j'adresse un remerciement particulier pour les suggestions et les remarques judicieuses qu'il m'a indiqué, pour une relecture très rigoureuse de ma thèse et pour l'intérêt qu'il a manifesté en participant en qualité de membre invité à ce jury.

Je conclurais en remerciant de tout cœur mes amis et collègues de département Génie Industriel pour leur aide, leur soutien et leur gentillesse.

# TABLE DES MATIÈRES

<b>Table des matières</b>	<b>vi</b>
<b>Liste des figures</b>	<b>x</b>
<b>Liste des tableaux</b>	<b>xi</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Positionnement du problème</b>	<b>5</b>
1.1 INTRODUCTION . . . . .	7
1.2 LOGISTIQUE ET CHAÎNE LOGISTIQUE . . . . .	7
1.3 DÉFINITIONS DE LA CHAÎNE LOGISTIQUE . . . . .	7
1.4 LES ACTEURS DE LA CHAÎNE LOGISTIQUE . . . . .	8
1.5 LES STRUCTURES TYPIQUES DE LA CHAÎNE LOGISTIQUE . . . . .	8
1.6 LES FONCTIONS DE LA CHAÎNE LOGISTIQUE . . . . .	9
1.6.1 Approvisionnement . . . . .	9
1.6.2 Production . . . . .	10
1.6.3 Stockage . . . . .	10
1.6.4 Distribution et transport . . . . .	10
1.6.5 Vente . . . . .	11
1.7 GESTION DE LA CHAÎNE LOGISTIQUE . . . . .	11
1.8 NIVEAUX DE DÉCISION DANS LA GESTION DE LA CHAÎNE LOGISTIQUE . . . . .	12
1.8.1 Le niveau stratégique . . . . .	12
1.8.2 Le niveau tactique . . . . .	13
1.8.3 Le niveau opérationnel . . . . .	13
1.9 LA NOTION D'INTÉGRATION DANS LA CHAÎNE LOGISTIQUE . . . . .	13
1.9.1 Les mécanismes de l'intégration de la chaîne logistique . . . . .	14
1.9.2 Les dimensions de l'intégration dans la chaîne logistique . . . . .	15
1.10 PLANIFICATION DE LA CHAÎNE LOGISTIQUE OU <i>Supply Chain Planning SCP</i> . . . . .	16
1.11 OPTIMISATION DES FONCTIONS DE LA CHAÎNE LOGISTIQUE . . . . .	16
1.11.1 Optimisation séquentielle . . . . .	16
1.11.2 Optimisation intégrée . . . . .	18
1.12 LES MODÈLES D'INTÉGRATION DES FONCTIONS DE LA CHAÎNE LOGISTIQUE . . . . .	19
CONCLUSION . . . . .	21

<b>2</b>	<b>Etat de l'art : Planification de la production et distribution</b>	<b>23</b>
2.1	INTRODUCTION . . . . .	25
2.2	PROBLÈME DE DIMENSIONNEMENT DE LOTS OU <i>lot sizing problem</i> LSP . . . . .	25
2.3	CLASSIFICATION DES MODÈLES DE LOT SIZING . . . . .	26
2.4	PROBLÈME DE LOT SIZING À UN PRODUIT . . . . .	27
2.4.1	Problème de lot sizing à un produit sans capacité . . . . .	28
2.4.2	Problème de lot sizing à un produit avec capacité . . . . .	28
2.5	PROBLÈME DE LOT SIZING À PLUSIEURS PRODUITS . . . . .	29
2.5.1	Les méthodes exactes . . . . .	29
2.5.2	Les méthodes heuristiques . . . . .	29
2.5.3	Les méta-heuristiques . . . . .	30
2.6	PROBLÈME DE LOT SIZING COORDONNÉ SANS CAPACITÉ ( <i>Coordinated uncapacitated Lot-Size Problem</i> CULSP) . . . . .	30
2.7	PROBLÈME DE LOT SIZING COORDONNÉ AVEC CAPACITÉ <i>Coordinated Capacitated Lot-Size Problem</i> CCLSP . . . . .	31
2.8	LE PROBLÈME DE LOT SIZING À PLUSIEURS NIVEAUX OU <i>Multi-Level Lot Sizing Problem</i> (MLLSP) . . . . .	31
2.9	LE PROBLÈME D'UN ENTREPÔT ET PLUSIEURS DÉTAILLANTS ( <i>One-Warehouse Multi-Retailer Problem</i> ) OWMR . . . . .	32
2.10	LE PROBLÈME PRODUCTION-ROUTAGE ( <i>Production Routing Problem</i> ) PRP . . . . .	33
2.11	CONTRIBUTION DE LA THÈSE . . . . .	39
	CONCLUSION . . . . .	39
<b>3</b>	<b>Approche polyédrale pour la résolution du problème</b>	<b>41</b>
3.1	INTRODUCTION . . . . .	43
3.2	APERÇU SUR LES MÉTHODES EXACTES DE RÉOLUTION . . . . .	43
3.2.1	La programmation linéaire . . . . .	43
3.2.2	La programmation linéaire en nombres entiers . . . . .	43
3.2.3	La relaxation lagrangienne . . . . .	46
3.2.4	La génération de colonnes . . . . .	46
3.2.5	La programmation dynamique . . . . .	46
3.3	FORMULATION DU PROBLÈME . . . . .	47
3.3.1	Formulation agrégée ou <i>Aggregated Formulation</i> AGF . . . . .	47
3.3.2	Formulation à stock d'échelon ou <i>Echelon Stock Formulation</i> ESF . . . . .	49
3.3.3	Formulation basée sur le problème de localisation ou <i>Facility Location Formulation</i> FLF . . . . .	50
3.3.4	Formulation basée sur le problème du plus court chemin ou <i>Shortest Path Formulation</i> (SPF) . . . . .	51
3.4	DÉVELOPPEMENT DES INÉGALITÉS VALIDES . . . . .	53
3.4.1	Inégalités déterminant le nombre maximum de véhicules . . . . .	53
3.4.2	Inégalités éliminant la symétrie . . . . .	54
3.4.3	Inégalités de véhicule plein . . . . .	54

3.4.4	Extension des inégalités valides ( $l, S$ ) . . . . .	55
3.4.5	Inégalités valides réduisant le nombre de variables . . . . .	56
3.5	RÉSULTATS NUMÉRIQUES . . . . .	58
3.5.1	Effet de chaque famille d'inégalités valides sur la formulation AGF . . . . .	59
3.5.2	Effet de chaque famille d'inégalités valides sur la formulation ESF . . . . .	59
3.5.3	Comparaison des formulations AGF et ESF . . . . .	63
3.5.4	Evolution du Gap . . . . .	63
	CONCLUSION . . . . .	65
<b>4</b>	<b>Méta-heuristiques à base d'algorithme génétique</b>	<b>71</b>
4.1	INTRODUCTION . . . . .	73
4.2	CONCEPTS DE BASE D'UN ALGORITHME GÉNÉTIQUE . . . . .	73
4.3	CONCEPTION DES APPROCHES À BASE D'ALGORITHME GÉNÉTIQUE . . . . .	74
4.3.1	Première approche (Hybridation AG et une recherche locale) . . . . .	75
4.3.2	Deuxième approche . . . . .	85
4.3.3	Troisième approche . . . . .	86
4.3.4	Quatrième approche . . . . .	87
4.4	RÉSULTATS EXPÉRIMENTAUX . . . . .	88
4.4.1	Résultats expérimentaux sur les petites et moyennes instances . . . . .	88
4.4.2	Résultats expérimentaux sur les larges instances . . . . .	89
	CONCLUSION . . . . .	96
<b>5</b>	<b>Méta-heuristiques à base de recherche tabou et de recuit simulé</b>	<b>99</b>
5.1	INTRODUCTION . . . . .	101
5.2	LA RECHERCHE TABOU : PRINCIPES DE BASE . . . . .	101
5.2.1	Le voisinage . . . . .	101
5.2.2	La mémoire . . . . .	102
5.2.3	Aspiration . . . . .	102
5.2.4	Intensification . . . . .	102
5.2.5	Diversification . . . . .	102
5.2.6	Algorithme général de la recherche tabou . . . . .	102
5.3	DÉVELOPPEMENT DE L'APPROCHE À BASE DE LA RECHERCHE TABOU . . . . .	103
5.3.1	Définition de la solution . . . . .	103
5.3.2	Définition du voisinage . . . . .	103
5.3.3	Définition du mouvement . . . . .	103
5.3.4	Génération de la solution initiale . . . . .	104
5.3.5	Détermination des autres variables . . . . .	104
5.3.6	Evaluation d'une solution . . . . .	105
5.3.7	Structure de la liste tabou . . . . .	105
5.3.8	Détermination du seuil de faisabilité de la production . . . . .	105
5.3.9	Algorithme de l'approche à base de recherche tabou . . . . .	106

5.4	LE RECUIR SIMULÉ : PRINCIPES DE BASE . . . . .	107
5.4.1	Principe de base du recuit simulé . . . . .	107
5.4.2	Paramètres du recuit simulé . . . . .	107
5.4.3	Algorithme général de recuit simulé . . . . .	107
5.4.4	Conception de l'approche à base de recuit simulé . . . . .	108
5.5	HYBRIDATION DE LA RECHERCHE TABOU AVEC LE RECUIR SIMULÉ . . . . .	109
5.6	RÉSULTATS DE SIMULATION . . . . .	109
5.6.1	Résultats obtenus par l'approche à base de recherche tabou . . . . .	109
5.6.2	Résultats obtenus par le recuit simulé . . . . .	114
5.6.3	Comparaison des résultats donnés par la recherche tabou et le recuit simulé . . .	114
5.6.4	Résultats fournis par l'approche hybride . . . . .	114
	CONCLUSION . . . . .	115
	<b>Conclusion générale</b>	<b>119</b>
	<b>Bibliographie</b>	<b>123</b>

# LISTE DES FIGURES

1	Structure de la chaîne logistique . . . . .	2
1.1	Types des structures de la chaîne logistique . . . . .	9
1.2	Maison du Supply Chain Management. Source : Stadtler et Kilger [107] . . . . .	13
1.3	Matrice de planification de la chaîne logistique (SCP-Matrix). Source : Stadtler et Kilger [107] . . . . .	17
2.1	Du LSP au PIDP . . . . .	25
2.2	Exemple de classification des modèles de lot sizing [116] . . . . .	27
3.1	Evolution du Gap entre 5 et 100 secondes pour l'instance $6 \times 10 \times 3$ . . . . .	65
4.1	Organigramme de la première approche . . . . .	75
4.2	Exemple d'un individu . . . . .	76
4.3	Population . . . . .	76
4.4	Opération de réparation . . . . .	77
4.5	Croisement vertical . . . . .	85
4.6	Croisement horizontal . . . . .	85
4.7	Comparaison du Gap moyen . . . . .	90
5.1	Effet de la taille de la liste tabou sur le Gap moyen . . . . .	110
5.2	Effet de variation du nombre de périodes T sur le Gap moyen . . . . .	111
5.3	Effet de la variation du nombre de détaillants sur le Gap moyen . . . . .	112
5.4	Effet de la variation du nombre de véhicules sur le Gap moyen . . . . .	112
5.5	Comparaison du Gap moyen avec l'hybridation . . . . .	116

# Liste des tableaux

3.1	Paramètres des instances générées . . . . .	59
3.2	Les trois familles d'instances . . . . .	60
3.3	CPU (en secondes) en utilisant la formulation AGF sur la première famille d'instances . . . . .	60
3.4	Le Gap obtenu par la formulation AGF sur la première famille d'instances . . . . .	61
3.5	Pourcentage des instances résolues à l'optimalité par la formulation AGF avec chaque inégalité valide sur la première famille d'instances . . . . .	61
3.6	CPU (en secondes) par l'utilisation de la formulation ESF sur la première famille d'instances . . . . .	62
3.7	Le Gap obtenu par l'utilisation de la formulation ESF sur la première famille d'instances . . . . .	62
3.8	Pourcentage des instances résolues à l'optimalité par la formulation ESF avec chaque inégalité valide sur la première famille d'instances . . . . .	63
3.9	Comparaison des deux formulations sans inégalités valides sur les trois familles d'instances . . . . .	64
3.10	Comparaison des deux formulations avec toutes les inégalités valides sur les trois familles d'instances . . . . .	64
3.11	Résultats moyens sur les trois familles d'instances en utilisant la formulation AGF	66
3.12	Résultats moyens sur les trois familles d'instances en utilisant la formulation ESF	68
4.1	Gap obtenu par la première approche sans amélioration sur les petites et moyennes instances . . . . .	91
4.2	Gap obtenu par la première approche avec amélioration sur les petites et moyennes instances . . . . .	92
4.3	Gap obtenu par la deuxième approche sur les petites et moyennes instances . . . . .	93
4.4	Gap obtenu par la troisième approche sur les petites et moyennes instances . . . . .	94
4.5	Gap obtenu par la quatrième approche sur les petites et moyennes instances . . . . .	95
4.6	Gap obtenu par la première et la quatrième approche sur les instances larges . . . . .	96
4.7	Comparaison de la première et la quatrième approche avec CPLEX en terme de l'objectif et du Gap . . . . .	97
5.1	Effet de la variation de la taille de la liste tabou sur le Gap moyen . . . . .	110
5.2	Gap obtenu par la recherche tabou avec $\alpha = 5\%$ . . . . .	113
5.3	Variation du Gap moyen en fonction des paramètres $T$ , $J$ et $K$ . . . . .	114

5.4	Résultats obtenus par le recuit simulé . . . . .	115
5.5	Résultats obtenus par l'hybridation Tabou-Recuit . . . . .	117

# LISTE DES ABRÉVIATIONS

<b>AGF</b>	Aggregated Formulation.
<b>APS</b>	Advanced Planning and Scheduling.
<b>B &amp; B</b>	Branch and Bound (séparation et évaluation).
<b>B &amp; C</b>	Branch and Cut (méthode des coupes).
<b>CCLSP</b>	Coordinated Capacitated Lot Size Problem.
<b>CLSP</b>	Capacitated Lot Sizing Problem.
<b>CPFR</b>	Collaborative Planning, Forecasting and Replenishment.
<b>CRP</b>	Continuous Replenishment Policy.
<b>CULSP</b>	Coordinated uncapacitated Lot Sizing Problem.
<b>DJRP</b>	Dynamic Joint Replenishment Problem.
<b>DLSP</b>	Discrete Lot Sizing and Scheduling Problem.
<b>ELSP</b>	Economic Lot Scheduling Problem.
<b>EOQ</b>	Economic Order Quantity.
<b>ERP</b>	Enterprise Resource Planning.
<b>ESF</b>	Echelon Stock Formulation.
<b>FLF</b>	Facility Location Formulation.
<b>GA</b>	Genetic Algorithm (algorithme génétique).
<b>GAMS</b>	General Algebraic Modeling System.
<b>JIT</b>	Just In Time (juste à temps).
<b>LB</b>	Lower Bound.
<b>LSP</b>	Lot Sizing Problem.
<b>LTL</b>	Less Than TruckLoad.
<b>MFFD</b>	Modified First Fit Decreasing.
<b>MLLSP</b>	Multi Level Lot Sizing Problem (problème de lot sizing à plusieurs niveaux).
<b>OWMR</b>	One-Warehouse-Multi-Retailer.
<b>PDL</b>	Problème de Dimensionnement de Lots.
<b>PIDP</b>	Production-Inventory-Distribution Problem.
<b>PL</b>	Programmation Linéaire.
<b>PLNE</b>	Programmation Linéaire en Nombres Entiers.
<b>PRP</b>	Production Routing Problem.
<b>QR</b>	Quick Replenishment.
<b>SC</b>	Supply Chain.
<b>SCM</b>	Supply Chain Management (gestion de la chaîne logistique).

<b>SA</b>	Simulated Annealing (recuit simulé).
<b>SPF</b>	Shortest Path Formulation.
<b>TL</b>	Truckload.
<b>TS</b>	Taboo Search (la recherche taboue).
<b>TSP</b>	Traveling Salesman Problem.
<b>UB</b>	Upper Bound.
<b>ULSP</b>	Uncapacitated Lot Sizing Problem.
<b>VI</b>	Valid Inequalities (inégalités valides).
<b>VMI</b>	Vendor Managed Inventory.
<b>VRP</b>	Vehicle Routing Problem.
<b>WW</b>	Wagner-Whitin.

# INTRODUCTION GÉNÉRALE

Dans un environnement de plus en plus concurrentiel, l'enjeu essentiel des entreprises est l'amélioration continue de leurs performances. Celles-ci dépendent essentiellement de la production des biens et services devant satisfaire les exigences des clients dans les meilleurs délais et aux moindres coûts, tout en maximisant les profits. L'amélioration des performances engage toutes les entreprises qui participent dans le processus globale allant de l'approvisionnement des matières premières, en passant par la transformation de ces matières en produits finis, en arrivant à la distribution des produits finis aux clients. Cela pousse ces entreprises ayant des intérêts communs à se regrouper dans un réseau appelé chaîne logistique.

La gestion de la chaîne logistique ou *Supply Chain Management* (SCM), s'est imposée aujourd'hui comme un concept clé et un domaine d'intérêt croissant aussi bien pour les entreprises que pour les chercheurs. Son objectif est l'optimisation globale des fonctions de la chaîne et l'intégration des niveaux de décision et les différentes activités qui en découlent.

Les principales fonctions de la chaîne logistique sont la production, le stockage et la distribution. Traditionnellement, ces fonctions sont gérées indépendamment les unes des autres. Dans ce type de gestion, chaque fonction est responsable de l'optimisation de sa propre activité. La tendance vers le *JIT* et le *Lean Manufacturing*, et par conséquent vers la réduction des stocks entraîna une interaction plus étroite entre ces fonctions. L'idée d'intégrer ces fonctions pour optimiser la chaîne logistique s'est imposée. On parle alors de de l'optimisation intégrée. Plusieurs exemples tirés de la littérature montrent que l'intégration entre au moins deux fonctions peut améliorer l'efficacité de la chaîne logistique.

Il existe de nombreux modèles, dans la littérature, impliquant l'intégration des fonctions de la chaîne logistique. Ils consistent à intégrer les activités d'au moins deux fonctions telle que la production, le stockage et la distribution, afin de minimiser le coût total engendré par ces fonctions. Ces modèles peuvent être appliqués dans le cadre de différents mécanismes de coordination qui sont proposés pour intégrer les fonctions d'une chaîne logistique. Parmi ces mécanismes figurent essentiellement le *Vendor Managed Inventory (VMI)* qui consiste à confier au fournisseur la responsabilité de gérer les stocks de ses détaillants, en fonction des informations qu'il reçoit, de manière à faciliter le réapprovisionnement.

C'est dans le contexte de *VMI* que nous étudions dans cette thèse, l'optimisation intégrée des fonctions production, stockage et distribution d'une chaîne logistique composée d'un fournisseur et un ensemble de détaillants situés dans une région formant ce qu'on appelle un *cluster*. Le fournisseur, selon le *VMI*, est responsable de la production et l'approvisionnement des stocks

des détaillants, en utilisant une flotte de véhicules de capacité limitée. La demande de chaque détaillant est dynamique sur un horizon discret et fini. La capacité de production et les capacités de stockage chez chaque détaillant sont limitées. Nous considérons également comme hypothèse dans notre cas, d'une part, qu'un véhicule peut visiter plusieurs détaillants dans une tournée, mais un détaillant peut être visité au plus par un véhicule dans une période d'autre part. La particularité de cette configuration, comme illustrée dans la figure 1, est que, la distance entre les détaillants est négligeable par rapport à la distance reliant le producteur et la région des détaillants ("*cluster*").

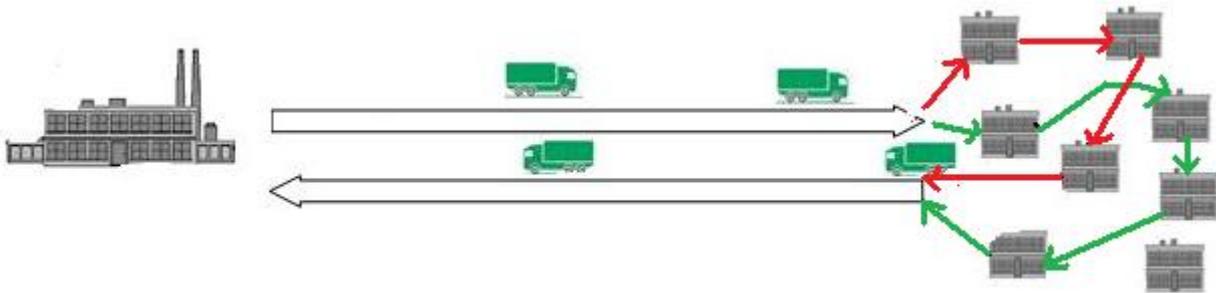


FIGURE 1 – Structure de la chaîne logistique

L'étude d'une chaîne avec une telle configuration est motivée par l'émergence des *clusters* dans les chaînes logistiques. Cette configuration est devenue très courante suite au regroupement de nombreux détaillants dans le milieu urbain créant ainsi des marchés intéressants pour des biens homogènes ou hétérogènes. Les *clusters* sont en émergence. En raison de la demande consolidée des biens et des services, les *clusters* attirent l'attention des fournisseurs pour répondre à cette grande demande centrée. Cela est bénéfique aussi bien pour les fournisseurs que pour les détaillants situés dans le *cluster*, particulièrement en termes de coût de transport (utilisation des véhicules pleins).

Il existe plusieurs exemples pour cette configuration. A titre d'exemple, pour la production et la distribution de l'eau minérale, des véhicules pleins traversent de longues distances entre l'usine de production et les épiceries d'une ville, alors que le passage d'une épicerie à une autre se fait par le parcours de distances beaucoup plus courtes, avant de retourner à l'usine de production. Une autre application est dans le transport maritime. Dans ce cas, la route d'un navire est prédéfinie, par conséquent un coût majeur fixe est connu pour chaque navire. Les coûts mineurs fixes restants sont liés à chaque port visité. Une application similaire est la livraison des détaillants sur une autoroute. Le véhicule parcourt une longue distance sur l'autoroute avant de sortir vers une voie de service pour y livrer les détaillants qui s'y trouvent.

Le problème que nous abordons dans ce travail de recherche consiste à déterminer les quantités à produire et à livrer à chaque période et l'affectation de ces quantités à livrer aux véhicules, d'une manière satisfaisant les demandes de chaque détaillant dans chaque période et à respecter les capacités de production, de stockage aux détaillants et les capacités des véhicules. L'objectif est

de minimiser le coût total de notre chaîne logistique. Ainsi la particularité de la structure considérée fait que pour évaluer le coût total, il faut, en plus du coût de production et de stockage, prendre en considération le coût, que nous supposons fixe, de déplacement de chaque véhicule du producteur vers le *cluster*, le coût de passage d'un détaillant à un autre dans le *cluster*, qui est négligeable, et enfin un coût de service qui dépend de chaque détaillant livré. Résoudre de manière optimale un tel problème intégré est en général difficile en raison de sa nature combinatoire.

Ce problème est une extension d'un problème prouvé NP-difficile, connu sous le nom *One-Warehouse-Multi-Retailer* (OWMR) qui sera détaillé plus loin. Ce qui justifie que notre problème est aussi NP-difficile. Le problème le plus proche du notre est le PRP *Production Routing Problem*. Dans le PRP, la partie production est un problème de dimensionnement de lots (*lot sizing Problem* LSP) comme dans notre cas, mais pour la partie distribution, il s'agit d'un problème de tournée de véhicule (*Vehicle Routing Problem* VRP) à chaque période. Cependant dans notre problème, la partie distribution à chaque période n'est pas un VRP. Ceci rend les méthodes proposées pour le PRP, inefficaces pour notre problème. Aussi, nous proposons dans cette thèse des approches de résolution qui exploitent les particularités de la structure considérée. Pour ce faire, nous avons organisé cette thèse de la manière suite :

Dans le premier chapitre, nous présentons des notions et des concepts de chaîne logistique, de gestion de la chaîne logistique (SCM) et les principales fonctions d'une chaîne logistique. Nous nous intéressons à l'intégration de ces fonctions en donnant ces différents mécanismes et dimensions. Puis, nous effectuons une analyse de la littérature sur les modèles d'intégration. Enfin, nous présentons l'objectif de ce travail de recherche en le positionnant par rapport à cette littérature.

Le deuxième chapitre est dédié à un état de l'art des différents problèmes qui sont en étroite relation avec le notre en montrant les relations existantes entre eux. Nous commençons d'abord, par la littérature liée au problème de production, ensuite, celle qui est étroitement liée à notre problème de distribution. Nous discutons ensuite les recherches pertinentes qui intègrent les deux problèmes (production et distribution). Nous terminons par une explication concise des différences entre notre modèle et ceux existants dans la littérature, ce qui permet de révéler les principales contributions de notre travail.

Dans le troisième chapitre, nous développons une méthode exacte pour la résolution de notre problème. Nous présentons d'abord un aperçu général des méthodes exactes proposées dans la littérature. ensuite, nous introduisons différentes formulations de programmation linéaire en nombres entiers. Après, nous développons plusieurs familles d'inégalités valides pour renforcer deux formulations. Enfin, nous effectuons une étude de calcul en comparant chacune des formulations ainsi que l'effet de l'ajout des inégalités.

Dans le quatrième chapitre, nous détaillons quatre approches à base d'algorithme génétique pour résoudre le problème. La première approche est une hybridation d'un algorithme génétique

avec une recherche locale. Les trois autres approches sont une inter-opération entre l'algorithme génétique et la programmation linéaire en nombres entiers.

Dans le dernier chapitre de ce mémoire, nous proposons trois autres approches. La première est à base de recherche Tabou, la deuxième est à base de Recuit Simulé et la troisième est une hybridation entre les deux premières.

Enfin, nous terminons cette thèse par une conclusion générale qui reprend les principaux résultats obtenus et des perspectives de recherche à envisager pour des travaux futurs.

# POSITIONNEMENT DU PROBLÈME



## RÉSUMÉ

*C*E chapitre a pour but de positionner ce travail de thèse dans le vaste domaine de recherche sur les chaînes logistiques et leur gestion. Nous débutons par des définitions des termes chaîne logistique (Supply Chain SC en Anglais), et gestion de la chaîne logistique (Supply Chain Management SCM en Anglais). Puis, nous donnons les principales fonctions d'une chaîne logistique et nous expliquons comment elles sont intégrées en donnant les différents mécanismes et dimensions de cette intégration. Cette dernière a conduit à la notion d'optimisation intégrée. Nous donnons à la fin de ce chapitre les différentes classifications des modèles d'optimisation intégrée proposés dans la littérature.



## 1.1 INTRODUCTION

Comme nous l'avons souligné dans l'introduction générale de cette thèse, l'intégration des différentes fonctions de la chaîne logistique constitue un défi actuel majeur. La problématique de recherche que nous avons exposé consiste à intégrer les décisions de production de stockage et de distribution dans un seul modèle en minimisant le coût total lié à ces fonctions. Pour répondre à cette problématique, il apparaît judicieux de positionner cette problématique par rapport à un axe de recherche très célèbre qui est la gestion de la chaîne logistique. Nous aborderons ainsi plus précisément ce que est la chaîne logistique et la gestion de la chaîne logistique. Nous nous intéressons ensuite à la notion d'intégration, sa relation étroite avec la gestion de la chaîne logistique et ses différents mécanismes et dimensions. Nous terminons ce chapitre par une synthèse des modèles d'intégration des fonctions de la chaîne logistique qui sont parus dans la littérature.

## 1.2 LOGISTIQUE ET CHAÎNE LOGISTIQUE

La notion de chaîne logistique inclut le terme "logistique" qui vient du mot grec "*logistikos*" qui signifie l'art du raisonnement et du calcul. La logistique est apparue en premier lieu dans un contexte militaire relatif essentiellement à tout ce qui est nécessaire (physiquement) à l'application sur le terrain des décisions stratégiques et tactiques. Après la seconde guerre mondiale, la logistique militaire c'est transposée vers le domaine industriel, et c'est ainsi qu'on a vu l'apparition de ce qu'on appelle logistique industrielle. Celle-ci repose plus particulièrement sur les activités de soutien à la production. Le concept de logistique a évolué depuis, avec l'évolution du marché et des systèmes industriels.

Dans les sections suivantes, nous évoquons les principaux concepts d'une chaîne logistique (ou Supply Chain, le terme anglo-saxon équivalent plus largement utilisé), en commençant par les différentes définitions qui s'y rapportent.

## 1.3 DÉFINITIONS DE LA CHAÎNE LOGISTIQUE

Il n'existe pas une définition universelle de la chaîne logistique. La littérature en propose un large panel, certaines adoptent la vision *produit* d'autres la vision *entreprise* ou encore la vision *processus* comme critère pour identifier les acteurs de la chaîne logistique.

Parmi les définitions existantes, nous en présentons trois :

Pour Christopher [35], la chaîne logistique peut être considérée comme le réseau d'entreprises qui participent, en amont et en aval, aux différents processus et activités qui créent de la valeur sous forme de produits et de services apportés au consommateur final. En d'autres termes, une chaîne logistique est composée de plusieurs entreprises, en amont (fourniture de matières et composants) et en aval (distribution), et du client final.

Selon Lee et Billington [71], la chaîne logistique est un réseau d'installations qui assure les fonctions d'approvisionnement en matières premières, de transformation de ces matières premières en composants puis en produits finis, et de distribution des produits finis vers le client.

Beamon [17] définissent la chaîne logistique comme étant un processus intégré dans lequel différentes entités (fournisseurs, producteurs, distributeurs et détaillants) travaillent ensemble dans un effort d'acquérir les matières premières, de convertir ces matières en produits finis spécifiques et de livrer ces produits aux clients .

## 1.4 LES ACTEURS DE LA CHAÎNE LOGISTIQUE

Toute organisation qui participe à l'acheminement des flux du point de départ jusqu'à sa destination dans les meilleures conditions est appelée maillon ou acteur dans la chaîne logistique. Dans n'importe quelle structure de la chaîne logistique, les organisations prennent des rôles spécifiques : fournisseurs, producteurs, prestataires, détaillants ou client final. Selon Mentzer et al. [81], une chaîne logistique peut prendre trois formes possibles :

*Une chaîne logistique directe* : dans ce cas, le réseau se limite à l'entreprise productrice, son fournisseur direct et son client final. C'est la forme minimale de la chaîne logistique.

*Une chaîne logistique étendue* : ce type de réseau inclut trois autres catégories d'acteurs à la structure minimale de la chaîne logistique : le fournisseur du fournisseur ou le fournisseur final en amont ainsi que le client du client ou le client final en aval.

*Une chaîne logistique globale* : cette forme de réseau tient compte de toutes les organisations impliquées dans la chaîne logistique. Ce type de réseau est très complexe à étudier. Cette complexité est due d'une part au nombre de maillons présents et d'une autre part à la variété des relations existantes.

## 1.5 LES STRUCTURES TYPIQUES DE LA CHAÎNE LOGISTIQUE

La structure d'une chaîne logistique dépend de sa nature et des objectifs souhaités lors de sa conception. Les structures typiques de chaînes logistiques usuelles rencontrées dans la littérature sont décomposées par Huang et al. [61] en : Série; Divergente; Dyadique; Convergente et Réseau. Elles sont présentées sur la figure 1.1.

*Structure en série* : La structure en série correspond à un procédé de fabrication linéaire où chaque entité de la chaîne alimente une seule autre entité en aval.

*Structure divergente* : Une structure est dite divergente si un fournisseur alimente plusieurs clients. Elle permet de modéliser un réseau de distribution.

*Structure convergente* : Une structure est dite divergente si un client est alimenté par plusieurs fournisseurs. Elle représente aussi un processus d'assemblage.

*Structure réseau* : La structure réseau est la composition d'une structure convergente et divergente permettant de prendre en compte des chaînes logistiques plus complexes.

*Structure dyadique* : La structure dyadique peut être vue comme un cas particulier d'une chaîne en série limitée à deux étages. Elle peut servir de base à l'étude de relations client/fournisseur ou donneur d'ordre/sous-traitant.

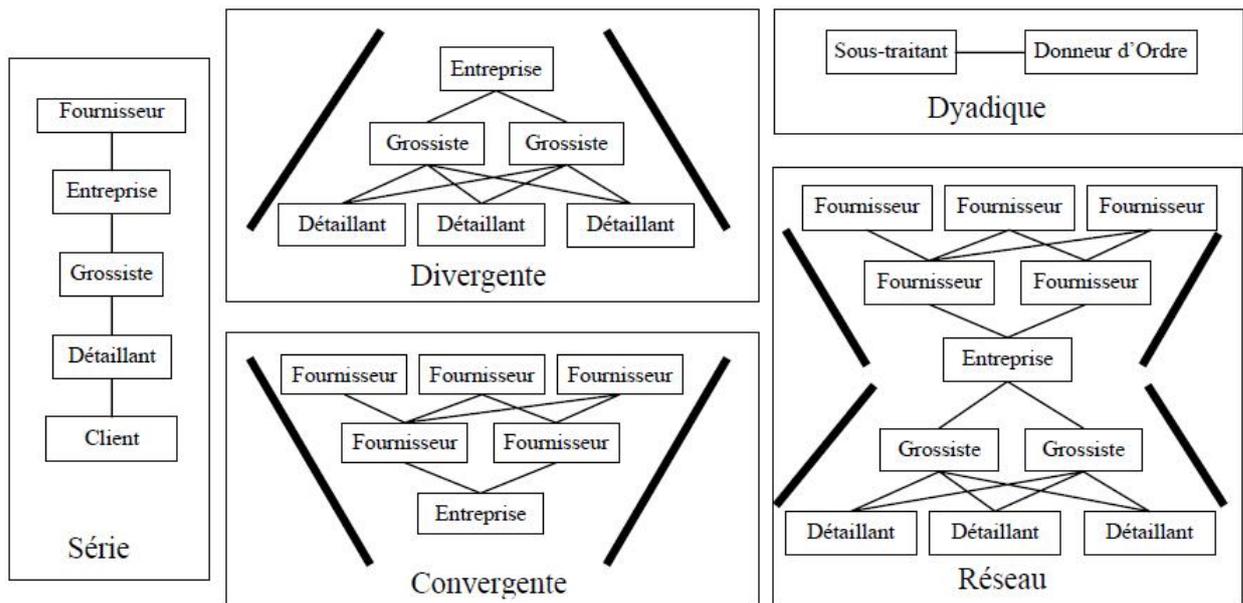


FIGURE 1.1 – Types des structures de la chaîne logistique

## 1.6 LES FONCTIONS DE LA CHAÎNE LOGISTIQUE

Les chaînes logistiques sont généralement complexes et caractérisées par de nombreuses activités réparties sur plusieurs fonctions qui vont de l'achat des matières premières à la vente des produits finis en passant par la production, le stockage et la distribution. Chacune de ces fonctions joue un rôle majeur dans la performance globale de la chaîne logistique. Aussi nous présentons dans ce qui suit les principales fonctions qui peuvent constituer une chaîne logistique.

### 1.6.1 Approvisionnement

La fonction approvisionnement consiste à fournir tout ce qui est nécessaire à l'entreprise. Deux grandes phases sont ici à distinguer. La première phase consiste à sélectionner les fournisseurs de l'entreprise. Le choix des fournisseurs peut se faire sur différents critères comme la qualité, le prix, les délais de réapprovisionnement des matières premières ou composants, mais aussi leur capacité de production, leur facilité à accepter une demande très variable, leur possibilité de

faire évoluer techniquement les composants. Il est possible de sélectionner un fournisseur unique par produit ou, au contraire, des sources multiples qui se partagent la demande, en minimisant ainsi le risque de rupture de livraison. Les fournisseurs étant déterminés, la seconde phase de la fonction approvisionnement consiste à passer les commandes des composants à ces fournisseurs en fonction de la production à réaliser. Il s'agit aussi de vérifier que ces composants sont livrés dans de bonnes conditions, c'est-à-dire de vérifier que la livraison comporte les bons composants, de qualité requise, en quantité conforme et au bon moment. La fonction approvisionnement regroupe ainsi toutes les relations avec les fournisseurs pour assurer les niveaux de stocks en composants nécessaires et suffisants

### 1.6.2 Production

La fonction de production est au cœur de la chaîne logistique, elle englobe l'ensemble des activités qui transforment des matières premières et composants en produits finis. Les décisions de production comprennent l'allocation des capacités de production, le dimensionnement de lots des différents produits à fabriquer, les périodes auxquelles ces lots doivent être produits, et les machines sur lesquelles la production doit avoir lieu. Elles visent en même temps une meilleure qualité, le plus court délai et le moindre coût et faire preuve de plus de flexibilité et de réactivité. L'entreprise doit gérer les compétences quelle détient pour fabriquer, développer ou transformer les matières premières en produits ou services.

### 1.6.3 Stockage

Les produits sont stockés à tous les niveaux de la chaîne logistique et on peut dire que pratiquement chaque entité faisant partie de la chaîne logistique est responsable de l'entreposage des produits. On trouve les stocks de matières premières, les stock des composants, les stocks des en-cours et finalement le stock des produits finis. Les stocks sont donc partagés entre les différents acteurs : les fournisseurs, les producteurs et les distributeurs. Le stockage garantit l'intégrité physique et la sécurité des produits, et de leur conditionnement dans les différents entrepôts et magasins jusqu'à ce qu'ils soient distribués aux clients. Il est évident que plus on a de stocks, plus la chaîne logistique est réactive aux fluctuations des demandes sur le marché. Cependant, avoir des stocks engendre des coûts et des risques surtout dans le cas de produits périssables ou bien de produit qui risque d'être obsolète suite à la sortie sur le marché de nouveau produit de la même gamme. La gestion des stocks est l'une des clés de la réussite et l'optimisation de toute chaîne logistique. En outre, avec l'avènement des techniques de management dite de juste à temps ou (*Just In Time JIT*), beaucoup d'entreprises tendent à avoir un stock quasi nul, autrement dit, ce qu'il faut pour produire et satisfaire les commandes. Mais ceci n'est évidemment pas sans risques. Donc la question qui se pose est de trouver un équilibre entre une meilleure réactivité et la réduction des coûts.

### 1.6.4 Distribution et transport

La fonction transport intervient tout au long de la chaîne, le transport des matières premières, le transport des composants entre les usines, le transport des composants vers les centres

d'entreposage ou vers les centres de distribution, ainsi que la livraison des produits finis aux clients. La performance de la chaîne logistique est liée aussi au choix des modes de transport. Les modes de transport aériens sont les plus rapides et les plus coûteux, mais permettent de réagir très vite et ainsi de satisfaire les demandes non prévisibles. Les modes de transport terrestres sont plus efficaces d'un point de vue coûts engendrés mais moins rapides. L'ensemble des partenaires peuvent choisir de combiner ces modes de transport et de les adapter à certaines situations selon l'importance de la demande et le gain total engendré.

Les problèmes liés à la distribution et au transport peuvent être vus sous plusieurs angles. On peut chercher à trouver les meilleures routes possibles pour visiter les points de collecte et/ou de distribution ( problèmes de tournées des véhicules ou *Vehicle routing problems*), ou bien, comme dit plus haut, chercher les meilleurs modes de transports, ou bien les quantités des produits qui doivent être transportées aux clients tout en minimisant le coût global des transports et des stocks. Les entreprises doivent aussi choisir entre différentes stratégies, par exemple, la possession d'une propre flotte de véhicules ou opter pour un tiers prestataire logistique, faisant expéditions avec véhicule complet (*Truckload* ou TL en anglais) ou chargement partiel(*less than truckload* ou LTL en anglais) et ainsi de suite. En effet, des études montrent que les coûts de transport et distributions constituent une part importante des coûts opérationnels globaux d'une chaîne logistique, et affectent aussi la satisfaction des clients, ce qui rend leur optimisation un défi majeur pour les entreprises.

### 1.6.5 Vente

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

## 1.7 GESTION DE LA CHAÎNE LOGISTIQUE

Le terme gestion de la chaîne logistique ou *Supply Chain Management* (SCM) est apparu durant les années 80. Cette période est caractérisée par une très forte internationalisation des entreprises, et une concentration de leurs activités. Ceci a conduit à la fragmentation des processus et à la spécialisation des acteurs, par conséquent à une globalisation des échanges. Ces échanges ont entraîné une augmentation de la complexité des flux dans la chaîne logistique. Ils rendent plus difficile la coordination des acteurs de la chaîne logistique. Aussi pour contourner le problème de coordination, de nouveaux défis sont apparus pour intégrer des entreprises indépendantes et coordonner les flux des matières, d'informations et financiers. Depuis, le concept SCM a trouvé sa place aussi bien au niveau académique que professionnel. A l'instar du concept de la chaîne logistique, le concept du SCM a lui aussi donné lieu à plusieurs définitions.

Simchi-Levi et al. [105] définissent le SCM comme un ensemble d'approches utilisées pour intégrer les fournisseurs, les producteurs, les dépôts et les centres de distribution, pour que le produit fini soit fabriqué et distribué avec la qualité requise, dans les délais exigés, de façon à minimiser le coût total en satisfaisant le niveau de service demandé.

Selon Thomas et Griffin [110], le SCM est la gestion des flux de matière et des flux d'informations à la fois à l'intérieur et entre les entités de la chaîne logistique tels que les fournisseurs, les centres de fabrication et d'assemblage et les sites de distribution.

Stadtler et Kilger [107], définissent le SCM comme étant la tâche d'intégration des différentes organisations qui composent la chaîne logistique, et de la coordination des flux de matières, d'informations et financiers afin de satisfaire la demande des clients finaux et d'améliorer la compétitivité de la chaîne logistique globale.

Les différents aspects du SCM présentés dans la définition de Stadtler et Kilger [107], sont illustrés par ce qu'ils appellent la maison de SCM, présentée en figure 1.2. Le toit de cette maison correspond aux objectifs du SCM en termes de réponse aux besoins des clients et de compétitivité de la chaîne logistique. Ces objectifs reposent sur deux piliers qui sont, d'une part l'intégration du réseau formé par les différents partenaires de la chaîne et, d'autre part, la coordination des différents acteurs du réseau. Le pilier de l'intégration concerne la création de la chaîne logistique et des partenariats entre les différents acteurs. Le pilier coordination concerne la gestion des trois flux de matière, d'information et financier. Les fondations de la maison reprennent globalement les différents aspects de la gestion industrielle qui vont favoriser le développement du SCM [51].

## 1.8 NIVEAUX DE DÉCISION DANS LA GESTION DE LA CHAÎNE LOGISTIQUE

Les décisions prises dans la gestion de la chaîne logistique sont généralement structurées en trois niveaux hiérarchiques : le niveau stratégique, le niveau tactique et le niveau opérationnel. Chacun de ces niveaux est lié à un aspect temporel différent.

### 1.8.1 Le niveau stratégique

L'effet des décisions stratégiques est à long terme. Elles concernent la conception de la structure de la chaîne : le choix des fournisseurs, le choix des sites de production ou des sites de stockage (capacité, dimensions), le choix du mode de transport, la conception de nouveaux produits, l'extension à de nouveaux marchés ou de nouvelles branches d'activité. Elles concernent aussi la détermination des modalités de relations entre les partenaires de la chaîne (contrats, partage de gain, partage d'information).

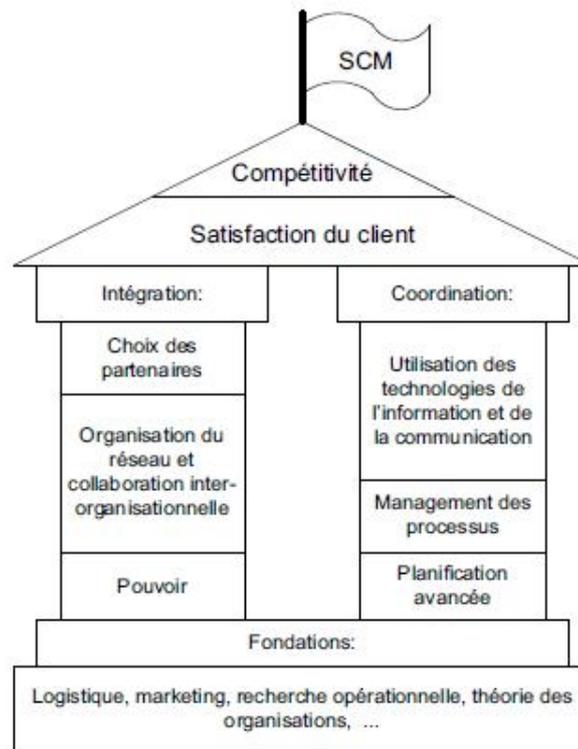


FIGURE 1.2 – *Maison du Supply Chain Management. Source : Stadler et Kilger [107]*

### 1.8.2 Le niveau tactique

Il s'agit généralement des décisions à moyen terme (de quelques semaines à quelques mois). Elles portent sur les problèmes liés à la planification des ressources de l'entreprise. Ces décisions concernent, par exemple, l'allocation des sources d'approvisionnement aux usines, le dimensionnement de lots de production, la gestion des stocks, le choix des transporteurs.

### 1.8.3 Le niveau opérationnel

Le niveau opérationnel concerne les décisions à court terme qui affectent la façon dont les produits se déplacent le long de la chaîne logistique et assure son fonctionnement quotidien. Ces décisions doivent prendre en compte les décisions stratégiques et tactiques existantes. Elles concernent par exemple l'optimisation des flux, l'ordonnancement et le pilotage des systèmes de production, la programmation des tournées de véhicules, etc... .

## 1.9 LA NOTION D'INTÉGRATION DANS LA CHAÎNE LOGISTIQUE

L'un des piliers de la maison du SCM est l'intégration (figure 1.2). Elle est probablement l'un des mots les plus fréquents du vocabulaire ou le maître mot de la logistique et de la gestion de la chaîne logistique. L'intégration des chaînes logistiques apparaît comme un idéal vers lequel de nombreuses initiatives tendent. En effet, il est considéré comme acquis qu'intégrer une chaîne logistique contribue à améliorer sa performance et, par conséquent, la performance des entreprises

qui y participent. Mais, quoi et comment intégrer ?. Telles sont les questions auxquelles répondent les deux sections suivantes :

### 1.9.1 Les mécanismes de l'intégration de la chaîne logistique

Pour répondre aux besoins d'intégration, des nouvelles relations de partenariats entre fournisseurs et clients sont à définir. Ces relations doivent être durables, et permettre aux entreprises de s'intégrer et coordonner la prise de décisions au niveau global de la chaîne, ceci dans le souci d'être réactif et efficace face aux mouvements imprévisibles des marchés. Différents mécanismes de coordination ont été proposés pour intégrer les entités d'une chaîne logistique. Parmi ces mécanismes figurent essentiellement les types suivants :

#### *Collaborative Planning, Forecasting and Replenishment (CPFR)*

Le CPFR a été créé aux Etats-Unis en 1996 grâce aux efforts conjugués d'une quarantaine d'entreprises réunies au sein du sous-comité CPFR du VICS (*Voluntary Interindustry Commerce Standards association*). Son principe consiste en un système de pilotage collaboratif Client/Fournisseur qui permet d'élaborer les prévisions des ventes, les plannings de réapprovisionnement, de production et de distribution pour assurer un rechargement optimal des stocks. En CPFR, les fonctions achats, marketing, ventes et service client prennent part à la collaboration client/fournisseur. Il nécessite une bonne organisation des ressources et une confiance mutuelle entre les partenaires. Le partage d'informations concerne les commandes, les niveaux de stocks, les ventes, les prévisions, les promotions ainsi que les stratégies commerciales.

Le CPFR a pour objectif d'arriver à la réduction des stocks, l'augmentation du taux de service client, l'augmentation des ventes, la diminution du temps de réponse à la demande, l'amélioration du temps de cycle, la réduction des besoins en capacité de production, l'amélioration de la précision des prévisions et la réduction des coûts [95].

#### *Quick Replenishment (QR)*

En 1984, le Quick Replenishment (QR) a été créé par le "*Crafted With Pride in the USA Council*", qui est un conseil d'entreprises. Ce conseil était formé de leaders de l'industrie textile. La mise en place du QR a permis d'améliorer les temps de réponse à la demande, les délais de réapprovisionnement, les stocks de sécurité et la rotation des stocks [95].

#### *Continuous Replenishment Policy (CRP)*

La politique du rechargement continu (*Continuous Replenishment Policy* CRP) a été développée dans les années 1990. Dans cette politique, le fournisseur ne se base plus sur l'historique de ventes du donneur d'ordre, mais sur sa consommation en temps réel ainsi que ses niveaux de stocks. C'est le passage d'une politique à flux poussés vers une autre à flux tirés basée sur la demande du client [95].

### *Vendor Managed Inventory (VMI)*

Le mécanisme de coordination *Vendor Managed Inventory (VMI)* consiste à confier au fournisseur la responsabilité de gérer les stocks de ces clients, en fonction des informations qu'il reçoit, de manière à faciliter le réapprovisionnement.

En gestion traditionnelle, le manque de coordination force les clients à commander par grandes quantités de manière à optimiser le remplissage des véhicules. Cependant en VMI, le fournisseur reçoit des informations sur la demande plus régulièrement et c'est lui qui décide quand et combien livrer à chacun de ses clients. Ceci permet une meilleure planification de la production et de la distribution. Ainsi, le VMI permet une intégration entre les partenaires et les livraisons de plusieurs clients peuvent être groupées avec de meilleurs remplissages des véhicules. Le VMI est caractérisé par de courtes périodes de réapprovisionnements, et une quantité optimale de réapprovisionnement plus petite [95].

Le *Supply Chain Council* a défini le VMI comme suit : "un concept pour la planification et le contrôle des stocks, dans lequel le fournisseur accède aux informations du stock du client, et il est responsable de maintenir le niveau du stock demandé par le client. Le réapprovisionnement est effectué par le fournisseur grâce à des suivis réguliers de l'état du stock".

Le VMI est largement utilisé dans diverses industries. Ses avantages nous ont motivé à étudier dans ce travail une chaîne logistique composée d'un fournisseur (producteur) et un ensemble de clients (détaillants ou donneurs d'ordre) qui utilisent le VMI comme mécanisme de coordination.

#### **1.9.2 Les dimensions de l'intégration dans la chaîne logistique**

L'intégration dans une chaîne logistique peut être réalisée selon différents aspects. Pour Shapiro [100], il y a quatre dimensions de l'intégration :

1. L'intégration fonctionnelle : Elle concerne les différentes fonctions exercées par une organisation, par exemple, l'achat, la production, l'entreposage et le transport.

2. L'intégration spatiale : Ce type d'intégration se fait par un groupe cible d'entités de la chaîne : les fournisseurs, les usines de production, les entrepôts et les marchés.

3. L'intégration hiérarchique : Elle est relative à l'intégration des décisions qui se chevauchent dans les horizons de planification stratégiques, tactiques et opérationnels. Par exemple, il est important de localiser stratégiquement les usines de production à l'égard des fournisseurs pour les aider à optimiser leurs politiques d'approvisionnement.

4. L'intégration entreprise : Elle souligne l'importance des décisions de planification stratégiques et tactiques comme, l'intégration de la gestion de la chaîne logistique avec la gestion de la demande et la gestion financière, afin de maximiser les revenus et d'augmenter les rendements

sur les investissements à long terme.

## 1.10 PLANIFICATION DE LA CHAÎNE LOGISTIQUE OU *Supply Chain Planning SCP*

Sous l'approche d'intégration des différentes fonctions de la chaîne logistique, un des blocs de construction de la maison du SCM (1.2) est le bloc planification avancée ou *Advanced planning* dont l'objectif est l'élaboration d'un plan intégré de toute la chaîne logistique. Un tel plan comprend les quantités appropriées concernant, les matières premières à approvisionner, les produits à fabriquer dans les usines, les produits à distribuer et les produits à vendre aux clients. Rohde et al. [94] proposent une matrice appelée *SCP-Matrix*, permettant d'assurer une planification avancée. Cette matrice donne une classification des décisions de planification en deux dimensions : horizon de planification et fonction de la chaîne logistique (voir la figure 1.3). Le support informatique du SCP est appelé *Advanced Planning and Scheduling (APS) systems*, qui est un ensemble de modules de logiciels prédéfinis conformément à la Matrice *SCP*. Les noms des modules changent d'un APS à un autre selon les vendeurs, mais les caractéristiques suivantes doivent être incluses dans tout APS : 1) l'outil doit assurer une planification à long terme, à moyen terme et à court terme, 2) il doit permettre de générer et d'évaluer différents scénarios, 3) il doit tenir compte des différentes contraintes simultanément 4) il doit utiliser des méthodes mathématiques très sophistiquées pour résoudre des problèmes d'optimisation complexes [52].

## 1.11 OPTIMISATION DES FONCTIONS DE LA CHAÎNE LOGISTIQUE

### 1.11.1 Optimisation séquentielle

Les fonctions de la chaîne logistique ont été gérées indépendamment les unes des autres pendant de longues années. Dans cette vision traditionnelle chaque fonction est une entité indépendante, et est responsable de l'optimisation de ses propres activités, autrement dit chaque fonction élabore sa propre planification séparément. Un exemple peut être donné comme suit. Une fois que la production est planifiée, le planning associé sert pour calculer le niveau de stockage des matières premières et des produits finis. Puis un planning de transport est préparé en fonction de ces quantités produites et stockées sans pouvoir modifier les plannings de production. On parle ainsi d'une optimisation séquentielle [6]. Ce mode d'optimisation présente un certain nombre d'inconvénients, nous énumérons quelques uns dans ce qui suit :

*1- Coûts élevés* : En raison de l'absence de coordination entre les fonctions, il y aura des ventes perdues en raison de ruptures de stock, ou des coûts de possession élevés dus aux stocks excédentaires. Cela se produit lorsque la demande et la production ne sont pas synchronisées et la fabrication se fait uniquement sur de mauvaises estimations.

*2- Manque de partage des données* : Bien que les données nécessaires sont présents dans toute

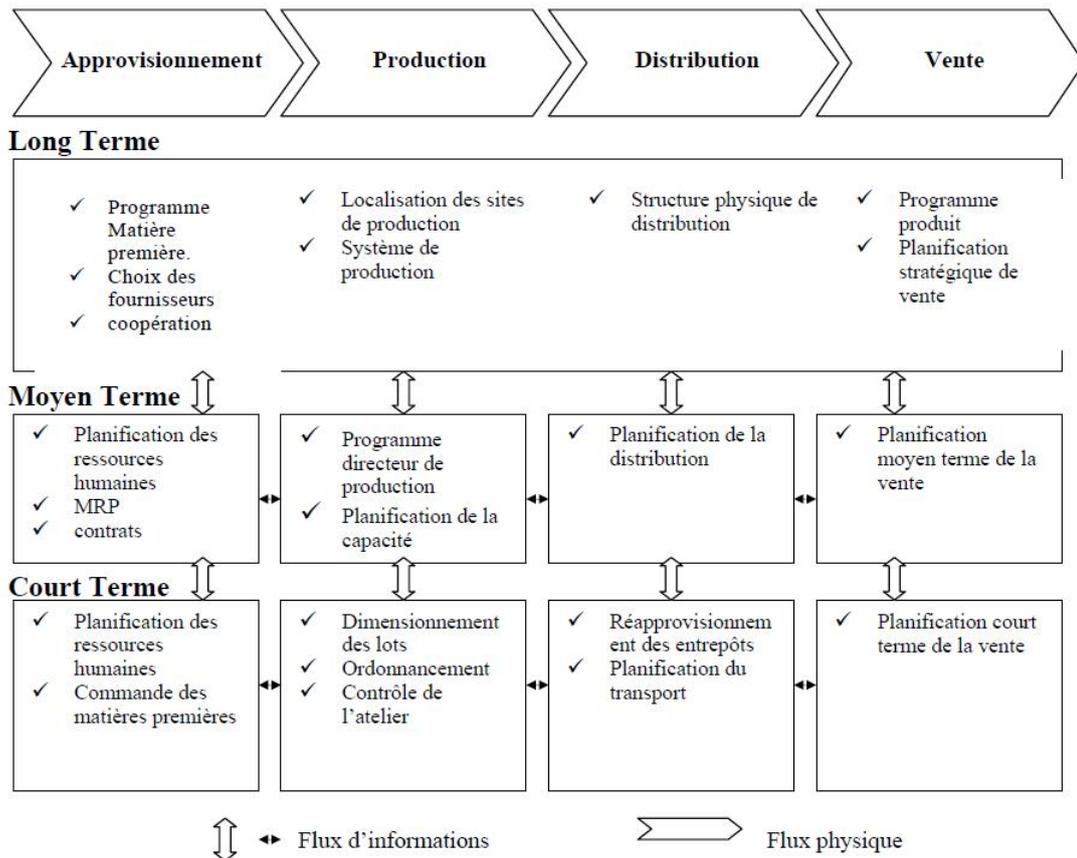


FIGURE 1.3 – Matrice de planification de la chaîne logistique (SCP-Matrix). Source : Stadler et Kilger [107]

la chaîne en divers points, elles ne sont pas utilisées aux bons endroits et aux bons moments, créant ainsi une illusion du manque de données. Cela conduit à des hypothèses non raisonnables, et à des mauvaises stratégies.

3- *Conflicts internes* : Différentes fonctions au sein de la chaîne d'une organisation peuvent se retrouver en conflit sur certaines ressources au même temps créant ainsi une concurrence interne non nécessaire.

4- *Duplication d'efforts* : L'absence de coordination peut également conduire à l'exécution de tâches similaires dans plusieurs points le long de la chaîne. Cela encourt des coûts supplémentaires. Par exemple, le maintien et la mise à jour des bases de données concernant les stocks des produits finis fait de façon indépendante dans les lieux de production et les lieux de distribution mènera forcément à la duplication. Par contre, on évitera cela si cette mise à jour est faite une seule fois en temps réel par l'utilisation d'une base de données uniques.

5- *Manque de vision* : La concentration sur la résolution des problèmes individuels de chaque fonction à court terme, peut donner une solution rapide à un problème temporaire, mais cela ne donne pas un avantage à une vision stratégique. Ses conséquences seront évidentes à long terme.

La tendance vers la fabrication en *JIT* et *Lean Production*, et par conséquent vers l'intégration des différentes fonctions a conduit à la notion d'optimisation intégrée.

### 1.11.2 Optimisation intégrée

La première raison qu'on peut donner dans ce sens est qu'un optimum global est rarement obtenu en rassemblant des optima locaux. Avec cette approche, les contraintes et les coûts des fonctions qui sont coordonnées ou intégrées sont mis dans le même modèle pour être optimisés en une seule fois.

Une autre raison pour mener une approche intégrée est l'augmentation des coûts de transport et de stockage vis-à-vis du coût de production qui est due à la délocalisation des entreprises vers les pays à bas coûts de main d'œuvre, ou pour des raisons écologiques et fiscales. Cette délocalisation a engendré l'éloignement des centres de production de ceux de consommation. Ceci incite donc les entreprises à donner plus d'importance à la prise en compte des coûts et des contraintes de toutes les fonction simultanément, afin de minimiser le coût total engendré. Le concept même de *supply chain* a émergé avec ces besoins [6].

Un facteur important qui facilite l'intégration est le développement des technologies de l'information et de la communication tel que l'EDI (*Electronic Data Interchange*), l'Internet, la technologie RFID (*Radio Frequency Identification*), les ERP (*Enterprise Resource Planning*) et les APS (*Advanced Planning Systems*). Ces outils permettent d'avoir un échange d'informations et une visibilité détaillée des différentes entités de la chaîne pour prendre une décision globale.

En revanche, pour mettre en œuvre une telle approche en situation réelle, plusieurs difficultés peuvent être rencontrées telles que :

- *La difficulté liée au partage du gain total obtenu.* Les différentes forces de pouvoir entrent ainsi dans le jeu, ce qui empêche son utilisation dans les chaînes où différents acteurs et centres d'intérêt interviennent dans la gestion des différents maillons. Dans le cas d'un grand producteur où le transport et le stockage des composants et des produits finis sont souvent sous-traités. L'acteur le plus puissant impose ses contraintes, ce qui pousse les autres acteurs à prendre en compte des contraintes en dehors de leurs propres activités. Ceci crée une gestion intégrée imposée [6].

- *La difficulté du partage de l'information.* Ce partage de l'information nécessaire pour optimiser le fonctionnement de la globalité de la chaîne peut être difficile malgré la puissance des outils informatiques pour l'échange de l'information. Les différents centres d'intérêt dans la chaîne font que cet échange est biaisé par des acteurs qui auront tendance à ne pas partager les informations cruciales même si ce partage mène à une réduction significative du coût total.

- *La difficulté de résolution des modèles mathématiques.* Malgré la puissance des logiciels

commerciaux, cette intégration conduit à considérer des systèmes complexes, de grande taille et difficiles à résoudre en un temps raisonnable. Cette complexification est due à l'augmentation des contraintes à prendre en compte lors d'une modélisation mathématique et aussi à la configuration de la fonction objectif qui devient plus compliquée à optimiser. Cette difficulté fait qu'il y a de plus en plus de travaux de recherches menés sur ce problème, en grande partie par des spécialistes en recherche opérationnelle [6].

Notre travail se situe dans le cadre de cette dernière difficulté. Nous nous intéressons au développement des méthodes de résolution efficaces pour résoudre un problème d'optimisation intégrée des fonctions production, stockage et distribution dans une chaîne logistique particulière que nous allons décrire par la suite.

## 1.12 LES MODÈLES D'INTÉGRATION DES FONCTIONS DE LA CHAÎNE LOGISTIQUE

L'intégration des fonctions de la chaîne logistique peut se faire sur le plan stratégique, tactique ou opérationnel, où chaque niveau a un ensemble de besoins et d'objectifs différents (Erengüç et al. [42]). Vu l'intérêt croissant dû à l'importance pratique de l'intégration et de ses effets potentiels sur la performance de la chaîne logistique, de nombreux modèles dans la littérature impliquant l'intégration des fonctions de la chaîne logistique sont apparus. Ils consistent à intégrer les activités de deux fonctions ou plus, afin de minimiser à titre d'exemple le coût total engendré par ces fonctions. Nous présentons dans ce qui suit les différentes classifications des modèles d'intégration.

Bhatnagar et al. [20] propose deux niveaux d'intégration : l'intégration entre les différentes fonctions comme la production, la distribution, le stockage et le marketing, qu'ils appellent problème d'intégration générale; et l'intégration au sein d'une même fonction des différents échelons de l'organisation, appelée problème d'intégration multi-usines.

Thomas et Griffin [110] classifient et répertorient les différents modèles d'intégration en trois catégories : la coordination entre le fournisseur et le client, la production et la distribution, le stockage et la distribution.

Beamon [17] donne une classification des modèles d'intégration dans une chaîne logistique multi-échelons en quatre catégories : Les modèles analytiques déterministes, les modèles analytiques stochastiques, les modèles économiques et les modèles de simulation.

Erengüç et al. [42] identifient les décisions pertinentes à prendre dans l'optimisation intégrée dans une chaîne logistique constitué de trois grandes fonctions( approvisionnement, production et distribution).

Sarmiento et Nagi [99] ont donné une classification des travaux de recherche en fonction des différents niveaux d'intégration : La distribution et le stockage ; le stockage, la distribution et le stockage ; la production, le stockage, la distribution et le stockage. Ils les différencient aussi selon le mode de transport choisis, le nombre de clients, le mode de livraison (tournées des véhicules ou direct), les types de modèles mathématiques construits (stochastique ou déterministe, horizon fini ou infini).

Chen [33] a donné un état de l'art des modèles qui intègrent explicitement les fonctions production et distribution. Il a classifié les divers problèmes existants selon trois dimensions : (A) niveau de décision (A1 : tactique, A2 : opérationnel), (B) structure d'intégration( B1 : intégration production-transport sortant, B2 : intégration transport entrant-production, B3 : intégration transport entrant-production-transport sortant) et (C) les paramètres du problème (C1 : Une seule période, C2 : horizon infini avec une demande constante, C3 : horizon fini mais avec plusieurs périodes et demande dynamique). Les cinq classes des modèles sont décrits comme suit :

Classe 1 : Problèmes de production-transport. - A1, B1, C1

Classe 2 : Problèmes d'intégration de dimensionnement de lots avec la livraison des produits finis - A1, B1, C2

Classe 3 : Problèmes d'intégration de l'approvisionnement des matières premières avec le dimensionnement de lots - A1, B2, C2

Classe 4 : Problèmes d'intégration tactique production-distribution - A1, B1 ou B3, C1 ou C3

Classe 5 : Problèmes d'intégration traitement-livraison des commandes - A2, B1, C3.

Chen [34] a donné un état de l'art des modèles d'intégration des décisions d'ordonnancement production-distribution au niveau opérationnelle. Basé sur le mode de livraison, il a classé les modèles en cinq classes :

1- les modèles avec livraison individuelle et immédiate. Dans ces modèles, chaque commande doit être livrée individuellement et immédiatement après son achèvement.

2- les modèles avec une livraison de lots à un seul client par mode d'expédition directe. Les commandes proviennent d'un seul client peuvent être groupées dans des lots de livraison.

3- les modèles avec livraison de lots à plusieurs clients par mode d'expédition directe. Cette classe de modèles est similaire à la deuxième, avec une différence, qu'il y a plusieurs clients, et uniquement les commandes d'un même client peuvent être groupées dans un lot.

4- les modèles avec livraison de commandes pour plusieurs clients avec routage. Dans ce cas, il y a plusieurs clients et les commandes destinées à des clients différents peuvent être groupées dans une tournée.

5- modèles avec des dates de départ pour la livraison fixes.

Fahimnia et al. [43] classifient les modèles d'intégration production distribution, en fonction de leurs degrés de complexité, en sept catégories :

1. Modèles mono-produit ;
2. Modèles multi-produit, mono-usine ;
3. Modèles multi-produit, multi-usine, mono ou sans dépôt ;
4. Modèles multi-produit, multi-usine, multi-dépôt, mono ou sans client ;
- Modèles multi-produit, multi-usine, multi-dépôt, multi-client, mono-chemin de transport ;
6. Modèles multi-produit, multi-usine, multi-dépôt, multi-client, multi-chemin de transport, sans de période de temps ;
7. Modèles multi-produit, multi-usine, multi-dépôt, multi-client, multi-chemin de transport, multi-période de temps ;

Fahimnia et al. [43] a donné aussi une classification des modèles en fonction des approches de résolution suivantes : (i) techniques mathématiques (modèles de programmation linéaire, modèles de programmation mixte en nombres entiers, et les modèles de relaxation de Lagrange), (ii) des techniques heuristiques, (iii) Les modèles de simulation, et (iv) les algorithmes génétiques.

## CONCLUSION

Ce chapitre nous a permis d'introduire des notions et des concepts sur la chaîne logistique et la gestion de la chaîne logistique (SCM). L'un des piliers du SCM et l'intégration et l'un des mécanismes de l'intégration est le VMI. Ce dernier est largement utilisé dans diverses relations fournisseur-donneur d'ordres. Ses avantages nous ont motivé à étudier dans ce travail une chaîne logistique composée d'un producteur(fournisseur) et un ensemble de détaillants(donneurs d'ordre) qui utilisent le VMI comme mécanisme de coordination. Notre objectif est le développement des méthodes de résolution efficaces pour résoudre un problème d'optimisation intégrée des fonctions production stockage et distribution dans cette chaîne logistique particulière que nous allons décrire par la suite.



# ÉTAT DE L'ART : PLANIFICATION DE LA PRODUCTION ET DISTRIBUTION

# 2

## RÉSUMÉ

*C*e chapitre se veut d'être un état de l'art des différents problèmes de lot sizing d'une manière générale et plus particulièrement du problème connu dans la littérature par le PRP ou Production Routing Problem. Ainsi nous présentons d'abord une définition brève du lot sizing suivi de la présentation de ces différents extensions avec à chaque fois les travaux correspondants. Nous terminons par la présentation de notre problème.



## 2.1 INTRODUCTION

L'émergence de la gestion de la chaîne logistique et de l'intégration dans cette chaîne des fonctions production et distribution a donné lieu à beaucoup de travaux de recherche.

Ce chapitre se veut être un état de l'art des différents travaux en relation avec notre problème. En effet, notre problème est une extension du problème connu sous le nom de *One-Warehouse-Multi-Retailer* (OWMR) qui est une extension du problème de dimensionnement de lots ou *lot sizing*. Cependant le OWMR est lui même un cas particulier du problème connu dans la littérature par le nom de *Production Routing Problem* (PRP) qui est à son tour un cas particulier du problème général appelé *Production-Inventory-Distribution Problem* (PIDP) (figure 2.1). Ainsi nous présentons en premier lieu la littérature liée à notre problème de production, ensuite, celle qui est étroitement liée à notre problème de distribution. Après, nous discutons des recherches pertinentes qui intègrent les deux problèmes (production et distribution). A la fin, nous expliquons les différences entre notre problème et ceux existants dans la littérature. Ces différences constituent nos contributions majeures dans ce travail.

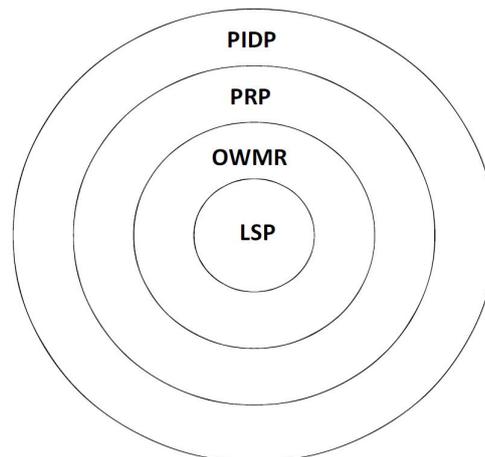


FIGURE 2.1 – Du LSP au PIDP

## 2.2 PROBLÈME DE DIMENSIONNEMENT DE LOTS OU *lot sizing problem* LSP

Le problème de dimensionnement de lots (PDL) ou *lot sizing problem* (LSP) dans la littérature anglo-saxonne consiste à déterminer la taille optimale des lots de production d'un produit, de manière à trouver un compromis entre le coût de production et le coût de stockage. Il est souvent trop coûteux de lancer la production d'un produit donné à chaque période à cause des coûts de setup. En contre partie en lançant la production moins fréquemment et en produisant de grandes quantités cela génèrera des coûts de stockage élevés.

## 2.3 CLASSIFICATION DES MODÈLES DE LOT SIZING

Plusieurs classifications des problèmes du lot sizing ont été proposées, nous pouvons citer les travaux de Drexl et Kimms [39], Staggemeier et Clark [108], Karimi et al [67], Wolsey [117] et Brahimi et al [27]. A titre d'exemple, Karimi et al [67] présentent une synthèse de la littérature pour un PDL à un niveau et ses différentes extensions. Brahimi et al [27] donnent une revue de la littérature pour le LSP à un produit avec capacité. Un schéma de classification des modèles de dimensionnement a été donné par [116] (voir Figure 2.2). Différents critères interviennent dans la classification des problèmes de dimensionnement de lots, notamment :

1- *La demande constante ou variable* : une demande est dite constante si elle est répartie uniformément sur tout l'horizon. Dans le cas contraire, on parle de demande variable. On discrétise dans ce cas l'horizon en périodes et on agrège la demande sur celles-ci. Cette distinction fondamentale a permis de définir deux approches de résolution très différentes : *les modèles à demande constante* ont pour caractéristique de fournir comme solution un cycle de production, reproductible au cours de l'horizon, *les modèles à demande variable* déterminent des quantités à produire sur des périodes préalablement définies sur l'horizon de planification.

2- *La nomenclature des produits* : La nomenclature des produits définit la relation parent-composant. S'il n'existe pas de relation parent-composant entre les produits, on parle de problèmes à un niveau, sinon on parle de problèmes multi-niveaux.

3- *Les coûts* : La notion de coût dans ce cas est relative aux coûts de production unitaires, aux coûts de setup de production qui sont liés à la préparation des ressources de production, aux coûts de stockage unitaires.

4- *Les temps de lancement ou de préparation* : C'est le temps nécessaire pour reconfigurer une ressource qui exécute différents types de produits. Ainsi il est parfois nécessaire de reconfigurer celle-ci à chaque changement de produits.

5- *Les capacités des ressources utilisées* : Si la capacité de production est illimitée ou ignorée, le problème est dit sans capacité. Dans le cas contraire, on parle d'un problème avec capacité.

6- *Objectifs* : Identifie l'objectif à atteindre par la résolution du problème. En général on cherche à minimiser l'ensemble des coûts. Mais il existe d'autres objectifs qui consistent à maximiser le niveau de service, etc.

7- *La modélisation de l'horizon de planification* : L'horizon peut être fini ou infini. De manière générale, les modèles discrets sont sur un horizon fini alors que les modèles continus sont basés sur un horizon infini. Pour l'horizon fini, la littérature distingue deux grandes classes de modèles : les modèles dits de *big bucket* et de *small bucket*. Les modèles de *big bucket* sont souvent appelés problèmes dynamiques de dimensionnement de lots ou *dynamic lot sizing problems* dont

le premier est celui de Wagner-Whitin [115]. Les modèles de *small bucket* sont appelés *Discrete Lot Sizing and Scheduling Problem (DLSP)*. Pour l'horizon infini, un des plus anciens modèles est le modèle de quantité économique de commande ou *Economic Order Quantity (EOQ)* qui a été étendu par la suite pour donner le *Economic Lot Scheduling Problem (ELSP)*.

8- *Type d'information* : Dans les modèles déterministes chaque paramètre a une valeur fixe et connue. Les modèles stochastiques contiennent des paramètres représentés par des variables aléatoires.

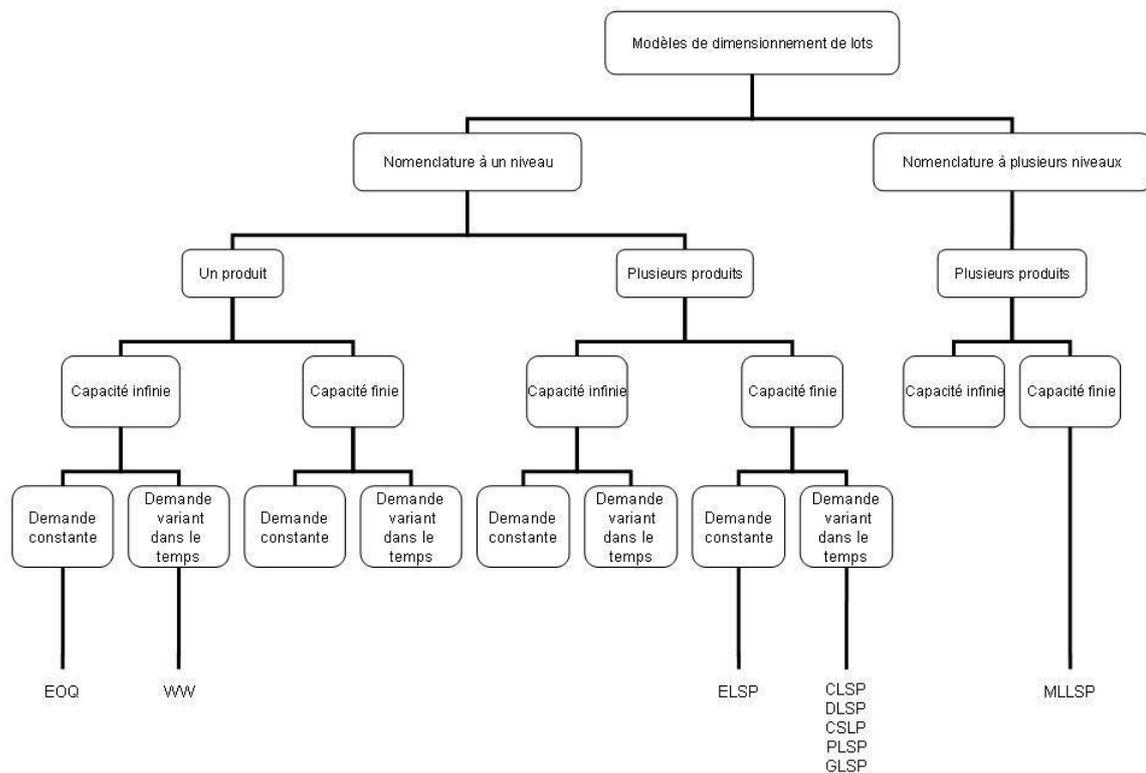


FIGURE 2.2 – Exemple de classification des modèles de lot sizing [116]

Après une brève présentation des travaux relatifs au LSP d'une manière générale, nous nous intéressons maintenant aux travaux relatifs aux problèmes de lot sizing à un seul niveau et à périodes longues *big time bucket*, autrement dit aux travaux de LSP ayant la même structure que notre cas d'étude.

## 2.4 PROBLÈME DE LOT SIZING À UN PRODUIT

Le problème du lot-sizing à un seul produit a reçu une attention particulière en raison de sa simplicité et de son importance comme étant un sous-problème de problèmes de lot-sizing plus complexes.

### 2.4.1 Problème de lot sizing à un produit sans capacité

Ce problème également connu sous l'appellation anglo-saxonne *Uncapacitated Lot Sizing Problem* (ULSP), est à l'origine formulé dans les années 50 par Wagner et Whitin [115]. Le ULSP est un problème polynomial et a été résolu par la programmation dynamique par Wagner et Whitin en  $O(T^2)$  (où  $T$  est le nombre de périodes). Cela a été suivi par les travaux d'Aggarwal et Park [5], Federgruen et Tzur [44] et Wagelmans et al. [114] qui ont proposé des algorithmes différents, mais d'une complexité  $O(T \log T)$ .

De nombreuses extensions ont été proposées dans la littérature. Zangwill [121] a étendu le modèle de Wagner et Whitin pour le cas des ventes différées (*backlogging*), en proposant un algorithme de programmation dynamique en  $O(T^2)$ . Federgruen et Tzur [45] ont proposé des algorithmes en  $O(T \log T)$ . Pour le cas des demandes perdues (*lost sales*), le problème a été résolu par Aksen et al. [7]. Brahimi et al. [28] ont étudié le problème avec fenêtres de temps, et ont proposé un algorithme polynomial en  $O(T^4)$ .

### 2.4.2 Problème de lot sizing à un produit avec capacité

Dans le cas où la capacité est limitée, on parle de problème de dimensionnement de lots avec capacité ou *Capacitated Lot Sizing Problem* (CLSP). Ce problème est fréquemment rencontré dans la pratique. Bitran et Yanasse [21] ont démontré que ce problème est NP-difficile. Mais le CLSP peut être résolu d'une manière polynomiale dans des cas particuliers. Par exemple, si la capacité est constante, le problème peut être résolu par un algorithme en  $O(T^4)$  proposé par Florian et Klein [48], celui-ci a été amélioré en  $O(T^3)$  par van Hoesel et Wagelmans [111]. Pochet [118] a proposé une formulation étendue compacte avec  $O(T^3)$  variables et contraintes. Si la capacité varie et est non décroissante avec le temps, la fonction de coût est non spéculative et le coût de setup non décroissant, le problème est aussi polynomiale. Pochet et Wolsey [85] ont proposé une reformulation compacte mixte en nombres entiers dont la relaxation linéaire résout ce problème.

Le problème avec des capacités générales est beaucoup plus difficile. Les méthodes exactes de résolution de ce problème, selon [25], sont la programmation dynamique ([31] et [101]), la séparation et évaluation ou branch & bound ([13] et [77]) et les méthodes de coupes ([74], [76] et [82]). Ces méthodes sont souvent développées pour des problèmes faciles ou pour résoudre des problèmes difficiles de petites tailles afin de générer des benchmarks pour des méthodes heuristiques. Elles ont pour objectif l'amélioration des formulations du problème et la description de l'enveloppe convexe de l'ensemble des solutions réalisables en utilisant les reformulations et les inégalités valides de sorte que les solveurs commerciaux comme CPLEX ou XPRESS-MP soient capables de résoudre des instances réelles.

## 2.5 PROBLÈME DE LOT SIZING À PLUSIEURS PRODUITS

Dans le cas où la capacité est supposée illimitée, un problème à plusieurs produits est réductible à  $N$  problèmes à un produit et sans capacité où  $N$  est le nombre de produits. Si la capacité est supposée limitée, ce problème devient très complexe, et il est largement étudié par plusieurs auteurs. Comme cité dans [73], diverses méthodes exactes et heuristiques ont été proposées pour le résoudre. On trouve :

### 2.5.1 Les méthodes exactes

On peut citer :

- des méthodes à base de Branch & Bound qui ont été développées par Gelders et al. [53] et Hindi [58],
- des méthodes à base de Branch & Cut qui ont été proposées par Leung et al. [74], Pochet et Wolsey [84], Barany et al. [14] et Belvaux et Wolsey [18],
- des méthodes à base de reformulation du modèle en plus court chemin : Eppen et Martin [40] utilisent une méthode à base de Branch & Bound pour résoudre ce problème.

### 2.5.2 Les méthodes heuristiques

Des heuristiques pour ce problème ont été développées par beaucoup d'auteurs. Maes et Van Wassenhove [78] ont proposé une classification des heuristiques en deux catégories : les heuristiques de bon sens et les heuristiques à base de programmation mathématique.

Les heuristiques de bon sens sont généralement des algorithmes gloutons. Elles abordent le problème, soit période par période, soit de façon itérative, soit produit par produit.

Les heuristiques à base de programmation mathématique ont été classées par Salomon [97] en trois catégories :

- *les heuristiques à base de relaxation* : elles s'appuient sur la relaxation des contraintes de capacités : on les dualise dans la fonction objectif (relaxation lagrangienne). C'est la méthode qu'utilise l'heuristique de Thizy et Van Wassenhove [109]. Le problème initial se décompose alors en plusieurs problèmes sans capacité avec un seul produit qu'on peut résoudre avec l'algorithme de Wagner et Whitin [115].

- *les heuristiques à base de programmation linéaire* : Ces heuristiques sont basées sur des formulations alternatives du problème (formulation sous forme de problème de plus court che-

min) dont les relaxations linéaires fournissent de meilleures bornes ou présentent une meilleure structure. Si les solutions obtenues ne sont pas réalisables, on les perturbe de différentes manières pour trouver des solutions réalisables (Gilbert et Madan [54]).

- *les heuristiques à base de génération de colonnes* : Les heuristiques à base de génération de colonnes sont basées sur le modèle approximatif de Manne (Chen et Thizy [32] et Huisman et al. [62]).

### 2.5.3 Les méta-heuristiques

Les méta-heuristiques ont été utilisées plus récemment pour résoudre ce problème :

- Salomon et al. [98] ont utilisé la recherche Tabou et le recuit simulé en codant leur solution grâce à un voisinage basé sur une matrice de setup (binaire) et en utilisant un algorithme glouton pour lisser la production sur le planning déterminé.

- Haase et Kohlmorgen [57] ont utilisé les algorithmes génétiques, ils génèrent leur population initiale grâce à une version modifiée de l'algorithme de Dixon-Silver [38] en introduisant un caractère aléatoire au choix des produits à planifier pour une période.

- Hindi [59] propose une recherche tabou pour résoudre ce problème.

Pour plus de détail, voir Jans et Degraeve [63] qui ont proposé un état de l'art sur l'application des méta-heuristiques aux problèmes de lot sizing.

## 2.6 PROBLÈME DE LOT SIZING COORDONNÉ SANS CAPACITÉ (*Coordinated uncapacitated Lot-Size Problem* CULSP)

Le CULSP est l'une des extensions du problème de lot sizing à plusieurs produits. L'objectif du CULSP est de minimiser un coût total qui comprend un coût de setup majeur engendré dans chaque période si un élément de la famille de produits est fabriqué/réapprovisionné, un coût de setup mineur dépend de chaque produit fabriqué/réapprovisionné à chaque période et les coûts de stockage. Ce problème est appelé dans certaines littératures *Dynamic Joint Replenishment Problem* (DJRP). La quantité commandée de chaque produit n'est pas limitée. Le DJRP est NP-complet (Arkin et al. [10]). Les méthodes exactes pour la résolution du DJRP peuvent être classées en quatre catégories selon Boctor et al [22] : La programmation dynamique (Zangwill [120], Veinott [113], Kao [66] et Silver [103]), branch & bound (Erenguc [41], Federgruen et Tzur [46], Kirca [68] et Robinson et Gao [91]), branch and cut (Raghavan et Rao [89], Raghavan [88]), Dantzig-Wolfe decomposition (Raghavan et Rao [90]). De plus plusieurs heuristiques ont été proposées pour la résolution du DJRP (Fogarty et Barringer [49],

Silver et Kelle [104] et Atkins et Iyogun [12]). Boctor et al. [22] ont développé deux nouvelles formulations en PLNE qu'ils prouvent efficaces par rapport aux formulations classiques. Un état de l'art sur ce problème est donné par Robinson et al. [93].

## 2.7 PROBLÈME DE LOT SIZING COORDONNÉ AVEC CAPACITÉ *Coordinated Capacitated Lot-Size Problem CCLSP*

Si on considère le CULSP ou le DJRP avec une seule ressource (un véhicule par exemple) de capacité limitée, on obtient ce qu'on appelle le *Coordinated Capacitated Lot-Size Problem* (CCLSP). Pour la résolution du CCLSP, Robinson et Lawrence [92] proposent la relaxation lagrangienne pour trouver une solution heuristique et optimale. Narayanan et Robinson [83] proposent deux heuristiques pour ce type de problème, la première appelée *six-phases heuristic* (SPH), la deuxième est le recuit simulé. SPH est utilisé pour trouver une solution initiale en supposant que chaque demande est satisfaite par un approvisionnement unique et il cherche, d'une manière itérative, à maximiser le gain associé à la consolidation des demandes, ensuite le recuit simulé est utilisé pour explorer d'autres domaines de solutions. Un état de l'art sur ce problème est donné par Robinson et al. [93].

## 2.8 LE PROBLÈME DE LOT SIZING À PLUSIEURS NIVEAUX OU *Multi-Level Lot Sizing Problem* (MLLSP)

Dans les problèmes de dimensionnement de lots à un niveau, on ne considère que les demandes indépendantes (demandes externes). Dans le MLLSP la production de chaque produit génère des demandes dépendantes (demandes internes) pour ces composants. Le MLLSP est souvent classé en se basant sur la structure de la nomenclature des produits. Parmi ces structures, on trouve la structure en série, la structure d'assemblage et la structure générale [25]. Le MLLSP avec structure en série est étudié par plusieurs auteurs. On trouve entre autre : Zangwill [121], Lee et al. [70], van Hoesel et al. [112], Jaruphongsa et al. [64], Zhang et al. [122]. Zangwill [121] a étudié un problème de  $L$  échelons, en développant un algorithme de programmation dynamique de complexité  $O(LT^4)$  et van Hoesel et al. [112], montrent que pour  $L = 2$ , l'algorithme devient  $O(T^3)$ . Ils développent un algorithme polynomial pour le cas avec capacité de production constante. Lee et al [70] considèrent le problème de lot sizing à deux niveaux (*two level lot sizing*), le premier échelon est un entrepôt et le deuxième est un centre de distribution. Le coût de transport dans ce cas est modélisé sous une forme appelée *stepwise cargo cost* dans laquelle on trouve un coût fixe plus un coût en fonction du nombre de cargos utilisés. Ils identifient plusieurs propriétés du problème qui mènent à un algorithme de résolution polynomiale. Jaruphongsa et al. [64] ont étudié le problème de lot sizing multi échelons pour déterminer le plan d'approvisionnement d'un premier échelon qui est un entrepôt qui à son tour livre un centre de distribution (deuxième échelon) en utilisant deux modes de transport *Less than-Truck Load* (LTL) et *Full-Truck Load*

(FTL) de fonctions de coût différentes. En exploitant certaines propriétés du problème, ils développent un algorithme de programmation dynamique de complexité  $O(T^5)$ . Zhang et al. [122] considèrent le problème de lot sizing de deux échelons avec modes de transport multiples. Ils présentent un algorithme de complexité  $O(T^5)$ . Melo et Wolsey [79], proposent un algorithme de programmation dynamique de  $O(T^2 \log T)$  et une formulation étendue.

## 2.9 LE PROBLÈME D'UN ENTREPÔT ET PLUSIEURS DÉTAILLANTS (*One-Warehouse Multi-Retailer Problem*) OWMR

Dans le cas d'un problème de lot sizing à deux niveaux où le deuxième niveau comporte plusieurs clients, on obtient ce qu'on appelle *one-warehouse multi-retailer problem*, avec une demande dynamique. Dans le problème OWMR un entrepôt s'approvisionne d'un produit auprès de son fournisseur pour satisfaire les demandes externes de ses détaillants. Le produit peut être stocké à l'entrepôt ou chez les détaillants. Le OWMR est une extension du DJRP. L'approvisionnement de l'entrepôt ou de chaque détaillant engendre un coût de commande fixe à chaque période. Les autres coûts sont, le coût d'achat et les coûts de stockage. Le problème OWMR consiste à déterminer, pour chaque période, les quantités à approvisionner du fournisseur à l'entrepôt, et les quantités à livrer de l'entrepôt à chaque détaillant. Parmi les travaux qui considèrent ce type de problème on trouve Federgruen et Tzur [47], Chan et al. [29], Levi et al. [75], Jin et Muriel [119]. Federgruen et Tzur [47] ont développé une *time-partitioning heuristic*, dans laquelle le problème est divisé en sous problèmes d'intervalles plus petits. Chaque sous problème est modélisé par la formulation à stock d'échelon et résolu par un algorithme de branch & bound basé sur la relaxation lagrangienne. Chan et al. [29], considèrent le coût de commande comme une fonction par morceaux et la politique ZIO (*zero-inventory ordering*) c.à.d le fournisseur et les clients commandent uniquement si leur stock est à zéro. Ils trouvent que le coût de cette politique est au plus  $4/3$  fois supérieur au coût de la politique optimale. Levi et al. [75], ont étudié le problème avec les coûts de stockage d'une structure générale et des coûts fixes de livraison des détaillants stationnaires. Ils ont utilisé la relaxation linéaire pour développer un algorithme d'approximation qui donne une solution égale à 1.8 fois la solution optimale dans le pire des cas. Jin et Muriel [119] proposent deux algorithmes de décomposition lagrangienne, pour un problème avec des coûts constants et des capacités d'approvisionnement limitées pour l'entrepôt et les détaillants. Un état de l'art récent sur le OWMR peut être trouvé dans Solyali et Süral [106] qui présentent des étendues et des classifications des différentes variantes du problème OWMR avec demande dynamique.

Un problème particulier proche de OWMR est celui étudié par Anily et Tzur [8]. Ils considèrent un système de livraison de plusieurs produits, d'un fournisseur à un entrepôt, par une flotte de véhicules homogènes, un coût fixe pour chaque véhicule utilisé dans une période et un coût de stockage lié à chaque produit. Mais le coût fixe de commande de chaque produits et le coût de stockage à l'entrepôt ne sont pas considérés. Ils utilisent la programmation dynamique dont

la solution optimale est polynomiale pour un nombre fixe de produits et exponentielle autrement.

Un autre problème est étudié par Kang et Kim [65] pour le problème de distribution d'un seul produit par un fournisseur à plusieurs clients concentrés dans une région. Dans ce cas, le coût majeur est le coût d'utilisation d'un véhicule et le coût mineur est le coût de service de chaque client. Ils ont suggéré une heuristique à deux phases : la première phase détermine les quantités à livrer à chaque client sans attribution de véhicules aux clients en résolvant des problèmes de lot sizing indépendants. La deuxième phase détermine le nombre minimum de véhicules à utiliser pour regrouper les livraisons inférieures à la capacité d'un véhicule en appliquant l'algorithme appelée *Modified First Fit Decreasing* (MFFD). Cette configuration est la plus proche de notre problème du côté détaillants. Cependant, du côté entrepôt, notre problème est différent en ce sens que [65] ne considère pas le coût de setup d'approvisionnement et le coût de stockage à l'entrepôt. Les capacités de production et les capacités de stockage aux détaillants ne sont pas aussi considérées.

Melo et Wolsey [80] considèrent le problème avec plusieurs producteurs et plusieurs produits, mais avec des livraisons directes entre les producteurs et les détaillants. Ils développent des heuristiques hybrides basées sur des formulations PLNE.

## 2.10 LE PROBLÈME PRODUCTION-ROUTAGE (*Production Routing Problem*) PRP

Le PRP est une extension du OWMR. Ce dernier est considéré comme une intégration entre le problème de lot sizing et la livraison directe, c.à.d. les produits sont transportés directement de l'entrepôt (producteur) vers les détaillants. Dans le PRP, la livraison des détaillants se fait par des tournées effectuées par une flotte de véhicules ; chaque véhicule peut visiter plusieurs détaillants dans une tournée. Le PRP est une intégration de deux problèmes classiques qui sont le problème de lot sizing et le problème de tournées de véhicules ou *Vehicle Routing Problem* (VRP).

Ce problème a attiré l'attention de plusieurs auteurs ces dernières années. Chandra et Fisher [30] étaient les premiers à étudier la possibilité d'intégrer le lot sizing avec le VRP. Ils montrent le gain de la coordination entre la production et la distribution. Ils comparent deux approches. La première est la résolution séparée des problèmes de production et de distribution. La deuxième est la résolution des deux problèmes en un seul modèle. Les deux approches sont testées avec les différentes valeurs des paramètres du modèle qui comporte : la longueur de l'horizon de planification, le nombre de produits, le nombre de clients, le coût de setup, le coût de stockage à l'usine et le coût de transport. Des instances jusqu'à 50 clients, 10 produits et 10 périodes sont utilisées pour tester l'approche. Ils obtiennent une réduction de 3% à 20% sur le coût total avec une approche intégrée.

Fumero et Vercellis [50] considèrent le même problème que celui de Chandra et Fischer

[30]. Ils utilisent la relaxation lagrangienne pour résoudre le problème, mais seulement pour de petites instances, avec un maximum de 12 clients, 10 produits, et huit périodes. Les auteurs ont aussi une approche découplée du problème où ils prennent les décisions de production sans tenir compte du reste du système. Ils comparent à la fin de leur article le gain apporté par la coordination.

Lei et al. [72] proposent une approche de résolution en deux phases avec plusieurs producteurs et une flotte hétérogène. Dans la première phase, le problème PLNE est résolu en réduisant les contraintes de routage à des livraisons directes. La solution obtenue donne un plan de production et la quantité à livrer à chaque client dans chaque période. Cette solution est réalisable pour le problème original. Dans la deuxième phase, une heuristique de routage pour consolider les affectations des véhicules peu remplis (*less-than transporter loads LTL*) pour obtenir un programme de livraison efficace. Ils ont appliqué cette approche à un problème réel avec des instances qui comportent 12 périodes, deux usines, 13 clients et trois véhicules hétérogènes.

Bertazzi et al. [19] considèrent un problème sans capacité de production, avec un coût fixe pour chaque véhicule utilisé dans tout l'horizon de production et la quantité à livrer à chaque client est déterminée selon deux politiques : si un client est livré, on remplit son stock au maximum (*order-up-to level policy*) et si on remplit le stock de chaque client livré au maximum sauf le dernier client reçoit la quantité résiduelle dans le véhicule (*full-fill-dump policy*). Ils ont développé une heuristique basée sur la décomposition du problème en un problème de production et un problème de distribution.

Archetti et al. [9] comparent la première politique de Bertazzi et al. [19] et la politique de livraison d'une quantité inférieure ou égale à la capacité de stockage du client, appelée *maximum level policy* (ML). Ils proposent une heuristique basée sur un PLNE pour résoudre le problème avec capacité de production illimitée. L'heuristique est composée de trois étapes : Tout d'abord, les quantités de production sont supposées illimitées et le problème de distribution, qui consiste à déterminer les niveaux des stocks chez les clients et les décisions de routage, est optimisé en résolvant un PLNE où chaque client est ajouté itérativement à la solution. Le plan de production est déterminé dans la deuxième étape par la résolution d'un problème lot sizing sans capacité (ULSP). Dans la troisième étape, l'algorithme explore de nouvelles améliorations en retirant et en insérant deux détaillants. L'étape de l'amélioration est répétée jusqu'à ce qu'il n'y a pas de réduction supplémentaire du coût total. Cette heuristique est testée sur des instances larges (de 50 à 100 détaillants et 6 périodes). Ils ont aussi développé une approche de *branch and cut* pour le PRP avec un seul véhicule. Ils ont testé l'approche sur des instances de 14 détaillants et 6 périodes.

Boudia et al. [23] proposent la procédure de recherche adaptative gloutonne et randomisée (*greedy randomized adaptive search procedure* GRASP) réactive avec le *path-relinking*. Ainsi dans une première phase, une solution initiale est générée par l'algorithme glouton randomisé (*randomized heuristic algorithm*). Dans la deuxième phase une recherche locale est appliquée

pour trouver un optimum local dans le voisinage de la solution construite par la première phase. Ensuite, ils ont développé une procédure de *path-relinking* dans laquelle, les solutions obtenues par le GRAPS sont classées en fonction de leur coût total. Un nombre limité de solutions est stocké dans un pool de solutions d'élite. Ensuite, chaque deux solutions dans le pool sont choisies pour créer une nouvelle solution en transférant des quantités de livraison dans une de ces solutions à une autre période, selon les quantités de livraison dans l'autre solution pour réduire les différences entre ces deux solutions. Ce processus pourrait améliorer légèrement la solution obtenue par le GRASP. Les auteurs n'ont pas pris en considération le coût de stockage chez les détaillants dans ce PRP. Cet algorithme est évalué sur des instances, de 50, 100 et 200 détaillants et 20 périodes.

Boudia et Prins [24] ont repris le même problème que Boudia et al. [23] et lui ont appliqué un algorithme dit *Memetic Algorithm with Population Management (MA|PM)*. Un algorithme mémétique est une modification l'algorithme génétique qui utilise une certaine forme de recherche locale pour améliorer les solutions. Dans ce travail, une solution initiale est générée par une heuristique simple qui fixe au préalable un plan de production, dans chaque période, égale à la demande totale. Puis, une heuristique est utilisé pour générer le plan de livraison et ensuite le plan de production est ajusté par l'algorithme Wagner-Whitin. Après, l'algorithme utilise le croisement pour générer de nouveaux individus. Dans l'étape suivante, les auteurs ont adopté la recherche locale de Boudia et al. [23] pour améliorer les individus générés, ils ont appliqué la méthode de gestion de la population pour la sélection où les nouveaux individus ne sont acceptés que s'ils améliorent la solution actuelle avec un certain seuil. Cet algorithme est évalué sur les mêmes instances de [23]. Les résultats montrent un gain de 23% de l'approche intégrée par rapport à l'approche découplée.

Bard et Nananukul [15] développent une procédure de recherche tabou réactive. Dans leur algorithme, une solution initiale est créée par la résolution d'un PRP modifié obtenu en supprimant les contraintes de routage. Ensuite, les décisions de routage sont obtenues par la résolution d'un VRP avec capacité, par une procédure basée sur la recherche tabou. La solution produite est améliorée par la procédure de recherche locale en utilisant le mouvement d'échange ou le mouvement de transfert. Le mouvement d'échange examine la possibilité d'échanger une quantité maximale entre deux clients dans deux périodes. Le mouvement de transfert cherche la quantité de livraison d'un client qui peut être combinée avec une autre quantité de livraison dans une période précédente afin de réduire le coût de transport. Les mouvements qui mènent à l'amélioration de la solution sont stockés dans la liste tabou et les solutions non faisables ne sont pas autorisées.

Shiguemoto et Armentano [102] proposent aussi un algorithme de recherche tabou, avec un mécanisme de relaxation qui permet l'évaluation de solutions non faisables pour guider la recherche vers les solutions faisables. Cet algorithme commence par une solution initiale (quantités à livrer) égale aux demandes. Puis une heuristique est appliquée afin de déterminer les routes par période. Après, l'algorithme de Wagner et Whitin [115] est utilisé pour élaborer le plan de production. Des mouvements sont examinés, et celui qui conduit vers l'amélioration du

coût total est exécuté puis stocké dans la liste tabou. Chaque mouvement a trois composantes : (1) le transfert d'une quantité maximale d'une période à une autre sans violer les capacités de stockage ; (2) l'insertion de cette quantité dans une route, (3) la détermination d'un nouveau plan de production. Enfin, une stratégie de diversification est effectuée. Cet algorithme est appliqué sur les instances de Bertazzi et al. [19]. Les résultats indiquent que cet algorithme donne une réduction de coût total moyen de 48-50% si on le compare à l'algorithme de décomposition heuristique de Bertazzi et al. [19].

Armentano et al. [11] présentent deux variantes de la recherche tabou pour le PRP : l'une avec deux phases, appelée, *construction and short-term memory*, et l'autre qui intègre une mémoire longue utilisée dans la *path relinking*. Cette approche accepte également quelques solutions non faisables, ce qui facilite le passage vers de bonnes solutions. La première variante *construction and short-term memory* est déterminée de la même manière que dans Shiguemoto et Armentano [102], mais avec la prise en considération des contraintes supplémentaires (capacité de production et longueur de trajet). Enfin, la procédure *path relinking* est proposée pour intégrer les stratégies d'intensification et de diversification dans la méthode de résolution. Toutes ces procédures sont testées sur des instances avec plusieurs produits générées par les auteurs, et aussi sur les instances avec un seul produit de Boudia et al. [23]. Les deux variantes donnent de bons compromis entre la qualité de solution obtenues et le temps de calcul. En outre, ces approches dépassent le *Memetic Algorithm* développé par Boudia et Prins [24] et la recherche tabou réactif proposé par Bard et Nananukul [15].

Bard et Nananukul [16] proposent une approche hybride qui combine une méthode exacte avec une heuristique dans un cadre de branch & price pour un PRP multi véhicules. Pour améliorer la procédure de branch & price, l'approche se compose de quatre phases. Dans la première phase, une solution initiale est générée par l'heuristique de recherche tabou présentée par Bard et Nananukul [15]. En second lieu, au cours du processus de génération de colonnes, les sous problèmes sont résolus par un algorithme heuristique basé sur la séparation au lieu d'être résolus à l'optimalité. Dans la troisième phase, le schéma de branchement est modifié pour brancher les variables en groupes. Une procédure heuristique d'arrondissement est utilisée dans la dernière phase. Avec ces modifications, les performances du processus de branch & price, sont sensiblement améliorées. Les tests effectués sur des instances, avec un maximum de 50 clients et 8 périodes, montrent que cette approche donne de meilleures solutions que celles obtenues par le solveur commercial CPLEX (les coûts de la solution sont améliorés de 12,2% en moyenne) dans une heure de temps d'exécution.

Ruokokoski et al. [96] ont développé une méthode exacte de branch & cut pour le PRP avec capacités de production et de véhicule illimitées. Ils introduisent, en plus de la formulation de base, des formulations renforcées du problème de lot sizing telles que la formulation basée sur le problème du plus court chemin (*shortest path formulation*) et la formulation basée sur le problème de localisation (*facility location formulation*). Les résultats montrent que, lorsque les contraintes d'élimination des sous-tours sont négligées, les valeurs de la relaxation linéaire

des deux dernières formulations sont améliorées par rapport à celle de la formulation de base. De plus dans les mêmes conditions, la formulation basée sur le problème de localisation donne de meilleures performances avec l'ajout des inégalités valides pour le problème de routage. Ils ont également adapté les inégalités valides appelées *generalized comb and 2-matching* (qui ont été développée initialement pour le problème de voyageur de commerce généralisé GTSP). Les résultats montrent que lorsque toutes les inégalités valides sont utilisées, la relaxation linéaire de la formulation basée sur le problème de localisation donne une meilleure borne inférieure. L'algorithme développé peut résoudre des problèmes avec 80 clients et 8 périodes dans un temps d'exécution de deux heures.

Adulyasak et al. [4] ont comparé les performances de deux formulations. La première est la formulation sans indice de véhicule. Les auteurs ont également développé des inégalités pour renforcer la partie routage de la formulation. La deuxième est la formulation ayant un indice de véhicule. Cette formulation est améliorée grâce à l'ajout des inégalités pour éliminer la symétrie qui peut être créée en raison du fait que les véhicules sont identiques. Les deux formulations sont aussi améliorées par les inégalités valides de Archetti et al. [9]. La formulation avec indice de véhicule est résolue par un algorithme de Branch & Cut similaire à celui de Ruokokoski et al. [96]. Les résultats obtenus ont montré que la formulation ayant un indice des véhicules est meilleure dans la recherche des solutions optimales. Des instances jusqu'à 35 clients, 3 périodes et 3 véhicules ont été résolus à l'optimalité en 2h. La formulation sans indice de véhicule, cependant, peut produire une meilleure borne inférieure au nœud racine et trouver de meilleures bornes inférieures pour les instances qui ne sont pas résolus à l'optimalité en 2h. Les auteurs ont également testé les performances de la formulation avec un indice de véhicule avec une machine huit cores en utilisant le calcul parallèle et des instances avec un maximum de 50 clients, 3 périodes et 3 véhicules ont été résolus à l'optimalité en 12 h.

Adulyasak et al. [3] ont appliqué l'heuristique ALNS *Adaptive Large Neighborhood Search* à un PRP qu'ils décomposent en plusieurs sous-problèmes, plus faciles à résoudre. Tout d'abord, plusieurs solutions initiales différentes sont générées par la résolution des sous problèmes décomposés, de façon similaire à l'approche de Bard et Nananukul [15]. Puis, les auteurs ont développé deux types d'opérateurs, appelés sélection et transformation. A chaque itération, un type d'opérateur est sélectionné aléatoirement. L'opérateur de sélection est appliqué en premier pour créer une liste de nœuds candidats (combinaisons client-période) et l'opérateur de transformation est appliquée à retirer et insérer les nœuds candidats dans la liste à la solution actuelle. Après, quand une nouvelle solution est trouvée au cours du processus de transformation, un problème de flux maximum est résolu pour optimiser les quantités de production, de stockage et de livraison. Le critère d'arrêt de leur algorithme est basé sur nombre maximum d'itérations. Afin d'évaluer l'efficacité de la méthode proposée, des calculs sont effectués sur des instances proposées par Archetti et al. [9] et Boudia et al. [23].

Absi et al. [1] ont introduit récemment une approche efficace pour résoudre le PRP sans capacité de production, basée sur deux phases. La première est la phase de lot sizing qui consiste

à résoudre une formulation PLNE où les coûts de routage sont remplacés par un coût fixe approximative de visite de chaque client. La deuxième est la phase de routage qui est appelée pour calculer des routes des véhicules à chaque période. Les coûts sont actualisés d'une manière à avoir un coût minimum d'insertion correspondant à la solution en cours et l'itération suivante est effectuée selon cette nouvelle configuration. Un mécanisme de diversification qui modifie les coûts de visite est également appliqué lorsque la solution actuelle n'est pas améliorée pour un certain nombre d'itérations. Les auteurs ont proposé deux variantes de cette approche itérative. Dans la première variante, une capacité de véhicule individuel est prise en compte dans la résolution du PLNE et chaque route est déterminée par une heuristique de TSP, tandis que dans la deuxième variante, la capacité globale du véhicule considérée et les routes sont déterminées par une heuristique de VRP. Les résultats de l'application de cette approche montrent que la deuxième variante est meilleure que la première et peut généralement produire de meilleures solutions dans un temps de calcul réduit. Les résultats montrent aussi que cette approche dépasse les heuristiques d'Archetti al. [9] et de Adulyasak et al. [3] en utilisant les instances de Archetti al. [9].

Lei et al. [72] proposent une approche de solution en deux étapes pour résoudre un PRP avec plusieurs producteurs et véhicules hétérogènes. Dans la première étape, le modèle est résolu comme un PLNE avec les contraintes du modèle original, sauf les contraintes de routage qui sont simplifiées à des livraisons directes. Cette étape détermine à chaque période, les quantités de production, les niveaux de stocks dans les usines et centres de distribution, et le nombre de livraisons par véhicule. Ensuite, un algorithme heuristique pour consolider les véhicules non pleins est proposé afin d'éviter des livraisons directes inefficaces dans la première étape en déterminant les routes par véhicule à chaque période pour chaque usine. En comparaison avec CPLEX, l'approche proposée fournit la même, ou une meilleure solution dans la plupart des instances générées par une seule usine, jusqu'à 12 centres de distribution, 2 véhicules et 4 périodes. En outre, la méthode de résolution est validée dans un chaîne logistique réelle relative à une société de produits chimiques.

Brahimi et Aouam [26] considèrent un PRP avec *backordering* et plusieurs produits. Pour la formulation, ils proposent deux modèles PLNE. Une heuristique de *relax-and-fix* combinée avec une heuristique de recherche locale est utilisée pour la résolution du problème. Les résultats de cette heuristique d'hybridation sont meilleurs que ceux obtenus par un solveur commercial, en terme de qualité et en temps d'exécution.

Pour une revue de la littérature récente sur le PRP, voir Adulyasak et al. [4] et Diaz-Madronero et al. [37].

## 2.11 CONTRIBUTION DE LA THÈSE

Dans tous les travaux traitants le PRP, les clients sont dispersés, par conséquent, le coût de transport est le coût de passage d'un client à un autre, par contre, dans notre problème, les clients sont concentrés dans une région et le coût de passage d'un client à un autre est négligé. Nous avons introduit un coût de service qui dépend uniquement du client visité en plus d'un coût fixe pour chaque véhicule utilisé dans chaque période. La majorité des PRP que nous avons répertorié sont mono véhicule et dans les travaux qui considèrent plusieurs véhicules, le coût fixe d'utilisation d'un véhicule n'est pas pris en considération. A notre connaissance, il n'existe pas de travaux qui traitent un problème d'intégration production, stockage et distribution sous les mêmes hypothèses considérées dans notre problème.

Dans tous les travaux traitants le PRP, le sous problème de distribution est un VRP à chaque période, sur lequel sont basées toutes les méthodes de résolution proposées. Dans notre problème, il n'existe pas de VRP, ce qui rend les méthodes proposées pour le PRP, inefficaces. Aussi, nous proposons dans cette thèse des approches de résolution qui exploitent les particularités de notre problème.

## CONCLUSION

Nous avons présenté dans ce chapitre un état de l'art des différents travaux relatifs à notre problème. Nous avons commencé par le problème de lot sizing et ses différentes extensions. Une de ces extensions est le *one-warehouse multi-retailer problem* qui est un problème de lot sizing à deux niveaux avec plusieurs clients dans le deuxième niveau. Dans le OWMR le transport vers les clients se fait par des livraisons directes. Mais dans la réalité, et pour minimiser les coûts de transport, on peut livrer plusieurs clients dans une tournée, ce qui intègre les décisions de routage des véhicules avec les décisions de production qui est le cas d'un PRP. Notre problème est cas particulier de ce dernier. Aussi, une importance particulière a été accordée, dans l'état de l'art, à ce problème.

Nous avons répertorié plus de 17 études portant sur le PRP. Ce problème est NP-Difficile et la majorité des méthodes de résolution sont des heuristiques. L'idée générale derrière toutes ces heuristiques et méta-heuristiques est de décomposer le problème en deux sous-problèmes et les résoudre séquentiellement par des méthodes spécifiques à chaque sous problème. Très peu d'auteurs ont utilisé des méthodes exactes. Ce qui nous a motivé à proposer, dans le chapitre suivant, une méthode exacte, qui consiste à donner plusieurs formulations, et pour les renforcer nous développons des inégalités valides dont certaines sont nouvelles et d'autres sont adaptées de la littérature.



# APPROCHE POLYÉDRALE POUR LA RÉSOLUTION DU PROBLÈME

## RÉSUMÉ

*DANS ce chapitre nous nous intéressons au développement d'une méthode exacte pour la résolution de notre problème. Nous présentons d'abord un aperçu général des méthodes exactes proposées dans la littérature. Ensuite, nous introduisons différentes formulations de programmation linéaire en nombres entiers. Nous développons par la suite plusieurs familles d'inégalités valides pour renforcer deux formulations. Enfin, nous effectuons une étude de calcul en comparant chacune des formulations sur un grand nombre d'instances et nous discutons aussi l'effet de l'ajout des inégalités.*



## 3.1 INTRODUCTION

En raison de la complexité des problèmes d'intégration production distribution, peu d'études ont mis en place des méthodes exactes. Ce qui nous a motivé à proposer, dans ce chapitre, l'approche polyédrale comme méthode exacte, pour résoudre notre problème. Les approches polyédrales doivent leur succès grandissant à leurs bons résultats quand elles sont appliquées au problème du voyageur de commerce. Ces résultats sont dus à une bonne connaissance de l'enveloppe convexe des solutions. Avant de commencer cette approche, nous introduisons dans la section 3.2 un bref rappel sur les méthodes exactes. Dans la section 3.3, nous modélisons notre problème en quatre formulations de programmation linéaire en nombres entiers. Pour renforcer ces formulations, nous développons dans la section 3.4 des inégalités valides dont certaines sont nouvelles et d'autres sont adaptées de la littérature. Pour monter l'intérêt de l'approche polyédrale proposée, nous analysons dans la section 3.5 les résultats expérimentaux obtenus.

## 3.2 APERÇU SUR LES MÉTHODES EXACTES DE RÉOLUTION

Nombreuses sont les méthodes proposées dans la littérature pour résoudre les problèmes d'intégration production-distribution. Elles sont généralement classées selon deux dimensions : les méthodes exactes et les méthodes approchées. Une méthode exacte permet de garantir l'optimalité de la solution obtenue, à la différence d'une méthode approchée qui tente de s'en approcher le plus possible. Nous donnons dans ce chapitre un aperçu des méthodes exactes qui sont souvent développées pour des problèmes faciles ou pour résoudre des problèmes difficiles de petites tailles.

### 3.2.1 La programmation linéaire

La programmation linéaire regroupe l'ensemble des techniques mathématiques d'optimisation qui consistent à maximiser (ou minimiser) une fonction objectif linéaire de variables soumises à un ensemble de contraintes exprimées sous forme d'équations ou d'inéquations linéaires. On note que l'ensemble des solutions admissibles (qui respectent les différentes contraintes du problème) forment dans ce cas une enveloppe convexe nommé polytope (polyèdre borné) et que l'ensemble des sommets du polytope constitue l'ensemble dominant de solutions parmi toutes les solutions admissibles.

### 3.2.2 La programmation linéaire en nombres entiers

Lorsqu'un problème d'optimisation peut être modélisé comme un programme linéaire dont les variables sont contraintes à être entières alors le programme est dit linéaire en nombre entiers (PLNE). Quand certaines variables seulement sont contraintes à être entières, on parle alors de programme linéaire en variables mixtes (PLNE mixte). Il est souvent difficile à résoudre un PLNE du fait que l'espace de recherche n'est plus convexe mais discret. Il existe principalement deux approches de recherche de la solution optimale entière des PLNE : Les méthodes arborescentes par séparation et évaluation (*branch and bound* et les méthodes de coupes *branch and cut*).

**a) La méthode par séparation et évaluation ou branch & bound**

C'est une méthode de recherche arborescente qui énumère à travers des nœuds des sous-ensembles de solutions. Le principe est d'éliminer de l'arbre de recherche les branches prouvées non optimales afin d'éviter l'énumération exhaustive des solutions. Pour chaque sous-ensemble, une borne inférieure et une borne supérieure du critère sont calculées. Si à un nœud donné, la borne inférieure est supérieure à la meilleure borne supérieure trouvée jusqu'ici (pour un problème de minimisation), alors il est certain qu'il n'existe pas de solutions meilleure dans ce sous-ensemble issu du nœud (le branchement s'arrête au niveau du nœud). Les méthodes de recherche arborescente sont généralement des méthodes exponentielles.

**b) Les méthodes de coupes ou *branch and cut***

Les méthodes de coupes ont pour but de trouver le plus petit polytope contenant toutes les solutions entières du PLNE. En pratique, ces méthodes consistent à itérer le processus suivant jusqu'à obtenir une solution entière : résoudre la relaxation linéaire du problème (relâcher les contraintes d'intégrité), puis ajouter des inégalités valides (les coupes) au problème pour réduire l'enveloppe des solutions sans éliminer des solutions entières admissibles. Si toutes les coupes sont générées, alors l'enveloppe convexe des solutions du problème est identifiée. La difficulté principale de cette méthode réside dans le choix des coupes efficaces pour une convergence rapide de l'algorithme.

***i) Inégalités valides***

Les modèles PLNE sont résolus d'une manière exacte par des algorithmes de programmation linéaire (tel que le simplexe) combinés avec des méthodes de branch & bound. L'efficacité de ces algorithmes implémentés dans les solveurs commerciaux dépend essentiellement de la qualité des bornes calculées au niveau des nœuds de l'arbre de recherche. Les chercheurs se sont intéressés aux méthodes polyédrales, qui permettent d'améliorer les bornes calculées par les algorithmes de programmation linéaire. Le principe de ces méthodes est d'ajouter de l'information au modèle dans le but de l'améliorer. L'amélioration du modèle se traduit par l'ajout de contraintes linéaires (inégalités valides, ou IV). Ces IV sont redondantes dans le modèle original, mais qui permettent d'améliorer les bornes par la réduction de l'espace des solutions de la relaxation linéaire de la formulation originale tout en n'éliminant aucune solution réalisable du problème entier. La difficulté majeure de l'approche polyédrale réside dans l'identification des ensembles (ou familles) d'IV. Ainsi et suite à plusieurs études polyédrales effectuées sur des problèmes difficiles connus dans la littérature; des familles d'IV pour des problèmes spécifiques ont été proposés. Ces familles d'IV ont été utilisées pour développer des algorithmes pour le problème initial. Elles ont également été adaptées pour d'autres formulations.

*ii) Séparation des inégalités valides*

La méthode de *branch & cut* consiste à ajouter à des nœuds de l'arbre de recherche des inégalités valides (des coupes) afin d'améliorer la qualité de la borne inférieure. Le processus de génération des inégalités non satisfaites par la relaxation linéaire au nœud courant est appelé séparation.

Notons que les cas, où le sous-problème de séparation peut être résolu polynomialement, sont malheureusement peu fréquents. Dans la majorité des cas il faut se contenter de mettre en œuvre des heuristiques pour le résoudre si l'on ne veut pas compromettre l'efficacité algorithmique du processus.

Dans les solveurs commerciaux, l'utilisation des coupes porte sur des structures spécifiques connues par l'étude de problèmes combinatoires et des coupes générales. Malheureusement, les problèmes en nombres entiers issus d'applications industrielles concrètes se caractérisent : soit par une absence de structure particulière ; soit par le fait qu'ils apparaissent comme combinant, de façon complexe, plusieurs types de structures connues souvent associées à des parties sans structures particulières.

Pour de tels problèmes que nous qualifierons de programmes en nombres entiers généraux, l'approche résultant de l'analyse de problèmes combinatoires structurés apparaît souvent insuffisante. En effet, même si on peut toujours exploiter la connaissance des inégalités valides pour les parties structurées du problème, celles-ci ne correspondent qu'à des sous-ensembles de variables et de contraintes considérés indépendamment les uns des autres, et de ce fait l'appréhension globale du problème est perdue.

*iii) Les coupes classiques utilisées dans les solveurs*

Les coupes classiques, qu'on trouve dans la plus part des solveurs commerciaux, sont, soit des coupes générales élaborés en procédant à l'arrondissement sur les parties entières des inégalités valides pour la relaxation continue (coupes MIR par exemple) ou en travaillant sur les parties fractionnaires des inégalités valides pour la relaxation continue (coupes de *Gomory* par exemple), soit des coupes portant sur des structures spécifiques connues par l'étude de problèmes combinatoires (coupes de couverture et coupes de couverture de flux définies originalement pour la structure de sac à dos et la structure dite *single node flow* respectivement)

pour les problèmes MIP de structures générales, il y'a plusieurs approches d'élaboration des inégalités valides ou classes des inégalités valides dans la littérature, l'une des approches est de commencer par la relaxation du modèle original à des modèles simples ayant des inégalités valides connues. Il y'a deux types de relaxation : les relaxations de bas niveau (*low-level relaxation*) et les relaxations de haut niveau (*high-level relaxation*) Pochet et Wolsey [87]. Les inégalités valides basées sur la relaxation de bas niveau mènent aux coupes classiques (citées dans la paragraphe précédent) qui sont générées automatiquement par la plus part des solveurs. Ils sont

basées sur une seule contrainte ou sur les bornes simples d'une variable. Les relaxations de haut niveau s'intéressent à un ensemble de contraintes et de variables. Un exemple de la relaxation de haut niveau est la relaxation du problème *single-item uncapacitated lot-sizing*, qui mènent à des formulations plus fortes de l'enveloppe convexe des solutions de certains problèmes de planification de la production.

### 3.2.3 La relaxation lagrangienne

C'est une méthodologie générale qui permet d'obtenir des bornes inférieures de bonne qualité pour certains problèmes d'optimisation combinatoire. L'idée de la technique consiste à supprimer (relaxer) certaines contraintes (en principe celles qui rendent le problème difficile) en les introduisant dans la fonction objectif sous forme d'une pénalité (combinaison linéaire des contraintes relaxées) si les contraintes relaxées ne sont pas respectées. Les coefficients de cette combinaison sont appelés multiplicateurs de Lagrange, et la nouvelle fonction objectif est appelée le Lagrangien du problème. La résolution du nouveau problème se fait généralement en utilisant la méthode du sous-gradient ou bien la méthode de génération de colonnes.

### 3.2.4 La génération de colonnes

Dans la pratique, de nombreux problèmes d'optimisation combinatoire sont complexes et de grande taille. Ils possèdent un grand nombre de variables (colonnes). Ainsi leur résolution par les logiciels existants, de résolution des programmes linéaires n'est pas possible. Pour pouvoir les résoudre, on utilise la méthode de génération de colonnes. Cette méthode consiste à décomposer le problème original, par la méthode de *Dantzig-Wolfe*, en un ou plusieurs sous-problèmes plus faciles à résoudre, l'ensemble étant généralement coordonné par un programme linéaire appelé maître. Le principe de l'algorithme de génération de colonnes consiste à résoudre le problème original ( $P$ ) avec un sous-ensemble de colonnes (variables) de taille réduite, puis à l'alimenter itérativement avec de nouvelles colonnes susceptibles d'améliorer la solution courante, jusqu'à atteindre l'optimalité. La rapidité de la méthode dépend à la fois, du nombre d'itérations nécessaire jusqu'à preuve d'optimalité (c.à.d de l'efficacité de l'algorithme générateur dans le choix de la colonne améliorante), et du temps de calcul consacré à chaque étape de résolution.

### 3.2.5 La programmation dynamique

La programmation dynamique n'est pas à proprement parler un algorithme mais un principe général applicable à de nombreux problèmes d'optimisation avec contraintes, possédant une certaine propriété dite de décomposabilité. En général, les problèmes se ramènent au choix d'une suite de décisions séquentielles pour un objectif donné. Le résultat définit la valeur optimale du critère. Pour obtenir la solution optimale, un parcours à rebours des décisions qui ont menées à la valeur optimale est effectué. Ces méthodes ont une complexité polynomiale ou exponentielle, selon le problème.

### 3.3 FORMULATION DU PROBLÈME

Nous rappelons que le problème considéré dans cette thèse est un problème d'intégration production, stockage et distribution dans une chaîne logistique composée d'un producteur qui fabrique un seul type de produit avec une capacité de production limitée et de capacité de stockage illimitée et qui livre un groupe de détaillants avec une flotte de véhicules de capacité limitée. La capacité de stockage de chaque détaillant est limitée. Un détaillant peut être livré au plus par un seul véhicule dans une période. Un véhicule peut faire au plus une tournée par période. La demande de chaque détaillant, dans une période, est déterministe et peut être satisfaite soit du stock soit de la livraison de la même période. Les coûts considérés sont : le coût fixe de lancement de production, le coût de stockage au producteur, un coût fixe d'utilisation d'un véhicule, un coût fixe de service de chaque détaillant et le coût de stockage chez chaque détaillant. L'objectif est de déterminer pour chaque période, les quantités à produire, les quantités à livrer à chaque détaillant et l'affectation des véhicules au détaillants, de manière à minimiser le coût total considéré. Pour modéliser ce problème, nous présentons quatre formulations PLNE en se basant sur des formulations conçues initialement pour le problème de lot sizing.

#### 3.3.1 Formulation agrégée ou *Aggregated Formulation* AGF

La formulation agrégée est la plus utilisée pour la modélisation des problèmes de lot sizing. Les notations de cette formulation sont présentées ci-dessous.

##### Ensemble d'indices

$\mathcal{T} = \{1, \dots, T\}$  : ensemble de périodes

$\mathcal{J} = \{1, \dots, J\}$  : ensemble de détaillants

$\mathcal{K} = \{1, \dots, K\}$  : ensemble de véhicules

##### Les indices

$t = 1..T$  : indice des périodes

$j = 1..J$  : indice des détaillants

$k = 1..K$  : indice des véhicules

##### Les paramètres

$d_{jt}$  : la demande du détaillant  $j$  dans la période  $t$

$c_j^C$  : coût de service du détaillant  $j$

$W$  : capacité de chaque véhicule

$S^P$  : coût fixe de setup de production

$h^P$  : coût de stockage unitaire chez le producteur

$h_j^C$  : coût de stockage unitaire chez le détaillant  $j$

$I_j^{max}$  : capacité de stockage du détaillant  $j$

$R$  : coût fixe d'utilisation de chaque véhicule

$Q$  : capacité de production

##### Les variables de décision

$y_t$  : variable binaire égale à 1 s'il y a une production à la période  $t$

$q_t^P$  : la quantité à produire à la période  $t$

$x_{jkt}$  : variable binaire égale à 1 si le détaillant  $j$  est visité par le véhicule  $k$  dans la période  $t$

$I_t^P$  : le niveau du stock au producteur à la période  $t$

$I_{jt}^C$  : le niveau du stock au détaillant  $j$  à la période  $t$

$q_{jkt}^C$  : la quantité à livrer au détaillant  $j$  par le véhicule  $k$  dans la période  $t$

$q_{kt}^V$  : la quantité à livrer par le véhicule  $k$  dans la période  $t$

$v_{kt}$  : variable binaire égale à 1 si le véhicule  $k$  est utilisé dans la période  $t$

### La formulation AGF

$$\text{Minimiser } \sum_{t=1}^T \left( S^P \cdot y_t + h^P \cdot I_t^P + \sum_{k=1}^K R \cdot v_{kt} + \sum_{j=1}^J \sum_{k=1}^K c_j^C \cdot x_{jkt} + \sum_{j=1}^J h_j^C \cdot I_{jt}^C \right) \quad (3.1)$$

sout les contraintes suivantes :

$$I_t^P = I_{t-1}^P + q_t^P - \sum_{j=1}^J \sum_{k=1}^K q_{jkt}^C \quad \forall t \in \mathcal{T} \quad (3.2)$$

$$I_{jt}^C = I_{j,t-1}^C - d_{jt} + \sum_{k=1}^K q_{jkt}^C \quad \forall j \in \mathcal{J}, \quad \forall t \in \mathcal{T} \quad (3.3)$$

$$I_{jt}^C \leq I_j^{Max} \quad \forall j \in \mathcal{J}, \quad \forall t \in \mathcal{T} \quad (3.4)$$

$$q_t^P \leq Q \cdot y_t \quad \forall t \in \mathcal{T} \quad (3.5)$$

$$\sum_{j=1}^J q_{jkt}^C \leq W \cdot v_{kt} \quad \forall k \in \mathcal{K}, \quad \forall t \in \mathcal{T} \quad (3.6)$$

$$q_{jkt}^C \leq W \cdot x_{jkt} \quad \forall j \in \mathcal{J}, \quad \forall k \in \mathcal{K}, \quad \forall t \in \mathcal{T} \quad (3.7)$$

$$\sum_{k=1}^K x_{jkt} \leq 1 \quad \forall j \in \mathcal{J}, \quad \forall t \in \mathcal{T} \quad (3.8)$$

$$q_t^P, I_{j,t}^C, I_t^P, q_{jkt}^C \geq 0 \quad \forall j \in \mathcal{J}, \quad \forall k \in \mathcal{K}, \quad \forall t \in \mathcal{T} \quad (3.9)$$

$$y_t, x_{jkt}, v_{kt} \in \{0,1\} \quad \forall j \in \mathcal{J}, \quad \forall k \in \mathcal{K}, \quad \forall t \in \mathcal{T} \quad (3.10)$$

La fonction objectif (3.1) minimise le coût total composé des coûts suivants : le coût de setup de production, le coût de stockage au producteur, le coût de transport et le coût de stockage aux détaillants. Le coût du transport est composé du coût fixe d'utilisation des véhicules et du coût de service des détaillants. Les contraintes (3.2) et (3.3) sont, respectivement, des contraintes d'équilibrage du stock au producteur et aux détaillants. Les contraintes de respect des capacités de stockage, de production et des véhicules sont représentées, respectivement, par les inégalités (3.4), (3.5) et (3.6). Le but des contraintes (3.7) est de relier la variable continue  $q^C$  avec la variable binaire  $x$ . Les contraintes (3.8) assurent que le détaillant peut être visité au plus une fois par période. En fin, les contraintes (3.9) et (3.10) sont des contraintes de non négativité et d'intégrité.

### 3.3.2 Formulation à stock d'échelon ou *Echelon Stock Formulation* ESF

L'idée de l'*echelon stock*, introduite par Clark et Scarf [36], est souvent utilisée pour modéliser les problèmes de lot sizing multi-niveaux. Dans le problème OWMR, cette formulation est utilisée par Federgruen et Tzur [47] et Solyali et Süral [106]. Dans la formulation à stock échelon, la variable indiquant le niveau du stock au producteur est éliminée et remplacée par la variable suivante :

$$E_t^P = I_t^P + \sum_{j=1}^J I_{jt}^C$$

qui représente le stock total du produit dans la chaîne.

La formulation mathématique dans ce cas devient :

$$\text{Minimiser } \sum_{t=1}^T \left( S^P \cdot y_t + h^P \cdot E_t^P + \sum_{k=1}^K R \cdot v_{kt} + \sum_{j=1}^J \sum_{k=1}^K c_j^C \cdot x_{jkt} + \sum_{j=1}^J (h_j^C - h^P) \cdot I_{jt}^C \right) \quad (3.11)$$

Sous les contraintes (3.3)–(3.8) et :

$$E_t^P = E_{t-1}^P + q_t^P - \sum_{j=1}^J d_{jt} \quad \forall t \in \mathcal{T} \quad (3.12)$$

$$E_t^P \geq \sum_{j=1}^J I_{jt}^C \quad \forall t \in \mathcal{T} \quad (3.13)$$

$$q_t^P, I_{j,t}^C, E_t^P, q_{jkt}^C \geq 0 \quad \forall j \in \mathcal{J}, \quad \forall k \in \mathcal{K}, \quad \forall t \in \mathcal{T} \quad (3.14)$$

$$y_t, x_{jkt}, v_{kt} \in \{0,1\} \quad \forall j \in \mathcal{J}, \quad \forall k \in \mathcal{K}, \quad \forall t \in \mathcal{T} \quad (3.15)$$

Les contraintes (3.12) sont les nouvelles equations d'équilibrage du stock. Les contraintes (3.13) assurent que le stock échelon est supérieur ou égal à la somme des stocks au niveau des détaillants. Les contraintes (3.14) et (3.15) sont les contraintes de non-négativité et d'intégrité.

### 3.3.3 Formulation basée sur le problème de localisation ou *Facility Location Formulation* FLF

Cette formulation est donnée pour la première fois par Krarup et Bilde [69] pour modéliser le ULSP (uncapacitated lot sizing problem). Sa relaxation LP donne une solution optimale pour le ULSP. Elle est utilisée ensuite pour des extensions du ULSP. L'inconvénient de cette formulation est le grand nombre de variables.

Pour cette formulation, en plus des notations de la première formulation, on ajoute les notations suivantes :

$p = 1..T$  et  $e = 1..T$  : autres indices de périodes

$Z_{jkpet}$  : la variable qui signifie, la partie de la demande du client  $j$  de la période  $t$  produite à la période  $p$  et transportée à la période  $e$  par le véhicule  $k$ .

$H_{jpet}$  : coût de stockage d'une unité de la demande du client  $j$  de la période  $t$ , produite à la période  $p$ , transportée par le véhicule  $k$  à la période  $e$

Le problème est modélisé mathématiquement dans ce cas par :

$$\text{Minimiser } \sum_{t=1}^T \left( S^P \cdot y_t + \sum_{k=1}^K R \cdot v_{kt} + \sum_{j=1}^J \sum_{k=1}^K c_j^C \cdot x_{jkt} + \sum_{j=1}^J \sum_{k=1}^K \sum_{p=1}^e \sum_{e=p}^t H_{jpet} \cdot Z_{jkpet} \right) \quad (3.16)$$

Sous les contraintes :

$$\sum_{k=1}^K \sum_{p=1}^e \sum_{e=p}^t Z_{jkpet} = d_{jt} \quad \forall j \in \mathcal{J}, \quad \forall t \in \mathcal{T} \quad (3.17)$$

$$\sum_{j=1}^J \sum_{p=1}^e \sum_{t=e}^T Z_{jkpet} \leq W \cdot v_{ke} \quad \forall e \in \mathcal{T}, \quad \forall k \in \mathcal{K} \quad (3.18)$$

$$\sum_{t=e}^T \sum_{p=1}^e Z_{jkpet} \leq W \cdot x_{jke} \quad \forall j, \quad \forall e \in \mathcal{T}, \quad \forall k \in \mathcal{K} \quad (3.19)$$

$$\sum_{k=1}^K \sum_{t=s}^T \sum_{p=1}^e \sum_{e=p}^s Z_{jkpet} \leq I_j^{Max} \quad \forall j \in \mathcal{J}, \quad \forall s \in \mathcal{T} \quad (3.20)$$

$$\sum_{j=1}^J \sum_{k=1}^K \sum_{e=p}^t \sum_{t=e}^T Z_{jkpet} \leq Q \cdot y_t \quad \forall p \in \mathcal{T} \quad (3.21)$$

$$\sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^T x_{jkt} \leq 1 \quad \forall j \in \mathcal{J}, \quad \forall t \in \mathcal{T} \quad (3.22)$$

$$Z_{jkpet} \geq 0 \quad \forall j, \quad \forall k \in \mathcal{K}, \quad \forall p \in \mathcal{T} \quad \forall e \in \mathcal{T} \quad \forall t \in \mathcal{T} \quad (3.23)$$

$$y_t, \quad x_{jkt}, \quad v_{kt} \in \{0,1\} \quad \forall j \in \mathcal{J}, \quad \forall k \in \mathcal{K}, \quad \forall t \in \mathcal{T} \quad (3.24)$$

La fonction objectif (3.16) minimise toujours le coût total composé du coût de setup de production, le coût de stockage, le coût d'utilisation des véhicules, et le coût de service. Les contraintes (3.17) assurent la satisfaction de toutes les demandes. Les contraintes (3.18), (3.20), et (3.21) sont respectivement des contraintes de respect des capacités, des véhicules, de stockage et de production. Le lien entre les variables binaires  $x_{jkt}$  et les variables continues  $Z_{jkpet}$  est établie à l'aide des contraintes (3.19). Les contraintes (3.22) assurent qu'un véhicule au plus peut visiter un détaillant dans une période. Les contraintes (3.23) et (3.24) sont des contraintes de non négativité et d'intégrité.

### 3.3.4 Formulation basée sur le problème du plus court chemin ou *Shortest Path Formulation (SPF)*

Les nouvelles notations utilisées par cette formulation sont :

$$D_{jet} = \sum_{l=e}^t d_{jl} : \text{demande cumulative du détaillant } j \text{ de la période } e \text{ à la période } t$$

$U_{jket}$  : fraction de la demande du détaillant  $j$  de la période  $e$  à la période  $t$  transportée par le véhicule  $k$  dans la période  $e$ .

$$H_{jet} = \sum_{l=e}^t h_j^C D_{jle} : \text{est le coût de stockage de } D_{jet} \text{ transportée}$$

dans la période  $e$ .

La modélisation mathématique est donnée comme suit :

$$\text{Minimiser } \sum_{t=1}^T \left( S^P \cdot y_t + h^P \cdot I^P(t) + \sum_{k=1}^K R \cdot v_{kt} + \sum_{j=1}^J \sum_{k=1}^K c_j^C \cdot x_{jkt} + \sum_{j=1}^J \sum_{k=1}^K \sum_{e=1}^t H_{jet} \cdot U_{jket} \right) \quad (3.25)$$

Sous les contraintes :

$$I_t^P = I_{t-1}^P + q_t - \sum_{j=1}^J \sum_{k=1}^K \sum_{e=t}^T U_{jkte} \cdot D_{te} \quad \forall t \quad (3.26)$$

$$q_t^P \leq Q \cdot y_t \quad \forall t \quad (3.27)$$

$$\sum_{j=1}^J \sum_{e=t}^T U_{jkte} \cdot D_{te} \leq W \cdot v_{k,t} \quad \forall k, \forall t \quad (3.28)$$

$$\sum_{e=t}^T U_{jkte} \leq x_{jkt} \quad \forall j, \forall k, \forall t \quad (3.29)$$

$$\sum_{k=1}^K \sum_{e=t}^T U_{jkte} = \sum_{k=1}^K \sum_{e=1}^{t-1} U_{jket-1} \quad \forall j, \forall t \geq 2 \quad (3.30)$$

$$\sum_{k=1}^K \sum_{t=1}^T U_{jk1t} = 1 \quad \forall j \quad (3.31)$$

$$\sum_{k=1}^K \sum_{t=1}^s \sum_{e=s}^T U_{jkte} \leq I_j^{Max} \quad \forall j, \forall s \quad (3.32)$$

$$\sum_{k=1}^K x_{jkt} \leq 1 \quad \forall j, \forall t \quad (3.33)$$

$$U_{jket}, \quad q_t^P, \quad I_t^P \geq 0 \quad \forall j, \forall k, \forall e \quad \forall t \quad (3.34)$$

$$y_t, \quad x_{jkt}, \quad v_{kt} \in \{0, 1\} \quad \forall j, \forall k, \forall t \quad (3.35)$$

La fonction objectif (3.25) minimise le coût total de production, de transport et de stockage. L'équilibrage du stock au producteur est exprimé par les contraintes (3.26) . Les contraintes

(3.27), (3.28) et (3.32) sont respectivement des contraintes de respect des capacités des véhicules, de stockage au détaillants et de production. Les contraintes (3.31) indiquent que la somme des arcs sortant du premier nœud est égale à 1. Les contraintes (3.30) vérifient que la somme des arcs sortant égale à la somme des arcs entrants dans les autres nœuds. Les contraintes (3.29) sont utilisées pour relier la variable continue  $U$  à la variable binaire  $x$ . Les contraintes (3.33) assurent qu'un véhicule au plus peut visiter un détaillant dans une période. Les contraintes (3.34) et (3.35) donnent les domaines de définition des variables.

### 3.4 DÉVELOPPEMENT DES INÉGALITÉS VALIDES

Une façon communément utilisée pour renforcer les formulations est l'ajout des inégalités valides qui permettent d'améliorer les bornes par la réduction de l'espace des solutions de la relaxation linéaire de la formulation originale tout en n'éliminant aucune solution réalisable du problème entier.

Ayant modélisé notre problème avec les quatre formulations AGF, ESF, FLF et SPF, nous présentons dans cette section le développement des inégalités valides pour les deux formulations AGF et ESF. Le premier type d'inégalités valides s'intéressent au nombre maximum de véhicules à utiliser dans une période. La deuxième famille d'inégalités valides cherchent à éliminer la symétrie. Le troisième type des inégalités valides sont basées sur une propriété d'optimalité qui signifie qu'un véhicule plein peut visiter seulement des détaillants avec zero stock. Le quatrième type des inégalités valides sont inspirées des inégalités valides  $(l, S)$  connues pour le problème de lot sizing. La dernière famille d'inégalités valides visent à réduire le nombre de variables.

#### 3.4.1 Inégalités déterminant le nombre maximum de véhicules

Si nous supposons que dans une période  $t$  tous les détaillants sont visités, et que le stock initial de chaque détaillant est nul et le stock final est au maximum, le nombre maximum de véhicules nécessaires dans cette période est donné par les inégalités valides suivantes :

$$\sum_{k=1}^K v_{kt} \leq \lceil \sum_{j=1}^J (I_j^{max} + d_{jt}) / W \rceil \quad \forall t \in \mathcal{T} \quad (3.36)$$

où  $\lceil \cdot \rceil$  représente le plus petit entier supérieur.

On suppose que les stocks des détaillants sont vides au début de la période  $t$  et pleins à la fin. Alors la quantité maximale qu'on peut livrer dans la période  $t$  est égale à la somme des demandes de la période  $t$  de tout les détaillants plus le remplissage de leurs stocks au maximum.

L'inégalité (3.36) signifie que le nombre maximum de véhicules à utiliser, dans une période  $t$ ,

est égale à la quantité maximale qu'on peut livrer divisée par la capacité d'un véhicule. L'objectif de cette inégalité est de donner une borne supérieure à la variable  $v$ .

### 3.4.2 Inégalités éliminant la symétrie

L'objectif de ces inégalités est d'éviter la résolution des problèmes similaires dans l'arbre de recherche *branch and bound* due au fait que les véhicules sont identique (véhicules homogènes).

Nous supposons que les véhicules sont indexés de 1 à  $K$  et que le véhicule  $k$  doit être utilisé avant le véhicule  $k + 1$ . Ceci est donné par les inégalités suivantes :

$$v_{k+1,t} \leq v_{kt} \quad 1 \leq k \leq K-1 \quad \text{et} \quad \forall t \in \mathcal{T} \quad (3.37)$$

Cette inégalité signifie qu'on ne peut pas utiliser le véhicule  $k + 1$  si on n'a pas utilisé le véhicule  $k$ . Cela permet au solveur de ne pas passer d'une solution à une autre solution similaire en permettant  $k$  et  $k + 1$ .

### 3.4.3 Inégalités de véhicule plein

Nous introduisons une variable binaire  $z_{jt}$  qui est égale à 1 si le niveau du stock de détaillant  $j$  à la période  $t$  est strictement positif, et 0 sinon. Pour relier la variable  $z_{jt}$  à la variable  $I_{jt}^C$ , nous modifions la contrainte de capacité de stockage comme suit :

$$I_{jt}^C \leq I_j^{max} \cdot z_{jt}$$

**Hypothèse :** Nous supposons que le coût de stockage unitaire à chaque détaillant est supérieur au coût de stockage unitaire au producteur. Sous cette hypothèse, nous avons établi le théorème suivant :

**Théorème :** Si un détaillant est visité en période  $t$  et si son niveau de stock à la période  $t - 1$  est strictement positif, alors le véhicule visitant ce détaillant doit être plein au moment de son démarrage de l'usine de producteur.

Ceci donne l'inégalité valide suivante :

$$q_{kt}^V \geq W \cdot (x_{jkt} + z_{jt-1} - 1) \quad (3.38)$$

**Preuve :** Nous supposons qu'à la période  $t$ , un véhicule  $k$  visite plusieurs détaillants. Si tous les détaillants ont un niveau de stock zéro à  $t - 1$ , l'inégalité est vérifiée. Si un détaillant  $j$  a un niveau de stocks strictement positif à  $t - 1$ , alors  $x_{jkt} = 1$  et  $z_{jt} = 1$ , on aura trois cas possibles :

1. La quantité chargée dans ce véhicule  $q_{kt}=W$ , alors la propriété est vérifiée.
2. La quantité chargée est inférieure à  $W$ , et  $r = (W - q_{kt}) < I_{jt-1}$ , soit la capacité résiduelle dans le véhicule est inférieure au niveau du stock pour ce détaillant. Cette solution n'est pas optimale et la propriété est non vérifiée. Nous pouvons trouver une meilleure solution vérifiant cette propriété, en retardant  $r$  unités de la dernière livraison avant  $t$  pour être livrées à  $t$ . Ensuite, le niveau du stock pour le détaillant  $j$  à la période  $t - 1$  devient  $I_{jt-1}^* = I_{jt-1} - r > 0$ , et la quantité chargée dans ce véhicule devient  $q_{kt}^* = q_{kt} + r = W$ , ce qui vérifie la propriété. Cette solution est meilleure parce que les  $r$  unités dans la première solution sont stockées au détaillant dans la première solution, mais dans la deuxième solution, les  $r$  unités sont stockées au producteur. Ce qui diminue le coût de stockage et les autres coûts sont inchangés.
3. La quantité chargée est inférieure à  $W$ , et  $r = (W - q_{kt}) > I_{jt-1}$ , c.à.d. la capacité résiduelle dans le véhicule est plus grande que le niveau du stock de ce détaillant. Cette solution n'est pas optimale aussi et la propriété n'est pas vérifiée. Nous pouvons trouver une meilleure solution qui vérifie la propriété en retardant l'ensemble du stock  $I_{jt-1}$  de la dernière livraison avant  $t$  pour être livré à  $t$ . Ensuite, le niveau du stock pour le détaillant  $j$  à la période  $t - 1$  devient 0, et la quantité chargée dans ce véhicule devient  $q_{kt}^* = q_{kt} + I_{jt-1} < W$ , ce qui vérifie la propriété. Cette solution est meilleure car le stockage de  $s = I_{jt-1}$  unités dans la première solution est chez le détaillant, mais dans la deuxième solution, le stockage de  $s = I_{jt-1}$  unités est chez le producteur.

#### 3.4.4 Extension des inégalités valides $(l, S)$

Les inégalités valides  $(l, S)$ , développées par Barany et al. [14], donnent une description complète de l'enveloppe convexe du problème *Uncapacitated Lot Sizing Problem* (ULS). Elle sont définies comme suit :

$$\sum_{t \in S} x_t + \sum_{t \in \bar{S}} d_{tl} \cdot y_t \geq d_{1l}$$

où  $l \in \{1, 2, \dots, T\}$ ,  $S \subset \{1, 2, \dots, l\}$ ,  $\bar{S} = \{1, 2, \dots, l\} \setminus S$ , c.à.d  $\bar{S}$  est constitué de tous les éléments de  $\{1, 2, \dots, l\}$  n'appartenant pas à  $S$ , et

$$d_{tl} = \sum_{k=t}^l d_k$$

Pochet et Wolsey [86] ont montré que, si les coûts ont une structure de *Wagner-Whitin* (*WW*) (c.à.d  $p_t + h_t \geq p_{t+1}$  avec  $p_t$  et  $h_t$  les coûts unitaires de production et de stockage, respectivement,

dans la période  $t$ ), il est possible d'obtenir une solution optimale pour le problème ULS par l'ajout d'un sous ensemble des inégalités valides  $(l, S)$  à la relaxation linéaire du problème.

Comme le problème considéré dans ce travail est une extension du problème de lot sizing à un seul produit, nous pouvons faire une extension des inégalités valides  $(l, S)$  à notre problème comme suit :

On considère trois périodes  $u, t', l$ , tel que  $1 \leq u \leq t' \leq l \leq T$ . Si nous sommes les contraintes (3.2) sur  $t \in [u, t']$  et l'équation (3.3) pour chaque  $j \in J$  et pour chaque  $t \in [u, l]$  nous obtenons :

$$I_{t'}^P = I_{u-1}^P - \sum_{t=u}^{t'} \sum_{j=1}^J \sum_{k=1}^K q_{jkt}^C + \sum_{t=u}^{t'} q_t^P$$

et

$$\sum_{j=1}^J I_{jl}^C = \sum_{j=1}^J I_{ju-1}^C - \sum_{j=1}^J \sum_{t=u}^l d_{jt} + \sum_{j=1}^J \sum_{t=u}^l \sum_{k=1}^K q_{jkt}^C$$

et la somme, terme par terme, des deux équations nous donne :

$$I_{u-1}^P + \sum_{j=1}^J I_{ju-1}^C + \sum_{t=u}^{t'} q_t^P + \sum_{t=t'+1}^l \sum_{j=1}^J \sum_{k=1}^K q_{jkt}^C = \sum_{t=u}^l \sum_{j=1}^J d_{jt} + I_{t'}^P + \sum_{j=1}^J I_{jl}^C$$

Nous avons  $I_{t'}^P \geq 0$ ,  $I_{jl}^C \geq 0$ , et on se basant sur les contraintes (3.5) et (3.6), nous obtenons les inégalités suivantes :

$$I_{u-1}^P + \sum_{j=1}^J I_{ju-1}^C + Q \sum_{t'=u}^t y_{t'} + W \sum_{j=1}^J \sum_{t'=t+1}^l \sum_{k=1}^K x_{jkt} \geq \sum_{j=1}^J \sum_{t'=u}^l d_{jt'} \quad \forall u \leq t' \leq l \leq T \quad (3.39)$$

De la même manière, la sommation des contraintes (3.6) sur  $t \in [u, l]$ , nous donne les inégalités suivantes :

$$I_{ju-1}^C + W \sum_{t=u}^l \sum_{k=1}^K x_{jkt} \geq \sum_{t=u}^l d_{jt} \quad \forall u \leq l \leq T \quad \forall j \in \mathcal{J} \quad (3.40)$$

### 3.4.5 Inégalités valides réduisant le nombre de variables

Notez que, en raison de contraintes (3.8), au plus un véhicule  $k$  peut être affecté à un détaillant  $j$  dans une période. Il est toujours possible de construire une solution optimale dans laquelle, si le détaillant 1 est livré à la période  $t$ , alors il est toujours livré par le premier véhicule, c.à.d.  $\sum_{k=3}^K x_{2kt} = 0$ .

Suivant le même raisonnement, il existe une solution optimale dans laquelle le détaillant 2 est livré par le premier véhicule potentiellement avec le détaillant 1 ou sans le détaillant 1 par le deuxième véhicule, c.à.d.  $\sum_{k=3}^K x_{2kt} = 0$ . L'expression générale de la contrainte peut être donnée de la manière suivante :

$$\sum_{k=i+1}^K x_{ikt} = 0 \quad \forall i \leq K, \forall t$$

Cela contribue à réduire le nombre de variables binaires et donc la recherche dans l'espace de solution pour le solveur standard.

Il est possible de faire une extension des résultats précédents en considérant que, si le détaillant 1 n'est pas livré dans la période  $t$ , c.à.d.  $x_{11t} = 0$ , alors, il y a une solution optimale dans laquelle le détaillant 2 utilise le véhicule 1 s'il est livré à la période  $t$ , soit  $x_{22t} \leq x_{11t}$ , ce qui garantit que le détaillant 2 ne soit pas livré par le véhicule 2 si le détaillant 1 n'est pas livré par le véhicule 1.

L'inégalité valide ci-dessus pourrait être écrite différemment en considérant que, si aucun détaillant d'ordre inférieur à  $j$  n'a été livré par le premier véhicule à la période  $t$ , alors  $j$  ne peut pas être attribué au deuxième véhicule. C'est à dire :

$$x_{j2t} \leq \sum_{i=1}^{j-1} x_{i1t} \quad \forall j, \quad \forall t$$

Par conséquent, l'inégalité valide la plus générale devient :

$$\sum_{l=k}^K x_{jlt} \leq \sum_{i \in \mathcal{J}(j-1)} \sum_{l=1}^{\min(i,k-1)} x_{ilt} \quad \forall j, \quad \forall k \geq 2, \forall \mathcal{J}(j-1) \subseteq \{1, \dots, j-1\}$$

s.t.  $|\mathcal{J}(j-1)| = j - k + 1, \forall t$  (3.41)

où  $\mathcal{J}(j)$  désigne un sous-ensemble des détaillants  $\{1, \dots, j\}$ .

Pour  $k = 2$  :

$$\sum_{l=2}^K x_{jlt} \leq \sum_{i=1}^{j-1} x_{i1t} \quad \forall j, \quad \forall t \quad (3.42)$$

Pour  $k = 3$  :

$$\sum_{l=3}^K x_{jlt} \leq \sum_{i \in \mathcal{J}(j-1)} \sum_{l=1}^{\min(i,2)} x_{ilt} \quad \forall j, \quad \forall \mathcal{J}(j-1) \text{ s.t. } |\mathcal{J}(j-1)| = j - 2, \quad \forall t$$

Ce qui peut être écrit :

$$\sum_{l=3}^K x_{jlt} \leq \sum_{i=1, i \neq jj}^{j-1} \sum_{l=1}^{\min(i,2)} x_{ilt} \quad \forall j, \quad \forall jj = 1, \dots, j-1, \quad \forall t \quad (3.43)$$

Des expérimentations préliminaires ont montré que les contraintes (3.42) sont efficaces mais les contraintes (3.43) n'ont pas donné de bons résultats, elles ralentissent considérablement le solveur standard. Par conséquent, seules les contraintes (3.42) sont considérées dans la section 3.5.

### 3.5 RÉSULTATS NUMÉRIQUES

Toutes les instances générées sont résolues en utilisant le système de modélisation GAMS et le solveur IBM ILOG CPLEX 12.4, sur un PC avec CPU Core 2 Duo (2.93 GHz) et 3 GB RAM. La configuration par défaut d'IBM ILOG CPLEX n'est pas modifiée pendant les essais. L'exécution du programme est arrêtée lorsque l'optimum est atteint ou après une heure d'exécution.

Comme nous avons cité auparavant, il n'existe pas de travaux traitant un problème sous les mêmes hypothèses, par conséquent il n'existe pas d'instances dans la littérature qui correspondent à notre problème. Pour cela, nous avons choisi les instances du problème le plus proche du notre à savoir le PRP. Les instances utilisées par Adulyasak et al. [2] et Archetti et al. [9] pourraient être particulièrement pertinentes. Les tests numériques sont effectués sur des instances générées en faisant varier les paramètres suivants :

- Nombre de périodes ( $T$ ),
- Nombre de détaillants ( $J$ ),
- Nombre de véhicules ( $K$ ),
- Capacité de production ( $Q$ ),
- Capacité de chaque véhicule ( $W$ ).

Pour chaque paramètre, nous avons considéré les valeurs données dans le tableau 3.1. Nous obtenons au total 108 combinaisons groupées en trois familles selon le tableau (3.2). Pour chaque combinaison, cinq problèmes sont générés aléatoirement en choisissant les autres paramètres entre une valeur minimale  $s1$  et une valeur maximale  $s2$  suivants une loi de distribution uniforme ( $U(s1, s2)$ ) comme suit :

Demande de chaque détaillant  $j$  chez chaque période  $t$  :  $d_{jt} = U[5, 25]$ .

Coût de stockage unitaire chez chaque détaillant  $j$  :  $h_j^C = U[1, 5]$ .

Capacité de stockage à chaque détaillant  $j$  :  $I_j^{max} = \bar{d} \times U[2, 6]$  ; où  $\bar{d} = (\sum_{j=1}^J \sum_{t=1}^T d_{jt} / T \times J)$  est la demande moyenne.

Coût fixe de service de chaque détaillant  $j$  :  $C_j^C = U[100, 500]$ .

Coût de stockage unitaire au producteur :  $h^P = 1$ .

Coût de setup de production :  $S^P = 2000$ .

Coût fixe d'utilisation de chaque véhicule dans chaque période :  $R = 1000$ .

TABLE 3.1 – Paramètres des instances générées

Paramètre	Valeur possible
T	3, 6, 9
J	5, 10, 15, 20
K	2, 3, 4
Q	$2\alpha$ , $1.5\alpha$ ( $\alpha = \sum_{j=1}^J \sum_{t=1}^T d_{jt} / T$ )
W	$2\beta$ , $1.5\beta$ ( $\beta = \max_{t=1..T} (\sum_{j=1}^J d_{jt}) / K$ )

Un total de 108 combinaisons possibles sont examinées avec 5 problèmes générés pour chaque combinaison, ce qui donne un ensemble de 540 problèmes. Les deux formulations AGF et ESF, présentées dans les sections 3.3.1 et 3.3.2, sont comparées avec et sans inégalités valides.

### 3.5.1 Effet de chaque famille d'inégalités valides sur la formulation AGF

Les tableaux 3.3, 3.4 et 3.5 montrent l'impact de chaque famille d'inégalités valides sur la formulation agrégée AGF. Cette comparaison est basée sur le temps moyen CPU (Tableau 3.3), le Gap moyen quand la solution optimale n'est pas atteinte après une heure d'exécution (Tableau 3.4), et le pourcentage des instances résolues à l'optimalité en moins d'une heure de temps, par rapport au nombre total d'instances. Dans chaque tableau, la colonne (*No VI*) correspond au cas où aucune inégalité valide n'est ajoutée à la formulation. Les colonnes de (*VI 3.36*) à (*VI 3.42*) montrent les résultats obtenus avec l'ajout des inégalités valides (3.36) à (3.42) séparément. A la fin, la dernière colonne montre l'effet de l'ajout de toutes les inégalités valides en même temps.

Le tableau 3.3 montre que les meilleures inégalités valides sont les inégalités d'élimination de la symétrie (3.37) et les inégalités du véhicule plein (3.38) qui diminuent le CPU de 36% et 32% respectivement. La prise en compte de toutes les inégalités valides réduit encore le CPU de 56 %. La formulation est très sensible au nombre de périodes ( $T$ ) avec ou sans inégalités valides. Elle est également sensible à l'augmentation du nombre de véhicules, en particulier lors de l'ajout des inégalités (3.37) ou (3.38). En intégrant toutes les inégalités valides, le Gap moyen est réduit de 0,33 % à 0,03 %. Lorsque les inégalités valides sont utilisées individuellement, le meilleur Gap de 0,1 % est obtenu avec les inégalités valides (3.40). Le nombre de problèmes pour lesquels des solutions optimales sont trouvées en moins d'une heure augmente de plus de 22 % lorsque toutes les inégalités valides sont utilisées. Lors de l'ajout des inégalités valides individuellement, la meilleure augmentation est obtenue avec les inégalités valides (3.37), (3.38) et (3.42). Le tableau 3.11 donne les résultats détaillés sur les trois familles d'instances en utilisant la formulation AGF. Les colonnes (*CPU*), (*Gap (%)*) et (*#Opt*) représentent respectivement, le CPU, le Gap et le nombres de problèmes résolus à l'optimalité dans moins d'une heure, où toutes les inégalités sont ajoutées.

### 3.5.2 Effet de chaque famille d'inégalités valides sur la formulation ESF

Les tableaux 3.6, 3.7 et 3.8 récapitulent l'effet de chaque famille des inégalités valides sur la formulation ESF. Le tableau 3.7 montre que la famille des inégalités valides (3.40) a un effet

TABLE 3.2 – Les trois familles d'instances

Famille 1	T	J	K	C	W	Famille 2	T	J	K	C	W	Famille 3	T	J	K	C	W
1	3	5	2	$2\alpha$	$2\beta$	37	3	5	2	$1.5\alpha$	$2\beta$	73	3	5	2	$2\alpha$	$1.5\beta$
2	3	10	2	$2\alpha$	$2\beta$	38	3	10	2	$1.5\alpha$	$2\beta$	74	3	10	2	$2\alpha$	$1.5\beta$
3	3	15	2	$2\alpha$	$2\beta$	39	3	15	2	$1.5\alpha$	$2\beta$	75	75	15	2	$2\alpha$	$1.5\beta$
3	3	15	2	$2\alpha$	$2\beta$	40	3	15	2	$1.5\alpha$	$2\beta$	76	3	15	2	$2\alpha$	$1.5\beta$
5	6	5	2	$2\alpha$	$2\beta$	41	6	5	2	$1.5\alpha$	$2\beta$	77	6	5	2	$2\alpha$	$1.5\beta$
6	6	10	2	$2\alpha$	$2\beta$	42	6	10	2	$1.5\alpha$	$2\beta$	78	6	10	2	$2\alpha$	$1.5\beta$
7	6	15	2	$2\alpha$	$2\beta$	43	6	15	2	$1.5\alpha$	$2\beta$	79	6	15	2	$2\alpha$	$1.5\beta$
8	6	15	2	$2\alpha$	$2\beta$	44	6	15	2	$1.5\alpha$	$2\beta$	80	6	15	2	$2\alpha$	$1.5\beta$
9	9	5	2	$2\alpha$	$2\beta$	45	9	5	2	$1.5\alpha$	$2\beta$	81	9	5	2	$2\alpha$	$1.5\beta$
10	9	10	2	$2\alpha$	$2\beta$	46	9	10	2	$1.5\alpha$	$2\beta$	82	9	10	2	$2\alpha$	$1.5\beta$
11	9	15	2	$2\alpha$	$2\beta$	47	9	15	2	$1.5\alpha$	$2\beta$	83	9	15	2	$2\alpha$	$1.5\beta$
12	9	15	2	$2\alpha$	$2\beta$	48	9	15	2	$1.5\alpha$	$2\beta$	84	9	15	2	$2\alpha$	$1.5\beta$
13	3	5	3	$2\alpha$	$2\beta$	49	3	5	3	$1.5\alpha$	$2\beta$	85	3	5	3	$2\alpha$	$1.5\beta$
14	3	10	3	$2\alpha$	$2\beta$	50	3	10	3	$1.5\alpha$	$2\beta$	86	3	10	3	$2\alpha$	$1.5\beta$
15	3	15	3	$2\alpha$	$2\beta$	51	3	15	3	$1.5\alpha$	$2\beta$	87	3	15	3	$2\alpha$	$1.5\beta$
16	3	15	3	$2\alpha$	$2\beta$	52	3	15	3	$1.5\alpha$	$2\beta$	88	3	15	3	$2\alpha$	$1.5\beta$
17	6	5	3	$2\alpha$	$2\beta$	53	6	5	3	$1.5\alpha$	$2\beta$	89	6	5	3	$2\alpha$	$1.5\beta$
18	6	10	3	$2\alpha$	$2\beta$	54	6	10	3	$1.5\alpha$	$2\beta$	90	6	10	3	$2\alpha$	$1.5\beta$
19	6	15	3	$2\alpha$	$2\beta$	55	6	15	3	$1.5\alpha$	$2\beta$	91	6	15	3	$2\alpha$	$1.5\beta$
20	6	15	3	$2\alpha$	$2\beta$	56	6	15	3	$1.5\alpha$	$2\beta$	92	6	15	3	$2\alpha$	$1.5\beta$
21	9	5	3	$2\alpha$	$2\beta$	57	9	5	3	$1.5\alpha$	$2\beta$	93	9	5	3	$2\alpha$	$1.5\beta$
22	9	10	3	$2\alpha$	$2\beta$	58	9	10	3	$1.5\alpha$	$2\beta$	94	9	10	3	$2\alpha$	$1.5\beta$
23	9	15	3	$2\alpha$	$2\beta$	59	9	15	3	$1.5\alpha$	$2\beta$	95	9	15	3	$2\alpha$	$1.5\beta$
24	9	15	3	$2\alpha$	$2\beta$	60	9	15	3	$1.5\alpha$	$2\beta$	96	9	15	3	$2\alpha$	$1.5\beta$
25	3	5	4	$2\alpha$	$2\beta$	61	3	5	4	$1.5\alpha$	$2\beta$	97	3	5	4	$2\alpha$	$1.5\beta$
26	3	10	4	$2\alpha$	$2\beta$	62	3	10	4	$1.5\alpha$	$2\beta$	98	3	10	4	$2\alpha$	$1.5\beta$
27	3	15	4	$2\alpha$	$2\beta$	63	3	15	4	$1.5\alpha$	$2\beta$	99	3	15	4	$2\alpha$	$1.5\beta$
28	3	15	4	$2\alpha$	$2\beta$	64	3	15	4	$1.5\alpha$	$2\beta$	100	3	15	4	$2\alpha$	$1.5\beta$
29	6	5	4	$2\alpha$	$2\beta$	65	6	5	4	$1.5\alpha$	$2\beta$	101	6	5	4	$2\alpha$	$1.5\beta$
30	6	10	4	$2\alpha$	$2\beta$	66	6	10	4	$1.5\alpha$	$2\beta$	102	6	10	4	$2\alpha$	$1.5\beta$
31	6	15	4	$2\alpha$	$2\beta$	67	6	15	4	$1.5\alpha$	$2\beta$	103	6	15	4	$2\alpha$	$1.5\beta$
32	6	15	4	$2\alpha$	$2\beta$	68	6	15	4	$1.5\alpha$	$2\beta$	104	6	15	4	$2\alpha$	$1.5\beta$
33	9	5	4	$2\alpha$	$2\beta$	69	9	5	4	$1.5\alpha$	$2\beta$	105	9	5	4	$2\alpha$	$1.5\beta$
34	9	10	4	$2\alpha$	$2\beta$	70	9	10	4	$1.5\alpha$	$2\beta$	106	9	10	4	$2\alpha$	$1.5\beta$
35	9	15	4	$2\alpha$	$2\beta$	71	9	15	4	$1.5\alpha$	$2\beta$	107	9	15	4	$2\alpha$	$1.5\beta$
36	9	15	4	$2\alpha$	$2\beta$	72	9	15	4	$1.5\alpha$	$2\beta$	108	9	15	4	$2\alpha$	$1.5\beta$

TABLE 3.3 – CPU (en secondes) en utilisant la formulation AGF sur la première famille d'instances

AGF		CPU (Secondes)							
		No VI	VI 3.36	VI 3.37	VI 3.38	VI 3.39	VI 3.40	VI 3.42	All VI
T	3	1.35	1.16	0.67	0.79	1.53	2.03	1.11	0.49
	6	1089.33	1074.15	637.49	757.18	1195.37	1063.31	705.06	357.82
	9	2955.27	2871.68	1925.93	2021.43	2992.20	2311.81	2305.88	1423.96
J	5	829.16	769.90	443.69	438.10	853.35	550.59	652.70	159.96
	10	1068.18	1050.53	712.98	770.90	1158.80	990.91	826.49	519.71
	15	1647.03	1694.32	1136.58	1129.24	1719.72	1548.06	1186.35	804.80
	20	1850.24	1747.90	1125.54	1367.63	1853.60	1413.30	1350.54	891.89
K	2	583.43	534.94	48.05	114.50	598.24	175.13	149.37	17.90
	3	1594.84	1534.48	818.19	921.25	1585.74	1376.75	1079.01	509.65
	4	1867.68	1877.57	1697.85	1743.65	2005.13	1825.27	1783.67	1254.71
Moyenne		1348.65	1315.66	926.47	886.99	1396.37	1125.71	1004.02	594.09

TABLE 3.4 – *Le Gap obtenu par la formulation AGF sur la première famille d'instances*

AGF		Gap (%)							
		No VI	VI 3.36	VI 3.37	VI 3.38	VI 3.39	VI 3.40	VI 3.42	All VI
T	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	6	0.05	0.09	0.04	0.03	0.11	0.06	0.03	0.01
	9	0.93	0.84	0.42	0.42	1.00	0.24	0.71	0.09
J	5	0.27	0.25	0.07	0.07	0.30	0.05	0.17	0.00
	10	0.30	0.26	0.15	0.15	0.30	0.14	0.20	0.05
	15	0.33	0.33	0.15	0.15	0.40	0.10	0.29	0.04
	20	0.39	0.39	0.24	0.22	0.48	0.10	0.31	0.04
K	2	0.04	0.04	0.00	0.00	0.05	0.00	0.00	0.00
	3	0.32	0.24	0.09	0.06	0.36	0.07	0.18	0.01
	4	0.62	0.65	0.36	0.38	0.70	0.22	0.55	0.09
Moyenne		0.33	0.31	0.15	0.15	0.37	0.10	0.24	0.03

TABLE 3.5 – *Pourcentage des instances résolues à l'optimalité par la formulation AGF avec chaque inégalité valide sur la première famille d'instances*

AGF		Pourcentage des solutions optimales							
		No VI	VI 3.36	VI 3.37	VI 3.38	VI 3.39	VI 3.40	VI 3.42	All VI
T	3	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	6	80.00	78.33	85.00	88.33	71.67	80.00	88.33	95.00
	9	26.67	26.67	55.00	55.00	21.67	48.33	46.67	71.67
J	5	82.22	82.22	91.11	91.11	77.78	91.11	88.89	100.00
	10	75.56	77.78	82.22	86.67	71.11	77.78	82.22	88.89
	15	64.44	60.00	73.33	75.56	60.00	66.67	73.33	84.44
	20	53.33	53.33	73.33	71.11	48.89	68.89	68.89	82.22
K	2	90.00	90.00	100.00	100.00	88.33	100.00	100.00	100.00
	3	63.33	61.67	83.33	85.00	60.00	71.67	78.33	93.33
	4	53.33	53.33	56.67	58.33	45.00	56.67	56.67	73.33
Moyenne		68.89	68.33	81.11	79.44	64.44	76.11	78.33	88.89

important sur le CPU qui est diminué de 1040.38 secondes à 666.45 secondes, soit une réduction de plus de 35%. Le tableau 3.7 à son tour montre aussi que la formulation ESF est sensible à la longueur de l'horizon de planification et au nombre de détaillants. En ce qui concerne le nombre de véhicules, la formulation ESF est plus sensible lorsque les inégalités valides (3.38), (3.40) et (3.42) sont ajoutées.

TABLE 3.6 – CPU (en secondes) par l'utilisation de la formulation ESF sur la première famille d'instances

ESF		CPU (secondes)							
		No VI	VI 3.36	VI 3.37	VI 3.38	VI 3.39	VI 3.40	VI 3.42	All VI
T	3	1.44	1.18	0.77	0.86	1.21	0.52	0.81	0.38
	6	846.85	847.66	762.03	695.08	812.07	478.84	706.30	392.53
	9	2272.85	2272.22	2040.31	2130.47	2323.91	1519.99	2022.41	1351.48
J	5	564.15	565.14	508.33	573.29	592.87	398.27	534.63	170.05
	10	886.26	885.47	802.85	862.26	834.54	510.68	751.17	468.32
	15	1302.70	1275.55	1300.34	1076.96	1309.05	830.58	1087.54	813.35
	20	1408.40	1435.24	1125.95	1256.04	1446.47	926.28	1266.02	874.13
K	2	161.44	165.38	144.96	106.88	208.26	13.93	32.00	16.08
	3	1041.78	1037.64	886.63	924.04	1045.27	441.64	883.26	430.51
	4	1917.91	1918.04	1771.51	1795.50	1883.66	1543.79	1814.26	1297.80
Moyenne		1040.38	1040.35	934.37	942.14	1045.73	666.45	909.84	581.46

En termes de Gap moyen, le tableau 3.7 montre que les inégalités valides (3.40) ont un meilleur effet sur la formulation ESF que la formulation AGF, contrairement aux inégalités valides (3.37) et (3.38) qui ont une performance légèrement meilleure lorsqu'elles sont ajoutées à la formulation AGF.

TABLE 3.7 – Le Gap obtenu par l'utilisation de la formulation ESF sur la première famille d'instances

ESF		Gap (%)							
		No VI	VI 3.36	VI 3.37	VI 3.38	VI 3.39	VI 3.40	VI 3.42	All VI
T	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	6	0.05	0.05	0.04	0.06	0.08	0.02	0.03	0.02
	9	0.95	0.90	0.57	0.63	0.99	0.20	0.67	0.10
J	5	0.21	0.21	0.06	0.18	0.20	0.03	0.11	0.00
	10	0.27	0.26	0.19	0.29	0.28	0.06	0.17	0.04
	15	0.38	0.37	0.29	0.23	0.36	0.05	0.31	0.07
	20	0.47	0.41	0.26	0.22	0.58	0.15	0.34	0.04
K	2	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
	3	0.20	0.15	0.05	0.10	0.24	0.07	0.14	0.00
	4	0.80	0.79	0.55	0.59	0.81	0.14	0.57	0.11
Moyenne		0.33	0.31	0.20	0.23	0.36	0.07	0.23	0.04

Les inégalités (3.39) donnent presque le même Gap comme avec la formulation AGF. Toutefois, le tableau 3.8, montre que le pourcentage des problèmes résolus de façon optimale augmente considérablement en ajoutant ces inégalités et les meilleurs sont les inégalités valides (3.40). Le tableau 3.12 montrent des résultats détaillés sur toutes les familles d'instances en utilisant la formulation ESF. Les colonnes 6, 7 et 8 représentent respectivement, le CPU, le Gap et le

nombre de problèmes pour lesquels des solutions optimales ont été trouvées dans moins d'une heure, sans l'ajout des inégalités valides. Les colonnes 9, 10 et 11 représentent respectivement, le CPU, le Gap et le nombre de problèmes pour lesquels des solutions optimales ont été trouvées dans moins d'une heure, avec l'ajout des inégalités valides.

TABLE 3.8 – *Pourcentage des instances résolues à l'optimalité par la formulation ESF avec chaque inégalité valide sur la première famille d'instances*

ESF		Pourcentage des solutions optimales							
		No VI	VI 3.36	VI 3.37	VI 3.38	VI 3.39	VI 3.40	VI 3.42	All VI
T	3	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	6	81.67	85.00	86.67	86.67	88.33	91.67	86.67	91.67
	9	46.67	45.00	58.33	51.67	43.33	65.00	51.67	73.33
J	5	88.89	88.89	93.33	88.89	86.67	91.11	88.89	97.78
	10	82.22	82.22	82.22	80.00	84.44	91.11	86.67	93.33
	15	66.67	68.89	68.89	77.78	68.89	82.22	73.33	82.22
	20	66.67	66.67	82.22	71.11	68.89	77.78	68.89	80.00
K	2	100.00	98.33	100.00	100.00	98.33	100.00	100.00	100.00
	3	78.33	80.00	88.33	85.00	78.33	95.00	85.00	96.67
	4	50.00	51.67	56.67	53.33	55.00	61.67	53.33	68.33
Moyenne		76.11	76.67	81.67	79.44	77.22	85.56	79.44	88.33

### 3.5.3 Comparaison des formulations AGF et ESF

Dans cette section, les deux formulations sont comparées avec et sans inégalités valides. La comparaison est basée sur le CPU, le Gap moyen et le pourcentage des solutions optimales trouvées dans moins d'une heure du temps d'exécution.

Sans ajout des inégalités valides, les résultats dans le tableau 3.9 montre que la formulation ESF est meilleure en termes de CPU et le nombre de problèmes résolus optimalement, mais elle donne le même Gap moyen. La sensibilité aux paramètres du problème est presque la même avec les deux formulations. Cependant, le nombre de problèmes résolus avec AGF diminue plus rapidement quand la longueur de l'horizon augmente.

Lorsque toutes les inégalités valides sont ajoutées, les deux formulations deviennent très proches en terme de performances. Par exemple, les pourcentages de problèmes résolus à l'optimalité sont très proches (86% pour AGF contre 85% pour le ESF). Il est intéressant de noter que les inégalités valides ont un effet important sur les deux formulations et surtout sur la formulation AGF. Le CPU moyen a diminué de 38% pour la formulation ESF et de 50% pour la formulation AGF. Le pourcentage des problèmes résolus de façon optimale a augmenté d'environ 22% et 12% pour les formulations AGF et ESF, respectivement.

### 3.5.4 Evolution du Gap

Il est important d'analyser l'évolution du Gap en fonction du temps et d'identifier à quel moment commence la stagnation. La figure 4.7 représente l'évolution du Gap pour une instance

TABLE 3.9 – Comparaison des deux formulations sans inégalités valides sur les trois familles d'instances

No VI		CPU (secondes)		Gap (%)		% optimal	
		AGF	ESF	AGF	ESF	AGF	ESF
T	3	60.73	57.61	0.00	0.00	100.00	100.00
	6	1207.95	955.08	0.10	0.09	75.00	81.67
	9	2956.21	2376.12	0.90	0.94	25.00	43.89
J	5	839.98	613.87	0.20	0.14	81.48	86.67
	10	1219.15	962.36	0.32	0.32	71.85	80.74
	15	1682.19	1318.45	0.39	0.35	59.26	68.15
	20	1891.86	1623.72	0.40	0.55	53.33	64.44
K	2	593.39	214.99	0.04	0.01	88.89	97.22
	3	1623.53	1185.79	0.33	0.27	61.67	77.78
	4	2008.65	1988.03	0.63	0.76	48.89	50.00
Q	1.5	1303.58	709.90	0.27	0.21	68.89	87.04
	2	1461.00	1549.31	0.36	0.48	65.28	62.96
W	1.5	1572.66	1346.30	0.39	0.38	61.67	67.78
	2	1326.45	1021.25	0.30	0.32	68.89	78.61
Moyenne		1408.52	1129.60	0.33	0.34	66.48	75.00

TABLE 3.10 – Comparaison des deux formulations avec toutes les inégalités valides sur les trois familles d'instances

All VI		CPU (secondes)		Gap (%)		% optimal	
		AGF	ESF	AGF	ESF	AGF	ESF
T	3	26.51	35.15	0.00	0.00	100.00	100.00
	6	518.87	504.13	0.02	0.02	91.67	90.00
	9	1584.69	1584.87	0.20	0.16	66.67	65.00
J	5	148.22	185.41	0.00	0.00	100.00	98.52
	10	666.79	628.79	0.12	0.09	85.19	87.41
	15	899.00	873.32	0.09	0.07	80.74	80.00
	20	1126.08	1144.68	0.08	0.07	78.52	74.07
K	2	77.68	82.58	0.00	0.00	100.00	99.44
	3	632.53	586.99	0.04	0.03	90.56	90.00
	4	1419.86	1454.59	0.18	0.14	67.78	65.56
Q	1.5	523.22	517,54	0.04	0.03	88.89	88.89
	2	803.43	803,30	0.09	0.07	84.72	83.06
W	1.5	1012.76	1025.15	0.15	0.11	80.56	77.78
	2	558.65	549.50	0.03	0.04	88.89	88.61
Moyenne		710.02	708.05	0.07	0.06	86.11	85.00

typique entre 5 et 100 secondes. Elle montre que le solveur donne de très bons Gap dans les 30 premières secondes lorsque les inégalités valides sont ajoutés. Sans inégalités valides, le solveur commence avec un Gap élevé et son amélioration se fait lentement. Même après 100 secondes, le Gap reste encore plus élevé que celui obtenu avec les inégalités valides en 30 secondes.

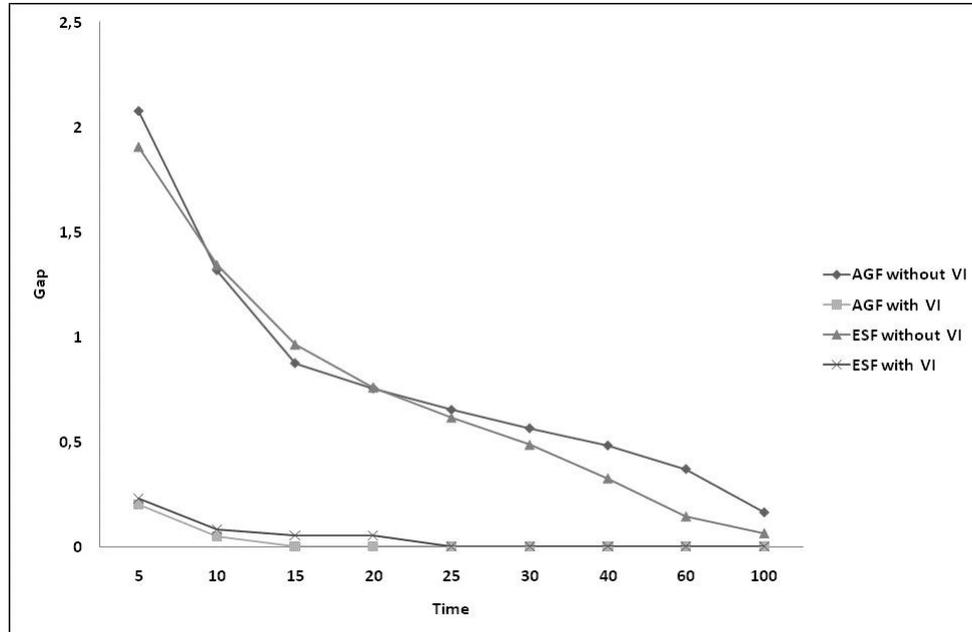


FIGURE 3.1 – Evolution du Gap entre 5 et 100 secondes pour l'instance  $6 \times 10 \times 3$

## CONCLUSION

Dans ce chapitre, nous avons présenté des formulations de programmation linéaire en nombres entiers, et nous avons proposé des inégalités valides pour résoudre le problème en utilisant un solveur standard. L'ajout des inégalités valides pour les formulations réduit considérablement le temps de calcul nécessaire pour la résolution des grandes instances du problème. En outre, ces inégalités ont aidé à combler l'écart entre la formulation agrégée AGF et la formulation à stock échelon ESF. Nous avons également analysé l'influence de la variation de chaque paramètre sur les deux formulations.

Les résultats de ce chapitre peuvent être adaptés pour la résolution d'un PIDP plus général. Les formulations et les inégalités valides peuvent être étendus à des problèmes avec multi-produits.

Malgré les résultats intéressants trouvés dans ce chapitre, cette approche ne sera pas adaptée pour les très grandes instances du problème. Dans les chapitres suivants, nous allons développer et tester différentes méta-heuristiques.

TABLE 3.11 – Résultats moyens sur les trois familles d'instances en utilisant la formulation AGF

					No VI			All VI		
T	J	K	$\alpha$	$\beta$	CPU	Gap	#Opt	CPU	Gap	#Opt
3	5	2	2	2	0.15	0	5	0.08	0	5
3	10	2	2	2	0.24	0	5	0.17	0	5
3	15	2	2	2	0.28	0	5	0.15	0	5
3	20	2	2	2	0.31	0	5	0.21	0	5
6	5	2	2	2	4.87	0	5	0.92	0	5
6	10	2	2	2	8.25	0	5	2.7	0	5
6	15	2	2	2	25.71	0	5	2.49	0	5
6	20	2	2	2	37.55	0	5	4.35	0	5
9	5	2	2	2	277.37	0	5	8.91	0	5
9	10	2	2	2	551.22	0	5	28.06	0	5
9	15	2	2	2	2976.48	0.22	2	68.21	0	5
9	20	2	2	2	3143.27	0.23	2	98.54	0	5
3	5	3	2	2	0.3	0	5	0.23	0	5
3	10	3	2	2	0.88	0	5	0.52	0	5
3	15	3	2	2	3.75	0	5	0.97	0	5
3	20	3	2	2	1.98	0	5	0.76	0	5
6	5	3	2	2	75.15	0	5	5.19	0	5
6	10	3	2	2	286.15	0	5	51.09	0	5
6	15	3	2	2	1988.89	0.02	4	91.74	0	5
6	20	3	2	2	2666.09	0.11	2	452.73	0	5
9	5	3	2	2	3314.9	0.63	2	197.31	0	5
9	10	3	2	2	3600	0.74	0	1360.41	0.01	4
9	15	3	2	2	3600	0.94	0	1319.18	0	5
9	20	3	2	2	3600	1.35	0	2635.7	0.1	2
3	5	4	2	2	0.54	0	5	0.31	0	5
3	10	4	2	2	1.28	0	5	0.53	0	5
3	15	4	2	2	3.58	0	5	0.83	0	5
3	20	4	2	2	2.96	0	5	1.13	0	5
6	5	4	2	2	189.13	0	5	15.56	0	5
6	10	4	2	2	1565.62	0.15	4	213.24	0	5
6	15	4	2	2	2649.1	0.17	3	2159.65	0.12	3
6	20	4	2	2	3600	0.12	0	1294.13	0.02	4
9	5	4	2	2	3600	1.83	0	1211.12	0	5
9	10	4	2	2	3600	1.85	0	3020.65	0.44	1
9	15	4	2	2	3600	1.66	0	3600	0.25	0
9	20	4	2	2	3600	1.68	0	3539.42	0.24	1
3	5	2	1.5	2	0.21	0	5	0.1	0	5
3	10	2	1.5	2	0.42	0	5	0.21	0	5
3	15	2	1.5	2	5.4	0	5	0.69	0	5
3	20	2	1.5	2	2.04	0	5	0.74	0	5
6	5	2	1.5	2	3.78	0	5	0.49	0	5
6	10	2	1.5	2	18.67	0	5	2.17	0	5
6	15	2	1.5	2	29.3	0	5	3.68	0	5
6	20	2	1.5	2	73.49	0	5	6.01	0	5
9	5	2	1.5	2	142.1	0	5	5.19	0	5
9	10	2	1.5	2	881.24	0	5	24.14	0	5
9	15	2	1.5	2	2675.99	0.13	2	70.02	0	5
9	20	2	1.5	2	3009.51	0.19	2	292.17	0	5
3	5	3	1.5	2	0.37	0	5	0.23	0	5
3	10	3	1.5	2	4.96	0	5	1.21	0	5
3	15	3	1.5	2	5.13	0	5	1.09	0	5
3	20	3	1.5	2	205.01	0	5	26.08	0	5
6	5	3	1.5	2	108.97	0	5	2.51	0	5
6	10	3	1.5	2	157.73	0	5	6.96	0	5
6	15	3	1.5	2	985.16	0.17	4	36.49	0	5
6	20	3	1.5	2	1977.81	0.01	4	38.11	0	5

(Suite Table3.11) : Résultats moyens sur les trois familles d'instances en utilisant la formulation AGF

					No VI			All VI		
T	J	K	$\alpha$	$\beta$	CPU	Gap	#Opt	CPU	Gap	#Opt
9	5	3	1.5	2	3087.6	0.37	1	66.58	0	5
9	10	3	1.5	2	3600	0.74	0	514.14	0	5
9	15	3	1.5	2	3600	0.96	0	601.55	0	5
9	20	3	1.5	2	3600	1.04	0	2777.53	0.05	2
3	5	4	1.5	2	0.46	0	5	0.25	0	5
3	10	4	1.5	2	6.68	0	5	4.08	0	5
3	15	4	1.5	2	38.84	0	5	6.11	0	5
3	20	4	1.5	2	67.14	0	5	10.24	0	5
6	5	4	1.5	2	324.44	0	5	11.09	0	5
6	10	4	1.5	2	1824.8	0.17	4	133.91	0	5
6	15	4	1.5	2	3083.32	0.23	1	313.08	0	5
6	20	4	1.5	2	3008.16	0.1	1	2305.78	0.04	3
9	5	4	1.5	2	3600	1.13	0	773.26	0	5
9	10	4	1.5	2	3600	1.43	0	3600	0.55	0
9	15	4	1.5	2	3600	1.83	0	3600	0.4	0
9	20	4	1.5	2	3600	1.34	0	3600	0.24	0
3	5	2	2	1.5	0.23	0	5	0.16	0	5
3	10	2	2	1.5	0.36	0	5	0.28	0	5
3	15	2	2	1.5	1.74	0	5	1.98	0	5
3	20	2	2	1.5	3.65	0	5	2.75	0	5
6	5	2	2	1.5	1.85	0	5	1.08	0	5
6	10	2	2	1.5	21.01	0	5	5.84	0	5
6	15	2	2	1.5	59.71	0	5	14.75	0	5
6	20	2	2	1.5	90.48	0	5	69.36	0	5
9	5	2	2	1.5	534.22	0	5	27	0	5
9	10	2	2	1.5	1193.84	0.04	4	87.87	0	5
9	15	2	2	1.5	1987.2	0.27	3	589.83	0	5
9	20	2	2	1.5	3600	0.27	0	1375.23	0	5
3	5	3	2	1.5	0.41	0	5	0.21	0	5
3	10	3	2	1.5	1.08	0	5	0.87	0	5
3	15	3	2	1.5	9.38	0	5	7.43	0	5
3	20	3	2	1.5	48	0	5	62.89	0	5
6	5	3	2	1.5	133.13	0	5	9.13	0	5
6	10	3	2	1.5	1139.14	0	5	73.08	0	5
6	15	3	2	1.5	3406.31	0.33	1	1306.17	0	5
6	20	3	2	1.5	2936.84	0.07	2	1004.24	0	5
9	5	3	2	1.5	3501.82	0.51	1	607.38	0	5
9	10	3	2	1.5	3600	1.13	0	2968.45	0.4	1
9	15	3	2	1.5	3600	1.13	0	3600	0.36	0
9	20	3	2	1.5	3600	1.5	0	2943.02	0.46	4
3	5	4	2	1.5	0.71	0	5	0.4	0	5
3	10	4	2	1.5	53.59	0	5	22.94	0	5
3	15	4	2	1.5	308.48	0	5	134.39	0	5
3	20	4	2	1.5	1405.83	0.03	4	663.16	0	5
6	5	4	2	1.5	229.98	0	5	19.19	0	5
6	10	4	2	1.5	3600	0.68	0	2279.87	0.02	4
6	15	4	2	1.5	3600	0.69	0	3142.5	0.29	1
6	20	4	2	1.5	3600	0.44	0	3600	0.23	0
9	5	4	2	1.5	3546.79	1.05	1	1038.03	0	5
9	10	4	2	1.5	3600	1.81	0	3600	1.78	0
9	15	4	2	1.5	3600	1.88	0	3600	1.07	0
9	20	4	2	1.5	3600	2.37	0	3600	0.72	0
Moyenne					1408.52	0.33	3.32	710.02	0.07	4.31

TABLE 3.12 – Résultats moyens sur les trois familles d'instances en utilisant la formulation ESF

					No VI			All VI		
T	J	K	$\alpha$	$\beta$	CPU	Gap	#Opt	CPU	Gap	#Opt
3	5	2	2	2	0.13	0	5	0.09	0	5
3	10	2	2	2	0.2	0	5	0.15	0	5
3	15	2	2	2	0.23	0	5	0.2	0	5
3	20	2	2	2	0.31	0	5	0.18	0	5
6	5	2	2	2	1.38	0	5	0.78	0	5
6	10	2	2	2	5.84	0	5	2.08	0	5
6	15	2	2	2	6.02	0	5	2.67	0	5
6	20	2	2	2	11.55	0	5	4.51	0	5
9	5	2	2	2	31.04	0	5	9.85	0	5
9	10	2	2	2	119.92	0	5	21.27	0	5
9	15	2	2	2	674.19	0	5	62.56	0	5
9	20	2	2	2	1086.49	0	5	88.61	0	5
3	5	3	2	2	0.2	0	5	0.13	0	5
3	10	3	2	2	0.68	0	5	0.35	0	5
3	15	3	2	2	1.7	0	5	0.76	0	5
3	20	3	2	2	2.13	0	5	0.55	0	5
6	5	3	2	2	21.61	0	5	3.83	0	5
6	10	3	2	2	53.22	0	5	27.54	0	5
6	15	3	2	2	238.35	0	5	62.93	0	5
6	20	3	2	2	1220.92	0.03	4	399.09	0	5
9	5	3	2	2	1293.23	0	5	157.85	0	5
9	10	3	2	2	2469.29	0.3	3	1131.84	0	5
9	15	3	2	2	3600	0.59	0	1124.8	0	5
9	20	3	2	2	3600	1.43	0	2256.45	0.05	3
3	5	4	2	2	0.58	0	5	0.16	0	5
3	10	4	2	2	1.12	0	5	0.4	0	5
3	15	4	2	2	3.84	0	5	0.72	0	5
3	20	4	2	2	6.14	0	5	0.84	0	5
6	5	4	2	2	129.16	0	5	16.05	0	5
6	10	4	2	2	1726.04	0.1	4	208.46	0	5
6	15	4	2	2	3600	0.39	0	2465.52	0.17	2
6	20	4	2	2	3148.05	0.12	1	1516.9	0.03	3
9	5	4	2	2	3600	1.9	0	1341.72	0.03	4
9	10	4	2	2	3600	2.03	0	2822.8	0.4	2
9	15	4	2	2	3600	2.43	0	3600	0.42	0
9	20	4	2	2	3600	2.68	0	3600	0.27	0
3	5	2	1.5	2	0.11	0	5	0.06	0	5
3	10	2	1.5	2	0.27	0	5	0.24	0	5
3	15	2	1.5	2	2.38	0	5	0.49	0	5
3	20	2	1.5	2	2.88	0	5	0.5	0	5
6	5	2	1.5	2	0.65	0	5	0.44	0	5
6	10	2	1.5	2	5.99	0	5	2.45	0	5
6	15	2	1.5	2	9.25	0	5	3.09	0	5
6	20	2	1.5	2	30.82	0	5	6.27	0	5
9	5	2	1.5	2	17.67	0	5	6.12	0	5
9	10	2	1.5	2	91.58	0	5	20.96	0	5
9	15	2	1.5	2	223.84	0	5	72.01	0	5
9	20	2	1.5	2	1197.95	0.03	4	518.89	0	5
3	5	3	1.5	2	0.16	0	5	0.11	0	5
3	10	3	1.5	2	2.05	0	5	0.67	0	5
3	15	3	1.5	2	1.61	0	5	1.49	0	5
3	20	3	1.5	2	34.78	0	5	3.87	0	5
6	5	3	1.5	2	11.57	0	5	2.31	0	5
6	10	3	1.5	2	45.35	0	5	6.09	0	5
6	15	3	1.5	2	392.43	0	5	26	0	5
6	20	3	1.5	2	1300.7	0	5	123.94	0	5

(Suite Table 3.12) : Résultats moyens sur les trois familles d'instances en utilisant la formulation ESF

					No VI			All VI		
T	J	K	$\alpha$	$\beta$	CPU	Gap	#Opt	CPU	Gap	#Opt
9	5	3	1.5	2	523.92	0	5	55.67	0	5
9	10	3	1.5	2	2564.36	0.21	4	350.83	0	5
9	15	3	1.5	2	3600	1	0	598.89	0	5
9	20	3	1.5	2	3600	1.9	4	2698.22	0.07	2
3	5	4	1.5	2	0.3	0	5	0.15	0	5
3	10	4	1.5	2	4.71	0	5	1.18	0	5
3	15	4	1.5	2	14.74	0	5	1.61	0	5
3	20	4	1.5	2	198.56	0	5	3.84	0	5
6	5	4	1.5	2	250.13	0	5	9.36	0	5
6	10	4	1.5	2	1530.98	0	5	92.19	0	5
6	15	4	1.5	2	2660.3	0.11	3	296.6	0	5
6	20	4	1.5	2	3356.74	0.18	1	2238.82	0.03	3
9	5	4	1.5	2	3600	0.78	0	688.24	0	5
9	10	4	1.5	2	3600	2.45	0	3600	0.55	0
9	15	4	1.5	2	3600	2.15	0	3600	0.34	0
9	20	4	1.5	2	3600	2.5	0	3600	0.22	0
3	5	2	2	1.5	0.18	0	5	0.13	0	5
3	10	2	2	1.5	0.26	0	5	0.25	0	5
3	15	2	2	1.5	1.63	0	5	1.35	0	5
3	20	2	2	1.5	5.51	0	5	1.4	0	5
6	5	2	2	1.5	2.18	0	5	0.78	0	5
6	10	2	2	1.5	8.88	0	5	4.57	0	5
6	15	2	2	1.5	24.39	0	5	12.45	0	5
6	20	2	2	1.5	67.81	0	5	25.3	0	5
9	5	2	2	1.5	82.79	0	5	35.06	0	5
9	10	2	2	1.5	179.15	0	5	62.06	0	5
9	15	2	2	1.5	1119.34	0.03	4	560.9	0	5
9	20	2	2	1.5	2726.86	0.12	2	1444.02	0.01	4
3	5	3	2	1.5	0.32	0	5	0.19	0	5
3	10	3	2	1.5	0.86	0	5	0.84	0	5
3	15	3	2	1.5	86.71	0	5	11.39	0	5
3	20	3	2	1.5	286.63	0	5	45.95	0	5
6	5	3	2	1.5	39.53	0	5	10.22	0	5
6	10	3	2	1.5	242.94	0	5	69.06	0	5
6	15	3	2	1.5	1170.01	0	5	658.43	0	5
6	20	3	2	1.5	2744.44	0.05	2	1099.44	0.01	4
9	5	3	2	1.5	2984.51	0.25	2	719.2	0	5
9	10	3	2	1.5	3354.08	0.82	1	2878.31	0.4	2
9	15	3	2	1.5	3600	1.28	0	3600	0.3	0
9	20	3	2	1.5	3600	1.83	0	3004.29	0.32	1
3	5	4	2	1.5	0.7	0	5	0.22	0	5
3	10	4	2	1.5	32.92	0	5	18.17	0	5
3	15	4	2	1.5	167.27	0	5	142.33	0	5
3	20	4	2	1.5	1211.23	0.08	4	1024.41	0	5
6	5	4	2	1.5	382.49	0	5	23.5	0	5
6	10	4	2	1.5	2743.09	0.24	2	2054.64	0.02	4
6	15	4	2	1.5	3600	1.32	0	3072.52	0.3	1
6	20	4	2	1.5	3600	0.56	0	3600	0.23	0
9	5	4	2	1.5	3600	0.97	0	1923.9	0.1	4
9	10	4	2	1.5	3600	2.59	0	3600	1.1	0
9	15	4	2	1.5	3600	0.28	0	3600	0.23	0
9	20	4	2	1.5	3600	3.4	0	3600	0.77	0
Moyenne					1129.6	0.34	3.75	708.05	0.05	4.25



# MÉTA-HEURISTIQUES À BASE D'ALGORITHME GÉNÉTIQUE

# 4

## RÉSUMÉ

*DANS ce chapitre nous présentons quatre approches à base d'algorithme génétique pour résoudre le même problème traité dans le chapitre 3. La première approche est une hybridation d'un algorithme génétique avec une recherche locale. Les trois autres approches sont une inter-opération entre l'algorithme génétique et la programmation linéaire en nombres entiers. Nous commençons d'abord par une description du principe générale d'un algorithme génétique et ces différentes phases. Nous expliquons ensuite la conception de chaque approche. Le chapitre se termine par une comparaison sur ces quatre approches.*



## 4.1 INTRODUCTION

Dans le chapitre précédent, nous avons développé une méthode exacte pour la résolution de notre problème. Les résultats expérimentaux montrent l'efficacité de la méthode sur des petites et moyennes instances. Nous rappelons que notre problème est NP-difficile, et la résolution de ce type de problèmes pour des instances larges, devient difficile ou impossible. Ce qui justifie notre recours au développement des méthodes approchées dans les chapitres suivants. Nous commençons dans ce chapitre par le développement de quatre méta-heuristiques à base d'algorithme génétique. Nous décrivons d'abord le principe des algorithmes génétiques et ses différentes phases avant de présenter chaque approche en détail. Dans la première approche, la démarche consiste à partir de composants de base de l'algorithme génétique à améliorer la solution par une méthode de recherche locale. Dans les trois autres approches, la démarche consiste à faire une inter-opération entre l'algorithme génétique et la programmation linéaire en nombres entiers. Le chapitre se termine par une comparaison entre ces différentes approches d'abord sur des petites et moyenne instances, ensuite sur des instances larges.

## 4.2 CONCEPTS DE BASE D'UN ALGORITHME GÉNÉTIQUE

Les algorithmes génétiques ou *Genetic Algorithms* (GA) sont des méthodes d'optimisation heuristiques robustes et efficaces pour la résolution des problèmes d'optimisation complexes. Leur succès pratique vient du fait qu'ils représentent des outils d'optimisation adaptés à des fonctions, des problèmes difficiles, complexes, irréguliers. Les algorithmes génétiques font partie de la famille des algorithmes évolutionnaires. Ils s'inspirent de l'évolution naturelle des espèces. Avec ce type de méthodes, il ne s'agit pas de trouver une solution exacte mais de trouver une bonne solution satisfaisante dans un temps de calcul raisonnable. La première description du processus des algorithmes génétiques a été donnée par Holland [60], puis Goldberg [56] les a utilisé pour résoudre des problèmes concrets d'optimisation.

Les algorithmes génétiques travaillent sur un ensemble de solutions candidates, appelé *population* d'individus ou chromosomes qui sont constitués d'un ensemble d'éléments, appelés *gènes*, qui peuvent prendre plusieurs valeurs, appelées *bits*. Un chromosome est une représentation ou un codage d'une solution du problème donné. Une première population est choisie soit aléatoirement, soit par des heuristiques ou par des méthodes spécifiques au problème, soit encore par mélange de solutions aléatoires et heuristiques. Cette population doit être suffisamment diversifiée pour que l'algorithme ne reste pas bloqué dans un optimum local. C'est ce qui se produit lorsque trop d'individus sont semblables. Les algorithmes génétiques génèrent de nouveaux individus, de telle sorte qu'ils soient plus performants que leurs prédécesseurs. Le processus d'amélioration des individus s'effectue par utilisation d'opérateurs génétiques, qui sont la sélection, le croisement et la mutation.

## Les étapes d'un algorithme génétique

Les grandes étapes d'un algorithme génétique sont présentées ci-dessous :

1. Représenter les solutions par un codage.
2. Générer la population initiale.
3. Evaluer le fitness de chaque chromosome dans la population.
4. Créer une nouvelle population par la répétition des opérations suivantes :
  - (a) Sélectionner les chromosomes de la population en fonction de leur fitness.
  - (b) Selon la probabilité de croisement, échanger des gènes entre deux chromosomes (parents) pour obtenir de nouveaux chromosomes (enfants).
  - (c) Selon la probabilité de mutation, modifier un ou plusieurs gènes de l'individu sélectionné.
5. Si le critère d'arrêt est atteint, arrêter l'algorithme et retourner la meilleure solution.
6. Aller à l'étape 3.

### 4.3 CONCEPTION DES APPROCHES À BASE D'ALGORITHME GÉNÉTIQUE

Malgré la simplicité apparente, concevoir un algorithme génétique efficace est une tâche difficile, car les algorithmes génétiques sont très sensibles aux choix des paramètres et à la représentation des solutions. L'expérience montre que les grandes réussites sont fondées sur une très bonne connaissance du problème à traiter, et sur une bonne compréhension des mécanismes évolutionnaires.

Dans cette section, nous évaluons quatre heuristiques à base d'algorithme génétique. Le point commun de toutes ces heuristiques est d'une part le codage d'un seul type de variables qui sont les setups de livraison des détaillants sous forme d'une matrice des binaires, et d'autre part, le mécanisme de reproduction d'une nouvelle population. Ce qui différencie ces heuristiques entre elles est la manière de déterminer les autres variables à partir de la matrice des setups de livraison.

Les nouvelles notations concernant les paramètres de ces heuristiques sont :

- $pop$  : population
- $P$  : taille de la population
- $p = 1..P$  : indice des individus dans la population
- $X_p$  : individu  $p$  de la population
- $x_{jt}$  : élément de la matrice  $X$  qui indique que si le détaillant  $j$  est livré dans la période  $t$  ou non
- $q_{kt}^V$  : la quantité à transporter par le véhicule  $k$  dans la période  $t$
- $q_{jt}$  : la quantité à livrer au détaillant  $j$  dans la période  $t$
- $QL_t$  : la quantité à livrer à tous les détaillants dans la période  $t$ .
- $nv_t$  : nombre de véhicules utilisés dans la période  $t$ .

- $\alpha_{jt}$  : est une période qui signifie que si le stock de détaillant  $j$  est plein à la fin de la période  $t$ , alors, la prochaine livraison peut se faire au plus tard à la période  $\alpha_{jt}$

### 4.3.1 Première approche (Hybridation AG et une recherche locale)

Au cours des dernières années, l'intérêt pour l'hybridation des approches a considérablement attiré l'attention des chercheurs pour résoudre les problèmes combinatoires. Les meilleurs résultats trouvés pour de nombreux problèmes d'optimisation au niveau académique ou pratique sont obtenus par des algorithmes hybrides. Par conséquent, plusieurs métaheuristiques à base de population (par exemple les algorithmes génétiques) sont hybridées avec des heuristiques de recherche locale. L'intérêt de l'hybridation est de combiner la diversification obtenue par les métaheuristiques à base de population et l'intensification obtenue par les heuristiques de recherche locale.

La première approche dans le cadre de notre travail a consisté à une hybridation entre les AG et une recherche locale. L'organigramme de cette première approche est donné par la figure 4.1. Les principales étapes seront décrites dans les sections suivantes.

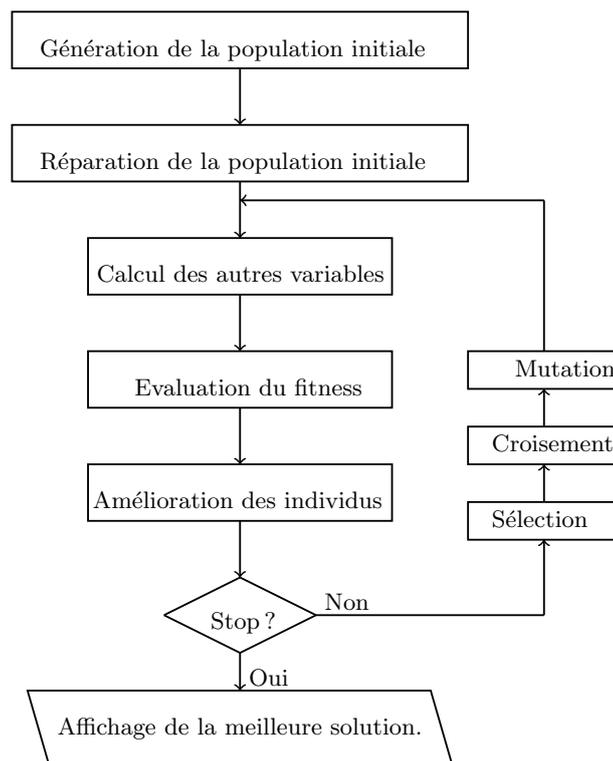


FIGURE 4.1 – Organigramme de la première approche

#### a) Représentation d'une solution

Pour concevoir un algorithme génétique, on doit d'abord donner un schéma de codage (représentation) de l'espace des variables de décision du problème. En effet un bon codage des solutions assure le succès des algorithmes génétiques. Dans le cas de notre problème, la solution

complète du problème est la détermination des quantités à produire dans chaque période et des quantités à livrer par chaque véhicule à chaque détaillant dans chaque période. L’idée principale dans ce cas, est le codage des variables de setups de livraison des détaillants sous forme d’une matrice bidimensionnelle  $X$  de dimension  $[J \times T]$  dont les éléments prennent les valeurs 0 ou 1. Ainsi chaque élément  $x_{jt}$  de la matrice indique si le détaillant  $j$  est livré dans la période  $t$  ou non. Ceci est la variable clé, car à partir d’elle nous pouvons déterminer les autres variables selon l’approche considérée.

**b) Génération de la population initiale**

Dans le cas de notre problème la population initiale est générée aléatoirement sous forme d’une matrice binaire de dimension  $(J \times T \times P)$ . La figure 4.3 est un exemple d’une population.

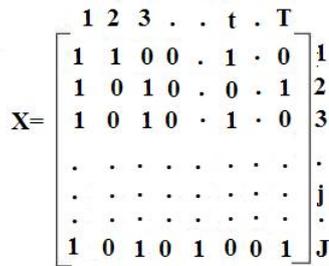


FIGURE 4.2 – Exemple d’un individu

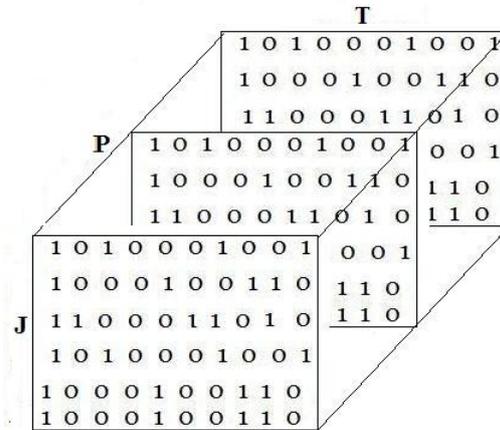


FIGURE 4.3 – Population

**c) Procédure de réparation**

Les chromosomes de la population initiale doivent satisfaire dans le cas de notre problème à trois types de contraintes, les contraintes de capacité de production, les contraintes de capacité de stockage aux détaillants et enfin les contraintes de capacité des véhicules. Ainsi les chromosomes générés aléatoirement ne sont pas forcément réalisables. Pour traiter l’infaisabilité, nous procédons comme suit :

Pour la violation de la capacité de production, nous introduisons dans la fonction objectif une pénalité de surproduction. Pour les deux autres contraintes, nous appliquons une procédure de réparation pour réparer chaque chromosome non faisable dans la population initiale selon l’algorithme 2.

Nous supposons que nous avons déjà calculé  $\alpha$  selon l’algorithme (1). Chaque ligne du chromosome qui correspond à un détaillant  $j$  est réalisable si une livraison est faite dans la période  $t$ , la prochaine livraison sera au plus tard dans la période  $\alpha_{jt}$ . Si il n’y a pas de livraison entre

**Algorithme 1** : Calcul de  $\alpha$ **Données** :  $T, N, d, W, I^{max}$ **Résultat** :  $\alpha$  $\alpha_{jt} \leftarrow T + 1$  (initialisation de  $\alpha$ )**pour**  $j = 1$  à  $N$  **faire**  **pour**  $t = 1$  à  $T - 1$  **faire**     $S \leftarrow 0$ ;     $t' \leftarrow t + 1$ ;    **tant que**  $S \leq \min(I_j^{max}, W)$  **et**  $t' \leq T - 1$  **faire**       $S \leftarrow S + d_{jt'}$ ;       $t' \leftarrow t' + 1$ ;    **fin**    **si**  $S > \min(I_j^{max}, W)$  **alors**       $\alpha_{jt} \leftarrow t'$     **finsi**  **fin****fin**

$t + 1$  et  $\alpha_{jt}$ , nous introduisons un 1 au hasard. Un exemple de fonctionnement de l'algorithme de réparation est présenté par la figure 4.4.

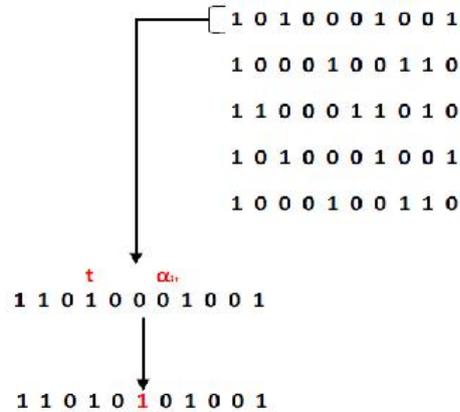


FIGURE 4.4 – Opération de réparation

**d) Détermination des autres variables**

On utilise la matrice  $X$  des setups de livraison aux détaillants, qui est dans notre cas l'individu ou le chromosome, pour déterminer les autres variables qui sont : les quantités à livrer, l'attribution de ces quantités aux véhicules, et les quantités de production, nous utilisons les étapes suivantes :

**Algorithme 2** : Réparation de la population initiale

---

**Données** :  $T, J, P, pop, \alpha$   
**Résultat** :  $pop$

```

pour  $p = 1$  à  $P$  faire
  pour  $j = 1$  à  $N$  faire
     $x \leftarrow pop(j, \cdot, p)$  ;
     $t1 \leftarrow 1$  ;
     $s \leftarrow 0$  ;
    tant que  $t1 < T$  et  $s = 0$  faire
      si  $\alpha_{jt1} = T + 1$  alors
         $s \leftarrow 1$ 
      finsi
       $t2 \leftarrow \alpha_{jt1}$  ;
      si  $x(t2) = 1$  alors
         $t1 \leftarrow \alpha_{jt2}$  ;
      sinon
        tant que  $t2 > t1$  et  $x(t2) = 0$  faire
           $t2 \leftarrow t2 - 1$  ;
          si  $t2 = t1$  alors
             $k$  : Une position aléatoire entre  $t1 + 1$  et  $\alpha_{jt1}$  ;
             $x(k) \leftarrow 1$  ;
             $pop(j, k, p) \leftarrow 1$  ;
             $t1 \leftarrow k$  ;
          finsi
          si  $x(t2) = 1$  alors
             $t1 \leftarrow t2$ 
          finsi
        fin
      finsi
    fin
  finsi
fin

```

---

*i. Détermination des quantités à livrer*

Dans cette étape, une fois que les périodes de livraison de chaque détaillant sont connues grâce à la matrice  $X$ , nous pouvons tout d'abord déterminer les quantités à livrer formant une matrice  $ql$ . Pour cela nous pouvons utiliser la propriété connue sous le nom de *zero stock* qui est spécifique au problème ULSP. Pour chaque  $j$ , si  $x_{jt} = 0$ , alors  $ql_{jt} = 0$ . Pour chaque  $t1$  et  $t2$ , si  $x_{jt1} = 1$ ,  $x_{jt2} = 1$ , et  $x_{jt} = 0 : t1 < t < t2$ , alors  $ql_{jt1} = \sum_{t=t1}^{t2-1} d_{jt}$ .

*ii. Allocation des quantités à livrer aux véhicules*

Après avoir déterminé les quantités à livrer à chaque détaillant, nous obtenons dans chaque période un problème de *bin packing*, c.à.d les quantités à livrer dans chaque période doivent être attribuées à un nombre minimum de véhicules. Ce problème fait partie de la classe des problèmes

NP-difficiles. Pour le résoudre, nous avons choisi une heuristique approximative appelée *Best Fit Decreasing* BFD. Cette heuristique est utilisée souvent dans la résolution du problème de *bin packing* et elle donne de bons résultats. Le nombre de bins ne dépasse jamais plus de  $11/9OPT + 1$  (où  $OPT$  est le nombre optimal de bins dans une solution optimale).

### iii. Détermination des quantités à produire

Une fois que nous avons les quantités à livrer à chaque période, le problème de production est un problème de lot sizing à capacité constante avec comme demandes, les quantités à livrer agrégées  $QL$ , qui peuvent être calculées comme suit :

$$QL_t = \sum_{j=1}^J ql_{jt} \quad \forall t \in \mathcal{T}$$

Pour vérifier si on peut résoudre le problème de production avec  $QL$ , nous procédons d'abord au lissage (*smoothing*) de  $QL$ . Nous notons  $SQ$ , après le lissage. Résoudre un problème de production avec  $SQ$  est équivalent à la résolution du problème avec  $QL$ . Le lissage se fait par l'algorithme 3.

Nous constatons que

$$\sum_{l=1}^t SQ_l \leq t \times Q \quad \forall t \in \mathcal{T} \quad (4.1)$$

Si la relation (4.1) n'est pas vérifiée pour certaines périodes, cela veut dire qu'on ne peut pas avoir un plan de production faisable en terme de capacité de production. Nous supposons dans ce cas, que la relation (4.1) n'est pas vérifiée jusqu'à une période que nous notons  $t_{inf}$ , c.à.d

$$\sum_{l=1}^{\tau} SQ_l > \tau \times Q \quad \forall \tau : 1 \leq \tau < t_{inf} \quad (4.2)$$

et

$$\sum_{l=1}^{\tau} SQ_l \leq \tau \times Q \quad \forall \tau : t_{inf} \leq \tau \leq T \quad (4.3)$$

Dans le cas de l'infaisabilité de production, nous déterminons les quantités à produire comme suit :  $q_t^P = SQ_t \quad \forall t : 1 \leq t < t_{inf}$  et les quantités  $q_t^P$ , pour  $t_{inf} \leq t \leq T$ , sont obtenus par la résolution d'un problème de lot sizing à capacité constante, en utilisant l'algorithme de programmation dynamique développé par Florian et Klein [48].

**Algorithme 3** : Lissage de  $QL$ 


---

**Données** :  $T, Q, QL$   
**Résultat** :  $SQ$   
 $SQ \leftarrow QL$ ;  
**pour**  $t = T$  **à** 1 **faire**  
     $t' \leftarrow t - 1$  ;  
     $exces \leftarrow QL_t - Q$ ;  
    **tant que**  $exces > 0$  **faire**  
        **tant que**  $SQ_{t'} \geq Q$  **et**  $t' > 0$  **faire**  
             $t' \leftarrow t' - 1$   
        **fin**  
         $liss \leftarrow \min(exces, Q - QL_{t'})$ ;  
         $SQ_t \leftarrow SQ_t - liss$ ;  
         $SQ_{t'} \leftarrow SQ_{t'} + liss$ ;  
    **fin**  
**fin**

---

Après avoir calculé les quantités à livrer, affecter ces quantités aux véhicules et déterminer les quantités de production, nous pouvons facilement calculer la fonction objectif d'un individu  $X$  de la population.

Si un individu  $X$  engendre une violation de la capacité de production, nous introduisons dans la fonction objectif de cet individu une pénalité  $\beta \times \varphi$ , où

$$\varphi = \sum_{l=1}^{t_{inf}} SQ_l - t_{inf} \times Q \quad (4.4)$$

désigne le dépassement de la capacité de production, et  $\beta$  un paramètre positif.

Si un individu  $X$  engendre, dans une période, un nombre de véhicules supérieur à la taille de la flotte, nous introduisons aussi dans la fonction objectif de  $X$  une pénalité  $\gamma \times \psi$ , où

$$\psi = \sum_{t=1}^T \max\{nv_t - K, 0\} \quad (4.5)$$

$K$  est la taille de la flotte dans chaque période,  $nv_t$  est le nombre de véhicules utilisés dans la période  $t$ , et  $\gamma$  un paramètre positif.

**e) Heuristiques d'amélioration**

Des améliorations peuvent être apportées aux solutions obtenues par l'application de deux heuristiques de recherche locale. La première heuristique que nous proposons est le *Backward*

*Shift*, et la deuxième est le *Forward Shift*. Dans l'heuristique de *Backward Shift*, pour chaque détaillant  $j$  et pour chaque période  $t$ , nous essayons de déplacer une partie ou la totalité  $ql_{jt}$  à la période  $t' < t$  afin de réduire le nombre de véhicules utilisés ou de réduire le nombre de setups de livraison des détaillants. Dans le *Forward Shift*, nous essayons de retarder la livraison d'une partie ou toute la quantité  $ql_{jt}$  de la période  $t$  à la période  $t' : t' > t$  en vue de minimiser le coût de stockage ou de réduire le nombre de véhicules utilisés.

#### i) Heuristique de *Backward Shift*

L'heuristique de *Backward Shift* consiste à anticiper une partie ou la totalité d'une valeur de la matrice  $ql$  de manière à réduire le nombre de véhicules utilisés et/ou à réduire le nombre de setups de livraisons (minimiser le coût total de service des détaillants). On suppose qu'on dispose de la matrice des quantités à livrer  $ql$ , et que les détaillants sont ordonnés selon le critère suivant :  $c_j^C/h_j^C$ . L'avantage de l'anticipation est la réduction du nombre de setup, et l'inconvénient est la création des stocks. Donc si on a la possibilité d'anticiper la livraison d'une quantité, il est préférable que cette quantité à livrer soit celle du détaillant qui a un coût de setup élevé et un coût de stockage réduit, c.à.d un détaillant qui a  $c_j^C/h_j^C$  élevé.

Pour chaque détaillant  $j$  et pour chaque période  $t$  telle que,  $1 < t \leq T$ , nous testons deux scénarios, dans le premier scénario, nous essayons de déplacer toute ou une partie de la valeur  $ql_{jt}$  vers une période antérieure  $t'$  dans laquelle il y a déjà une livraison au détaillant  $j$  à la période  $t'$ , c.à.d.  $x_{jt'} = 1$ . On suppose que  $r_{jtt'}$  est la quantité que nous pouvons déplacer de la période  $t$  à la période antérieure  $t'$ , alors les niveaux de stock  $I_{j\tau}^C$  pour chaque  $\tau = t', \dots, t-1$ , augmenteront d'une quantité  $r_{jtt'}$ . Pour déterminer  $r_{jtt'}$  nous devons s'assurer que les contraintes (3.3) et (3.7) sont satisfaites après un tel déplacement, cela veut dire que nous devons avoir :

$$r_{jtt'} = \min \left\{ \min_{\tau=t', \dots, t-1} \{ I_j^{max} - I_{j\tau}^C \}, W - ql_{jt'}, ql_{jt} - \sum_{\tau=t'+1}^{t-1} r_{jt\tau} \right\} \quad (4.6)$$

A chaque fois  $r_{jtt'}$  est généré, les valeurs de  $I_{j\tau}^C$ ,  $\tau = t', \dots, t-1$  doivent être mises à jour afin d'actualiser les nouvelles limites pour le prochain déplacement.

Nous devons éviter de faire le transfert vers les périodes  $t'$  telle que :  $t' < t_{inf}$ , afin de ne pas aggraver la violation de la capacité de production. Nous devons aussi éviter les périodes dans lesquelles le nombre de véhicules utilisés dépasse la taille de la flotte.

La variation dans le coût due au déplaçant de  $r_{jtt'}$  est déterminée en tenant compte de la variation du coût de stockage et de la variation des coûts fixes liés au nombre de véhicules utilisés. Cette variation peut être exprimée par la formule suivante :

$$f(r_{jtt'}) = R \times (nv_{t'} - nv_t) + R \times (nv_t' - nv_t) + (t - t') \times h_j^C \times r_{jtt'} \quad (4.7)$$

où  $nv'$  est le nouveau nombre de véhicules utilisés après le déplacement.

Après la détermination des quantités qui pourraient être déplacées, le changement dans le coût du premier scénario est  $Cost^{S1}$  qui peut être exprimée comme suit :

$$Cost^{S1} = \sum_{t'=1}^t f(r_{jt'}) - c_j^C \times (1 - x'_{jt}) \quad (4.8)$$

Si la totalité de  $ql_{jt}$  est déplacée, alors  $x'_{jt}$  égale à zero, sinon elle prend la valeur 1.

Dans le deuxième scénario, nous cherchons à déplacer la totalité  $ql_{jt}$  à une période antérieure  $t'$  dans laquelle il n'y a pas de livraison du détaillant  $j$  c.à.d ( $x_{jt'=0}$ ), afin de réduire le nombre de véhicules utilisés. Nous calculons le changement dans le coût que nous appelons  $Cost^{S2}$  dans ce cas.

$$Cost^{S2} = R \times (nv'_{t'} - nv_t) + R \times (nv'_t - nv_t) + (t - t') \times h_j^C \times ql_{jt} \quad (4.9)$$

Nous choisissons le meilleur scénario et nous effectuons toutes les modifications correspondantes. Un pseudo-code général de l'heuristique *Backward Shift* est donné dans l'algorithme 4.

### ii) Heuristique de *Forward Shift*

Dans l'heuristique de *Forward Shift*, nous cherchons à retarder la livraison d'une partie ou la totalité d'une quantité à livrer  $ql_{jt}$  de la période  $t$  à la période  $t' > t$  afin de minimiser le coût de stockage et de réduire le nombre de véhicules utilisés. Nous choisissons cette fois ci, les détaillants dans l'ordre croissant de  $c_j^C/h_j^C$ . Si  $r_{jt'}$  est la quantité à transférer de la période  $t$  à une période ultérieure  $t'$ , alors les niveaux des stocks  $I_{j\tau}^C$ ,  $\tau = t', \dots, t-1$ , seront diminués de la quantité  $r_{jt'}$ . Afin d'assurer la satisfaction des besoins nets de la période  $t$  à la période  $t'-1$  et de ne pas dépasser la capacité des véhicules, nous devons avoir,

$$r_{jt'} = \min \left\{ \min_{\tau=t', \dots, t-1} \{I_{j\tau}^C\}, W - ql_{jt'}, ql_{jt} \right\} \quad (4.10)$$

Nous devons également éviter les périodes où le nombre de véhicules utilisés dépasse la taille de la flotte des véhicules.

Le gain obtenu par le déplacement de  $r_{jt'}$  de  $t$  à  $t'$  se calcule par la formule suivante :

$$cost^{FS} = R \times (nv'_{t'} - nv_t) + R \times (nv'_t - nv_t) + (t - t') \times h_j^C \times r_{jt'} - c_j^C \times (1 - x'_{jt}) \quad (4.11)$$

où  $nv'$  et  $x'$  sont respectivement le nouveau nombre de véhicule utilisés et le nouveau setup

**Algorithme 4** : Heuristique de *Backward Shift*


---

**Données** :  $T, J, ql, I^C, X, c^C, h^C, R, Q, I^{max}$   
**Résultat** :  $ql, I^C, X$

**pour**  $j = 1$  à  $J$  **faire**  
  Trouver  $\Pi$  la liste de toutes les périodes de livraison de détaillant  $j$ ;  
  **tant que**  $\Pi \neq \emptyset$  **faire**  
     $t \leftarrow$  la dernière période dans  $\Pi$ ;  
     $qshift \leftarrow 0$ ;  
     $rshift \leftarrow ql_{jt}$ ;  
     $t' \leftarrow t - 1$ ;  
    **tant que**  $rshift > 0$  et  $t' \geq 1$  **faire**  
      Calculer  $r_{jt'}$ ;  
      Calculer  $f_{jt'}$ ;  
       $qshift \leftarrow qshift + r_{jt'}$ ;  
       $rshift \leftarrow rshift - r_{jt'}$ ;  
       $t' \leftarrow t' - 1$ ;  
    **fin**  
    Calculer  $Cost^{S1}$ ;  
    Trouver la dernière période possible de non livraison avant  $t$ ;  
    Transférer  $ql_{jt}$  à cette période;  
    Calculer  $Cost^{S2}$ ;  
    Choisir le meilleur scénario selon  $Cost^{S1}$  et  $Cost^{S2}$ ;  
    Faire les modifications correspondantes selon le scénario choisi ;  
    Eliminer  $t$  de la liste  $\Pi$ ;  
  **fin**  
**fin**

---

de livraison après le déplacement.

**f) Evaluation de Fitness**

Pour évaluer le fitness de chaque individu  $X$  de la population, nous utilisons la valeur de la fonction objectif. S'il y a une violation de la capacité de production ou une violation de la taille de la flotte des véhicules, nous introduisons dans la fonction objectif cet individu des pénalités correspondant à chaque violation. Une solution avec le coût minimum représente la meilleure solution avec une plus grande valeur de fitness. Après avoir évalué tous les individus, nous mettons à jour la meilleure solution.

**g) Selection**

La sélection permet d'identifier les meilleurs individus d'une population et d'éliminer les mauvais. Il existe plusieurs techniques de sélection, telles que la sélection uniforme, par tournoi, par élitisme, par roulette. La plus couramment utilisée est la sélection par roulette de *wheel*.

---

**Algorithme 5** : Heuristique de Forward Shift

---

**Données** :  $T, J, ql, I^C, X, c^C, h^C, R, Q, I^{max}$ **Résultat** :  $ql, I^C, X$ **pour**  $t = 2$  **à**  $T$  **faire**  **pour**  $t' = t + 1$  **à**  $T$  **faire**    Trouver  $\Omega$  la liste des détaillants livrés dans la période  $t'$ ;    **tant que**  $\Omega \neq \emptyset$  **faire**      Choisir  $j$  de  $\Omega$  ;      Calculer  $r_{jtt'}$  ;      **si**  $r_{jtt'} > 0$  **alors**        Calculer  $cost^{FS}$  ;        **si**  $cost^{FS} < 0$  **alors**

Faire les modifications correspondantes;

**finsi**      **finsi**    **fin**  **fin****fin**

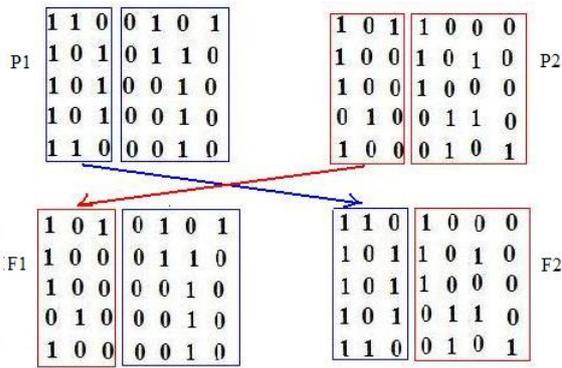
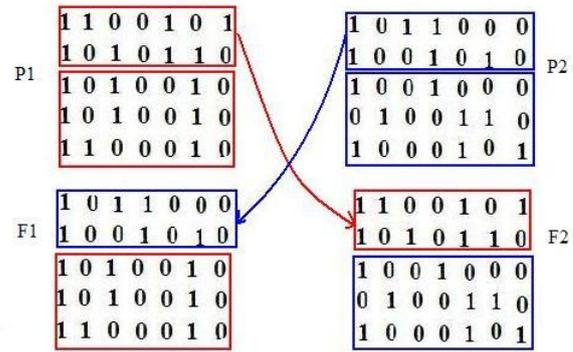
---

Nous avons choisi cette dernière. Elle consiste à associer à chaque individu un segment dont la longueur est proportionnelle à son fitness. Ces segments sont ensuite concaténés sur un axe que l'on normalise entre 0 et 1. On tire alors un nombre aléatoire de distribution uniforme entre 0 et 1, puis on fixe le segment sélectionné. Avec ce système, les segments sélectionnés, c'est-à-dire les bons individus, seront souvent choisis [56].

**h) Croisement**

L'opérateur de croisement recombine les gènes d'individus existant dans la population. Il est appliqué à deux parents avec une probabilité de croisement et génère deux enfants. Le croisement peut être simple, à un seul point, ou multiples, à deux ou plusieurs points.

Dans ce cas, nous avons choisi le croisement à un point qui est généré de façon aléatoire. Le découpage de la matrice peut être horizontal ou vertical. Donc le croisement horizontal (figure 4.6) est le croisement relatif aux détaillants et le croisement vertical (figure 4.5) aux croisement des périodes. Le premier enfant  $F1$  reçoit la première partie du premier parent  $P1$  et la deuxième partie du deuxième parent  $P2$ . Par contre, le deuxième enfant  $F2$  reçoit la deuxième partie du premier parent  $P1$  et la première partie du deuxième parent  $P2$ . Nous pouvons dans certains cas obtenir des enfants non faisables. Dans le *croisement des périodes*, la faisabilité peut être détruite localement au voisinage du point de croisement, alors l'algorithme de réparation doit être appliqué uniquement autour du point de croisement. Dans le *croisement des détaillants*, la faisabilité peut être détruite à tout point, alors la vérification de la faisabilité doit être globale.

FIGURE 4.5 – *Croisement vertical*FIGURE 4.6 – *Croisement horizontal*

### i) Mutation

La mutation est utilisée pour produire de petites perturbations sur les individus pour promouvoir la diversité de la population. Il existe plusieurs opérateurs de mutation, on peut citer les opérateurs suivants : changer la valeur d'une variable, choisir aléatoirement deux variables et les échanger ou perturber certaines valeurs en ajoutant du bruit de façon aléatoire.

Dans notre cas, une simple inversion des bits est utilisée comme opérateur de mutation. Chaque élément  $x_{jt}$  choisi est inversé en fonction de la probabilité de mutation. La mutation peut affecter la faisabilité d'un individu. L'algorithme de réparation est appliquée localement comme dans le cas de *croisement des périodes*.

### 4.3.2 Deuxième approche

Le principe de cette approche est de faire une inter-opération entre l'algorithme génétique et la programmation linéaire en nombres entiers PLNE. L'algorithme génétique génère à chaque itération, une population suivant les étapes décrites dans la première approche, cette dernière sera envoyée à un solveur standard pour déterminer les autres variables d'une manière exacte. Pour chaque individu  $X$  dans la population, le solveur détermine le reste des variables de décision par la résolution du sous-problème ( $SP1$ ) qui est dérivé de la formulation originale, où la contrainte (3.8) qui assure que le détaillant peut être visité au plus une fois par période, est remplacée par la contrainte (4.13), et la matrice  $X$  est utilisée comme paramètre (chaque élément  $x_{jt}$  de la matrice  $X$  désigne si un détaillant  $j$  est livré dans une période  $t$  ou non). Nous introduisons dans ce cas une autre variable binaire  $z_{jkt}$  qui est égale à 1 si le détaillant  $j$  est livré par le véhicule  $k$  dans la période  $t$ , 0 sinon. Son rôle est la détermination de l'affectation des détaillants aux véhicules. Elle est liée à  $x_{jt}$  (qui est considéré comme un paramètre dans le sous-problème  $SP1$ ) par la contrainte (4.13).

Le sous-problème ( $SP1$ ) est défini comme suit :

$$(SP1) \quad \text{Min} \quad \sum_{t=1}^T \left( S^P .y_t + h^P .I_t^P + \sum_{k=1}^K R.v_{kt} + \sum_{j=1}^J \sum_{k=1}^K c_j^C .z_{jkt} + \sum_{j=1}^J h_j^C .I_{jt}^C \right) \quad (4.12)$$

sous :

les contraintes (3.2)-(3.7), (3.9), (3.10) et la contrainte suivante :

$$\sum_{k=1}^K z_{jkt} = x_{jt} \quad \forall j \in \mathcal{J}, \quad \forall t \in \mathcal{T} \quad (4.13)$$

La contrainte (4.13) signifie que les périodes de livraison sont fixes.

A chaque itération, la solution obtenue est à son tour envoyée à l'algorithme génétique qui l'utilise pour calculer le fitness de la population actuelle. Par l'application des opérateurs génétiques (sélection, croisement et mutation), l'algorithme génétique génère une nouvelle population qui sera envoyée à nouveau au solveur standard. Ce processus est répété jusqu'à ce que le critère d'arrêt est atteint. Le temps d'exécution de solveur est limité, c.à.d le solveur doit retourner la solution trouvée après un temps fixe, même si elle n'est pas optimale.

### 4.3.3 Troisième approche

L'idée de la troisième approche est de permettre au solveur d'annuler certaines périodes de livraison proposées par l'AG, c.à.d éliminer des 1 dans la matrice  $X$  mais pas d'ajouter d'autres. Cela est due au fait que l'algorithme de réparation, qui est basé sur l'ajout des 1, peut engendrer des individus qui contiennent un nombre des 1 élevé.

Le principe de cette approche est le même que pour la seconde, sauf que la contrainte (3.8), qui assure qu'un détaillant peut être visité au plus une fois par période, est remplacée par la contrainte (4.15). Pour déterminer les autres variables le solveur résout le sous-problème (SP2) formulé comme suit :

$$(SP2) \quad \text{Min} \quad \sum_{t=1}^T \left( S^P .y_t + h^P .I_t^P + \sum_{k=1}^K R.v_{kt} + \sum_{j=1}^J \sum_{k=1}^K c_j^C .z_{jkt} + \sum_{j=1}^J h_j^C .I_{jt}^C \right) \quad (4.14)$$

Sous les contraintes :

(3.2)-(3.7), (3.9), (3.10) et la contrainte suivante :

$$\sum_{k=1}^K z_{jkt} \leq x_{jt} \quad \forall j \in \mathcal{J}, \quad \forall t \in \mathcal{T} \quad (4.15)$$

La contrainte (4.15) signifie que le nombre de setup de livraison peut être diminué, mais ne

peut pas être augmenté.

Après la résolution du sous-problème *SP2*, on fait une actualisation de la matrice  $X$  de la manière suivante :

$$\text{Si } \sum_{k=1}^K z_{jkt} = 0 \text{ alors } x_{jt} = 0.$$

Les nouvelles valeurs dans la matrice  $X$  ainsi que les valeurs des autres variables déterminées par le solveur sont utilisées par l'AG pour évaluer le fitness et par la suite la production d'une nouvelle population en utilisant les opérateurs génétiques.

#### 4.3.4 Quatrième approche

Dans les deux approches précédentes, il existe une variable binaire  $z_{jkt}$ , ceci rend la résolution des sous-problèmes difficile si la taille du problème augmente. L'idée de la quatrième approche est de réduire la taille du sous-problème à résoudre par le solveur.

Dans cette approche, le sous-problème (*SP3*) à résoudre par le solveur contient peut de variables binaires. Cette fois ci, le solveur détermine uniquement les quantités à produire et les quantités à livrer à chaque détaillant dans chaque période. Ensuite, on utilise l'heuristique BFD, pour attribuer les quantités à livrer aux véhicules. Après le calcul du fitness, une nouvelle population est produite par les opérateurs génétiques. Le sous-problème (*SP3*) à résoudre dans cette approche contient deux types de variables binaires :  $y_t$  et  $z_{jt}$ . Une nouvelle variable entière  $mv_t$  est introduite pour désigner le nombre minimum de véhicules à utiliser dans la période  $t$ . Le but de l'introduction de cette nouvelle variable est de forcer le solveur à réduire le nombre de véhicules. Les  $x_{jt}$  sont générés par l'algorithme génétique et sont considérés, dans le modèle (*SP3*), comme paramètres d'entrée.

$$(SP3) \quad \text{Min} \quad \sum_{t=1}^T \left( S^p \cdot y_t + h^p \cdot I_t^p + R \cdot mv_t + \sum_{j=1}^J c_j^C \cdot z_{jt} + \sum_{j=1}^J h_j^C \cdot I_{jt}^C \right) \quad (4.16)$$

Sous les contraintes :

(3.2)-(3.5) et la contrainte suivante :

$$\sum_{j=1}^J q_{jt}^c \leq W \cdot mv_t \quad \forall t \in \mathcal{T} \quad (4.17)$$

$$ql_{jt} \leq W \cdot z_{jt} \quad \forall j \in \mathcal{J}, \quad \forall t \in \mathcal{T} \quad (4.18)$$

$$z_{jt} \leq x_{jt} \quad \forall j \in \mathcal{J}, \quad \forall t \in \mathcal{T} \quad (4.19)$$

$$q_t^P, I_{jt}^C, I_t^P, ql_{jt}, mv_t \geq 0 \quad \forall j \in \mathcal{J}, \quad \forall t \in \mathcal{T} \quad (4.20)$$

$$y_t, z_{jt} \in \{0,1\}, mv_t \in \mathbb{N} \quad \forall j \in \mathcal{J}, \quad \forall t \in \mathcal{T} \quad (4.21)$$

La contrainte (4.17) détermine le nombre minimum de véhicules à utiliser dans la période  $t$ . Le rôle de la contrainte (4.18) est de relier la variable binaire  $z$  à la variable continue  $ql$ . La contrainte (4.19) montre que le solveur peut réduire le nombre de setup de livraison.

#### 4.4 RÉSULTATS EXPÉRIMENTAUX

Dans cette section, nous présentons les résultats expérimentaux issus de l'application des quatre approches à notre problème. Les algorithmes sont implémentés en *MATLAB* et les programmes *PLNE* sont résolues par *GAMS – CPLEX*. Tous les tests ont été effectués sur un ordinateur personnel équipé d'un processeur *Core 2 Duo (2.93 GHz) et 3 GB RAM*. Nous avons utilisé deux classes d'instances. La première classe est la première famille d'instances utilisées dans le troisième chapitre (voir tableau (3.2)). Elles sont considérées comme des petites et moyennes instances. La deuxième classe représente les instances qui sont considérées comme larges instances où le nombre de périodes  $T \in \{20, 30, 40\}$ , le nombre de clients  $N \in \{40, 60, 80\}$  et le nombre de véhicules  $K \in \{4, 6, 8\}$ , ce qui représente 27 instances. Les autres paramètres sont générés entre une valeur minimale  $s1$  et une valeur maximale  $s2$  suivant une loi de distribution uniforme ( $U(s1, s2)$ ) comme dans la section 3.5.

Après plusieurs tests préliminaires, les valeurs suivantes ont été choisies empiriquement comme paramètres de notre algorithme génétique : La taille de la population  $popsiz = 30$  pour les petites et moyennes instances et  $popsiz = 20$  pour les instances larges. La probabilité de croisement est égale à 0,80 et la probabilité de mutation est égale à 0,1 pour toutes les instances.

##### 4.4.1 Résultats expérimentaux sur les petites et moyennes instances

Le but de ces expérimentations est de tester l'efficacité des approches par rapport à l'optimum obtenu par *CPLEX* dans le chapitre 3. Nous utilisons une seule exécution pour chaque instance. La qualité de la solution trouvée par chaque approche, est évaluée en calculant le *gap* en utilisant la formule suivante :

$$gap = 100 \times (UB - OP) / UB,$$

où  $UB$  et  $OP$  représentent respectivement la meilleure borne trouvée et la valeur optimale de l'objectif trouvée par *CPLEX*. Nous avons fixé le temps d'exécution de chaque approche à 300 secondes et nous enregistrons le *gap* à chaque 50 secondes. Les tableaux (4.1, 4.2, 4.3, 4.4 et

4.5) récapitulent le *gap* résultant.

A partir des résultats obtenus on remarque l'avantage de l'heuristique d'amélioration sur la performance de la première approche. Le tableau (4.1) montre le *gap* de la première approche sans utiliser la procédure d'amélioration. Nous constatons que sur 36 instances, 4 sont résolues à l'optimalité et le *gap* moyen après 300 secondes est de 4.32. Le *gap* de certaines instances est plus de 9.

D'après les résultats présentés dans le tableau 4.2, il est clair que la procédure d'amélioration a un effet très significatif sur la réduction du *gap* et par conséquent sur le nombre d'instances résolues à l'optimalité, où on trouve 13 instances sur 36. Le *gap* moyen a diminué de 4.32% à 0.73% à 300 secondes. Il est déjà à 1.20% après 50 secondes. Le *gap* maximum ne dépasse pas 2% pour la majorité des instances et on trouve seulement 3 instances avec un *gap* entre 2% et 2.64%.

Le tableau 4.3, présente les résultats de la deuxième approche. Le nombre d'instances résolues à l'optimalité dépasse celui de la première (15 instances sur 36), mais le *gap* moyen est un peu élevé (0.96% dans ce cas). Nous constatons que cette approche est bonne pour les petites instances, où le nombre de périodes est petit ( $T = 3$  et  $T = 6$ ), par contre, si le nombre de périodes  $T = 9$ , le *gap* devient un peu élevé, il dépasse 2% pour la plupart des cas. Cela peut être expliqué par le fait que le sous problème à résoudre par CPLEX devient compliqué avec  $T = 9$ .

Les résultats de la troisième approche (tableau 4.4), sont légèrement meilleurs par rapport à la première approche en terme du *gap* moyen. Le nombre d'instances résolues à l'optimalité est le même (15 instances sur 36), et aucune valeur du *gap* n'a dépassé 2.92%.

La quatrième approche donne des résultats acceptables en terme de nombre d'instances résolues à l'optimalité (12 instances sur 36), mais elle n'est pas bonne par rapport aux autres approches, malgré le fait que le sous problème à résoudre par CPLEX dans ce cas n'est pas compliqué.

La figure 4.7 illustre l'évolution du *gap* moyen des 36 instances, pour les quatre approches, plus la première approche sans heuristique d'amélioration, sur 300 secondes. A partir de la figure 4.7, nous pouvons remarquer l'effet considérable de l'heuristique d'amélioration sur la première approche qui devient la meilleure. Les résultats de la deuxième et la troisième approche sont pratiquement équivalents.

#### 4.4.2 Résultats expérimentaux sur les larges instances

Afin d'évaluer l'efficacité des heuristiques, nous avons également effectué des tests sur des instances larges. Pour ce type d'instances, nous avons testé uniquement la première et la quatrième approche. L'exclusion de la deuxième et la troisième approche est due à la complexité des sous problèmes PLNE *SP1* et *SP2* à résoudre par CPLEX qui nécessite beaucoup de temps

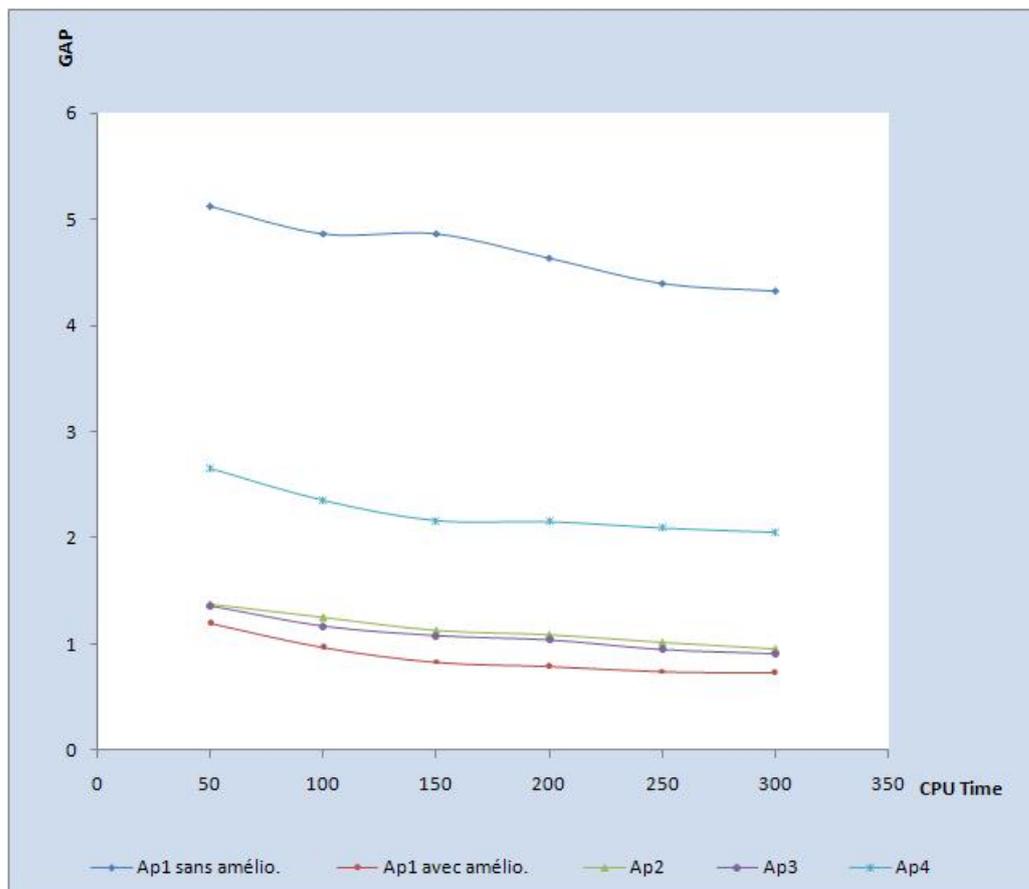


FIGURE 4.7 – Comparaison du Gap moyen

TABLE 4.1 – Gap obtenu par la première approche sans amélioration sur les petites et moyennes instances

Instances			Gap					
T	N	K	50 <sub>s</sub>	100 <sub>s</sub>	150 <sub>s</sub>	200 <sub>s</sub>	250 <sub>s</sub>	300 <sub>s</sub>
3	5	2	0.40	0.40	0.40	0.40	0.40	0.40
3	10	2	0.00	0.00	0.00	0.00	0.00	0.00
3	15	2	1.47	1.47	1.47	1.47	0.59	0.59
3	20	2	7.98	7.09	7.09	7.09	6.92	6.92
6	5	2	0.00	0.00	0.00	0.00	0.00	0.00
6	10	2	3.17	2.96	2.96	2.96	2.96	2.96
6	15	2	6.19	6.19	6.19	6.19	6.19	6.19
6	20	2	9.49	9.49	9.13	9.13	9.13	8.67
9	5	2	4.86	4.53	4.53	4.53	4.53	4.53
9	10	2	7.43	4.03	4.03	4.03	2.80	2.80
9	15	2	9.39	8.10	8.10	8.10	8.10	8.10
9	20	2	11.52	10.87	9.61	9.61	9.61	9.61
3	5	3	0.00	0.00	0.00	0.00	0.00	0.00
3	10	3	0.00	0.00	0.00	0.00	0.00	0.00
3	15	3	1.05	1.05	1.05	1.05	1.05	1.05
3	20	3	3.70	3.70	2.87	2.87	2.87	2.87
6	5	3	2.60	2.60	2.60	2.60	2.60	2.60
6	10	3	2.20	1.75	1.75	1.29	1.29	1.29
6	15	3	7.38	7.38	7.02	7.02	7.02	7.02
6	20	3	8.82	8.60	8.60	8.07	8.07	7.37
9	5	3	3.87	3.84	3.48	3.45	3.17	3.17
9	10	3	6.83	6.83	6.83	6.09	3.37	3.37
9	15	3	8.91	8.91	8.91	8.91	8.91	8.84
9	20	3	10.38	10.38	9.92	9.92	9.92	9.92
3	5	4	5.09	5.09	5.09	5.09	5.09	5.09
3	10	4	0.00	0.00	0.00	0.00	0.00	0.00
3	15	4	3.46	3.46	3.21	2.24	2.24	1.57
3	20	4	4.93	4.93	4.93	4.93	4.93	4.93
6	5	4	4.34	4.34	4.34	4.34	4.34	4.24
6	10	4	2.56	2.56	2.56	1.72	1.72	1.72
6	15	4	6.73	5.56	5.56	5.56	4.90	4.90
6	20	4	8.33	8.03	8.03	8.03	8.03	7.54
9	5	4	5.29	5.07	4.80	4.68	4.43	4.43
9	10	4	6.91	6.91	6.39	6.39	5.33	5.22
9	15	4	8.63	8.63	8.53	8.53	8.30	8.30
9	20	4	10.32	10.32	10.32	10.32	9.30	9.30
<b>Moyenne</b>			<b>5.12</b>	<b>4.86</b>	<b>4.73</b>	<b>4.63</b>	<b>4.39</b>	<b>4.32</b>

pour trouver une solution, ce qui ne permet pas de résoudre plusieurs sous problèmes dans une génération. Le temps réservé à chaque approche (300 secondes) sera consommé peut être par un sous problème. Par contre, dans la quatrième approche, le sous problème *SP3* à résoudre par CPLEX est plus facile, puisque il contient moins de variables binaires par rapport à *SP1* et *SP2*.

Le gap dans le tableau 4.6, est calculé par rapport à la meilleure solution trouvée par CPLEX

TABLE 4.2 – Gap obtenu par la première approche avec amélioration sur les petites et moyennes instances

Instances			Gap					
T	N	K	50 <sub>s</sub>	100 <sub>s</sub>	150 <sub>s</sub>	200 <sub>s</sub>	250 <sub>s</sub>	300 <sub>s</sub>
3	5	2	0.00	0.00	0.00	0.00	0.00	0.00
3	10	2	0.00	0.00	0.00	0.00	0.00	0.00
3	15	2	0.00	0.00	0.00	0.00	0.00	0.00
3	20	2	0.00	0.00	0.00	0.00	0.00	0.00
6	5	2	0.00	0.00	0.00	0.00	0.00	0.00
6	10	2	0.16	0.16	0.16	0.16	0.16	0.16
6	15	2	3.97	0.93	0.52	0.52	0.52	0.52
6	20	2	1.06	1.06	0.71	0.71	0.71	0.71
9	5	2	2.64	2.64	2.64	2.64	2.64	2.64
9	10	2	0.55	0.55	0.55	0.55	0.55	0.55
9	15	2	1.01	0.95	0.53	0.38	0.38	0.38
9	20	2	4.18	2.21	2.00	1.98	1.98	1.97
3	5	3	0.00	0.00	0.00	0.00	0.00	0.00
3	10	3	0.00	0.00	0.00	0.00	0.00	0.00
3	15	3	0.00	0.00	0.00	0.00	0.00	0.00
3	20	3	0.00	0.00	0.00	0.00	0.00	0.00
6	5	3	0.75	0.75	0.43	0.43	0.23	0.23
6	10	3	0.67	0.43	0.43	0.43	0.42	0.42
6	15	3	2.65	2.36	1.75	1.75	1.75	1.75
6	20	3	1.75	1.75	1.75	1.68	1.68	1.67
9	5	3	1.10	1.10	1.10	1.10	1.10	1.10
9	10	3	2.51	2.37	1.60	1.60	1.31	1.06
9	15	3	2.15	1.91	1.52	1.48	1.48	1.48
9	20	3	3.52	3.19	2.40	2.33	1.89	1.89
3	5	4	0.00	0.00	0.00	0.00	0.00	0.00
3	10	4	0.00	0.00	0.00	0.00	0.00	0.00
3	15	4	0.00	0.00	0.00	0.00	0.00	0.00
3	20	4	0.00	0.00	0.00	0.00	0.00	0.00
6	5	4	0.07	0.07	0.07	0.07	0.07	0.07
6	10	4	0.56	0.56	0.52	0.39	0.39	0.39
6	15	4	2.87	1.82	1.82	1.82	1.25	1.25
6	20	4	1.21	1.21	1.21	1.19	1.11	1.11
9	5	4	2.24	2.24	2.24	2.24	2.24	2.24
9	10	4	2.80	2.54	2.46	1.52	1.52	1.52
9	15	4	1.81	1.51	1.48	1.48	1.12	1.12
9	20	4	2.90	2.47	2.11	2.11	2.11	2.11
<b>Moyenne</b>			<b>1.20</b>	<b>0.97</b>	<b>0.83</b>	<b>0.79</b>	<b>0.74</b>	<b>0.73</b>

dans la limite de temps de 2 heures, ce qui explique que certaines valeurs du gap sont négatives. La lecture des résultats nous permet de dire que la première approche est meilleure par rapport à la quatrième. Les deux approches permettent d'améliorer certaines instances résolues par CPLEX en 2 heures.

Le tableau 4.7 présente la comparaison des valeurs de la fonction objectif. La colonne 4 du

TABLE 4.3 – Gap obtenu par la deuxième approche sur les petites et moyennes instances

Instances			Gap					
T	N	K	50 <sub>s</sub>	100 <sub>s</sub>	150 <sub>s</sub>	200 <sub>s</sub>	250 <sub>s</sub>	300 <sub>s</sub>
3	5	2	0.00	0.00	0.00	0.00	0.00	0.00
3	10	2	0.00	0.00	0.00	0.00	0.00	0.00
3	15	2	0.00	0.00	0.00	0.00	0.00	0.00
3	20	2	0.00	0.00	0.00	0.00	0.00	0.00
6	5	2	0.10	0.10	0.00	0.00	0.00	0.00
6	10	2	2.32	1.99	1.88	1.47	0.82	0.82
6	15	2	1.31	0.96	0.58	0.58	0.58	0.58
6	20	2	2.02	1.72	1.58	1.32	1.32	1.32
9	5	2	2.27	1.87	1.87	1.87	1.68	1.25
9	10	2	1.43	1.40	1.40	1.20	1.01	1.01
9	15	2	3.28	3.28	2.24	2.17	2.17	2.17
9	20	2	3.03	2.68	2.49	2.35	2.35	2.35
3	5	3	0.00	0.00	0.00	0.00	0.00	0.00
3	10	3	0.00	0.00	0.00	0.00	0.00	0.00
3	15	3	0.00	0.00	0.00	0.00	0.00	0.00
3	20	3	0.00	0.00	0.00	0.00	0.00	0.00
6	5	3	0.00	0.00	0.00	0.00	0.00	0.00
6	10	3	0.85	0.59	0.59	0.59	0.59	0.59
6	15	3	2.32	2.32	2.32	2.32	2.28	1.90
6	20	3	1.96	1.09	1.09	1.09	1.09	1.09
9	5	3	1.05	0.71	0.31	0.12	0.12	0.12
9	10	3	2.72	2.42	2.00	2.00	2.00	2.00
9	15	3	3.15	2.82	2.35	2.35	1.17	1.17
9	20	3	4.21	4.21	4.21	4.21	4.10	3.87
3	5	4	0.00	0.00	0.00	0.00	0.00	0.00
3	10	4	0.00	0.00	0.00	0.00	0.00	0.00
3	15	4	0.00	0.00	0.00	0.00	0.00	0.00
3	20	4	0.00	0.00	0.00	0.00	0.00	0.00
6	5	4	0.00	0.00	0.00	0.00	0.00	0.00
6	10	4	0.44	0.44	0.33	0.33	0.33	0.33
6	15	4	2.84	2.62	2.62	2.28	2.28	2.28
6	20	4	2.65	2.65	2.65	2.65	2.65	2.65
9	5	4	0.48	0.48	0.34	0.34	0.34	0.34
9	10	4	3.03	3.03	3.03	3.03	3.03	2.78
9	15	4	3.67	3.67	3.67	3.67	3.61	2.77
9	20	4	4.24	3.97	3.30	3.29	3.29	3.29
<b>Moyenne</b>			<b>1.37</b>	<b>1.25</b>	<b>1.13</b>	<b>1.09</b>	<b>1.02</b>	<b>0.96</b>

tableau donne la valeur de fonction objectif trouvée par CPLEX dans deux heures. Les colonnes 6, 7 et 8 présentent les valeurs de la fonction objectif trouvée par CPLEX, la première approche et la quatrième approche respectivement, après 300 secondes. La colonne 5 donne la borne inférieure trouvée par CPLEX après deux heures d'exécution. On constate que, à 300 secondes d'exécution, CPLEX n'arrive pas à trouver une solution pour plus de 48% d'instances (soit 13 instances sur 27), et pour le reste d'instances, les valeurs de la fonction objectif sont de mauvaise

TABLE 4.4 – Gap obtenu par la troisième approche sur les petites et moyennes instances

Instances			Gap					
T	N	K	50 <sub>s</sub>	100 <sub>s</sub>	150 <sub>s</sub>	200 <sub>s</sub>	250 <sub>s</sub>	300 <sub>s</sub>
3	5	2	0.00	0.00	0.00	0.00	0.00	0.00
3	10	2	0.00	0.00	0.00	0.00	0.00	0.00
3	15	2	0.00	0.00	0.00	0.00	0.00	0.00
3	20	2	0.00	0.00	0.00	0.00	0.00	0.00
6	5	2	0.00	0.00	0.00	0.00	0.00	0.00
6	10	2	2.08	2.08	2.08	2.08	2.08	2.08
6	15	2	0.92	0.67	0.67	0.04	0.04	0.04
6	20	2	1.68	1.05	0.90	0.90	0.90	0.90
9	5	2	2.27	2.27	2.27	2.27	2.27	2.27
9	10	2	1.11	1.02	1.02	1.02	0.75	0.75
9	15	2	2.58	2.08	1.96	1.96	1.74	1.74
9	20	2	4.47	3.14	2.76	2.44	2.43	2.26
3	5	3	0.00	0.00	0.00	0.00	0.00	0.00
3	10	3	0.00	0.00	0.00	0.00	0.00	0.00
3	15	3	0.00	0.00	0.00	0.00	0.00	0.00
3	20	3	0.00	0.00	0.00	0.00	0.00	0.00
6	5	3	0.00	0.00	0.00	0.00	0.00	0.00
6	10	3	0.61	0.61	0.55	0.55	0.55	0.55
6	15	3	2.64	2.64	2.25	2.25	1.99	1.94
6	20	3	1.66	1.54	1.54	1.49	1.37	1.36
9	5	3	0.23	0.23	0.23	0.23	0.19	0.19
9	10	3	2.92	2.53	2.53	2.48	2.13	1.90
9	15	3	3.70	3.45	2.92	2.60	2.60	2.60
9	20	3	4.25	3.87	3.87	3.87	3.87	2.95
3	5	4	0.00	0.00	0.00	0.00	0.00	0.00
3	10	4	0.00	0.00	0.00	0.00	0.00	0.00
3	15	4	0.00	0.00	0.00	0.00	0.00	0.00
3	20	4	0.00	0.00	0.00	0.00	0.00	0.00
6	5	4	0.00	0.00	0.00	0.00	0.00	0.00
6	10	4	0.76	0.41	0.13	0.13	0.13	0.13
6	15	4	3.95	2.06	2.06	2.06	0.74	0.74
6	20	4	3.10	3.10	3.10	2.74	2.56	2.33
9	5	4	0.56	0.44	0.32	0.32	0.18	0.18
9	10	4	2.18	2.08	1.88	1.88	1.88	1.88
9	15	4	3.48	3.41	2.99	2.99	2.99	2.92
9	20	4	3.86	3.40	2.99	2.99	2.88	2.88
<b>Moyenne</b>			<b>1.36</b>	<b>1.17</b>	<b>1.08</b>	<b>1.04</b>	<b>0.95</b>	<b>0.91</b>

qualité.

Le gap dans le tableau 4.7, est calculé par rapport à la borne inférieure trouvée par CPLEX après deux heures d'exécution. La colonne 9 du tableau donne le gap obtenu par CPLEX après deux heures d'exécution. Il varie entre 6% et 56%. Les colonnes 10, 11 et 12 représentent le gap obtenu après 300 secondes, par CPLEX, la première approche et la quatrième approche

TABLE 4.5 – Gap obtenu par la quatrième approche sur les petites et moyennes instances

Instances			Gap					
T	N	K	50 <sub>s</sub>	100 <sub>s</sub>	150 <sub>s</sub>	200 <sub>s</sub>	250 <sub>s</sub>	300 <sub>s</sub>
3	5	2	0.00	0.00	0.00	0.00	0.00	0.00
3	10	2	0.00	0.00	0.00	0.00	0.00	0.00
3	15	2	0.00	0.00	0.00	0.00	0.00	0.00
3	20	2	0.00	0.00	0.00	0.00	0.00	0.00
6	5	2	2.94	2.94	2.94	2.94	2.94	2.94
6	10	2	5.41	5.41	1.92	1.92	1.92	1.92
6	15	2	3.26	2.74	2.74	2.74	2.74	2.74
6	20	2	1.44	1.44	1.35	1.28	1.05	1.05
9	5	2	6.09	5.94	5.94	5.94	5.94	5.94
9	10	2	2.54	2.43	2.43	2.43	2.43	2.43
9	15	2	4.44	2.76	2.46	2.26	2.21	2.21
9	20	2	6.00	6.00	5.07	5.07	4.80	3.60
3	5	3	0.00	0.00	0.00	0.00	0.00	0.00
3	10	3	0.00	0.00	0.00	0.00	0.00	0.00
3	15	3	0.00	0.00	0.00	0.00	0.00	0.00
3	20	3	0.00	0.00	0.00	0.00	0.00	0.00
6	5	3	4.07	4.07	4.07	4.07	4.07	4.07
6	10	3	3.46	3.43	3.43	3.43	3.43	3.43
6	15	3	3.13	3.13	3.13	3.13	3.13	3.13
6	20	3	0.95	0.63	0.63	0.63	0.63	0.63
9	5	3	6.94	6.80	6.80	6.80	6.80	6.80
9	10	3	4.92	3.21	3.21	3.21	3.21	3.21
9	15	3	3.84	2.97	2.97	2.97	2.17	1.76
9	20	3	4.73	3.16	3.16	3.16	3.16	3.16
3	5	4	0.00	0.00	0.00	0.00	0.00	0.00
3	10	4	0.00	0.00	0.00	0.00	0.00	0.00
3	15	4	0.00	0.00	0.00	0.00	0.00	0.00
3	20	4	0.00	0.00	0.00	0.00	0.00	0.00
6	5	4	3.38	3.38	3.38	3.38	3.38	3.38
6	10	4	3.74	3.27	3.27	3.27	3.27	3.27
6	15	4	2.43	2.43	2.43	2.16	2.04	2.04
6	20	4	2.85	2.02	2.02	2.02	2.02	2.02
9	5	4	3.11	2.42	2.42	2.42	2.42	2.42
9	10	4	5.57	5.57	5.57	5.57	5.57	5.57
9	15	4	4.43	3.33	3.33	3.33	3.07	3.06
9	20	4	5.98	5.66	3.58	3.54	3.35	3.35
<b>Moyenne</b>			<b>2.66</b>	<b>2.36</b>	<b>2.17</b>	<b>2.16</b>	<b>2.10</b>	<b>2.06</b>

respectivement. Le gap obtenu par la première approche varie entre 9% et 14% et celui de la quatrième approche varie entre 11% et 17%. Cela veut dire que la première approche est aussi meilleure pour les instances larges. Une observation intéressante, est que le gap, pour les deux approches, n'augmente pas d'une manière considérable avec l'augmentation des paramètres  $T$ ,  $N$  et  $K$ .

TABLE 4.6 – Gap obtenu par la première et la quatrième approche sur les instances larges

Instances			Gap obtenu par la première approche						Gap obtenu par la quatrième approche					
T	N	K	50 <sub>s</sub>	100 <sub>s</sub>	150 <sub>s</sub>	200 <sub>s</sub>	250 <sub>s</sub>	300 <sub>s</sub>	50 <sub>s</sub>	100 <sub>s</sub>	150 <sub>s</sub>	200 <sub>s</sub>	250 <sub>s</sub>	300 <sub>s</sub>
20	40	4	5.10	4.52	3.23	3.23	3.23	3.23	4.08	4.04	3.42	3.42	3.42	3.42
20	60	4	5.17	3.50	3.50	3.50	3.50	3.50	5.96	5.59	5.22	4.59	4.34	4.32
20	80	4	3.00	1.73	1.73	0.82	0.82	0.82	5.13	4.16	3.29	3.29	2.67	2.67
30	40	4	6.74	6.74	6.74	6.74	6.29	6.29	4.28	3.26	3.22	3.21	2.92	2.70
30	60	4	8.24	8.01	7.83	7.83	7.21	7.13	7.08	5.19	5.19	4.88	4.78	4.64
30	80	4	7.24	6.77	6.61	6.39	6.35	6.35	9.09	4.62	3.61	3.61	3.61	3.56
40	40	4	6.50	6.44	6.31	6.26	5.61	5.61	4.35	3.56	2.81	2.80	2.80	2.57
40	60	4	1.56	1.15	0.95	0.95	0.76	0.62	-0.09	-2.42	-2.96	-2.96	-2.96	-2.96
40	80	4	2.74	2.72	2.72	2.30	1.49	1.49	2.10	0.40	-0.32	-0.32	-0.36	-0.42
20	40	6	-0.01	-0.24	-0.24	-0.24	-0.61	-0.61	-1.38	-2.24	-2.62	-3.07	-3.63	-3.63
20	60	6	6.86	6.49	6.29	6.29	6.29	6.15	5.78	5.46	5.29	4.70	3.87	3.62
20	80	6	5.33	5.20	5.20	4.29	4.29	4.29	4.84	4.38	4.28	4.08	3.92	3.73
30	40	6	3.76	3.76	3.65	3.56	3.02	2.68	1.26	0.91	0.91	0.91	0.12	-0.29
30	60	6	6.02	5.91	5.76	5.76	5.59	5.06	7.58	4.68	3.78	3.78	3.42	3.42
30	80	6	2.85	2.33	2.33	2.24	1.58	1.58	0.02	-0.11	-0.17	-0.23	-0.24	-0.24
40	40	6	5.00	4.77	4.72	4.59	4.59	4.44	3.76	2.57	2.57	2.22	2.00	2.00
40	60	6	-19.65	-19.65	-19.65	-19.69	-19.92	-20.05	-17.55	-21.24	-21.52	-21.52	-21.58	-21.77
40	80	6	6.31	5.49	5.25	4.32	4.32	4.32	7.95	7.95	4.56	3.25	3.25	2.86
20	40	8	2.22	2.17	2.17	2.09	2.09	1.70	0.75	0.08	-0.22	-1.56	-2.22	-2.22
20	60	8	6.09	6.09	6.05	5.54	5.53	5.53	3.94	3.17	2.70	2.70	2.70	2.70
20	80	8	5.95	5.64	5.38	4.37	4.37	4.37	5.38	3.52	3.11	2.68	2.68	2.68
30	40	8	-5.86	-6.58	-6.58	-6.58	-6.82	-6.82	-7.87	-9.54	-9.54	-10.32	-10.32	-10.32
30	60	8	-10.12	-10.78	-11.06	-11.06	-11.06	-11.06	-9.35	-12.20	-13.02	-13.02	-13.28	-13.28
30	80	8	-95.73	-95.75	-96.31	-96.31	-96.31	-96.31	-92.93	-98.36	-100.02	-100.83	-101.68	-101.68
40	40	8	5.05	5.05	4.68	3.43	3.43	3.43	2.91	2.19	1.81	1.70	1.41	1.37
40	60	8	-36.99	-37.89	-38.35	-38.62	-38.62	-39.20	-36.71	-40.93	-42.47	-42.53	-42.53	-42.53
40	80	8	-12.29	-13.16	-14.82	-14.82	-14.82	-14.82	-4.90	-13.69	-16.08	-16.38	-16.49	-17.01

## CONCLUSION

Dans ce chapitre, nous avons proposé quatre approches heuristiques à base d'algorithme génétique pour la résolution du problème. Nous avons, d'abord testé ces approches sur des petites et moyennes instances. Les résultats montrent que ces approches peuvent donner des solutions optimales ou proches de l'optimum dans ce cas. Nous avons testé uniquement la première et la quatrième approche sur des instances larges. Les résultats expérimentaux montrent l'efficacité de ces deux approches dans ce cas de figure d'instances, dans un temps réduit(300 secondes), en comparant la qualité des solutions trouvées avec la meilleure borne inférieure trouvée par CPLEX en deux heures d'exécution. On remarque que CPLEX n'arrive pas à trouver dans un temps raisonnable des solutions faisables pour plusieurs instances. Ceci explique la nécessité d'utiliser des méthodes approchées pour ce problème avec des instances larges. Ce constat nous a poussé à proposer d'autres approches heuristiques basées sur la recherche tabou et le recuit simulé dans le chapitre suivant.

TABLE 4.7 – Comparaison de la première et la quatrième approche avec CPLEX en terme de l'objectif et du Gap

T	N	K	Ob.Cplex <sub>2h</sub>	BI <sub>2h</sub>	Ob.Cplex <sub>300s</sub>	Ob.Ap <sub>1300s</sub>	Ob.Ap <sub>4300s</sub>	Gap.Cplex <sub>2h</sub>	Gap.Cplex <sub>300s</sub>	Gap.Ap <sub>1300s</sub>	Gap.Ap <sub>4300s</sub>
20	40	4	174650	163560	186264	180843	184038	6.35	12.19	9.56	11.13
20	60	4	231583	215766	242998	242048	244203	6.83	11.21	10.86	11.64
20	80	4	300589	270921	463098	308834	309873	9.87	41.50	12.28	12.57
30	40	4	255221	231179	512490	262315	273672	9.42	54.89	11.87	15.53
30	60	4	340282	310031	685050	356828	370846	8.89	54.74	13.11	16.40
30	80	4	429198	383832	647314	445025	462682	10.57	40.70	13.75	17.04
40	40	4	366448	331122	601060	376117	391934	9.64	44.91	11.96	15.52
40	60	4	508999	430053	PdS	494366	517043	15.51	PdS	13.01	16.82
40	80	4	615207	523049	PdS	612614	632535	14.98	PdS	14.62	17.31
20	40	6	206762	179862	206762	199528	206742	13.01	13.01	9.86	13.00
20	60	6	252548	233102	513860	262041	271151	7.7	54.64	11.04	14.03
20	80	6	318469	287641	654880	330800	336409	9.68	56.08	13.05	14.50
30	40	6	293181	257442	572490	292325	304643	12.19	55.03	11.93	15.49
30	60	6	370972	335025	745050	384100	394733	9.69	55.03	12.78	15.13
30	80	6	475370	410957	PdS	474210	489319	13.55	PdS	13.34	16.01
40	40	6	407596	364839	PdS	415921	429039	10.49	PdS	12.28	14.96
40	60	6	657540	466590	PdS	539983	549565	29.04	PdS	13.59	15.10
40	80	6	633783	559694	PdS	652458	676454	11.69	PdS	14.22	17.26
20	40	8	223092	198619	432955	218248	228165	10.97	54.12	8.99	12.95
20	60	8	273794	250768	PdS	281406	291560	8.41	PdS	10.89	13.99
20	80	8	339887	307258	PdS	349237	361402	9.6	PdS	12.02	14.98
30	40	8	355501	284792	PdS	322237	335834	19.89	PdS	11.62	15.20
30	60	8	469759	360775	805050	414687	426576	23.2	55.19	13.00	15.43
30	80	8	1013330	436948	PdS	502443	517713	56.88	PdS	13.04	15.60
40	40	8	445861	400562	PdS	452050	469564	10.16	PdS	11.39	14.70
40	60	8	813573	501324	PdS	570797	593913	38.38	PdS	12.17	15.59
40	80	8	810495	597335	PdS	692680	721814	26.3	PdS	13.76	17.25

PdS : pas de solution trouvée

BI : Borne Inférieure trouvée par Cplex après 2 heures d'exécution



# MÉTA-HEURISTIQUES À BASE DE RECHERCHE TABOU ET DE RECUIT SIMULÉ

## RÉSUMÉ

*DANS ce chapitre nous proposons la résolution de notre problème par trois approches, la première est à base de recherche tabou, la deuxième est à base de recuit simulé et la troisième est une hybridation entre les deux premières approches. Nous comparons ensuite ces approches. Nous montrons que l'hybridation entre les métaheuristiques permet d'obtenir des solutions de haute qualité.*



## 5.1 INTRODUCTION

Le but de ce chapitre est de développer des approches basées sur des méta-heuristiques encore plus pertinentes, qui sont la recherche tabou et le recuit simulé. Elles sont appelées méthodes de recherche à voisinage, pour lesquelles à chaque itération une meilleure solution est trouvée en explorant le voisinage de la solution courante. Ces dernières ont été appliquées avec succès à une grande variété des problèmes combinatoires, comme les problèmes d'intégration production distribution. Nous avons déjà cité Bard et Nananukul [15], Shiguemoto and Armentano [102] et Armentano et al. [11]. Avant de débiter le développement des approches, nous donnons les concepts de base de la recherche tabou et du recuit simulé dans les sections 5.2 et 5.4. Nous expliquons l'application de l'approche à base de la recherche tabou, pour résoudre notre problème, dans la section 5.3. L'algorithme de l'approche à base de recuit simulé pour résoudre le même problème est donné dans la section 5.4.3. Afin de pouvoir tirer avantage de chacune des deux premières approches, nous développons dans la section 5.5 une approche hybride dans laquelle la recherche tabou utilise comme solution initiale la meilleure solution trouvée par le recuit. Enfin, nous terminons ce chapitre par une évaluation de ces trois approches à l'aide des tests expérimentaux.

## 5.2 LA RECHERCHE TABOU : PRINCIPES DE BASE

La recherche tabou (RT) introduite par Glover [55] en 1986, est une métaheuristique à base d'une solution unique. La RT est une méthode de recherche locale avancée. Elle fait appel à un ensemble de règles et de mécanismes généraux pour guider la recherche de la solution de manière intelligente. La RT est basée sur l'utilisation de deux notions : le voisinage et la mémoire (liste tabou). Le principe de la RT est que pour un échantillonnage de solutions du voisinage de la solution courante  $s$ , elle retient la meilleure solution voisine  $s'$ , même si celle-ci est mauvaise par rapport à  $s$ . Ceci permet de donner au processus la possibilité d'exploration de l'espace de recherche afin de rencontrer l'optimum globale. Les solutions de mauvaise qualité peuvent avoir de bons voisinages et donc guider la recherche vers de meilleures solutions. Cependant, cette stratégie peut créer un phénomène de cyclage (i.e. on peut revisiter des solutions déjà parcourues plusieurs fois). Afin de pallier à ce problème, la RT utilise une mémoire permettant le stockage des dernières solutions rencontrées pour ne pas les visiter dans les prochaines itérations. Cette mémoire est appelée *liste tabou*, d'où le nom de la méthode. La taille de la liste tabou est limitée, ce qui empêche l'enregistrement de toutes les solutions rencontrées. C'est la raison pour laquelle la liste tabou procède comme une pile FIFO, où la plus ancienne solution sera éliminée pour laisser place à la dernière solution rencontrée.

### 5.2.1 Le voisinage

Le voisinage  $N(s)$  d'une solution est défini comme l'ensemble des solutions qui sont obtenues par une perturbation(mouvement) de  $s$ . La recherche tabou examine un échantillonnage de solutions de  $N(s)$  et retient la meilleure solution  $s'$  même si  $f(s') > f(s)$ . La recherche tabou ne s'arrête donc pas au premier optimum trouvé.

### 5.2.2 La mémoire

Elle est représentée par une liste tabou qui contient les mouvements qui sont temporairement interdits ou solutions interdites. Elle se détermine empiriquement, et varie avec les problèmes, mais c'est une donnée primordiale. Dans certains cas, il est nécessaire d'ajuster la longueur de la liste tabou selon la qualité de voisinage. En effet une liste trop petite peut conduire à un cycle, alors qu'une liste trop grande peut interdire des transformations intéressantes.

### 5.2.3 Aspiration

L'aspiration permet de révoquer les mouvements considérés comme tabou lorsqu'ils produisent une bonne solution. Les critères d'aspiration les plus connus sont les suivants :

- Meilleur global : Soient  $s$  la meilleure solution connue jusqu'à présent et  $s''$  une solution qui peut être obtenue à partir de la solution courante  $s'$  avec une transformation qui utilise des mouvements tabous. Si  $f(s'') < f(s)$  on lève les interdictions tabous ;
- Meilleur régional : quand la solution obtenue à partir des mouvements tabous produit la meilleure solution de la région courante ;
- Meilleur parmi les derniers : quand la solution obtenue à partir des mouvements tabous donne la meilleure solution parmi les  $k$  dernières solutions.

### 5.2.4 Intensification

L'intensification est une stratégie destinée à restreindre la trajectoire de recherche à une partie de l'espace des solutions jugée prometteuse. Cette approche est motivée par l'idée intuitive qui, si l'on exhibe une solution particulièrement intéressante au cours d'une exploration rapide, alors il y a de fortes chances pour qu'un examen plus poussé dans sa proximité fournisse de solutions encore meilleures.

### 5.2.5 Diversification

Afin d'éviter qu'une grande région de l'espace de recherche reste inconnue et inexplorée, il est important de diversifier la recherche en effectuant plusieurs lancements aléatoires. Un autre moyen, qui garantit l'exploration des régions non visitées est de pénaliser les mouvements ou les solutions fréquemment visitées. Cette pénalité est posée de telle sorte à assurer l'éloignement et l'évitement des régions courantes. Il est également possible d'employer une pénalité sur les mouvements fréquemment effectués pendant toute la procédure de recherche. Pendant cette phase de diversification, les solutions visitées ne sont pas obligatoirement réalisables

### 5.2.6 Algorithme général de la recherche tabou

Les principales étapes d'un algorithme de recherche tabou sont les suivantes :

1. Choisir une solution initiale  $s \in N(s)$ , poser  $TabList := \emptyset$  et  $s^* := s$  ;

2. Tant que le critère d'arrêt n'est satisfait faire
  - (a) Déterminer une solution  $s'$  qui maximise  $f(s')$  dans  $N\text{TabList}(s)$ , Avec  $N\text{TabList}(s) = \{s' \in V(S) \text{ tel que } s' \notin \text{TabList}\}$ ;
  - (b) Si  $f(s') > f(s^*)$  alors poser  $s^* := s'$ ;
  - (c) Poser  $s := s'$  et mettre à jour  $T$ ;
3. Fin du tant que.

### 5.3 DÉVELOPPEMENT DE L'APPROCHE À BASE DE LA RECHERCHE TABOU

Nous développons une approche à base de la recherche tabou spécifiquement pour notre problème. Celle-ci est détaillée dans l'algorithme (6).

Ces différentes étapes sont :

#### 5.3.1 Définition de la solution

Dans cette approche nous choisissons comme solution  $s$  la matrice des quantités à livrer aux détaillants dans chaque période notée dans notre cas  $ql$ . c'est la variable "clé" du problème. En effet c'est à partir d'elle qu'on pourra déterminer les autres variables du problème.

#### 5.3.2 Définition du voisinage

Nous définissons le voisinage  $N(s)$  d'une solution  $s$ , avec  $s = ql$ , comme l'ensemble des solutions produites chacune des mouvements décrits dans la section 5.3.3 permettant de passer de la matrice  $ql$  à une autre matrice  $ql'$  proche avec une faible modification de la structure de la matrice  $ql$ .

#### 5.3.3 Définition du mouvement

Nous définissons le mouvement dans la matrice des quantités à livrer  $ql$  (dans le cas de notre problème), comme étant le déplacement d'une quantité maximale  $r_{jt'} \leq ql_{jt'}$  de la période  $t'$  (période source) à la période  $t$  (période réceptrice), pour le détaillant  $j$  sans violer :

1. La capacité de stockage aux détaillants ;
2. La capacité de production ;
3. La capacité de chaque véhicule.

Deux types de mouvement peuvent être appliqué : *forward move* et *backward move*.

#### Forward move

Ce mouvement consiste à déplacer une quantité  $r_{jt'} \leq ql_{jt'}$  de la période  $t'$  à la période  $t > t'$  pour le détaillant  $j$ . La quantité  $r_{jt'}$  est donnée par la formule suivante :

$$r_{jt'} = \min \left\{ \min_{\tau=t', \dots, t-1} I_{j\tau}^C, ql_{jt'}, W - ql_{jt} \right\} \quad (5.1)$$

Après ce mouvement, les éléments concernés par la modification dans la nouvelle solution  $s'$  sont :

$$ql_{jt}^{s'} = ql_{jt}^s + r_{jt'}, \text{ et } ql_{jt'}^{s'} = ql_{jt'}^s - r_{jt'}.$$

Le mouvement a un effet sur le plan de production, les niveaux des stocks au producteur et aux détaillants, ce qui va influencer directement sur le coût total.

### Backward move

Le *Backward move* consiste à déplacer une quantité  $r_{jt'} \leq ql_{jt'}$  de la période  $t'$  à la période  $t < t'$  pour le détaillant  $j$ .

La quantité  $r_{jt'}$  est donnée par la formule suivante :

$$r_{jt'} = \min \left\{ \min_{\tau=t, \dots, t'-1} I_j^{max} - I_{j\tau}^C, W - ql_{jt}, ql_{jt'} \right\} \quad (5.2)$$

Après ce mouvement, les éléments concernés par la modification dans la nouvelle solution  $s'$  sont :

$$ql_{jt}^{s'} = ql_{jt}^s + r_{jt'}, \text{ et } ql_{jt'}^{s'} = ql_{jt'}^s - r_{jt'}.$$

### 5.3.4 Génération de la solution initiale

Dans le cas de notre problème, pour la solution initiale on avait le choix entre soit la génération aléatoire de la matrice  $ql$  soit considérer que la matrice  $ql$  est égale à la matrice des demandes. Après tests le premier choix a donné de mauvais résultats aussi bien en qualité qu'en temps d'exécution aussi nous avons opté pour le second choix à savoir  $ql = d$ .

### 5.3.5 Détermination des autres variables

La détermination des autres variables dans cette approche se fait de la même manière que celle de l'approche à base d'algorithme génétique, avec comme point de départ la matrice des quantités à livrer  $ql$ . Etant donné  $ql$ , l'affectation de ces quantités aux véhicules est un problème de *bin packing* à chaque période et sa résolution peut se faire par l'heuristique *Best Fit Decreasing BFD*. Les niveaux des stocks aux détaillants peuvent être déterminés d'une manière très simple en utilisant l'équation d'équilibre du stock. La détermination des quantités à produire est un problème de lot sizing à capacité constante dont une des méthodes de résolution est l'algorithme de programmation dynamique développé par Florian et Klein [48]. Pour plus de détails sur la détermination des autres variables, le lecteur peut consulter la section 4.3.1.

### 5.3.6 Evaluation d'une solution

Chaque solution est évaluée par la fonction du coût suivante :

$$f(s) = obj(s) + \varphi(s) + \psi(s)$$

où

$obj(s)$  est la valeur de la fonction objectif de la solution  $s$ , ou le coût ordinaire sans prendre en considération les pénalités. Ce coût est calculé par l'équation (3.1)

$\varphi(s)$  est la pénalité liée à la violation des capacités de production. Elle se calcule par la formule (4.4) donnée dans le chapitre 4.

$\psi(s)$  est la pénalité liée à la violation du nombre de véhicules. Elle se calcule aussi par la formule (4.5) donnée dans le chapitre 4.

### 5.3.7 Structure de la liste tabou

Nous considérons la liste tabou *TabList* comme une matrice à deux colonnes. La première colonne est relative aux détaillants  $j$  et la seconde à une période  $t$ . La gestion de la liste tabou est de type FIFO (*First In First Out*). Si un mouvement consiste à transférer une quantité  $r_{jt'}$  de la période  $t'$  (période source) à la période  $t$  (période réceptrice) pour le détaillant  $j$  dans la matrice des quantité à livrer  $ql$ , alors l'enregistrement de ce mouvement se fait par l'insertion d'une ligne de deux éléments dans la liste *TabList* où le premier élément est l'indice  $j$  et le deuxième et l'indice de la période réceptrice  $t$ . Donc n'importe quel mouvement qui consiste à transférer une quantité de la période  $t$  à n'importe quelle autre période pour le détaillant  $j$  est un mouvement interdit, c.à.d un mouvement qui considère  $t$  comme période source. alors la liste tabou n'enregistre pas les solutions déjà visitée mais uniquement les mouvements inverses des mouvements déjà effectués.

### 5.3.8 Détermination du seuil de faisabilité de la production

Dans le chapitre 4, nous avons déterminé la relation (4.1) qui permet de vérifier la faisabilité de la production. Si cette relation n'est pas vérifiée pour certaines périodes, cela veut dire qu'on ne peut pas avoir un plan de production faisable en terme de capacité de production. Nous supposons dans ce cas, que la relation (4.1) n'est pas vérifiée jusqu'à une période que nous avons noté  $t_{inf}$ . La détermination de celle ci se fait par la vérification des deux relations 4.2 et 4.2. L'objectif de la détermination de  $t_{inf}$  est d'éviter le transfert des quantités de livraison avant  $t_{inf}$  par le Backward move, qui aggrave l'infaisabilité du plan de production.

### 5.3.9 Algorithme de l'approche à base de recherche tabou

L'algorithme (6) présente le pseudo code de l'approche proposée. La méthode est exécutée pendant un temps limité. La solution courante  $s$  est modifiée à chaque itération par l'application de *Forward move* ou *Backward move* pour créer le voisinage  $N(s)$ . Le meilleur voisinage est retourné et le mouvement qui mène à celui ci est enregistré. La solution actuelle est mise à jour et le meilleur mouvement est inséré dans la liste tabou. La meilleure solution  $s^*$  peut également être mise à jour. Le choix de type du mouvement dépend de la position de la période de destination  $t$  par rapport à la période source  $t'$ .

---

**Algorithme 6 :** Pseudo code de l'algorithme de recherche tabou

---

```

TabList  $\leftarrow \emptyset$ ;
 $s \leftarrow d$ ;
Calculer  $f(s)$ ;
Bestcost  $\leftarrow f(s)$ ;
 $s^* \leftarrow s$ ;
temps  $\leftarrow 0$ ;
tant que temps < tempsmax faire
  pour  $t = t_{inf} + 1$  à  $T - 1$  faire
    pour  $t' = t + 1$  à  $T$  faire
      pour  $j = 1$  à  $J$  faire
        si  $ql_{jt'} > 0$  alors
           $s' \leftarrow \text{Backward move}(s)$  ;
          Insérer  $s'$  dans  $N(s)$  ;
        finsi
      fin
    fin
  fin
  pour  $t = 2$  à  $T$  faire
    pour  $t' = 1$  à  $t - 1$  faire
      pour  $j = 1$  to  $J$  faire
        si  $ql_{jt'} > 0$  alors
           $s' \leftarrow \text{Forward move}(s)$  ;
          Insérer  $s'$  dans  $N(s)$  ;
        finsi
      fin
    fin
  fin
  Trouver BestNeighbor dans  $N(s)$ ;
  si  $f(\text{BestNeighbor}) < \text{Bestcost}$  alors
     $s^* \leftarrow \text{BestNeighbor}$  ;
  finsi
  Mettre à jour la liste TabList;
   $s \leftarrow \text{BestNeighbor}$ ;
fin

```

---

## 5.4 LE RECUIT SIMULÉ : PRINCIPES DE BASE

Le recuit simulé est une autre méta-heuristique que nous allons appliquer pour la résolution de notre problème. Le choix de la méthode est justifié par sa simplicité et son paramétrage. Le recuit simulé a également été utilisé dans des hybridations avec la recherche tabou.

### 5.4.1 Principe de base du recuit simulé

Le recuit simulé (Simulated Annealing) est souvent présenté comme la plus ancienne des méta-heuristiques, en tout cas, la première à mettre spécifiquement en œuvre une stratégie d'évitement des minimas locaux.

Le recuit simulé repose sur une analogie avec la métallurgie et le recuit des métaux : un métal refroidi trop vite présente de nombreux défauts qui correspondent à des excédents d'énergie interne. L'objectif du recuit est de minimiser ces excédents de façon à obtenir une configuration d'énergie minimale. Pour le réaliser, on réchauffe le métal ce qui a pour effet d'augmenter encore l'énergie interne, mais un réglage judicieux de la température de refroidissement permet de sortir de l'état initial et d'obtenir finalement une énergie interne plus faible. L'application de ce principe à l'optimisation est le suivant : Il est possible, contrairement à un algorithme de recherche locale, d'accepter une dégradation de la fonction objectif avec une certaine probabilité, sachant que cette dégradation pourra entraîner une amélioration ultérieurement.

### 5.4.2 Paramètres du recuit simulé

La convergence du recuit simulé dépend du contrôle de la température  $T_m$  pour atteindre, le plus vite possible, une solution. Le programme de recuit doit préciser les valeurs des paramètres, de contrôle de la température, suivants : la température initiale et la loi de décroissance de la température.

La température  $T_m$  a un rôle très important au cours du processus de recuit simulé. En effet, une forte décroissance de température risque de piéger l'algorithme dans un minimum local, alors qu'une faible décroissance au début du processus entraîne une convergence très lente de l'algorithme.

La loi, selon laquelle la température décroît, est importante pour l'efficacité de l'algorithme puisqu'elle doit laisser le temps au système de tester le maximum de configurations pour être sûr d'obtenir le minimum global. Aussi, la température initiale doit aussi être suffisamment élevée pour que la descente en température soit aussi lente que possible.

### 5.4.3 Algorithme général de recuit simulé

Le recuit simulé applique itérativement l'algorithme de *Metropolis*, pour engendrer une séquence de configurations qui tend vers l'équilibre thermodynamique. L'énergie du système est

représentée par  $E$  et la température par  $Tm$ .

Pour un problème d'optimisation, on obtient une solution à partir d'une recherche locale quelconque, si cette solution a pour effet de diminuer la fonction objectif  $f$ , elle est acceptée; si elle provoque au contraire une augmentation de  $\Delta f$  de la fonction objectif, elle est acceptée tout de même, avec la probabilité  $e^{(\Delta f/Tm)T}$ . Cette méthode est présentée par l'algorithme (7).

---

**Algorithme 7** : Algorithme général du recuit simulé
 

---

Déterminer aléatoirement une solution initiale  $s_0$  et une température initiale  $Tm_0$  ;  
 $s \leftarrow s_0$  et  $Tm \leftarrow Tm_0$ ;  
 Calculer  $f(s)$ ;  
**répéter**  
   Choisir  $s' \in N(s)$ ;  
   Calculer  $\Delta f \leftarrow f(s') - f(s)$ ;  
   **si**  $\Delta f < 0$  **alors**  
      $s \leftarrow s'$ ;  
   **sinon**  
     Générer une probabilité  $p \in [0, 1]$  ;  
     **si**  $p \leq e^{(\Delta f/Tm)}$  **alors**  
        $s \leftarrow s'$ ;  
     **finsi**  
**finsi**  
 Mettre à jour  $Tm$ ;  
**jusqu'à**  $Tm$  est proche de 0;

---

#### 5.4.4 Conception de l'approche à base de recuit simulé

Dans cette approche, nous définissons une solution  $s$  et son voisinage  $N(s)$  de la même façon que dans la méthode de recherche tabou. Pour passer à une solution voisine, on génère aléatoirement les indices :  $j$  (indice détaillant),  $t$  (indice période réceptrice) et  $t'$  (indice période source). Si  $t < t'$ , on applique le *Backward move* et si  $t > t'$  on applique le *Forward move*. Après chaque type de mouvement, on fait,  $ql_{jt}^{s'} = ql_{jt}^s + r_{jtt'}$ , et  $ql_{j't'}^{s'} = ql_{j't'}^s - r_{jtt'}$ , dans la nouvelle solution  $s'$ . La quantité  $r_{jtt'}$  est déterminée par (5.1) ou (5.2), selon le type du mouvement. Le choix de la solution initiale est le même que dans la méthode de recherche tabou, nous choisissons  $s_0 = d$  (initialement les quantités à livrer prennent la valeur de la demande).

On constate ces dernières années que les métaheuristiques les plus performantes sont des méthodes hybrides, c'est-à-dire des méthodes combinant deux ou plusieurs métaheuristiques. Certaines méthodes auraient tendance à trop explorer l'espace de recherche (diversification), alors que d'autres, au contraire, auraient plutôt tendance à faire plus d'exploitation des solutions déjà rencontrées (intensification). L'hybridation des métaheuristiques fournit l'équilibre entre la diversification et l'intensification, donnant une métaheuristique performante. Dans cette pers-

pective, nous développons, dans la section suivante, une approche hybride des deux premières approches.

## 5.5 HYBRIDATION DE LA RECHERCHE TABOU AVEC LE RECUIT SIMULÉ

La finalité de l'approche proposée est de concevoir une collaboration entre les deux métaheuristiques à savoir la recherche tabou et le recuit simulé. Nous envisageons de les faire coopérer pour améliorer leurs résultats. Le schéma de coopération correspond à exécuter l'algorithme de recuit simulé pendant un laps de temps. Après, la recherche tabou utilise comme solution initiale la meilleure solution trouvée par le recuit. Les résultats de simulation, que nous allons voir dans la section suivante, montrent clairement l'efficacité de l'approche adoptée dans la résolution du problème.

## 5.6 RÉSULTATS DE SIMULATION

Pour illustrer l'efficacité et les performances des approches proposées, nous utilisons les petites et moyennes instances utilisées dans le quatrième chapitre. Les algorithmes sont implémentés en MATLAB et tous les tests ont été effectués sur un ordinateur personnel équipé d'un processeur *Core 2 Duo (2.93 GHz) et 3 GB RAM*. Des tableaux et des courbes sont utilisés pour montrer et analyser l'effet de chaque approche. Chaque algorithme est exécuté 300 secondes, ce qui permet de faire un *Benchmarking* entre les approches.

### 5.6.1 Résultats obtenus par l'approche à base de recherche tabou

Dans cette section nous présentons les résultats expérimentaux de la recherche tabou qui a l'avantage d'être facilement paramétrable. Il existe en effet un paramètre primordial qui est la taille de la liste tabou. Elle est généralement déterminée empiriquement et varie avec les problèmes. Dans notre cas, nous déterminons la taille de la liste tabou, que nous notons  $TL$ , de la manière suivante :  $TL = \alpha \times TV$ , où  $TV = T \times (T - 1) \times J$  est la taille du voisinage d'une solution ou le nombre de mouvements possibles.  $T$  est le nombre de périodes,  $J$  le nombre de détaillants et  $\alpha$  un pourcentage. En effet, la taille de la liste tabou est définie comme un pourcentage de la taille de voisinage de la solution courante.

#### a) Effet de la variation de la taille de la liste tabou sur le Gap moyen

Pour déterminer la meilleure taille de la liste tabou, nous avons varié le pourcentage  $\alpha \in \{0\%, 5\%, 10\%, 20\%, 30\%, 40\%, 50\%\}$ , et nous avons comparé les différentes valeurs du Gap obtenu par les simulations au bout de 50, 100, 150, 200, 250 et 300 secondes. Les résultats du Gap moyen des 36 instances sont présentés dans le Tableau 5.1.

La variation du Gap moyen en fonction du temps d'exécution pour différentes variantes de  $TL$  est illustrée par la figure 5.1. Le cas de  $\alpha = 0$  signifie que l'approche tabou est appliquée sans liste tabou, donc elle devient une méthode de recherche locale simple ou une méthode de

TABLE 5.1 – Effet de la variation de la taille de la liste tabou sur le Gap moyen

Taille	Gap						
	1 <sub>s</sub>	50 <sub>s</sub>	100 <sub>s</sub>	150 <sub>s</sub>	200 <sub>s</sub>	250 <sub>s</sub>	300 <sub>s</sub>
0%	30,48	25,42	24,38	23,2	21,84	20,44	14,69
5%	30,48	11,75	9,04	7,85	6,98	6,23	3,28
10%	30,48	12,99	10,52	8,68	8,04	7,65	3,45
20%	30,48	14,79	12,49	11,19	10,02	9,1	3,62
30%	30,48	15,08	12,93	11,63	10,35	9,26	3,47
40%	30,48	15,48	13,38	11,98	11,02	10,02	4,11
50%	30,48	11,67	8,72	7,63	6,86	6,23	5,75

descente. Dans ce cas, le Gap moyen est très élevé, il dépasse 14% à 300 secondes. Entre 50 et 250 secondes, les trois pourcentages :  $\alpha = 5\%$ ,  $\alpha = 10\%$  et  $\alpha = 50\%$  donnent des meilleurs Gap. Entre 250 et 300 secondes, on remarque que l'amélioration du gap continu pour toutes les valeurs de  $\alpha$  sauf pour  $\alpha = 50\%$  et bien sûr pour  $\alpha = 0$ , où il n'y a pas d'amélioration significative. On peut conclure que une taille de la liste tabou  $TL$  obtenue par un  $\alpha$  variant entre 5% et 10% donne le meilleur Gap moyen.

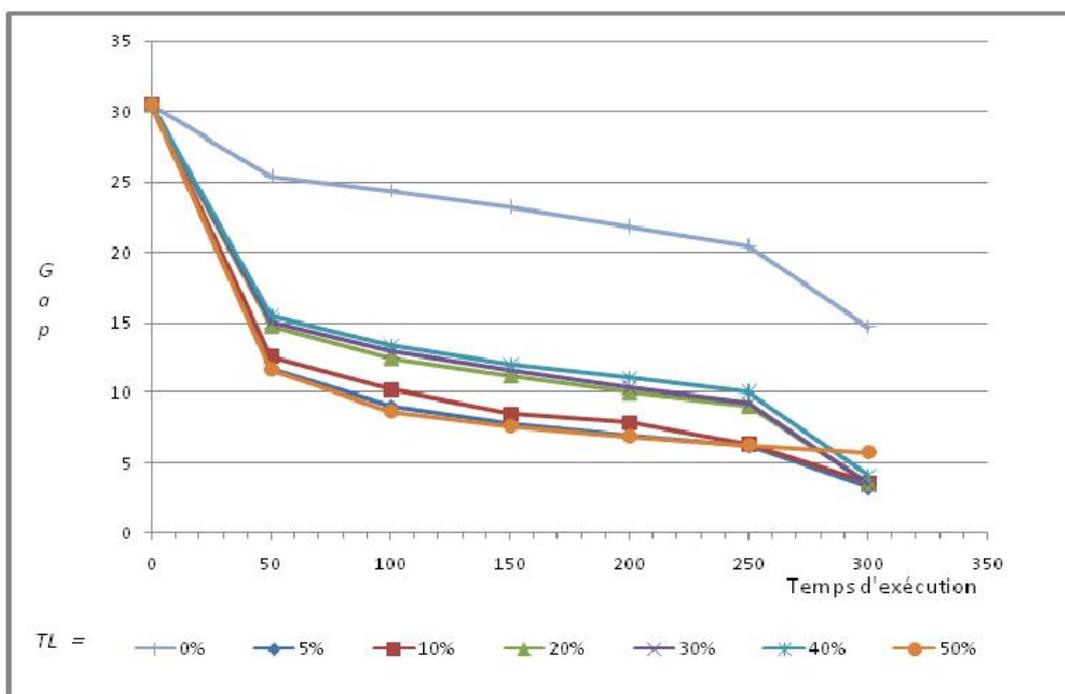


FIGURE 5.1 – Effet de la taille de la liste tabou sur le Gap moyen

### b) Effet de la variation du nombre de périodes $T$ sur le Gap moyen

Nous présentons dans la figure 5.2 les variations du Gap moyen en fonction de la taille de la liste tabou  $TL$  et pour différentes périodes  $T$ , après 300 secondes. On constate que pour  $T = 3$ , l'effet de la variation de la taille de liste tabou n'est pas très important et la meilleure taille est celle donnée par  $\alpha = 5\%$  et  $\alpha = 10\%$ . Pour  $T = 6$ , l'effet de la liste tabou commence à être significatif et la taille donnée par  $\alpha = 5\%$  reste toujours meilleure. Alors pour  $T = 9$ , l'effet de la

taille de la liste tabou est devenu très significatif et cette fois ci, le meilleur gap moyen est donné par  $\alpha = 10\%$ .

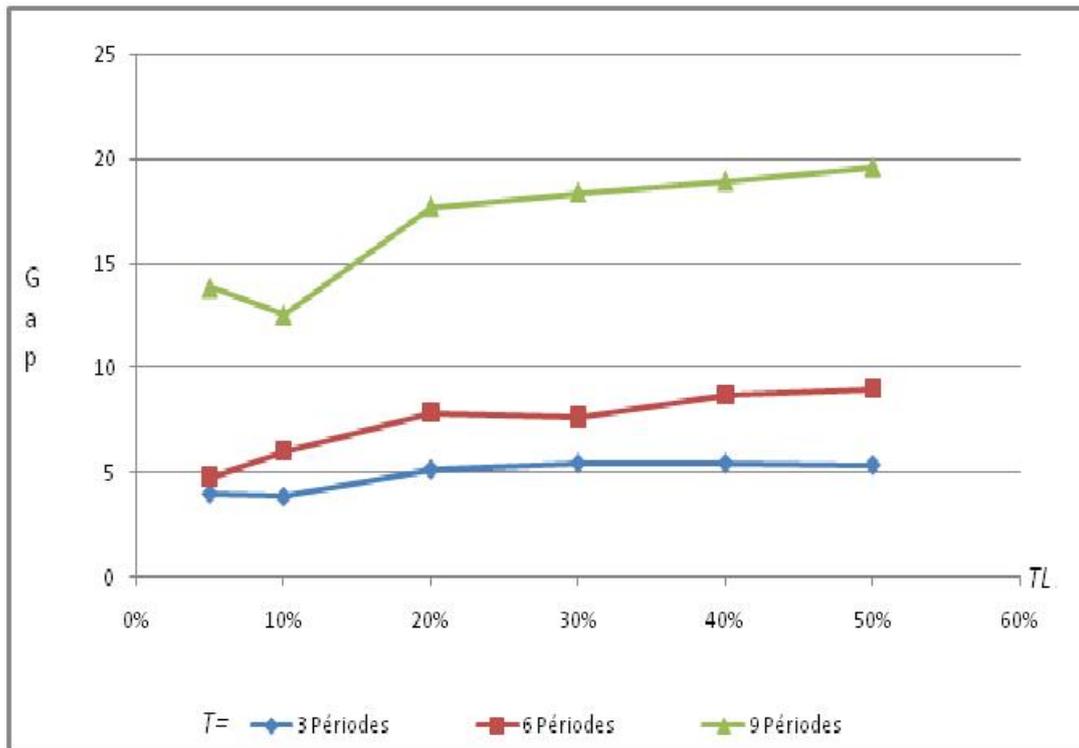


FIGURE 5.2 – Effet de variation du nombre de périodes  $T$  sur le Gap moyen

### c) Effet de la variation du nombre de détaillants sur le Gap moyen

La présentation graphique du gap moyen, après 300 secondes, en fonction de  $TL$  pour différentes valeurs du nombre de détaillants  $J$  est donnée par la figure 5.3. On constate que le meilleur gap est toujours donné pour la taille correspondant à  $\alpha = 5\%$ . Après, le Gap se dégrade avec l'augmentation de la taille. Dans le cas de  $J = 20$ , le Gap est très sensible à l'augmentation de  $\alpha$ .

### d) Effet de la variation du nombre de véhicules sur le Gap moyen

D'une manière analogue, nous représentons sur la figure 5.3 les variations du Gap moyen en fonction de  $\alpha$  mais pour différentes valeurs du nombre de véhicules  $K$ . On constate que la valeur de  $\alpha$  la plus appropriée est 5%, particulièrement pour  $K = 2$ . Les valeurs du gap pour  $K = 2$  et  $\alpha \neq 5\%$  dépassent celles pour  $K = 3$  et  $K = 4$ .

### e) Résultats détaillés obtenus en cas de $\alpha = 5\%$

Nous donnons dans le tableau 5.2 les résultats détaillés obtenus dans le cas où la liste tabou est donnée par  $\alpha = 5\%$ , qui est la meilleure d'après les résultats précédents. Une remarque très

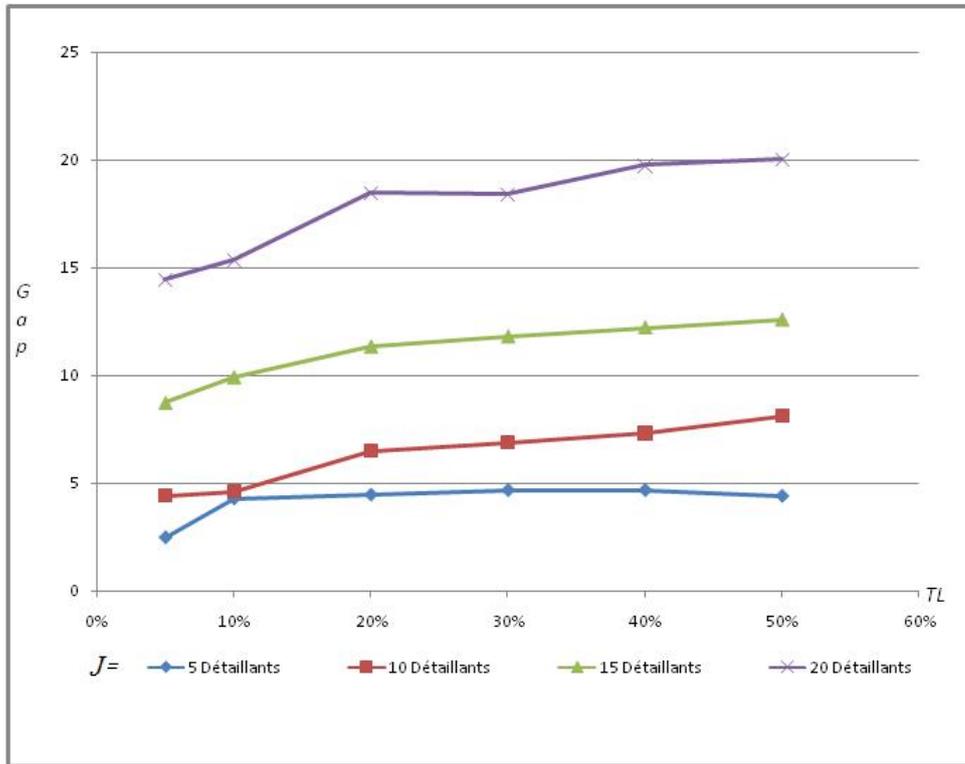


FIGURE 5.3 – Effet de la variation du nombre de détaillants sur le Gap moyen

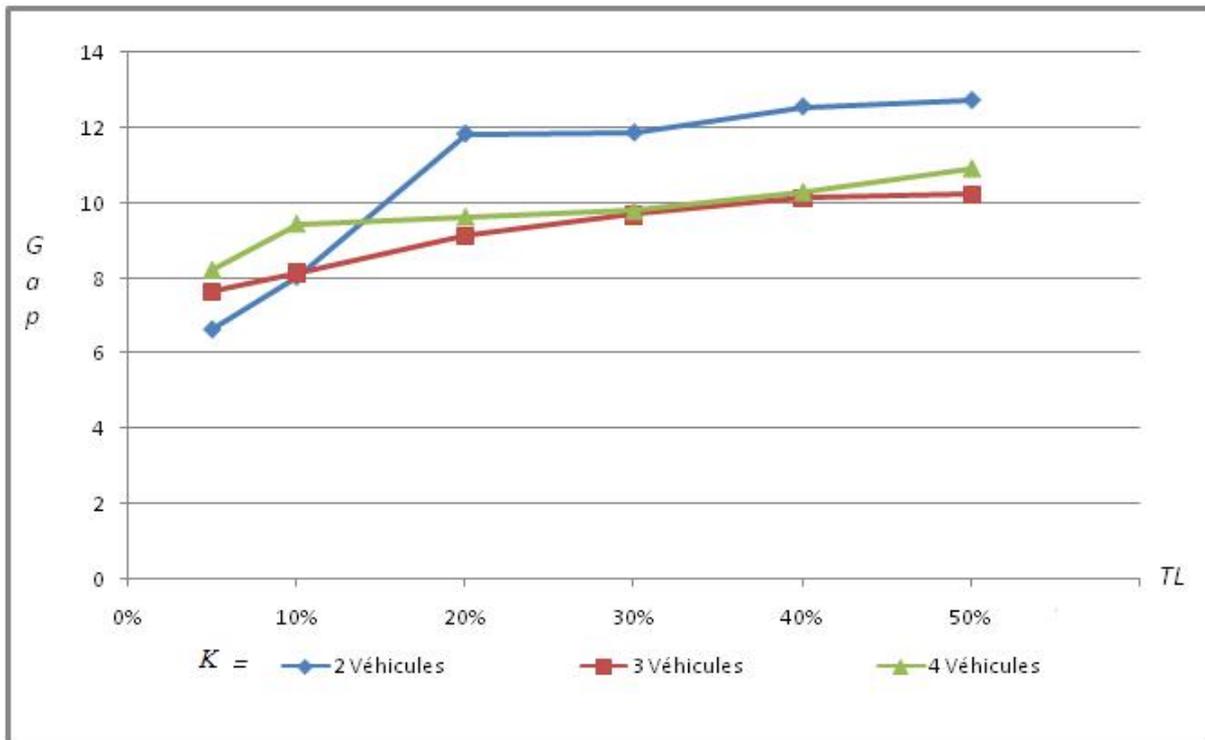


FIGURE 5.4 – Effet de la variation du nombre de véhicules sur le Gap moyen

importante que nous pouvons faire, est que la plus petite instance ( $T = 3$ ,  $J = 5$  et  $K = 2$ ) n'est pas résolue à l'optimum. Par contre, la plus grande instance ( $T = 9$ ,  $J = 20$  et  $K = 4$ ) est résolue à l'optimum. Sur les 36 instances, 10 sont résolues à l'optimum.

TABLE 5.2 – Gap obtenu par la recherche tabou avec  $\alpha = 5\%$ 

Instances			Gap					
T	J	K	50 <sub>s</sub>	100 <sub>s</sub>	150 <sub>s</sub>	200 <sub>s</sub>	250 <sub>s</sub>	300 <sub>s</sub>
3	5	2	4,73	4,73	4,73	4,73	4,73	4,73
3	10	2	0,06	0	0	0	0	0
3	15	2	0	0	0	0	0	0
3	20	2	0	0	0	0	0	0
6	5	2	3,56	1,64	0,85	0,05	0,05	0,05
6	10	2	3,44	3,04	2,72	2,72	2,72	2,72
6	15	2	2,12	1,94	1,81	1,8	1,8	1,67
6	20	2	21,6	13,66	12,97	12,86	12,77	11,88
9	5	2	7,6	5,83	5,34	5,2	5,2	5,2
9	10	2	20,05	6,9	5,67	3,9	3,9	3,9
9	15	2	28,17	23,29	19,47	13,24	7,94	5,99
9	20	2	36,39	33,68	30,81	27,98	25,03	0
3	5	3	0,01	0	0	0	0	0
3	10	3	0,26	0,26	0,26	0,26	0,26	0,26
3	15	3	5,13	5,13	5,13	5,13	5,13	5,13
3	20	3	7,29	7,29	7,29	7,29	7,29	7,29
6	5	3	2,25	0,67	0,67	0,67	0,67	0,67
6	10	3	5,44	5,44	5,44	5,44	5,44	5,35
6	15	3	10,35	8,46	6,78	6,32	6,32	5,65
6	20	3	24,02	11,96	7,09	5,43	4,03	2,6
9	5	3	5,02	4,6	4,39	4	3,71	3,71
9	10	3	20,81	11,13	7,67	6,91	5,79	0
9	15	3	30,43	22,34	17,41	12,87	8,97	6,45
9	20	3	38,21	36,19	33,82	28,79	25,58	0
3	5	4	1,1	0,06	0,06	0,06	0,06	0,06
3	10	4	3,41	3,39	3,39	3,24	3,24	3,24
3	15	4	12,08	12,08	12,08	12,08	12,08	12,08
3	20	4	14,63	14,63	14,63	14,63	14,63	14,63
6	5	4	1,05	0,77	0,77	0,53	0,53	0,53
6	10	4	2,5	2,2	1,72	1,72	1,72	1,57
6	15	4	6,34	3,25	3,25	3,24	3,24	3,24
6	20	4	20,33	11,72	6,24	5,2	3,31	0
9	5	4	6,55	5,32	4,88	4,73	3,37	3,37
9	10	4	19,59	11,3	8,08		6,79	5,99
9	15	4	25,18	21,38	18,21	16,18	13,02	0
9	20	4	33,29	31,14	29,08	27,05	24,81	0
<b>Moyenne</b>			<b>11,75</b>	<b>9,04</b>	<b>7,85</b>	<b>6,98</b>	<b>6,23</b>	<b>3,28</b>

Le tableau 5.3 nous permet de faire une autre remarque qui porte sur l'influence des paramètres  $T, J$  et  $K$  sur le Gap. Nous constatons que le Gap de toutes les instances avec  $T = 3$  est plus élevé par rapport aux autres instances où  $T = 6$  et  $T = 9$ . Ce qui explique la difficulté de la méthodes de la recherche tabou avec les petites instances. La variation du nombre de détaillants  $J$  a une influence au début de la simulation mais à la fin (à 300 secondes) cette influence commence à disparaître. Cette méthode n'est pas aussi très sensible à la variation du nombre de véhicules.

TABLE 5.3 – Variation du Gap moyen en fonction des paramètres  $T$ ,  $J$  et  $K$ 

Paramètres	Gap							
	$1_s$	$50_s$	$100_s$	$150_s$	$200_s$	$250_s$	$300_s$	
T	3	26,78	4,06	3,97	3,97	3,95	3,95	3,95
	6	30,99	8,58	5,4	4,19	3,83	3,55	2,99
	9	33,69	22,61	17,76	15,4	13,16	11,17	2,88
J	5	24,29	3,54	2,63	2,41	2,22	2,04	2,04
	10	29,09	8,4	4,85	3,88	3,48	3,32	2,56
	15	32,16	13,31	10,88	9,35	7,88	6,5	4,47
	20	36,42	21,75	17,81	15,77	14,36	13,05	4,05
K	2	30,26	10,64	7,89	7,03	6,04	5,35	3,01
	3	32,67	12,44	9,46	8	6,93	6,1	3,09
	4	28,54	12,17	9,77	8,53	7,98	7,23	3,73
Moyenne	<b>30,48</b>	<b>11,75</b>	<b>9,04</b>	<b>7,85</b>	<b>6,98</b>	<b>6,23</b>	<b>3,28</b>	

### 5.6.2 Résultats obtenus par le recuit simulé

L'application de la méthode de recuit simulé, sur les mêmes instances utilisées déjà dans la recherche tabou, pour le calcul du Gap a donné les résultats présentés dans le Tableau 5.4. Nous remarquons que le Gap est très sensible à la variation du nombre de périodes et le nombre de détaillants. Ce qui a donné des valeurs très élevées du Gap en augmentant  $T$  et  $J$ .

### 5.6.3 Comparaison des résultats donnés par la recherche tabou et le recuit simulé

Dans l'objectif de déterminer laquelle des deux méta-heuristiques (la recherche tabou et le recuit simulé) est la mieux adaptée à notre problème, nous avons comparé les valeurs du Gap moyen en fonction du temps d'exécution obtenu pour chacune des deux méthodes.

Nous pouvons constater clairement que l'approche de recherche tabou fait un mauvais démarrage, avec un gap moyen un peu élevé, mais après, elle converge très vite, et le Gap moyen atteint la valeur de 3.28, alors que l'approche de recuit simulé démarre avec un Gap moyen acceptable, mais il stagne avec le temps.

### 5.6.4 Résultats fournis par l'approche hybride

L'approche hybride que nous avons adopté consiste à exécuter séquentiellement les deux approches de telle manière que le résultat de l'algorithme du recuit simulé serve de solution initiale à celui de la recherche tabou. Le tableau 5.5

D'après le tableau 5.5 nous pouvons conclure que l'hybridation a permis d'améliorer clairement le Gap moyen. La figure 5.5 a illustré cette amélioration en comparant l'évolution du Gap moyen des trois approches.

TABLE 5.4 – Résultats obtenus par le recuit simulé

Instances			Gap					
T	J	K	50 <sub>s</sub>	100 <sub>s</sub>	150 <sub>s</sub>	200 <sub>s</sub>	250 <sub>s</sub>	300 <sub>s</sub>
3	5	2	0,08	0,01	0,01	0,01	0,01	0,01
3	10	2	2,4	2,4	2,24	1,62	1,21	1,21
3	15	2	5,94	5,94	5,45	2,9	2,9	2,9
3	20	2	9,93	9,51	9,03	9,03	9,03	9,03
6	5	2	5,09	5,09	3,97	3,97	3,97	3,97
6	10	2	9,75	8,6	8,57	8,57	6,54	6,54
6	15	2	14,44	12,37	12,37	11,9	11,9	11,9
6	20	2	18,18	18,03	18,03	17,34	17,34	17,34
9	5	2	12,4	12,17	12,17	12,11	12,03	10,85
9	10	2	18,16	17,66	16,73	16,58	16,55	16,55
9	15	2	21,74	21,31	21,31	21,08	21,08	20,98
9	20	2	24,01	23,28	22,22	22,22	22,22	22,22
3	5	3	0,01	0,01	0,01	0,01	0	0
3	10	3	2,16	0,9	0,05	0,05	0,02	0,02
3	15	3	4,23	2,64	2,64	2,64	2,61	2,61
3	20	3	6,11	5,01	5,01	3,74	3,74	2,62
6	5	3	4,5	4,5	4,5	2,89	2,89	2,27
6	10	3	11,7	8,28	7,71	7,71	7,26	7,26
6	15	3	11,33	10,59	10,44	10,44	10,44	10,44
6	20	3	17,47	16,11	14,14	13,82	13,28	12,43
9	5	3	10,65	10,65	10,46	10,46	10,33	10,33
9	10	3	16,83	16,08	15,24	15,24	15,2	0
9	15	3	20,25	19,88	19,78	19,08	18,29	18,29
9	20	3	23,64	22,77	22,42	22,42	21,41	21,16
3	5	4	0,09	0,06	0,06	0,04	0,04	0,02
3	10	4	0,99	0,13	0,13	0,02	0,02	0,02
3	15	4	7,05	5,63	2,28	2,28	2,28	2,28
3	20	4	8,13	8,13	8,13	7,91	7,91	7,91
6	5	4	3,94	3,94	3,94	3,94	3,71	3,71
6	10	4	6,35	6,35	5,97	5,97	5,1	5,1
6	15	4	13,02	12,8	12,39	12,39	12,39	12,37
6	20	4	14,33	13,78	13,78	13,78	13,7	13,49
9	5	4	12,79	10,58	9,96	9,96	9,96	9,96
9	10	4	13,54	13,54	13,54	13,54	13,54	12,42
9	15	4	18,54	17,93	17,19	17,19	17,09	16,44
9	20	4	21,89	21,6	20,53	20,07	20,04	20,03
<b>Moyenne</b>			<b>10,88</b>	<b>10,23</b>	<b>9,79</b>	<b>9,53</b>	<b>9,33</b>	<b>8,74</b>

## CONCLUSION DU CHAPITRE 5

Dans ce chapitre nous avons proposé trois approches pour résoudre le problème d'intégration production distribution. La première est à base de la recherche tabou, la deuxième est à base du recuit simulé et la troisième est une hybridation entre les deux premières. L'analyse des différents résultats montre que la recherche tabou est meilleure que le recuit simulé, et l'hybridation des deux a donné des bons résultats. Les performances les plus remarquables de ces approches sont

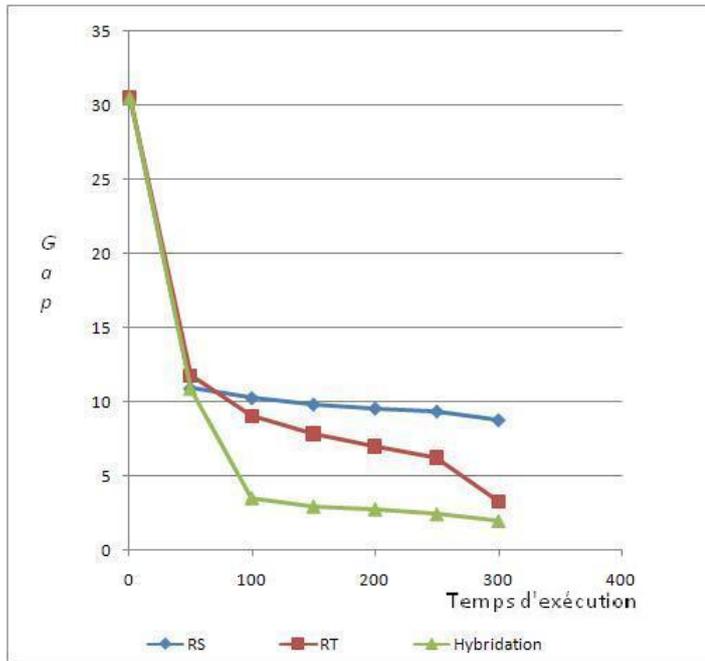


FIGURE 5.5 – Comparaison du Gap moyen avec l'hybridation

atteintes sur les instances avec un nombre de périodes élevé. Mais en terme du Gap moyen, les résultats obtenus par la première approche à base d'algorithme génétique développée dans le quatrième chapitre restent toujours les meilleurs. Cependant les algorithmes développés dans ce chapitre peuvent constituer une plate forme ou une bonne matière pour des recherches futures et la grande variabilité des résultats laisse espérer une marge de progression si on développe d'autres types de mouvements.

TABLE 5.5 – Résultats obtenus par l'hybridation Tabou-Recuit

Instances			Gap					
T	J	K	50 <sub>s</sub>	100 <sub>s</sub>	150 <sub>s</sub>	200 <sub>s</sub>	250 <sub>s</sub>	300 <sub>s</sub>
3	5	2	0,08	0,08	0,08	0,08	0,08	0,08
3	10	2	1,75	1,75	1,75	1,75	1,75	1,75
3	15	2	0,01	0,01	0,01	0,01	0,01	0,01
3	20	2	0	0	0	0	0	0
6	5	2	2,9	0,38	0,09	0,02	0,02	0,02
6	10	2	1,5	1,5	1,5	1,5	1,5	1,5
6	15	2	2,29	2,03	1,99	1,99	1,99	1,99
6	20	2	5,63	4,94	4,31	3,78	3,59	3,55
9	5	2	9,06	8,65	8,29	8,18	5,79	5,79
9	10	2	3,87	0,13	0,13	0,13	0,13	0,13
9	15	2	9,77	6,94	5,71	5,54	5,5	0
9	20	2	17,02	10,79	5,97	5,08	5,01	4,95
3	5	3	0,01	0,01	0,01	0,01	0,01	0,01
3	10	3	1,04	1,04	1,04	1,04	1,04	1,04
3	15	3	2,54	2,54	2,54	2,54	2,54	2,54
3	20	3	2,19	1,81	1,81	1,81	1,81	1,81
6	5	3	0,16	0,1	0,1	0,1	0,1	0,1
6	10	3	1,87	1,38	1,38	1,15	1,06	1,06
6	15	3	4,62	3,92	3,7	3,6	3,6	3,6
6	20	3	4,08	3,3	1,98	1,55	1,51	1,51
9	5	3	4,73	3,27	3,08	3,08	3,08	2,8
9	10	3	5,9	5,66	5,36	5,36	5,36	5,36
9	15	3	13,73	8,68	8,04	7,89	7,67	0
9	20	3	15,59	11,52	8,16	6,48	6,03	5,25
3	5	4	0,06	0,06	0,06	0,06	0,06	0,06
3	10	4	0,17	0	0	0	0	0
3	15	4	2,38	2,38	2,38	2,38	2,38	2,38
3	20	4	4,22	4,22	4,22	4,22	4,22	4,22
6	5	4	0,52	0,3	0,3	0,3	0,3	0,3
6	10	4	2,34	1,78	1,15	0,59	0,29	0,29
6	15	4	2,59	2,59	2,59	2,59	2,59	2,59
6	20	4	6,31	6,24	5,68	4,98	4,13	3,52
9	5	4	6,66	4,18	4,14	3,88	2,43	2,43
9	10	4	7,87	7,44	6,83	6,67	6,67	6,19
9	15	4	11,53	7,57	5,6	5,42	5,18	4,22
9	20	4	13,58	9,52	6,35	5,92	0	0
<b>Moyenne</b>			<b>4,68</b>	<b>3,52</b>	<b>2,95</b>	<b>2,77</b>	<b>2,43</b>	<b>1,97</b>



# CONCLUSION GÉNÉRALE

Au cours de cette thèse, nous avons étudié le problème d'intégration des fonctions production stockage et distribution dans une chaîne logistique composée d'un producteur et un ensemble de détaillants concentrés dans une région formant ce qu'on appelle un *cluster*. Le producteur est responsable de la production et de l'approvisionnement des stocks des détaillants, dans le cadre du VMI, en utilisant une flotte de véhicules de capacité limitée, afin de minimiser le coût total engendré par ces fonctions. Ce problème est un cas particulier d'un problème général qu'on appelle *Production-Inventory-Distribution Problem (PIDP)*.

Nous avons tout d'abord présenté un état de l'art sur les problèmes qui sont en étroite relation avec le notre. Nous avons commencé par le problème du lot sizing, problème élémentaire à partir duquel se sont formées plusieurs extensions. Nous avons accordé plus d'attention à l'une de ces extensions qui est le problème production-routage ou *Production Routing Problem (PRP)*. Nous avons étudié un cas particulier pratique d'un PRP, qui est le cas où les détaillants sont groupés dans une région (*cluster*). Ce problème est NP-Difficile. Nous nous sommes intéressés aux développements des méthodes de résolution efficaces pour résoudre ce problème.

Dans le chapitre 3, nous avons d'abord proposé quatre formulations de programmation linéaire en nombres entiers. Des tests préliminaires montrent que les deux formulations *Aggregated Formulation (AGF)* *Echelon Stock Formulation (ESF)* donnent des résultats plus intéressants que les deux formulations *Facility Location Formulation (FLF)* et *Shortest Path Formulation (SPF)*. Ceci nous a poussé à faire des tests très détaillés sur les deux premières formulations et de proposer dans une seconde étape des inégalités valides pour les renforcer. Certaines inégalités sont originales et d'autres sont adaptées de la littérature. L'ajout des inégalités valides pour les formulations a réduit considérablement le temps d'exécution nécessaire pour la résolution des grandes instances du problème.

Pour résoudre des problèmes de très grande taille, quatre nouvelles approches basées sur les algorithmes génétiques, ont été proposées. Nous avons d'abord effectué des tests sur des petites et moyennes instances en comparant les résultats obtenus par ces approches avec ceux obtenus par CPLEX. Les résultats montrent que ces approches donnent des solutions optimales ou proches de l'optimum. Pour le cas de très grandes instances, la première qui est une hybridation d'un algorithme génétique avec une recherche locale et la quatrième approche qui est une inter-opération entre l'algorithme génétique et la programmation linéaire en nombres entiers, permettent d'obtenir assez rapidement des solutions réalisables par rapport au CPLEX qui nécessite beaucoup de temps pour trouver une solution réalisable et pour certaines instances n'arrive pas à trouver une solution. Ces approches ont donné des solutions de bonne qualité

dans moins de 300 secondes en comparaison avec la meilleure solution et aussi avec la meilleur borne inférieure trouvée par CPLEX après 2 heures d'exécution.

Les résultats obtenus nous ont encouragé à proposer également trois approches heuristiques. La première basée sur la recherche Tabou, la deuxième sur Recuit Simulé. La troisième est une hybridation entre les deux premières. La première approche a donné des résultats intéressants mais surprenants vu que la qualité des solutions pour les instances avec un nombre de périodes élevé est meilleure que celle avec un nombre de périodes petit. La deuxième approche a donné des résultats moins bons, mais son hybridation avec la première approche a donné de bons résultats par rapport aux deux premières. Néanmoins, ces approches ont montré des faiblesses qui ouvrent la voie à de nouvelles recherches afin de les améliorer.

D'une manière générale, nous pouvons résumer les principales contributions de cette thèse dans les points suivants :

- L'étude de l'intégration des fonctions production stockage et distribution dans une chaîne logistique avec une configuration particulière (détaillants groupés ou cluster) qui est devenue très courante.

- La proposition de plusieurs formulations en PLNE et l'introduction de plusieurs inégalités valides dont certaines sont originales et d'autres sont adaptées de la littérature.

- La proposition de plusieurs approches à base de métaheuristiques dont certaines ont donné des résultats intéressants et d'autres donnent un espoir d'une grande marge de progression.

Les perspectives de recherche que nous comptons développer après la thèse et qui peuvent aussi être considérées par d'autres chercheurs sont :

- l'amélioration de l'approche polyédrale par le développement des nouvelles inégalités valides en tenant compte des caractéristiques du problème.

- le développement des nouvelles approches heuristiques et métaheuristiques pour la résolution de ce problème.

- la généralisation des approches développées aux chaînes logistiques avec plusieurs *clusters* où chaque véhicule peut livrer un ensemble de clients dans un cluster mais ne peut pas livrer les clients des autres clusters dans une tournée.

- la généralisation des approches développées aussi aux chaînes logistiques avec plusieurs produits.

---

- La prise en considération des demandes avec fenêtres de temps, ou des demandes sans split (c.a.d la demande de chaque client dans chaque période, doit être livrée intégralement).

- l'intégration des flux de retour peut être considérée. Ceci rend l'ordre de visite des clients par véhicule intéressant en raison de la quantité de retour chez chaque client et l'espace libre dans le véhicule.

- l'adaptation de ces approches aux cas des chaînes logistiques avec plusieurs centres de production.

- la prise en considération du problème de lot-sizing et l'ordonnancement dans le système de production.



# BIBLIOGRAPHIE

- [1] N. Absi, C. Archetti, S. Dauzère-Pérès, and D. Feillet. An iterative two-phase heuristic approach for the production routing problem. *Transportation Science*, 49(4) :784–795, 2014.
- [2] Y. Adulyasak, J. F. Cordeau, and J. Jans. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing*, 26(1) :103–120, 2013.
- [3] Y. Adulyasak, J. F. Cordeau, and J. Jans. Optimization-based adaptive large neighborhood search for the production routing problem. *Transportation Science*, 48(1) :20–45, 2014.
- [4] Y. Adulyasak, J. F. Cordeau, and J. Jans. The production routing problem : A review of formulations and solution algorithms. *Computers & Operations Research*, 55 :141–152, 2015.
- [5] A. Aggarwal and J.K. Park. Improved algorithms for economic lot-size problems. *Operations Research*, 41 :549–571, 1993.
- [6] A. Akbalik. *Optimisation de la gestion intégrée des flux physiques dans une chaîne logistique : extensions du problème de dimensionnement de lot*. PhD thesis, Institut National Polytechnique de Grenoble, France, 2006.
- [7] D. Aksen, K. Altinkemer, and S. Chand. The single-item lot-sizing problem with immediate lost sales. *European Journal of Operational Research*, 147(3) :558–566, 2003.
- [8] S. Anily and M. Tzur. Shipping multiple items by capacitated vehicles. *Transportation Science*, 39(2) :233–248, 2005.
- [9] C. Archetti, L. Bertazzi, G. Paletta, and M.G. Speranza. Analysis of the maximum level policy in a production-distribution system. *Computers & Operations Research*, 38(12) :1731–1746, 2011.
- [10] E. Arkin, D. Joneja, and R. Roundy. Computational complexity of uncapacitated multichelon production planning problems. *Operations Research Letters*, 8(2) :61–66, 1989.
- [11] V.A. Armentano, A.L. Shiguemoto, and A. Lokketangen. Tabu search with path relinking for an integrated production-distribution problem. *Computers & Operations Research*, 38(8) :1199–1209, 2011.
- [12] D. Atkins and P. Iyogun. A heuristic with lower bound performance guarantee for the multi-product lot-size problem. *IIE Transactions*, 20(4) :369–373, 1988.

- [13] K.R. Baker, P. Dixon, M.J. Magazine, and E.A. Silver. An algorithm for the dynamic lot-size problem with time-varying production capacity constraints. *Management Science*, 24 :1710–1720, 1978.
- [14] I. Barany, T.J. Van Roy, and L.A. Wolsey. Uncapacitated lot sizing : the convex hull of solutions. *Mathematical Programming Studies*, 22 :32–43, 1984.
- [15] J.F. Bard and N. Nananukul. The integrated production-inventory-distribution-routing problem. *Journal of Scheduling*, 12(3) :257–280, 2009.
- [16] J.F. Bard and N. Nananukul. A branch-and-price algorithm for an integrated production and inventory routing problem. *Computers & Operations Research*, 37(12) :2202–2217, 2010.
- [17] B. M. Beamon. Supply chain design and analysis : Models and methods. *International Journal of Production Economics*, 55 :281–294, 1998.
- [18] G. Belvaux and L. Wolsey. Lot-sizing problems : Modeling issues and a specialized branch and cut system bc-prod. *Management Science*, 46(5) :724–738, 2000.
- [19] L. Bertazzi, G. Paletta, and M.G. Speranza. Minimizing the total cost in an integrated vendor-managed inventory system. *Journal of Heuristics*, 11 :393–419, 2005.
- [20] R. Bhatnagar, P. Chandra, and S.K. Goyal. Models for multi-plant coordination. *European Journal of Operational Research*, 67 :141–160, 1993.
- [21] G.R. Bitran and H.H. Yanasse. Computational complexity of the capacitated lot size problem. *Management Science*, 28(10) :1174–1186, 1982.
- [22] F.F. Boctor, G. Laporte, and J. Renaud. Models and algorithms for the dynamic demand joint replenishment problem. *International Journal of Production Research*, 42(13) :2667–2678, 2004.
- [23] M. Boudia, M.A. Ould Louly, and C. Prins. A reactive GRASP and path relinking for a combined production-distribution problem. *Computers & Operations Research*, 34(11) :3402–3419, 2007.
- [24] M. Boudia and C. Prins. A memetic algorithm with dynamic population management for an integrated production-distribution problem. *European Journal of Operational Research*, 195(3) :703–715, 2009.
- [25] N. Brahimi. *Planification de la production : modèles et algorithmes pour les problèmes de dimensionnement de lots*. PhD thesis, Ecole des mines de Nantes, 2004.
- [26] N. Brahimi and T. Aouam. Multi-item production routing problem with backordering : a MILP approach. *International Journal of Production Research*, 54(4) :1076–1093, 2016.
- [27] N. Brahimi, S. Dauzère-Pérès, N.M. Najid, and A. Nordli. Single item lot sizing problems. *European Journal of Operational Research*, 168(1) :1–16, 2006.

- [28] N. Brahimi, S. Dauzère-Pérès, and N.M. Najid. Capacitated multi-item lot-sizing problems with time windows. *Operations Research*, 54(5) :951–967, 2006.
- [29] L. M. A. Chan, A. Muriel, Z. J. M. Shen, D. Simchi-Levi, and C. P. Teo. Effective zero-inventory ordering policies for the single-warehouse multi-retailer problem with piecewise linear cost structures. *Management Science*, 48 :1446–1460, 2000.
- [30] P. Chandra and M.L. Fischer. Coordination of production and distribution planning. *European Journal of Operational Research*, 72(3) :503–517, 1994.
- [31] H.D. Chen, D. Hearn, and C.Y. Lee. A new dynamic programming algorithm for the single item capacitated dynamic lot size model. *Journal of Global Optimization*, 4 :285–300, 1994.
- [32] J. Chen, W. et Thizy. Analysis of relaxation for the multi-item capacities lot-sizing problem. *Annals of Operations Research*, 26 :29–72, 1990.
- [33] Z. L. Chen. *Integrated Production and Distribution Operations : Taxonomy, Models, and Review*. In D. Simchi-Levi, S. D. Wu, & Z. J. Shen (Eds.), *Handbook of quantitative supply chain analysis : modeling in the e-business era*, pages 711–746. Kluwer, 2004.
- [34] Z. L. Chen. Integrated production and outbound distribution scheduling : Review and extensions. *Operations Research*, 58(1) :130–148, 2010.
- [35] M. G. Christopher. *Logistics and Supply Chain Management*. Pitman Publishing, 1992.
- [36] A.J. Clark and H. Scarf. Optimal policies for a multi-echelon inventory problem. *Management Science*, 6 :475–490, 1960.
- [37] M. Diaz-Madronero, D. Peidro, and J. Mula. A review of tactical optimization models for integrated production and transport routing planning decisions. *Computers & Industrial Engineering*, 88 :518–535, 2015.
- [38] P. Dixon and E. Silver. A heuristic solution procedure for the multi-item, single level, limited capacity, lot-sizing problem. *Journal of operations management*, 2 :23–39, 1981.
- [39] A. Drexl and A. Kimms. Lot sizing and scheduling : survey and extensions. *European Journal of Operations Research*, 99 :221–235, 1997.
- [40] G. Eppen and R. Martin. Solving multi-item capacitated lot-sizing problems using variable redefinition. *Operations Research*, 35(6) :832–848, 1987.
- [41] S. Erenguc. Multiproduct dynamic lot-sizing model with coordinated replenishment. *Naval Research Logistics*, 35 :1–22, 1988.
- [42] S. S. Erenguc, N. C. Simpson, and A. J. Vakharia. Integrated production/distribution planning in supply chains : An invited review. *European Journal of Operations Research*, 115 :219–236, 1999.

- [43] B. Fahimnia, R.Z. Farahani, R. Marian, and L. Luong. A review and critique on integrated production-distribution planning models and techniques. *Journal of Manufacturing Systems*, 32(1) :1–19, 2013.
- [44] A. Federgruen and M. Tzur. Simple forward algorithm to solve general dynamic lot sizing models with  $n$  periods in  $o(n \log n)$  or  $o(n)$  time. *Management Science*, 37(8) :909–925, 1991.
- [45] A. Federgruen and M. Tzur. Dynamic lot-sizing model with backlogging. a simple  $o(n \log n)$  algorithm and minimal forecast horizon procedure. *Naval Research Logistics*, 40(4) :459–478, 1993.
- [46] A. Federgruen and M. Tzur. The joint replenishment problem with the time-varying costs and demands : Efficient, asymptotic and epsilon-optimal solutions. *Operations Research*, 42 :1067–1086, 1994.
- [47] A. Federgruen and M. Tzur. Time-partitioning heuristics : Application to one warehouse, multi-item, multi-retailer lot-sizing problems. *Naval Research Logistics*, 46(5) :463–486, 1999.
- [48] M. Florian and M. Klein. Deterministic production planning with concave costs and capacity constraints. *Management Science*, 18 :18–20, 1971.
- [49] D. Forgarty and R. Barringer. Joint order release decisions under dependent demand. *Production and Inventory Management Journal*, 28(1) :55–61, 1987.
- [50] F. Fumero and C. Vercellis. Synchronized development of production, inventory, and distribution schedules. *Transportation Science*, 33(3) :330–340, 1999.
- [51] F. Galasso. *Aide à la planification dans les chaînes logistiques en présence de demande flexible*. PhD thesis, Laboratoire d’Analyse et d’Architecture des Systèmes - LAAS (Toulouse, France), 2007.
- [52] F. Galasso. *Use of Advanced Planning and Scheduling (APS) systems to support manufacturing planning and control processes*. PhD thesis, Chalmers University of Technology, Goteborg, Sweden, 2012.
- [53] L. Gelders, J. Maes, and L. Van Wassenhove. *A Branch and Bound Algorithm for the Multiitem Single Level Capacitated Dynamic Lotsizing Problem*,. in S. Axsaster et al. (eds.), *Multi-stage Production Planning and Inventory Control*, Lectures Notes in Economics and Mathematical Systems, 1986.
- [54] M. Gilbert, K. et Madan. A heuristic for a class of production planning and scheduling problems. *IIE Transactions*, 23 :282–289, 1991.
- [55] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13 :533–549, 1986.

- [56] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading MA Addison Wesley, 1989.
- [57] K. Haase and U. Kohlmorgen. Parallel genetic algorithm for the capacitated lot-sizing problem. *Operations research proceedings*, pages 370–375, 1995.
- [58] K. Hindi. Computationally efficient solution of multi-item capacitated lot sizing problems. *Computers and industrial Engineering*, 28(4) :709–719, 1995.
- [59] K. Hindi. Solving the clsp by a tabu search heuristic. *Journal of Operational Research Society*, 47(1) :151–161, 1996.
- [60] J. Holland. *Adaptation in natural and artificial systems*. University of Michigan Press, 1975.
- [61] G. Q. Huang, J. S .K. Lau, and K. L. Mak. The impacts of sharing production information on supply chain dynamics : a review of the literature. *International Journal of Production Research*, 41(7) :1483–1517, 2003.
- [62] D. Huisman, Peeters Jans, R., and A. M. Wagelmans. Combining column generation and lagrangian relaxation. *Technical report*, 2003.
- [63] R. Jans and Z. Degraeve. Modeling industrial lot sizing problems : a review. *International Journal of Production Research*, 46(6) :1619–1643, 2008.
- [64] W. Jaruphongsa, S. Cetinkaya, and C.-Y. Lee. Outbound shipment mode considerations for integrated inventory and delivery lot-sizing decisions. *Operations Research Letters*, 35(6) :813–822, 2007.
- [65] J.H. Kang and Y.D. Kim. Coordination of inventory and transportation managements in a two-level supply chain. *International Journal of Production Economics*, 123(1) :137–145, 2010.
- [66] E. Kao. A multi-product dynamic lot-size model with individual and joint set-up costs. *Operations Research*, 27(2) :279–289, 1979.
- [67] B. Karimi, S.M.T Fatemi Ghomi, and J.M. Wilson. The capacitated lot sizing problem : a review of models and algorithms. *Omega*, 31(5) :365–378, 2003.
- [68] Ö. Kirca. A primal-dual algorithm for the dynamic lot sizing problem with joint set-up costs. *Naval Research Logistics*, 42(5) :791–806, 1995.
- [69] J. Krarup and O. Bilde. *Optimierung bei graphentheoretischen and ganzzahligen probleme*, chapter Plant location, set covering and economic lot sizes : An  $O(mn)$ -algorithm for structured problems, pages 155–180. Collatz, L. et al., Birkhauser Verlag, Basel, 1977.
- [70] C.Y. Lee, S. Cetinkaya, and W. Jaruphongsa. A dynamic model for inventory lot sizing and outbound shipment scheduling at a third-party warehouse. *Operations Research*, 51(5) :735–747, 2003.

- [71] L.L. Lee and C. Billington. The evolution of supply chain management models and practice at hewlett-packard. *INTERFACES*, 25(5) :42–63, 1995.
- [72] L. Lei, Q. Wang, and C. Fan. Optimal business policies for a supplier-transporter-buyer channel with a price-sensitive demand. *Journal of the Operational Research Society*, 57 :281–289, 2006.
- [73] D. Lemoine. *Modèles Génériques et Méthodes de Résolution pour la Planification Tactique Mono-site et Multi-site*. PhD thesis, Université Blaise Pascal - Clermont-Ferrand II, 2008.
- [74] J. Leung, T. Magnanti, and R. Vachani. Facets and algorithms for the capacitated lot sizing. *Mathematical programming*, 45 :331–359, 1989.
- [75] R. Levi, R. Roundy, D.B. Shmoys, and M. Sviridenko. A constant approximation algorithm for the one-warehouse multiretailer problem. *Management Science*, 54 :763–776, 2008.
- [76] M. Loparic, H. Marchand, and L.A. Wolsey. Dynamic knapsack sets and capacitated lotsizing. *Mathematical Programming*, 95(1) :53–69, 2003.
- [77] V. Lotfi and Y.S. Yoon. An algorithm for the single-item capacitated lot-sizing problem with concave production and holding costs. *Journal of Operational Research Society*, 45(8) :934–941, 1994.
- [78] L. Maes, J. et Van Wassenhove. Multi-item single-level capacitated dynamic lot-sizing heuristics : A general review. *Journal of the Operational Research*, 39(11) :991–1004, 1988.
- [79] R.A. Melo and L.A. Wolsey. Uncapacitated two-level lot-sizing. *Operations Research Letters*, 38(4) :241–245, 2010.
- [80] R.A. Melo and L.A. Wolsey. MIP formulations and heuristics for two-level production-transportation problems. *Computers & Operations Research*, 39(11) :2776–2786, 2012.
- [81] J.T. Mentzer, W. Dewitt, J.S. Keebler, S. Min, N.W. Nix, C.D. Smith, and Z.G. Zacharia. Defining the supply chain management. *Journal of Business logistics*, 22(2) :1–20, 2001.
- [82] A.J. Miller, G.L. Nemhauser, and M.W. Savelsbergh. On the capacitated lot-sizing and continuous 0-1 knapsack polyhedra. *European Journal of Operational Research*, 125(2) :298–315, 2000.
- [83] A. Narayan and P. Robinson. Efficient and effective heuristics for the coordinated capacitated lot-size problem. *European Journal of Operational Research*, 203(3) :583–592, 2010.
- [84] Y. Pochet and L. Wolsey. Solving multi-item lot-sizing problems using strong cutting planes. *Management Science*, 37 :53–67, 1991.
- [85] Y. Pochet and L. A. Wolsey. Single item lot-sizing with non-decreasing capacities. *Mathematical Programming*, 121(1) :123–143, 2010.

- [86] Y. Pochet and L.A. Wolsey. Polyhedra for lot-sizing with Wagner-Whitin costs. *Mathematical Programming*, 67 :297–323, 1994.
- [87] Y. Pochet and L.A. Wolsey. *Production Planning by Mixed Integer Programming*. Springer, 2006.
- [88] P. Raghavan. *The multi-item lot sizing problem with joint replenishments*. PhD thesis, Leonard N. Stern School of Business, New Yoek University, 1993.
- [89] P. Raghavan and M. Rao. The multi-item lot sizing problem with joint replenishment : A polyhedral approach. Technical Report SOR-91-8, Leonard N. Stern School of Business, New York University, 1991.
- [90] P. Raghavan and M. Rao. Formulations for multi-item lot sizing problem with joint replenishment. Technical Report SOR-92-19, Leonard N. Stern School of Business, New York University, 1992.
- [91] E.P. Robinson and L. Gao. A dual ascent procedure for multi-product dynamic demand coordinated replenishment with backlogging. *Management Science*, 42(11) :1556–1564, 1996.
- [92] E.P. Robinson and F.B. Lawrence. Coordinated capacitated lot-sizing problem with dynamic demand : A lagrangian heuristic. *Decision Sciences Journal*, 35(1) :25–54, 2004.
- [93] P. Robinson, A. Narayanan, and F. Sahin. Coordinated deterministic dynamic demand lot sizing problem : a review of models and algorithms. *Omega*, 37(1) :3–15, 2009.
- [94] J. Rohde, H. Meyr, and M. Wagner. Die supply chain planning matrix. *PPS Management*, 5 :10–15, 2000.
- [95] A. Rouibi. *Impact du partage d'informations et du vendor managed inventory sur la performance des chaines logistiques*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne, France, 2012.
- [96] M. Ruokokoski, O. Solyali, J.F. Cordeau, R. Jans, and H. Süral. Efficient formulations and a branch-and-cut algorithm for a production-routing problem. Technical report, Les Cahiers du GERAD-G-2010-66, HEC Montréal, Canada, 2010.
- [97] M. Salomon. *Multi-stage Production Planning and Inventory Control*, volume 355. Lectures Notes in Economics and Mathematical Systems, 1991.
- [98] M. Salomon, R. Kuik, and L. VanWassenhove. Statistical search methods for lot-sizing problems. *Annals Operations Research*, 41 :453–468, 1993.
- [99] A.M. Sarmiento and R. Nagi. A review of integrated analysis of production-distribution systems. *IIE Transactions*, 11(3) :1061–1074, 1999.
- [100] J. F. Shapiro. *Modelling the Supply Chain*. Duxbury Press, 2001.

- [101] D.X. Shaw and A.P. Wagelmans. An algorithm for single-item capacitated economic lot sizing with piecewise linear production costs and general holding costs. *Management Science*, 44(6) :831–838, 1998.
- [102] A.L. Shiguemoto and V.A. Armentano. A tabu search procedure for coordinating production, inventory and distribution routing problems. *International Transactions in Operational Research*, 17(2) :179–195, 2010.
- [103] E. Silver. Coordinated replenishments of items under time varying demand : Dynamic programming formulation. *Naval Research Logistics*, 26(1) :141–151, 1979.
- [104] E. Silver and P. Kelle. More on joint order release decisions under dependent demand. *Production and Inventory Management Journal*, 29(2) :71–72, 1988.
- [105] D. Simchi-Levi, P. Kaminsky, and E.S. Levi. *Designing and Managing the supply Chain : Concepts, Strategies and case studies*. Mc Graw Hill, 2003.
- [106] O. Solyali and H. Süral. The one-warehouse multi-retailer problem : reformulation, classification, and computational results. *Annals of Operations Research*, 196(1) :517–541, 2012.
- [107] H. Stadtler and C. Kilger. *Supply Chain Management and Advanced Planning*. Springer, 2005.
- [108] A.T. Staggemeier and A.R. Clark. A survey of lot-sizing and scheduling models. *23rd Annual Symposium of the Brazilian Operational Research Society*, 10 :938–947, 2001.
- [109] L. Thizy, J. et Van Wassenhove. Lagrangean relaxation for the multi-item capacitated lotsizing problem : a heuristic implementation. *IIE Transactions*, 17 :308–313, 1985.
- [110] D. Thomas and P. Griffin. Coordinated supply chain management. *European Journal of Operational Research*, 94 :1–15, 1996.
- [111] C.P.M. van Hoesel and A.P.M. Wagelmans. An  $O(T^3)$  algorithm for the economic lot-sizing problem with constant capacities. *Management Science*, 42(1) :142–150, 1996.
- [112] S. van Hoesel, H.E. Romeijn, D.R. Morales, and A.P.M. Wagelmans. Integrated lot sizing in serial supply chains with production capacities. *Management Science*, 51(11) :1706–1719, 2005.
- [113] A.F. Veinott. Minimum concave-cost solution of leontief substitution models of multifacility inventory systems. *Operations Research*, 17(2) :262–291, 1969.
- [114] A. Wagelmans, S. Vanhoesel, and A. Kolen. Economic lot sizing an  $o(n \log n)$  algorithm that runs in linear time in the wagner-whitin case. *Operations Research*, 40(11) :145–156, 1992.
- [115] H.M. Wagner and T.M. Whitin. Dynamic version of the economic lot size model. *Management Science*, 5(1) :89–96, 1958.

- 
- [116] C. Wolosewicz. *Approche intégrée en planification et ordonnancement de la production*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne, 2008.
- [117] L.A. Wolsey. Solving multi-item lot-sizing problems with an mip solver using classification and reformulation. *Management Science*, 48(12) :1587–1602, 2002.
- [118] Pochet Y. Valid inequalities and separation for capacitated economic lot-sizing. *Operations Research Letters*, 7 :109–116, 1988.
- [119] J. Yue and A. Muriel. Single-warehouse multi-retailer inventory systems with full truckload shipments. *Naval Research Logistics (NRL)*, 56(5) :450–464, 2009.
- [120] W. Zangwill. A deterministic multiproduct multifacility production and inventory model. *Operations Research*, 14(3) :486–507, 1966.
- [121] W. Zangwill. A backloging model and a multi-echelon model of a dynamic economic lot size production system : a network approach. *Management Science*, 15 :506–527, 1969.
- [122] Y.Z. Zhang, R.Z. Yan, and W. Gao. An dynamic lot-sizing model with multi-mode shipments. In *The Eighth International Symposium on Operations Research and Its Applications (ISORA '09)*, volume 10, 2009.



