



**HAL**  
open science

# Etude et résolution de problèmes de planification dans des réseaux logistiques multi-échelons

Sona Kande

► **To cite this version:**

Sona Kande. Etude et résolution de problèmes de planification dans des réseaux logistiques multi-échelons. Gestion et management. Université de Technologie de Troyes, 2015. Français. NNT : 2015TROY0015 . tel-03359609

**HAL Id: tel-03359609**

**<https://theses.hal.science/tel-03359609>**

Submitted on 30 Sep 2021

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

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

Thèse  
de doctorat  
de l'UTT

**Sona KANDE**

**Etude et résolution de problèmes  
de planification  
dans des réseaux logistiques  
multi-échelons**

**Spécialité :**  
**Optimisation et Sûreté des Systèmes**

2015TROY0015

Année 2015

---

---

# THESE

*pour l'obtention du grade de*

## DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES Spécialité : OPTIMISATION ET SURETE DES SYSTEMES

*présentée et soutenue par*

**Sona KANDE**

*le 12 juin 2015*

---

---

### **Etude et résolution de problèmes de planification dans des réseaux logistiques multi-échelons**

---

---

### JURY

Mme N. SAUER	PROFESSEUR DES UNIVERSITES	Président
Mme L. BELGACEM	DOCTEUR, INGENIEUR DE RECHERCHE	Examinateur
Mme C. GUERET	PROFESSEUR DES UNIVERSITES	Examinateur
M. C. PRINS	PROFESSEUR DES UNIVERSITES	Directeur de thèse
M. M. SEVAUX	PROFESSEUR DES UNIVERSITES	Rapporteur
M. N. TCHERNEV	PROFESSEUR DES UNIVERSITES	Rapporteur

---

## Remerciements

J'adresse mes remerciements à Christian Prins et Lucile Belgacem, pour m'avoir encadré au cours de ces trois années, pour leurs conseils et leur soutien dans les moments les plus difficiles. Je tiens également à remercier Jean Philippe Casal, sans qui cette thèse n'aurait pas eu lieu, et qui m'a apporté une aide précieuse. Je vous remercie, tous les trois, pour votre confiance.

J'adresse également mes remerciements aux rapporteurs, Nikolay Tchernev et Marc Sevaux, pour leur intérêt vis-à-vis de mon travail, et pour avoir minutieusement étudié ce manuscrit, ainsi qu'à tous les membres du jury, notamment Christelle Gueret et Nathalie Sauer.

Je remercie la direction de l'entreprise FuturMaster, pour sa confiance et son soutien financier. Mes remerciements vont également à l'ensemble des collaborateurs de Futurmater, pour leur convivialité et leur soutien. J'exprime ma profonde gratitude spécialement à Ludovic Cheng et Thibault Lhermitte, qui m'ont permis de tenir et de finaliser ce travail. C'est un véritable plaisir de travailler à vos côtés, et j'espère que ce sera le cas pour plusieurs années encore.

Je remercie également tous mes collègues du laboratoire pour leur soutien et leur bonne humeur. Je remercie Matthieu Le Berre, Elyn Solano Charris, Syrine Roufaïda Ait Haddadene et Birome Holo Ba.

Je dédie ce manuscrit à mes chers et valeureux parents. Je ne les remercierai jamais assez pour tous les sacrifices consentis à notre égard, moi et mes frères et sœurs : Mama, Pape, Inna et Ibrahima ; sans qui ma vie ne serait pas aussi riche et joyeuse. Ma reconnaissance va à celui qui a assuré le soutien affectif de ce travail. Mon cher et tendre, Mohamed Amadou Diop : « ceci est aussi le fruit de tes efforts et surtout merci de me rendre aussi heureuse et épanouie ».

---

# Étude et résolution de problèmes de planification dans des réseaux logistiques multi-échelons

---

## Résumé

Les travaux de cette thèse concernent la résolution d'un problème de planification dans un réseau de distribution à deux échelons intégrant la gestion des stocks de produits périssables, le dimensionnement de lots, des alternatives d'approvisionnement. La livraison s'effectue directement entre un fournisseur et son client, sans tournées (de véhicules) avec une flotte homogène de véhicules. Nous proposons un programme linéaire mixte, une heuristique constructive (déterministe) et une heuristique réactive randomisée. Pour certaines instances, le solveur de programme linéaire mixte ne fournit pas une bonne solution réalisable dans la limite de temps définie ou prend beaucoup de temps. Les heuristiques proposées sont rapides mais ne donnent pas de bonnes solutions pour certaines instances. Pour améliorer la qualité des solutions des heuristiques, une descente à voisinage variable (VND), une recherche locale itérative (ILS) et une recherche locale itérative à démarrages multiples (MS-ILS) sont développées. Toutes ces méthodes ont été incluses dans un APS (Advanced Planning System) et sont comparées avec CPLEX sur des instances extraites de bases de données réelles. Un générateur aléatoire d'instances est conçu pour plus de diversité pour les tests. Une relaxation Lagrangienne est implémentée pour comparer les solutions des instances, pour lesquelles CPLEX ne fournit pas une bonne solution réalisable dans le temps imparti, avec les autres méthodes. Une heuristique Lagrangienne, utilisant la relaxation Lagrangienne et une heuristique de réparation, est également développée.

**Mots clefs** : Optimisation combinatoire, Algorithmes d'approximation, Algorithmes optimaux, Logistique (organisation), Planification

---

**Study and solving planning problems in multi-echelon supply networks**

---

**Abstract**

This work presents a planning problem in a distribution network incorporating a two levels inventory management problem of perishable products, lot-sizing, multi-sourcing and transport capacity with a homogeneous fleet of vehicles. A mixed integer linear programming (MILP) a greedy heuristic and a reactive randomized heuristic are developed to solve this real planning problem. There are some instances for which the solver CPLEX cannot give a good upper bound within the time limit and for other instances it takes a lot of time to solve the MILP. The heuristics are alternatives to CPLEX to quickly solve some large instances taking into account original and difficult constraints. For some instances the gap between the solutions of the solver (MILP) and the heuristics becomes quite significant. A variable neighborhood descent (VND), an iterated local search (ILS) and a multi-start iterated local search (MS-ILS) are implemented. These methods are included in an APS (Advanced Planning System) and compared with a MILP solver. The instances are derived from actual data or built using a random generator of instances to have a wider diversity for computational evaluation. A lagrangian relaxation is developed to compare the solutions of the instances, for which CPLEX cannot give a good upper bound within the time limit, with the other methods (greedy heuristic, VND, ILS and MS-ILS). A lagrangian heuristic is proposed ; the solution of lagrangian relaxation is used to build a feasible solution with a repair heuristic.

**Keywords** : Combinatorial optimization, Approximation algorithms, Exact methods, Business logistics, Planning



# Table des matières

<b>1</b>	<b>Introduction générale</b>	<b>15</b>
1.1	Enjeux . . . . .	15
1.2	Plan . . . . .	17
<b>2</b>	<b>État de l’art sur la planification de la chaîne logistique</b>	<b>19</b>
2.1	Introduction . . . . .	19
2.2	Les méthodes d’optimisation utilisées dans notre étude . . . . .	19
2.2.1	La relaxation Lagrangienne . . . . .	19
2.2.2	Méthodes approchées . . . . .	21
2.3	Gestion des stocks . . . . .	25
2.3.1	Politique de stock multi-échelons . . . . .	25
2.3.2	Gestion de produits périssables . . . . .	26
2.4	Dimensionnement de lots ( <i>lot-sizing</i> ) . . . . .	28
2.5	Planification de la distribution . . . . .	31
2.5.1	Modes de distribution . . . . .	31
2.5.2	Alternative d’approvisionnement : <i>multi-sourcing</i> . . . . .	35
2.5.3	DRP : <i>Distribution Requirements Planning</i> . . . . .	36
2.6	Gestion intégrée de la production et de la distribution . . . . .	36
2.6.1	Modélisation . . . . .	37
2.6.2	Résolution . . . . .	38
2.7	Conclusion . . . . .	41

<b>3</b>	<b>Modèle mathématique et heuristique constructive</b>	<b>43</b>
3.1	Introduction . . . . .	43
3.2	Description du problème . . . . .	44
3.3	Modèle linéaire mixte . . . . .	46
3.3.1	Notations . . . . .	46
3.3.2	Fonctions pour les coûts de stock . . . . .	49
3.3.3	Fonction-objectif et contraintes . . . . .	50
3.4	Heuristique constructive . . . . .	54
3.4.1	Approvisionnement minimum pour éviter la rupture . . . . .	55
3.4.2	Approvisionnement pour atteindre le stock objectif . . . . .	56
3.4.3	Déploiement du surplus des entrepôts . . . . .	57
3.5	Tests . . . . .	59
3.5.1	Instances de test . . . . .	59
3.5.2	Comparaison de l'heuristique avec le programme linéaire mixte . . . . .	60
3.5.3	Impacts des critères de priorité . . . . .	61
3.6	Conclusion . . . . .	64
<b>4</b>	<b>Heuristique réactive randomisée</b>	<b>65</b>
4.1	Introduction . . . . .	65
4.2	Méthode de randomisation . . . . .	66
4.2.1	Randomisation de la liste des couples produit-entrepôt . . . . .	66
4.2.2	Randomisation de la liste des couples produit-dépôt . . . . .	66
4.3	Heuristique Réactive Randomisée (HRR) . . . . .	67
4.4	Tests . . . . .	68
4.4.1	Instances de test . . . . .	69
4.4.2	Résultats de CPLEX et de l'heuristique déterministe pour les instances générées aléatoirement . . . . .	72
4.4.3	Résultats de l'heuristique réactive randomisée pour toutes les instances . . . . .	74
4.4.4	Analyse du grand écart moyen à l'optimum . . . . .	77
4.5	Conclusion . . . . .	77

<b>5</b>	<b>Métaheuristiques à base de voisinages</b>	<b>83</b>
5.1	Introduction . . . . .	83
5.2	Descente à voisinage variable . . . . .	84
5.2.1	Mouvements appliqués . . . . .	84
5.2.2	Réduction de l'espace d'exploration . . . . .	86
5.2.3	Tests . . . . .	88
5.3	Recherche locale itérative (Iterated local search - ILS) . . . . .	94
5.3.1	Perturbation . . . . .	94
5.3.2	Tests . . . . .	96
5.4	Recherche locale itérative à démarrages multiples (Multi-start iterated local search - MS-ILS) . . . . .	108
5.4.1	Implémentation . . . . .	108
5.4.2	Tests . . . . .	109
5.5	Conclusion . . . . .	113
<b>6</b>	<b>Relaxation Lagrangienne</b>	<b>127</b>
6.1	Introduction . . . . .	127
6.2	Tests préliminaires . . . . .	127
6.3	Principe de la relaxation Lagrangienne . . . . .	130
6.3.1	Fonction de Lagrange . . . . .	130
6.3.2	Méthode de sous-gradient . . . . .	130
6.4	Heuristique Lagrangienne . . . . .	134
6.4.1	Algorithme de l'heuristique Lagrangienne . . . . .	134
6.4.2	Heuristique de réparation . . . . .	135
6.5	Dualisation de la contrainte de taille de lot . . . . .	137
6.5.1	Conservation de l'égalité . . . . .	139
6.5.2	Transformation en deux inégalités . . . . .	139
6.6	Tests de dualisation de la contrainte de taille de lot . . . . .	139
6.6.1	Évaluation des instances non résolues pour le problème primal . . . . .	140
6.6.2	Évaluation de l'ensemble des instances générées aléatoirement . . . . .	144
6.7	Conclusion . . . . .	151

<b>7</b>	<b>Cas d'application : planification de la distribution de produits cosmétiques</b>	<b>157</b>
7.1	Introduction . . . . .	157
7.2	Présentation du contexte . . . . .	158
7.2.1	Le projet DRP Monde . . . . .	158
7.2.2	Les objectifs et les enjeux du projet . . . . .	159
7.3	Présentation de FuturMaster . . . . .	159
7.3.1	Définition et rôle d'un APS . . . . .	159
7.3.2	Quelques mots sur la société . . . . .	160
7.3.3	Les solutions standards FuturMaster . . . . .	161
7.4	Mise en œuvre d'un outil d'optimisation : de la théorie à la pratique . . . . .	162
7.4.1	Le recueil du besoin . . . . .	162
7.4.2	Appropriation de l'outil et accompagnement au changement . . . . .	163
7.4.3	Les difficultés liées aux données . . . . .	164
7.5	Apports de l'heuristique constructive . . . . .	165
7.6	Conclusion . . . . .	165
<b>8</b>	<b>Conclusion générale</b>	<b>167</b>
8.1	Contributions . . . . .	167
8.2	Perspectives . . . . .	169

# Table des figures

3.1	Réseau de distribution à deux échelons avec alternatives . . . . .	45
3.2	Politique de stock et coûts associés (la rupture est ajoutée pour les dépôts) . . . . .	50
3.3	Variabes pour linéariser les fonctions de coût de stock . . . . .	50
3.4	Algorithme . . . . .	58
4.1	Politique de stock et coûts associés (la rupture est ajoutée pour les dépôts) . . . . .	70
4.2	Interface du générateur aléatoire d'instances . . . . .	72
4.3	Écarts de H et HRR par rapport à LB pour les 20 instances des bases de données . . . . .	75
4.4	Écarts de H et HRR par rapport à LB pour les 40 instances du générateur . . . . .	76
4.5	Part de meilleures solutions fournies par réglage . . . . .	76
5.1	Orientation de l'exploration du voisinage . . . . .	87
5.2	Comparaison % Écart VND/LB et % Écart H/LB pour les instances issues des bases de données 1 et 2 . . . . .	89
5.3	Comparaison % Écart VND/LB et % Écart H/LB pour les instances générées aléatoirement . . . . .	90
5.4	Comparaison % Écart ILS/LB and % Écart VND/LB pour les instances générées aléatoirement . . . . .	98
6.1	Ratio moyen $LB_{primal}/LB$ : règle 1 utilisée pour la mise à jour du pas . . . . .	145
6.2	Écart moyen $HRe/LB_{primal}$ : règle 1 utilisée pour la mise à jour du pas . . . . .	145
6.3	Pourcentage du cumul de la durée moyenne : règle 1 utilisée pour la mise à jour du pas	145
6.4	Ratio moyen $LB_{primal}/LB$ : règle 2 utilisée pour la mise à jour du pas . . . . .	146
6.5	Écart moyen $HRe/LB_{primal}$ : règle 2 utilisée pour la mise à jour du pas . . . . .	146
6.6	Pourcentage du cumul de la durée moyenne : règle 2 utilisée pour la mise à jour du pas	147

6.7	Ratio moyen $LB_{\text{primal}}/LB$ : règle 3 utilisée pour la mise à jour du pas . . . . .	148
6.8	Écart moyen $HRe/LB_{\text{primal}}$ : règle 3 utilisée pour la mise à jour du pas . . . . .	148
6.9	Pourcentage du cumul de la durée moyenne : règle 3 utilisée pour la mise à jour du pas	148
6.10	Ratio moyen $LB_{\text{primal}}/LB$ : règle 4 utilisée pour la mise à jour du pas . . . . .	149
6.11	Écart moyen $HRe/LB_{\text{primal}}$ : règle 4 utilisée pour la mise à jour du pas . . . . .	150
6.12	Pourcentage du cumul de la durée moyenne : règle 4 utilisée pour la mise à jour du pas	150
7.1	Périmètre du projet DRP Monde . . . . .	158
7.2	Couverture fonctionnelle des APS . . . . .	159
7.3	Les activités de FuturMaster . . . . .	160
7.4	L'offre FuturMaster . . . . .	161
7.5	Efficacité de l'outil de planification . . . . .	163

# Liste des tableaux

2.1	Classification des articles sur les problèmes de <i>lot-sizing</i> . . . . .	31
2.2	Classification des articles selon le mode de distribution . . . . .	35
2.3	Classification des articles sur la planification intégrée de la production et de la distribution . . . . .	40
3.1	Description des instances du premier et du deuxième réseau . . . . .	60
3.2	Comparaison Heuristique-PL mixte . . . . .	62
3.3	Impact des critères de priorité . . . . .	63
4.1	Liste des réglages utilisés pour l’heuristique réactive randomisée . . . . .	69
4.2	Classification des produits . . . . .	71
4.3	Paramètres de génération aléatoire d’instances . . . . .	73
4.4	Résultats de CPLEX et de l’heuristique déterministe pour les instances générées aléatoirement . . . . .	79
4.5	Résultats de HRR pour les 20 instances des bases de données . . . . .	80
4.6	Résultats de HRR pour les 40 instances générées aléatoirement . . . . .	81
4.7	Synthèse des résultats de HRR . . . . .	82
4.8	Part de meilleures solutions de HRR fournies par réglage . . . . .	82
5.1	Ordre d’application des mouvements par degré d’efficacité . . . . .	87
5.2	Résultats de la VND pour les instances issues des bases de données 1 et 2 . . . . .	89
5.3	Résultats de la VND pour les instances générées aléatoirement . . . . .	91
5.4	Synthèse des résultats de la VND (avec réduction de l’espace d’exploration) . . . . .	92
5.5	Synthèse des résultats de la VND* (sans réduction de l’espace d’exploration) . . . . .	92

5.6	Synthèse des résultats VND <sub>oam</sub> avec un ordre aléatoire d'application des mouvements : test 1 . . . . .	93
5.7	Synthèse des résultats VND <sub>oam</sub> avec un ordre aléatoire d'application des mouvements : test 2 . . . . .	93
5.8	Résultats ILS pour les instances issues des bases de données 1 et 2 . . . . .	97
5.9	Résultats ILS pour les instances générées aléatoirement . . . . .	99
5.10	Synthèse des résultats de l'ILS avec 10 itérations (avec réduction de l'espace d'exploration) . . . . .	100
5.11	Synthèse des résultats de l'ILS* avec 10 itérations (sans réduction de l'espace d'exploration) . . . . .	100
5.12	Synthèse des résultats de l'ILS <sub>VND<sub>oam</sub></sub> (10 itérations) avec un ordre aléatoire d'application des mouvements pour chaque exécution de la VND . . . . .	102
5.13	Synthèse des résultats de l'ILS <sub>oam</sub> (10 itérations) avec un seul ordre aléatoire d'application des mouvements pour toutes les exécutions de la VND . . . . .	102
5.14	Résultats de l'ILS avec 20 et 30 itérations pour les instances issues des bases de données 1 et 2 . . . . .	104
5.15	Résultats de l'ILS avec 20 et 30 itérations pour les instances générées aléatoirement . . . . .	105
5.16	Synthèse des résultats de l'ILS avec 20 et 30 itérations . . . . .	106
5.17	Synthèse des résultats ILS avec 10 itérations et 5% de taux de perturbation . . . . .	107
5.18	Synthèse des résultats ILS avec 10 itérations et 20% de taux de perturbation . . . . .	107
5.19	Résultats MS-ILS pour les instances issues des bases de données 1 et 2 . . . . .	110
5.20	Résultats MS-ILS pour les instances générées aléatoirement . . . . .	112
5.21	Synthèse résultats MS-ILS . . . . .	113
5.22	Résultats VND <sub>oam</sub> pour les instances issues des bases de données 1 et 2 : test 1 . . . . .	114
5.23	Résultats VND <sub>oam</sub> pour les instances issues des bases de données 1 et 2 : test 2 . . . . .	115
5.24	Résultats VND <sub>oam</sub> pour les instances générées aléatoirement : test 1 . . . . .	116
5.25	Résultats VND <sub>oam</sub> pour les instances générées aléatoirement : test 2 . . . . .	117
5.26	Résultats ILS <sub>VND<sub>oam</sub></sub> pour les instances issues des bases de données 1 et 2 . . . . .	118
5.27	Résultats ILS <sub>VND<sub>oam</sub></sub> pour les instances générées aléatoirement . . . . .	119
5.28	Résultats ILS <sub>oam</sub> pour les instances issues des bases de données 1 et 2 . . . . .	120
5.29	Résultats ILS <sub>oam</sub> pour les instances générées aléatoirement . . . . .	121

5.30	Résultats ILS avec 5% de taux de perturbation pour les instances issues des bases de données 1 et 2 . . . . .	122
5.31	Résultats ILS avec 20% de taux de perturbation pour les instances issues des bases de données 1 et 2 . . . . .	123
5.32	Résultats ILS avec 5% de taux de perturbation pour les instances générées aléatoirement	124
5.33	Résultats ILS avec 20% de taux de perturbation pour les instances générées aléatoirement . . . . .	125
6.1	Synthèse des résultats de la relaxation des contraintes difficiles . . . . .	129
6.2	Résultats de CPLEX pour le problème primal . . . . .	141
6.3	Résultats de la relaxation de la contrainte de taille de lot . . . . .	141
6.4	Comparaison résultats heuristique réparation, VND, ILS et MS-ILS . . . . .	143
6.5	Tests sur la relaxation de la contrainte de capacité de transport . . . . .	152
6.6	Tests sur la relaxation de la contrainte de quantité minimale de transport . . . . .	153
6.7	Tests sur la relaxation de la contrainte de contrat de fraîcheur . . . . .	154
6.8	Tests sur la relaxation de la contrainte d'ordre imposé . . . . .	155
6.9	Tests sur la relaxation de la contrainte de taille de lot de transport . . . . .	156



# Chapitre 1

## Introduction générale

Cette thèse a été effectuée en collaboration avec la société FuturMaster, conceptrice d'un logiciel APS (*Advanced Planning System*) qui utilise la programmation linéaire pour calculer des solutions pour des problèmes de planification de la distribution. Le type de réseau étudié a été rencontré ces dernières années chez plusieurs clients de FuturMaster, notamment dans la distribution de produits lactés ou de boissons, et avec des cas pouvant atteindre de grandes tailles. Il a donc été décidé de développer un programme linéaire mixte spécifique pour ce type de réseau et d'inclure dans le logiciel une heuristique rapide, permettant d'obtenir des solutions quand le solveur n'arrive pas à trouver des solutions optimales dans des temps acceptables. Même pour des cas à portée du solveur, certains utilisateurs ne comprennent pas la construction du résultat et préfèrent une heuristique dont la logique peut être expliquée.

### 1.1 Enjeux

D'un point de vue scientifique, l'enjeu de cette thèse est d'apporter des avancées dans le domaine de l'optimisation de la chaîne logistique en étudiant la modélisation de ces problèmes, avec la prise en compte de nouvelles contraintes. Il s'agit des contraintes liées à la durée de vie des produits et à des contrats de fraîcheur. Ce type de contraintes commence à être étudié dans la littérature, mais de façon assez récente. Cela nécessite de prendre en compte une dimension supplémentaire permettant d'associer à chaque quantité de produit sur une étape du flux logistique et à chaque période une information de dates (date de production ou d'emballage, date limite d'utilisation, etc.). Cette gestion ajoute non seulement de la complexité au problème initial (l'optimisation ne porte plus uniquement sur les quantités à approvisionner mais aussi sur une notion de qualité des produits et des stocks), mais représente aussi un défi algorithmique pour stocker et exploiter au mieux ces informations.

Un autre aspect intéressant concerne la modélisation et la prise en compte des différents objectifs de la planification. Comment définir un plan de bonne qualité ? Dans la pratique et dans le contexte

d'un outil d'aide à la décision, cette qualité sera au final déterminée par les différents acteurs de la chaîne logistique : planificateurs, gestionnaires de stock, ordonnanceurs, transporteurs et commerciaux. Certains analyseront des critères bien définis, d'autres auront une vision plus subjective. Dans tous les cas chacun de ces acteurs possède ses propres critères et analysera tout ou seulement une partie du plan.

Cette complexité est amplifiée par le fait que nous considérons des réseaux multi-échelons, dans lesquels chaque niveau possède ses propres règles de gestion (stock de sécurité, niveau de stock maximum, minimum), avec une notion de risque associée au stockage d'un produit. La montée en stock peut en effet générer un risque de dégagement (perte du produit arrivé à péremption) et la descente du niveau de stock peut générer un risque de rupture (non-respect des demandes clients). Ces risques peuvent être évalués différemment en fonction du produit, du lieu de stockage, de la période. Pour que le plan fourni soit correct, il est nécessaire de quantifier tous ces éléments et de prioriser les différentes décisions : avantager un produit par rapport à un autre, ou un site par rapport à un autre, ou au contraire essayer d'équilibrer les niveaux de stock, décider si un risque de dégagement sur un produit est pire ou meilleur qu'un risque de rupture sur un autre produit.

Un autre défi que constitue cette thèse réside dans la complexité et la taille des problèmes à traiter. Nous nous intéressons à la résolution de problèmes réels d'optimisation de la chaîne logistique. Ces problèmes réels possèdent beaucoup de contraintes, et reposent en général sur des réseaux logistiques complexes en termes de flux et de multiplicité des niveaux. Les méthodes de résolution proposées doivent donc être suffisamment génériques pour pouvoir être appliquées à un grand nombre de cas clients. Elles doivent également prendre en compte de façon exhaustive toutes les contraintes et règles de gestion définies dans les bases de données des clients FuturMaster.

Les travaux doivent mener à terme à un outil de planification représentant une alternative à un module de FuturMaster reposant sur la programmation linéaire. Ce module a l'avantage d'être très puissant et très générique, mais présente aussi quelques inconvénients. Notamment, lorsque certaines données du problème sont incompatibles, rendant le problème infaisable, la résolution échoue et il est extrêmement difficile de localiser les données en cause. L'utilisateur a donc perdu du temps pour le lancement d'un calcul qui ne lui donne aucun résultat, et n'a aucun moyen de connaître la cause du blocage. Les approches heuristiques envisagées devront permettre de contourner cette difficulté : la détection d'une incompatibilité au cours du déroulement de l'algorithme devra générer un message d'alerte, et la(es) donnée(s) devront éventuellement être corrigées localement, mais le calcul ne doit en aucun cas être bloqué. Cette gestion et détection des données incohérentes amènera une énorme valeur ajoutée à l'outil.

## 1.2 Plan

Le réseau de distribution considéré comprend deux échelons, à savoir : des sites de production ou entrepôts qui fournissent des produits finis à plusieurs dépôts. Les produits sont périssables et ont une date limite d'utilisation. Un dépôt peut être approvisionné par plusieurs entrepôts en livraison directe ; il n'y a pas de tournées de véhicules. Le choix de la source d'approvisionnement est déterminé en fonction de la disponibilité et de la date limite d'utilisation des produits dans les stocks des entrepôts, des moyens de transport (flotte homogène), mais aussi des coûts de transport sur les arcs.

Le chapitre 2 est dédié à l'état de l'art sur la planification de la chaîne logistique. Le problème est décrit, puis modélisé sous forme de programme linéaire mixte dans le chapitre 3. Le modèle proposé intègre la traçabilité des lots de produits selon la date limite d'utilisation, à travers le flux (entrepôt-arc de transport-dépôt) ; ce qui n'est pas le cas du modèle déjà existant chez FuturMaster. Une heuristique constructive et les instances de tests issues de bases de données sont également présentées dans ce chapitre. Le chapitre 4 décrit une heuristique réactive randomisée. Pour plus de diversité pour les tests, un générateur aléatoire d'instances est développé dans ce chapitre. Trois métaheuristiques (VND, ILS et MS-ILS) sont proposées dans le chapitre 5. Une heuristique Lagrangienne, utilisant la relaxation Lagrangienne et une heuristique de réparation, est développée dans le chapitre 6. L'étude d'un cas d'application industrielle est présentée dans le chapitre 7. Il s'agit d'un projet comprenant la gestion des prévisions et la planification de la distribution de produits cosmétiques. Le dernier chapitre 8 résume les principales contributions de cette thèse ainsi que les perspectives des travaux.



## Chapitre 2

# État de l'art sur la planification de la chaîne logistique

### 2.1 Introduction

Un rappel est proposé sur les méthodes d'optimisation que nous avons utilisées pour résoudre le problème de planification dans des chaînes logistiques multi-échelons, dans la section 2.2. Les types de contraintes les plus significatifs sont ensuite présentés. La section 2.3 est dédiée à la littérature sur la gestion des stocks multi-échelons. Nous nous sommes particulièrement intéressés à la gestion des produits périssables. Les problèmes de *lot-sizing* sont discutés dans la section 2.4. La configuration d'un système de distribution dépend du mode de distribution. la littérature de la planification de la distribution, selon le mode de distribution utilisé, est étudiée dans la section 2.5. Les articles les plus récents traitent souvent de problème de planification de la distribution en intégrant la production (voir section 2.6). La dernière section 2.7 résume cet état de l'art.

### 2.2 Les méthodes d'optimisation utilisées dans notre étude

Le choix des méthodes d'optimisation utilisées dans notre étude, s'est fait en tenant compte de la littérature. La relaxation Lagrangienne et les méthodes approchées telles que les heuristiques, les recherches locales et les métaheuristiques sont très performantes pour la résolution des problèmes de planification dans des chaînes logistiques multi-échelons..

#### 2.2.1 La relaxation Lagrangienne

L'approche de la relaxation Lagrangienne actuellement utilisée, a été proposée par Held et Karp (1970) et Held et Karp (1971). Elle consiste à relaxer le problème en supprimant les contraintes qui le rendent difficile à résoudre, et à les introduire dans la fonction-objectif en attribuant un

poids à chaque contrainte. Chaque poids représente une pénalité qui est ajoutée à une solution qui ne satisfait pas la contrainte correspondante. Le problème transformé devient alors plus facile à résoudre de façon optimale. La valeur de la fonction-objectif du problème transformé est une borne inférieure pour le problème initial (dans le cas d'un problème de minimisation). Les pénalités sont modifiées, par exemple avec une méthode de sous-gradient, pour obtenir la borne inférieure la meilleure possible ou une solution de pénalité nulle, réalisable pour le problème initial. Le processus continue jusqu'à ce que l'écart entre les meilleures bornes inférieures et supérieures (du problème relaxé) soit nul (optimum) ou inférieur à un seuil fixé ou lorsque le temps de calcul maximum est atteint. Pour une explication plus détaillée sur l'utilisation de la relaxation Lagrangienne, le lecteur peut se référer aux articles de Geoffrion (1974) et de Fisher (1981).

Certains auteurs effectuent, à chaque itération de la méthode de sous-gradient, une procédure de réparation pour obtenir une borne supérieure. On obtient donc une suite de bornes inférieures et supérieures qui peuvent éventuellement conduire à une solution optimale. Ce genre de méthode est appelée heuristique Lagrangienne. Cette approche s'inscrit dans la catégorie des heuristiques à base mathématique.

La relaxation Lagrangienne est très utilisée pour résoudre les problèmes de gestion de la chaîne logistique, par exemple Nezhad *et al.* (2013) ont étudié un problème de localisation avec plusieurs produits fournis à partir d'une source unique dans un réseau de production/distribution sans capacité et en prenant en compte des coûts d'installation. Pour résoudre efficacement ce problème de décision, deux heuristiques basées sur la relaxation Lagrangienne sont proposées dont l'une intègre des coupes entières pour renforcer la formulation. Des opérateurs de recherche locale sont également intégrés dans ces méthodes pour améliorer les bornes supérieures. Les résultats obtenus montrent l'efficacité de ce type de méthode dans la résolution des problèmes combinatoires, en particulier les problèmes de localisation.

Plus récemment, Li *et al.* (2014) ont proposé une méthode hybride qui intègre la relaxation Lagrangienne, la recherche locale, la décomposition de Dantzig-Wolfe et un processus de parcours de réédition de liens. Leur article concerne un problème de localisation à deux échelons multi-produits à capacité limitée. La demande des clients doit être satisfaite par des usines via des dépôts (à ouvrir) sur une période. L'objectif est de minimiser les coûts d'ouverture des dépôts, de manutention et de transport. Des tests ont été effectués sur des instances générées aléatoirement. L'écart moyen, entre les meilleures bornes inférieures et supérieures des solutions fournies par la méthode hybride, est de 1.5%. Les temps de calcul sont de 15 minutes pour les instances avec 100 noeuds et de moins de 4 heures pour les instances avec 200 noeuds.

La relaxation Lagrangienne est utilisée surtout pour des problèmes d'optimisation formulés sous forme de programme linéaire en nombres entiers et pour des programmes non linéaires. A cause du saut de dualité, elle fournit rarement un optimum, mais on obtient de très bonnes bornes inférieures pour élaguer des solutions dans des méthodes arborescentes ou évaluer l'écart à l'optimum d'une

heuristique.

### 2.2.2 Méthodes approchées

Les limites des méthodes exactes amènent les chercheurs à développer des méthodes approchées pour trouver des solutions réalisables pour les problèmes de planification de la chaîne logistique à grande échelle.

Les méthodes approchées se décomposent en heuristiques constructives, recherches locales et métaheuristiques. Les premières consistent à construire une seule solution par une suite de décisions élémentaires et définitives comme l'heuristique *plus proche voisin* pour le problème du voyageur de commerce. Les recherches locales consistent à améliorer progressivement une première solution en recherchant des transformations améliorantes appelées mouvements, comme le croisement d'arêtes 2-OPT pour le problème du voyageur de commerce. Ces méthodes de type *descente*, telles que les méthodes de gradient en optimisation continue, convergent vers un optimum local, au sens des mouvements considérés. Les métaheuristiques remédient à ce point faible en incorporant divers mécanismes pour éviter d'être bloqué sur un optimum local. On distingue les métaheuristiques à base de voisinages ou parcours, qui génèrent une suite de solutions dans l'espace de recherche mais en travaillant sur une solution, et les méthodes à population.

#### Les heuristiques

Les heuristiques sont généralement des techniques basées sur l'expérience d'un problème en particulier et permettent de surmonter les lacunes (par exemple le temps de calcul peut être important) des méthodes exactes. Les heuristiques sont en général très rapides, contrairement aux méthodes exactes. Une heuristique est normalement utilisée pour trouver rapidement une solution que l'on espère être proche de la solution optimale. Les heuristiques sont très présentes dans la littérature des problèmes de planification dans la chaîne logistique. Cependant, leur principal inconvénient est qu'elles ne garantissent pas une solution optimale ou proche de celle-ci.

#### La recherche locale

La recherche locale est une approche utilisée pour trouver des solutions approximatives à des problèmes d'optimisation difficiles. Les approches de recherche locale remontent à la fin des années 1950, lorsque Bock (1958) et Croes (1958) ont développé leurs procédures d'échange de lien pour le problème du voyageur de commerce. Depuis, une grande variété d'algorithmes de recherche locale a été proposée. L'idée de base est de partir d'une solution initiale et de chercher des améliorations successives en examinant des solutions voisines. Dès 1998, Vaessens *et al.* (1998) proposent un modèle permettant de mettre en évidence les caractéristiques communes des algorithmes de recherche locale

les plus connus. Le modèle proposé conduit à une classification des algorithmes de recherche locale existants et offre la possibilité de s'adapter à de nouveaux types d'approches de recherche locale.

Fellows *et al.* (2012) ont remarqué que la recherche locale est l'une des approches les plus courantes dans la pratique pour résoudre des problèmes d'optimisation difficiles. Elle est utilisée comme un sous-programme dans plusieurs types d'heuristiques et de métaheuristiques. Les auteurs ont montré que pour plusieurs classes de graphes peu denses, une solution améliorée, dans le voisinage de  $k$ -échanges pour de nombreux problèmes, peut être trouvée beaucoup plus efficacement.

## Les métaheuristiques

L'avantage des métaheuristiques se trouve dans leur efficacité et leur applicabilité générale. Les heuristiques sont généralement développées pour résoudre un problème spécifique. Elles ne peuvent pas souvent, être généralisées à d'autres types de problèmes. Il est, généralement, moins difficile d'adapter une métaheuristique à un problème particulier, que de développer une heuristique spécialisée ; ce qui rend les métaheuristiques très attrayantes pour la mise en œuvre d'un logiciel à usage général. Ólafsson (2006) présentent une analyse sur les métaheuristiques, en se concentrant sur l'écart entre leur utilisation en pratique et dans la recherche.

Elles sont représentées essentiellement par les méthodes à base de voisinages ou de parcours comme le recuit simulé et la recherche tabou, et les méthodes à population comme les algorithmes génétiques. Les premières exploitent le voisinage de la solution courante, tandis que les secondes combinent des solutions pour en obtenir de nouvelles. D'après Jans et Degraeve (2007), les métaheuristiques telles que la recherche tabou, les algorithmes génétiques et le recuit simulé, sont devenus populaires et des outils efficaces pour résoudre des problèmes d'optimisation combinatoire difficiles. D'autres métaheuristiques à base de voisinages, utilisant le principe de la recherche locale, se sont révélées très efficaces sur les problèmes de planification dans des chaînes logistiques. Il s'agit de la recherche à voisinage variable (VNS), la recherche locale itérative (ILS) et la recherche locale itérative à démarrages multiples (MS-ILS). La MS-ILS correspond à un GRASP, dans lequel on remplace la recherche locale (LS) par une ILS.

### La recherche à voisinage variable (*Variable neighborhood search* - VNS)

Mladenović et Hansen (1997) ont proposé la recherche à voisinage variable (VNS) où plusieurs voisinages sont utilisés successivement. La méthode VNS ne suit pas une trajectoire unique mais explore des voisinages, de la solution courante, de plus en plus éloignés et passe d'une solution à une nouvelle en cas d'amélioration. La recherche locale est utilisée pour obtenir des optima locaux à partir de ces voisins. La méthode VNS est basée sur le principe d'explorer systématiquement différents voisinages, combiné à un mouvement de perturbation *shaking* pour échapper aux optima locaux. La descente à voisinage variable (*Variable neighborhood descent* - VND) est une simple

variante de la VNS, dans laquelle la phase de perturbation est omise. Contrairement à la VNS, la VND est, souvent, complètement déterministe. Hansen et Mladenović (2001) donnent plus de détails sur les principes et les applications de la méthode VNS.

### **La recherche locale itérative (*Iterated local search* - ILS)**

L'idée essentielle de la recherche locale itérative (ILS) est de concentrer la recherche non pas sur l'espace complet de solutions mais sur un sous-espace plus petit. Ce sous-espace est défini par les solutions qui sont localement optimales pour un moteur d'optimisation donné. À chaque itération, une copie de la meilleure solution connue est perturbée et la recherche locale est, généralement, utilisée pour l'améliorer. La force de la méthode est que les solutions générées sont dépendantes et tracent une trajectoire d'optima locaux proches les uns des autres dans l'espace des solutions. La méthode ILS est aussi très rapide : si la perturbation est assez faible, la nouvelle solution ressemble beaucoup à la solution-parent et est ré-optimisée en quelques mouvements de recherche locale.

La recherche locale itérative est une métaheuristique très utilisée dans la littérature. Lourenço *et al.* (2003) ont présenté un tutoriel sur la méthode ILS et ses applications à quelques problèmes tels que le problème du voyageur de commerce et l'ordonnancement. Ils ont également comparé l'ILS à d'autres métaheuristicques dans un autre article Lourenço *et al.* (2010). Cuervo *et al.* (2014) ont récemment développé une ILS pour le problème de tournées de véhicules avec des retours (VRPB). C'est une extension du problème de tournées de véhicules qui inclut deux types de clients : les consommateurs qui demandent des marchandises au dépôt et les fournisseurs qui envoient des marchandises au dépôt. De nouvelles meilleures solutions ont été trouvées par les auteurs pour deux instances dans l'un des ensembles de référence.

### **Le GRASP (*Greedy Randomized Adaptive Search Procedure*)**

La procédure aléatoire gloutonne de recherche adaptative (GRASP) est introduite par Feo et Bard (1989) puis la présentation générale de la méthode est faite par Feo et Resende (1995). C'est une méthode à démarrages multiples sans mémoire dans lequel la recherche locale est appliquée à différentes solutions initiales construites avec une heuristique gloutonne randomisée. Elle cherche à combiner les avantages des heuristiques gloutonnes, de la recherche aléatoire et des méthodes à base de voisinages.

Un algorithme GRASP répète un processus composé de deux étapes : la construction d'une solution suivie par une descente pour améliorer la solution construite. Durant l'étape de construction, une solution est itérativement construite. Chaque itération ajoute un élément dans la solution partielle courante. Pour déterminer l'élément qui sera ajouté, on utilise une liste des meilleurs candidats et on prend au hasard un élément dans cette liste. La liste des meilleurs candidats est dynamiquement mise à jour après chaque itération de construction. Cette étape de construction

continue jusqu'à ce qu'une solution complète soit obtenue. A partir de cette solution, une descente est appliquée pour améliorer la solution.

La procédure GRASP répète ces deux étapes et retourne à la fin la meilleure solution trouvée. Les deux paramètres de cette méthode sont donc la longueur de la liste de candidats et le nombre d'itérations autorisées.

### **La recherche locale itérative à démarrages multiples (*Multi-start iterated local search* - MS-ILS)**

La méthode ILS peut être piégée dans un grand groupe de bassins d'attraction ou dans un large bassin. Il est plus intéressant de relancer la recherche dans une région inexplorée plutôt que de passer du temps dans les itérations inutiles. Il s'agit de la recherche locale itérative à démarrages multiples (MS-ILS).

Nguyen *et al.* (2012) traitent le problème de localisation et de tournées de véhicules sur deux échelons (LRP-2E) qui consiste à concevoir un réseau de distribution avec deux types de tournées. Le premier échelon sert à approvisionner un ensemble de dépôts à localiser et le second échelon permet de fournir les clients à partir de ces dépôts. Les auteurs développent une MS-ILS basée sur trois heuristiques gloutonnes randomisées, une descente à voisinage variable et une liste tabou. Sur deux ensembles de 24 et 30 instances du LRP-2E, la méthode MS-ILS surpasse en moyenne deux algorithmes GRASP.

La méthode MS-ILS est aussi utilisée par Michallet *et al.* (2014) pour le problème de tournées de véhicules avec des contraintes de propagation de temps des services. Leur article traite d'un problème réel, rencontré par les entreprises de transport de marchandises de valeur. Chaque client doit absolument être livré dans son intervalle de temps défini et l'heure d'arrivée du livreur doit être irrégulière. Les heures de deux visites au même client doivent différer par une constante de temps donnée. Des évaluations ont été faites sur des cas provenant de références classiques pour le problème de tournées de véhicules avec fenêtres de temps, et sur deux cas réels. Pour un cas particulier avec une seule période (le problème de tournées de véhicules avec fenêtres de temps doux), la méthode MS-ILS est en concurrence avec deux algorithmes publiés et améliore six meilleures solutions connues.

De plus en plus, nous retrouvons dans la littérature des méthodes qui combinent plusieurs métaheuristiques. Villegas *et al.* (2010) propose deux métaheuristiques basées sur la méthode GRASP, la descente à voisinage variable (VND) et la recherche locale évolutive (Evolutionary Local Search - ELS). La méthode ELS a été proposée par Wolf et Merz (2007), pour un problème de télécommunications. Il s'agit d'une ILS avec plusieurs solutions construites en parallèle, à chaque itération. Les solutions sont générées à la suite d'une perturbation et d'une recherche locale. La meilleure solution générée remplace la meilleure solution courante en cas d'amélioration.

Villegas *et al.* (2010) ont étudié le problème d'acheminement d'un seul camion et d'une re-

morque avec des dépôts satellites. Un véhicule composé d'un camion avec une remorque amovible sert la demande d'un ensemble de clients accessibles uniquement par le camion sans la remorque. La contrainte d'accessibilité implique la sélection des emplacements, pour garer la remorque, avant d'effectuer les excursions vers les clients. L'expérimentation montre que la recherche locale évolutive à démarrage multiples (GRASP-ELS) surpasse un GRASP/VND. La méthode GRASP-ELS est également utilisée par Duhamel *et al.* (2011) pour une extension du problème de tournées de véhicules à capacité limitée où la demande des clients est composée d'éléments pondérés en deux dimensions (2L-CVRP). Les résultats montrent que leur méthode est très efficace et surpasse les meilleures méthodes publiées antérieurement sur le sujet.

## 2.3 Gestion des stocks

La politique d'approvisionnement pour un stock est déterminée par : la caractéristique de la demande, le délai d'approvisionnement, les divers coûts impliqués (coûts de passation de commande, de possession, de rupture, etc.) ainsi que le niveau de service à la clientèle. L'objectif de la gestion des stocks est de maintenir le stock d'articles à un niveau souhaité, que ce soient des matières premières, des en-cours (produits semi finis) ou des produits finis. Les articles stockés servent à satisfaire deux types de demandes, à savoir :

- une demande dépendante concernant les matières premières, les composants et les produits semi finis ; elle est déduite d'un autre article et fait intervenir la nomenclature, et
- une demande indépendante pour les produits finis et les pièces de rechange.

### 2.3.1 Politique de stock multi-échelons

Dans un réseau de distribution multi-échelons, seule la demande du niveau le plus bas est totalement indépendante. La demande d'un niveau supérieur découle du besoin du niveau inférieur, en plus d'un besoin indépendant propre à celui-ci. Pour être réaliste, la gestion des stocks sur plusieurs échelons doit prendre en compte des contraintes liant les différents niveaux.

Le problème de la gestion des stocks multi-échelons a été étudié dès les années 1950 par des chercheurs tels qu'Arrow *et al.* (1958) et Love (1979). Le principal défi de ces problèmes est de contrôler les niveaux des stocks, en déterminant la taille des ordres de commande pour chaque échelon, à chaque période, de manière à optimiser une fonction-objectif donnée. De nombreux chercheurs ont étudié des moyens de réduire le coût des stocks des fournisseurs ou des distributeurs, ou ont considéré soit le système de distribution ou le système de production.

Brecman *et al.* (1989) considèrent un réseau à trois niveaux : un entrepôt central fournit des produits à des entrepôts régionaux, qui à leur tour approvisionnent des centres de distribution. L'objectif est de minimiser la somme des coûts de possession de stock et de transport dans un environnement multi-échelons. Les retards de livraison sont permis mais pénalisés par un coût.

Des fenêtres de temps pour l'expédition et la réception sont pris en compte par Li *et al.* (2006). Le réseau comprend plusieurs détaillants qui se font livrer un seul type de produit, soit directement par un ensemble de producteurs, soit via des centres de transbordement. Ils modélisent les coûts de transport, de stockage et ceux relatifs aux contraintes temporelles.

Dans la plupart des articles, les auteurs considèrent un coût de stockage fixe dans le temps. C'est le cas de Zhou *et al.* (2013) qui appliquent une stratégie de réapprovisionnement commun dans un système de gestion des stocks multi-échelons. Ils proposent un modèle de gestion des stocks multi-échelons et multi-produits. Le modèle tient compte de plusieurs fournisseurs, un producteur et plusieurs distributeurs et acheteurs, d'un coût de stockage fixe dans le temps et un niveau de stock maximum.

Plus récemment, la gestion conjointe des stocks des systèmes de production et de distribution a été assez étudiée. Nous étudions, plus largement, ce sujet dans la section 2.6.

### 2.3.2 Gestion de produits périssables

Un produit ayant un taux de dégradation important influence forcément la gestion des stocks et du transport. L'effet de détérioration limite les possibilités de stockage et de transport en termes de quantité, de lieu et de condition.

Goyal et Giri (2001) présentent un état de l'art des articles publiés depuis le début des années 90 jusqu'en 2000. Selon les auteurs l'obsolescence intervient lorsque des articles perdent leur valeur à travers le temps à cause des changements de technologie ou l'introduction d'un nouveau produit par un concurrent. De tels produits sont alors soit soldés, soit perdus après la saison. La détérioration fait référence au dommage, à la péremption, à la sécheresse, la vaporisation des produits.

#### 2.3.2.1 Classification

Dans cette étude, nous nous intéressons à la détérioration ou plus exactement à la péremption des produits qui intervient un certain temps après la fabrication. Les produits périssables peuvent être subdivisés en deux catégories, en fonction de la caractéristique associée à leur durée de vie :

- fixe : lorsque les produits préservent leur utilité pour une période de temps déterminée, au-delà de laquelle, ils sont considérés comme inutiles ;
- variable, avec un taux de détérioration dépendant de l'âge du produit : leur utilité diminue au cours du temps, en fonction de l'âge du produit ;
- variable, avec un taux de détérioration dépendant du temps ou selon les conditions de stockage et de transport : Akkerman *et al.* (2011) traitent un problème de gestion de la production et de la distribution pour l'industrie alimentaire en prenant en compte la qualité du produit avec une durée de vie variable dépendant de la température de stockage et de transport. Les auteurs considèrent les coûts classiques de production, de transport, de stockage, de perte de

produits détériorés mais aussi celui généré par le maintien de la température adéquate des équipements de stockage et transport.

Selon Amorim *et al.* (2012) deux cas sont à considérer : des produits périssables ayant une durée de vie fixe et ceux avec une durée de vie fixe non indiquée. Lorsque la durée de vie est fixe, le repère visuel le plus commun pour le consommateur est la date renseignée sur l'inscription « à consommer avant ». Elle peut être définie comme étant la date jusqu'à laquelle le produit reste commercialisable et conserve toutes ses qualités. Dans ce cas, le consommateur peut adapter l'achat d'un produit en se basant sur la durée restante par rapport à cette date. D'autre part, lorsque la date de péremption d'un produit n'est pas indiquée, donc la durée de vie est perdue, le consommateur doit compter sur ses sens ou des sources d'information externes pour estimer la durée de vie restante. C'est le cas des fruits et légumes vendus au détail sans étiquetage.

### 2.3.2.2 Modélisation

Misra (1975) est le premier à étudier un modèle de quantité économique de production pour des produits périssables avec un taux de détérioration fixe et variable. Park (1983) propose un modèle de gestion de production et de stockage avec des matières premières périssables et un produit fini non périssable. Raafat *et al.* (1991) reprennent le modèle de Park (1983), en introduisant un taux de détérioration constant pour le produit fini, mais en considérant toujours que le périmètre de la production.

Kanchanasuntorn et Techanitisawad (2006) ont étudié l'effet de la péremption des produits sur le coût total, le profit net, le niveau de service, et le niveau de stock moyen dans un réseau de distribution et de gestion des stocks à deux échelons. Kanchanasuntorn et Techanitisawad (2006) ont modifié le modèle de simulation de Matta et Sinha (1995), en intégrant une durée de vie fixe et une politique de perte des ventes au niveau des points de vente. Dans la plupart des cas, les effets de la péremption des produits sont négatifs, en abaissant le taux de service et en augmentant le coût de l'ensemble du système ; ce qui entraîne une diminution du profit net. L'effet de la péremption est généralement accentué lorsqu'une politique de la perte des ventes est appliquée à chaque point de vente.

Selon van Donselaar *et al.* (2006), la péremption signifie la possible détérioration physique d'un produit et l'obsolescence intervient lorsque la demande du produit devient négligeable après un certain temps : par exemple un produit en fin de vie remplacé par un autre.

Dans le domaine agricole, Ahumada et Villalobos (2011) présentent un modèle opérationnel pour la planification à court terme. L'objectif est de permettre au producteur de maximiser ses revenus, en prenant des décisions de production et de distribution au cours de la saison des récoltes. Les questions les plus importantes, dans la planification de la récolte à court terme, sont la gestion des coûts du personnel, la préservation de la valeur des cultures périssables, et l'utilisation de modes de transport efficaces qui offrent le meilleur compromis entre le temps de mise à disposition sur le

marché (qualité des produits) et le coût. Les étapes du flux sont la récolte, le conditionnement et la distribution à travers des entrepôts et des centres de distribution.

Plus récemment, Yu *et al.* (2012) considèrent une chaîne d'approvisionnement basée sur la gestion partagée des approvisionnements (GPA) d'une matière première et d'un produit fini ayant chacun un taux de détérioration fixe. Cette différenciation est faite car les matières premières avec un taux de dégradation très important, comme le lait, sont souvent transformées en produits qui se détériorent plus lentement (poudre de lait, fromage). Le réseau est constitué d'un fabricant et de plusieurs détaillants. Des tests effectués pour le cas avec 3 détaillants montrent que lorsque le taux de détérioration du produit fini augmente (de 0 à 0.25% par unité et par période), le coût total augmente de plus de 40%. De même lorsque le taux de détérioration de la matière première augmente (de 0 à 0.25% par unité et par période), le coût total augmente, mais seulement de 3.4%.

Pour plus de détails sur les études effectuées sur la gestion des stocks de produits périssables, le lecteur peut consulter l'état de l'art récent de Bakker *et al.* (2012). Un état de l'art des articles publiés, entre janvier 2001 et décembre 2011 sur la gestion des stocks de produits périssables, est proposé. Les auteurs notent que les modèles à deux et plusieurs produits périssables sont très rares dans la littérature de la gestion des stocks, contrairement aux modèles à un seul produit. Ce déséquilibre peut, essentiellement, être attribué à la complexité des modèles multi-produits par rapport aux modèles à un seul produit.

## 2.4 Dimensionnement de lots (*lot-sizing*)

Les problèmes de *lot-sizing* sont en général des problèmes de planification de la production incluant des réglages (setup) entre la production des lots. La planification de la production permet de déterminer, sur un horizon limité, les quantités à produire à chaque période. Il s'agit d'utiliser efficacement les ressources disponibles dans le but de minimiser les coûts. Le livre de Buffa et Sarin (1987) fournit plus de précisions sur les concepts de la planification de la production.

Un lot représente une quantité d'unités d'un produit, afin de respecter une contrainte physique liée à la production ou au transport. Il est donc trop coûteux de produire chaque produit à chaque période. Par ailleurs, réduire les réglages en produisant d'importantes quantités peut engendrer des coûts de stockage élevés. L'objectif est de déterminer les périodes et les quantités à produire afin de satisfaire la demande tout en minimisant les coûts de production, de réglage et de stockage. D'autres coûts peuvent être considérés tels que : le coût de rupture de stock, le coût de transition, une pénalité de non satisfaction de la demande dans un intervalle de temps.

Dès 1958, Wagner et Whitin (1958) proposent un modèle de base du USILSP (*Uncapacitated Single Item Lot-Sizing Problem*), avec un produit et sans contrainte de capacité, pour un problème de planification de production, et un algorithme de programmation dynamique en  $O(T^2)$ . La programmation dynamique est une méthode implicite de dénombrement qui explore l'espace de la

solution d'un problème afin de trouver son optimum. Zangwill (1969) étend le modèle de Wagner et Whitin en considérant le retard des demandes (backlogging), propose un algorithme de programmation dynamique en  $O(T^2)$ . La complexité de l'algorithme de programmation dynamique pour le USILSP a été améliorée en  $O(T \log T)$ . Ce résultat a été trouvé indépendamment par van Hoesel (1991), Wagelmans *et al.* (1992), Federgruen et Tzur (1993) et Aggarwal et Park (1993).

Le problème de *lot-sizing* pour un seul produit, sans contrainte de capacité (USILSP), est très étudié dans la littérature. Brahim *et al.* (2006) présentent un état de l'art sur le problème de *lot-sizing* pour un seul produit avec et sans contrainte de capacité. Ces études sur le cas simple ont permis la résolution de cas plus complexes. Des extensions telles la prise en compte d'une capacité de production, plusieurs produits, plusieurs étapes de production, des fenêtres de temps, une capacité maximale de stockage ou encore la gestion de l'obsolescence des produits ont émergé ces dernières années.

Pour le problème à un seul produit avec une capacité finie (CSILSP *Capacitated Single Item Lot-Sizing Problem*) Florian *et al.* (1980), et Bitran et Yanasse (1982) ont prouvé qu'il est NP-difficile pour le cas général et d'autres cas particuliers. Chen *et al.* (1994) ont prouvé que le problème n'est pas NP-difficile au sens fort en proposant un algorithme pseudo-polynomial pour le résoudre.

Chen et Thizy (1990) ont montré que le problème considérant plusieurs types de produit et une contrainte de capacité (MCLS *Multi-Item Capacitated Lot-Sizing Problem*) est NP-difficile au sens fort.

Belvaux et Wolsey (2001) proposent une revue de la modélisation du problème de *lot-sizing* en programme mixte. Les modèles classiques sont présentés puis des extensions, qui reflètent des pratiques du problème de *lot-sizing*, sont intégrées. Quelques inégalités valides sont ajoutées au modèle. Les auteurs soulignent que la façon dont le modèle de *lot-sizing* est formulé est cruciale pour que les solveurs utilisés (par exemple XPRESS et CPLEX) en résolution de programme mixte, réussissent à générer des coupes. Ils proposent des formulations de différentes variantes du problème de *lot-sizing* permettant de générer facilement des coupes. Différentes méthodes ont été utilisées pour résoudre les problèmes de *lot-sizing*.

Dans un article de Brahim *et al.* (2003), les auteurs montrent l'importance de l'utilisation de la relaxation Lagrangienne pour résoudre le problème de *lot-sizing* avec capacité pour un seul produit (CSILSP) avec des fenêtres de temps.

Basnet et Leung (2005) présentent une extension du modèle de Wagner et Whitin (1958) en incluant la gestion des stocks de plusieurs produits, sans contrainte de capacité (MULS-SS *Multi-Item Uncapacitated Lot-Sizing Problem with Supplier Selection*). La demande des produits est connue sur un horizon de planification. Chaque produit a un coût de stockage par période et peut provenir d'un ensemble de fournisseurs agréés, dont chacun est caractérisé par un coût de transaction. La rupture de stock n'est pas autorisée. L'objectif est de déterminer les produits et les quantités à commander aux fournisseurs adéquats par période. Un algorithme de recherche énumérative et une

heuristique sont présentés pour résoudre le problème. Les auteurs notent que le temps de résolution de l'algorithme de recherche énumérative augmente fortement en fonction de la taille du problème.

Jans et Degraeve (2007) passent en revue les méta-heuristiques les plus utilisées pour résoudre des problèmes de *lot-sizing*. D'autres méthodes, telles que la programmation dynamique, les plans de coupe, la décomposition de Dantzig-Wolfe, la relaxation Lagrangienne et les heuristiques dédiées, sont également discutées.

Plus récemment, Absi *et al.* (2011) ont étudié le problème de *lot-sizing* pour un seul produit avec des fenêtres de temps en considérant la production, la perte de ventes, les avances et les retards de production sur un horizon de planification de  $T$  périodes (USILSP-TW-LS *Uncapacitated Single Item Lot-Sizing Problem with Time Windows Lost Sales*). Dans ce contexte, une demande non traitée dans sa fenêtre de temps peut être soit perdue (vente perdue), soit satisfaite par une production en avance ou satisfaite par une production effectuée après la date d'échéance de la demande. Les auteurs présentent plusieurs propriétés de la solution optimale pour les différentes variantes du problème lorsque les fenêtres de temps ne sont pas spécifiées par client. Ils proposent des algorithmes de programmation dynamique pour résoudre les problèmes examinés en  $O(T^2)$  où  $T$  représente le nombre de périodes de l'horizon de planification. Toutes les instances générées sont résolues avec un temps CPU inférieur à 1 seconde sur un PC avec un processeur Intel Dual Core à 2.53 GHz et 4 GB de RAM. Par conséquent, ces algorithmes peuvent être utilisés dans une méthode de décomposition comme la relaxation Lagrangienne.

Dans l'article de Absi *et al.* (2013), les auteurs étudient le problème de *lot-sizing* pour plusieurs produits avec une contrainte de capacité, des temps de setup et des pertes de ventes (MCLS-LS *Multi-Item Capacitated Lot-Sizing problem with setup times and Lost Sales*). Une relaxation Lagrangienne des contraintes de capacité a été utilisée pour obtenir de bonnes bornes inférieures. Un algorithme a ensuite été développé afin de résoudre le sous-problème avec un seul produit de manière optimale à l'aide d'une adaptation de l'algorithme de programmation dynamique en  $O(T^2)$  d'Aksen *et al.* (2003). Pour trouver des solutions réalistes, les auteurs ont utilisé une heuristique de lissage non-myopes ; une heuristique de sondage est utilisée à chaque étape afin d'évaluer et de sélectionner les déplacements les plus prometteurs. Une recherche locale et des procédures d'amélioration adaptées ont également permis d'améliorer la borne supérieure obtenue par la relaxation Lagrangienne. La méthode qui combine l'heuristique de sondage, la multiplicité des itérations pour la relaxation Lagrangienne et de recherche locale adaptative fournit de très bons résultats.

Le Tableau 2.1 présente une synthèse d'articles qui étudient les problèmes de *lot-sizing*. Les articles sont classés selon deux critères : le nombre de produits (un ou plusieurs types de produit) et la prise en compte, ou non, de la contrainte de capacité.

Produit	Capacité	Problème	Article	Méthode	Complexité
Mono	Non	USILSP	Wagner et Whitin (1958)	programmation dynamique	$O(T^2)$
		USILSP	Zangwill (1969)	programmation dynamique	$O(T^2)$
		USILSP	Federgruen et Tzur (1993)	programmation dynamique	$O(T \log T)$
		USILSP	Wagelmans <i>et al.</i> (1992)	programmation dynamique	$O(T \log T)$
		USILSP	Aggarwal et Park (1993)	programmation dynamique	$O(T \log T)$
		USILSP	van Hoesel (1991)	programmation dynamique	$O(T \log T)$
	Oui	USILSP	Absi <i>et al.</i> (2011)	programmation dynamique	$O(T^2)$
		CSILSP	Bitran et Yanasse (1982)	NP-difficile pour le cas général	
		CSILSP	Florian <i>et al.</i> (1980)	NP-difficile pour le cas général	
		CSILSP	Chen <i>et al.</i> (1994)	programmation dynamique	pseudo-polynomial
Multi	Non	MULS-SS	Basnet et Leung (2005)	recherche énumérative et heuristique	
	Oui	MCLS	Chen et Thizy (1990)	NP-difficile au sens fort	
		MCLS-LS	Absi <i>et al.</i> (2013)	relaxation Lagrangienne + recherche locale + heuristique	

TABLE 2.1 – Classification des articles sur les problèmes de *lot-sizing*

## 2.5 Planification de la distribution

Un réseau ou chaîne de distribution est un ensemble de lieux de stockage, de transit et de points de vente liés par des opérations logistiques, pour qu'un produit soit mis à disposition d'un client final. Dans cette section, nous présentons les différents types de mode de distribution, la distribution avec plusieurs sources possibles (alternative d'approvisionnement) et les principes de la planification des besoins de distribution (DRP).

### 2.5.1 Modes de distribution

La configuration d'un système de distribution dépend du mode de distribution utilisé, à savoir : la livraison directe, en entreposage, par transbordement (*cross-docking*) ou encore en tournées. Dans

chacune de ces configurations, l'approvisionnement d'un produit peut s'effectuer à partir d'un ou de plusieurs points en amont. Lorsque le choix de la source d'approvisionnement n'est pas exclusive, il s'agit du *multi-sourcing*.

Le Tableau 2.2 présente une synthèse d'articles qui étudient les problèmes de planification de la distribution. Les articles sont classés selon le mode de distribution utilisé.

#### 2.5.1.1 Livraison directe

Les produits sont livrés directement d'une usine (fournisseur) à un magasin de détail. L'avantage de livrer directement est d'éviter les coûts d'exploitation des centres de distributions ou des entrepôts. Le délai de livraison est réduit par rapport à celui d'une autre stratégie de distribution. L'effet de la mutualisation des risques (*Risk pooling*) de stockage est annulé. Cependant, le niveau de stock moyen est plus élevé pour assurer un bon niveau de service aux clients. Les coûts de transport augmentent car le fournisseur doit envoyer un camion à chaque client. Cette stratégie est efficace lors que la quantité de livraison remplit complètement un camion ou dans le cas où le délai de livraison est critique, notamment pour les produits périssables.

Zanoni et Zavanella (2007) étudient un réseau de distribution dans lequel un seul vendeur expédie un ensemble de produits périssables, avec une durée de vie fixe, vers une destination commune. L'objectif est de satisfaire la demande finale, en définissant les fréquences de livraison pour chaque type de produit afin de minimiser la somme des coûts de transport et de stockage. Des heuristiques existantes ont été modifiées pour résoudre le problème. Les résultats montrent que certaines sont très efficaces.

#### 2.5.1.2 Livraison en entreposage

Les produits sont livrés d'un point initial vers un point intermédiaire, puis sont livrés aux points finaux en se basant sur leurs demandes. Brecman *et al.* (1989) traitent la gestion des stocks de produits finis dans un environnement multi-échelon en considérant une version réduite du problème réel. En effet, 4 centres de distribution sont approvisionnés par 2 entrepôts régionaux, eux-mêmes sont fournis par un entrepôt central avec une capacité de transport limitée. Quant à Hindi *et al.* (1998), différents produits doivent être servis à des clients par un certain nombre d'usines, en passant par des centres de distribution intermédiaires. Chaque emplacement est caractérisé par un coût fixe pour ouvrir les centres de distribution, les dépenses d'exploitation et une capacité maximale. La demande de chaque client pour chaque produit est connue ainsi que les coûts d'expédition.

#### 2.5.1.3 Livraison par transbordement (*cross-docking*)

Les produits sont transportés vers une plate-forme logistique. Ils sont ensuite livrés aux points finaux, presque sans attente, après un regroupement des produits. Cette plate-forme n'a pas pour

vocation de stocker les produits. C'est un lieu pour faciliter le transfert des produits entre les moyens de transport venant des points initiaux et ceux allant aux points finaux. L'opération s'effectue de quai à quai. Il faut des installations particulières telles qu'un nombre important de quais, un espace au sol important pour une fluidité maximale, une intégration poussée au niveau des systèmes d'information entre le fournisseur, le prestataire logistique, le transporteur et le distributeur ou client final. Ce type d'activité logistique est souvent utilisé par les entreprises de messageries Express.

Apte et Viswanathan (2000) citent certains facteurs qui influent sur le choix d'une stratégie de distribution en *cross-docking*. Le premier facteur important est le taux de la demande de produits. S'il y a un déséquilibre entre le flux entrant et sortant, le *cross-docking* n'est pas efficace. Par conséquent, les produits qui sont le plus appropriés sont ceux qui ont des taux de la demande plus ou moins stable (épicerie et produits périssables régulièrement consommés). Pour ces produits, les conditions d'entreposage et de transport sont beaucoup plus prévisibles, et par conséquent, la planification et la mise en œuvre du *cross-docking* deviennent plus faciles. Le coût unitaire de rupture de stock est un autre facteur important. Le *cross-docking* minimise le niveau des stocks dans l'entrepôt, la probabilité d'avoir des ruptures de stock est donc plus importante. Le *cross-docking* est donc préférable pour les produits avec un taux stable de la demande et à faible unité de rupture de stock. L'entreposage est toujours préférable lorsque la demande est instable et le coût unitaire de rupture est élevé. Pour les autres cas, le *cross-docking* est pertinent lorsque les systèmes appropriés et des outils de planification sont mise en place pour maintenir le nombre de ruptures de stock à un niveau raisonnable.

Li *et al.* (2006) ont étudié un réseau de distribution dans lequel plusieurs détaillants se font livrer un seul type de produit directement par un ensemble de producteurs ou à travers des centres de transbordement. L'objectif est de trouver les meilleures localisations des usines et les itinéraires d'approvisionnement afin de minimiser les coûts de transport et de stockage.

D'après van Belle *et al.* (2012), le transbordement est utilisé dans l'objectif de réduire les coûts de stockage et de manutention (*picking*). C'est une stratégie logistique dans laquelle le fret est déchargé des véhicules entrants et (presque) directement chargé dans les véhicules sortants, avec peu ou pas de stockage entre les deux.

Pour plus de précision sur le *cross-docking*, le lecteur peut consulter l'état de l'art de van Belle *et al.* (2012).

#### 2.5.1.4 Livraison par tournées de véhicules

Dans un problème de tournées de véhicules, un dépôt livre plusieurs clients en tournées, avec une demande connue de chaque client. Une tournée commence et finit au dépôt, en visitant chaque client une seule fois, avec une flotte de véhicules à capacité finie. Le CVRP (*Capacited Vehicle Routing Problem*) désigne le problème classique. Les variantes du problème étudiées dans la littérature sont nombreuses. Nous en mentionnons quelques-unes :

- IRP** : *Inventory Routing Problem*, intégration de la gestion des stocks ;
- HVRP** : *Heterogeneous fleet VRP*, véhicules avec des capacités différentes ;
- VRPTW** : *VRP with Time windows*, intervalle de temps dans lequel il faut livrer le client ;
- VRPPD** : *VRP with Pick-up and Deliveries*, tournées de livraison/ramassage ou service mixte ;
- DCVRP** : *Distance-Constrained VRP*, une contrainte de distance maximale parcourue est imposée aux véhicules ;
- TDVRP** : *Time-Dependent VRP*, le temps de parcours sur un arc dépend de la date de départ ;
- MDVRP** : *Multi-Depot VRP*, plusieurs dépôts sont considérés et chaque véhicule est affecté à un seul dépôt ;
- PVRP** : *Periodic VRP*, un horizon multi-périodes est considéré ;
- VRPSD** : *VRP with Split deliveries*, la demande d'un client peut être scindée et satisfaite par plusieurs véhicules.

Les objectifs les plus communs sont la minimisation du nombre de véhicules utilisés, de la distance totale parcourue, de la durée totale des tournées, du coût total des tournées, des pénalités liées aux violations des contraintes (notamment dans le cas de fenêtres temporelles), ou encore la maximisation des gains engendrés par les tournées.

Le périmètre du VRP est aussi étendu à plusieurs échelons et est associé à d'autres problématiques de la chaîne logistique telles que la localisation des dépôts. Prins *et al.* (2012) traitent le problème de localisation et de tournées de véhicules sur deux échelons (LRP-2E) qui consiste à concevoir un réseau de distribution avec deux types de tournées. Le premier échelon sert à approvisionner un ensemble de dépôts à localiser et le second échelon permet de fournir les clients à partir de ces dépôts. Les auteurs développent une méta-heuristique MS-ILS (*multi-start iterated local search*), basée sur trois heuristiques gloutonne randomisée une descente à voisinage variable et une liste tabou..

De plus en plus, les variantes du VRP sont combinées pour obtenir des VRP, dits riches, qui reflètent des cas réels. Vidal *et al.* (2013) présentent un état de l'art des heuristiques et méta-heuristiques développées pour le MAVRP (*Multi-Attribute Vehicle Routing Problem*). Il s'agit de différentes variantes du VRP pour lesquelles des spécificités ou des contraintes sont ajoutées.

Récemment, l'intégration de contraintes d'ordre écologique, telles que les émissions de CO<sub>2</sub> et le taux de consommation de carburant, représente un réel intérêt pour les chercheurs. Xiao *et al.* (2012) proposent un modèle pour le FCVRP (*Fuel consumption Capacited Vehicle Routing Problem*) et montrent que le modèle peut réduire la consommation de carburant de 5% en moyenne par rapport au modèle CVRP. Kwon *et al.* (2013) présentent un modèle du problème de tournées de véhicules avec une flotte hétérogène fixe en considérant les émissions de carbone pour minimiser la somme des coûts d'exploitation. Des algorithmes de recherche tabou ont été développés avec trois méthodes de génération de voisinage. Ils montrent que la quantité d'émission de carbone peut être réduite, de manière significative.

Article	Produit	Échelons	Livraison	Source	Méthode
Brecman <i>et al.</i> (1989)	multi	3	entreposage	mono	heuristique
Hindi <i>et al.</i> (1998)	multi	3	entreposage	mono	<i>branch and bound</i>
Basnet et Leung (2005)	multi	2	directe	multi	recherche énumérative et une heuristique
Liu et Lin (2005)	mono	2	tournées	mono	méta-heuristique
Li <i>et al.</i> (2006)	mono	3	directe ou <i>cross-docking</i>	multi	métaheuristique
Zanoni et Zavanella (2007)	multi	2	direct	mono	heuristiques
Prins <i>et al.</i> (2012)	mono	3	tournées	mono	métaheuristique
Xiao <i>et al.</i> (2012)	mono	2	tournées	mono	métaheuristique
Kwon <i>et al.</i> (2013)	mono	2	tournées	mono	métaheuristique
Vidal <i>et al.</i> (2013)	état art		tournées		heuristique et métaheuristique

TABLE 2.2 – Classification des articles selon le mode de distribution

En pratique, les différentes stratégies de distribution sont souvent combinées dans un système global incluant plusieurs types de produit. Dans le réseau étudié par Cocco *et al.* (2013), trois stratégies de distribution peuvent être utilisées selon le type de produit : la livraison directe, l'entreposage ou le *cross-docking*. Pour les livraisons directes, les articles sont expédiés directement de l'usine vers le client final ; ce qui permet d'éliminer les frais d'exploitation d'un centre de distribution et de réduire les délais de livraison. L'entreposage permet de conserver des stocks de produits finis à grande rotation et le *cross-docking* est destiné aux articles à valeur élevée et à faible rotation.

### 2.5.2 Alternative d'approvisionnement : *multi-sourcing*

D'après le *business dictionary* <sup>(1)</sup>, les alternatives d'approvisionnement peuvent être définies comme étant « la répartition des activités ou des services (...) entre une combinaison de fournisseurs, à la fois internes et externalisées, afin de gagner plus de contrôle sur les coûts et la responsabilité, tout en réduisant la dépendance envers un fournisseur global ». En effet, cette multiplicité des sources garantit une meilleure réactivité face aux aléas tels que les situations de pénurie. Li *et al.*

1. <http://www.businessdictionary.com>

(2006) considèrent ce type d'approvisionnement. Par contre dans l'article de Hindi *et al.* (1998), chaque client doit être servi, pour tous les produits qu'il exige, par un seul centre de distribution : c'est une alternative exclusive.

### 2.5.3 DRP : *Distribution Requirements Planning*

Stenger et Cavinato (1979) ont formalisé les idées de Whybark (1975), sur l'utilisation des principes du MRP (Material Requirements Planning) pour la planification des besoins de distribution (DRP). C'est un processus de gestion des approvisionnements, sur plusieurs échelons, de tous les points de stock, d'un réseau de distribution. Sur un horizon de plusieurs semaines, par exemple, les besoins des entrepôts sont calculés et simulés en fonction de la demande prévisionnelle, laquelle est réactualisée hebdomadairement. Stenger et Cavinato (1979) notent que, si les prévisions de la demande du centre distribution sont exactes, des économies considérables peuvent être réalisées, en réduisant le niveau des stocks de sécurité, à chaque emplacement du réseau de distribution.

Selon Brecman *et al.* (1989), le DRP présente quelques lacunes. Bien que l'augmentation du niveau du stock de sécurité, permette de gérer les incertitudes, le processus de demandes découlant des échelons inférieurs suppose des conditions déterministes. En outre, le DRP ne considère pas directement la capacité des ressources de transport et de stockage.

Les articles les plus récents traitent souvent des problèmes de planification de la distribution en intégrant la production ou encore la gestion de produits avec une courte durée de vie.

## 2.6 Gestion intégrée de la production et de la distribution

Le réseau global d'une chaîne logistique est souvent composé d'usines, d'entrepôts ou/et de centres de distribution, et de clients finaux ou des détaillants. L'optimisation de la planification intégrée de la production et de la distribution consiste à optimiser, simultanément, les décisions concernant la production et la distribution, dont les périmètres sont traditionnellement considérés séparément. Ce problème présente un réel intérêt pour les industriels au cours des dernières années. Plusieurs raisons peuvent expliquer cette tendance : l'effet positif sur la rentabilité de la chaîne logistique à travers l'intégration globale des activités de production et de distribution, la réduction des délais, la réactivité face aux changements du marché et donc la réduction de la propagation des imprévus et des événements indésirables à travers le réseau. Les objectifs sont souvent la maximisation du profit ou la minimisation des coûts liés à la production, au transport et à la gestion des stocks.

Chandra et Fisher (1994) sont parmi les premiers à proposer une analyse détaillée du gain obtenu par la coordination de la planification de la production et de la distribution. Le schéma logistique est constitué d'une usine qui fabrique plusieurs produits au fil du temps et maintient un stock de

produits finis à l'usine. Les produits sont distribués directement par une flotte de véhicules non limitée vers plusieurs détaillants. La demande de chaque produit est connue pour chaque détaillant et à chaque période de l'horizon de planification. Les auteurs montrent que sur 132 instances de tests distinctes, la réduction du coût total d'exploitation découlant de la coordination varie de 3% à 20% sur des instances de petite taille (jusqu'à 10 produits à fournir à 50 clients sur un horizon de 10 périodes).

### 2.6.1 Modélisation

La programmation linéaire mixte prédomine dans la formulation des problèmes de planification intégrée de la production et de la distribution. Dans l'état de l'art présenté par Mula *et al.* (2010), 20 articles sur 44 utilisent la programmation linéaire mixte pour la formulation des problèmes de planification intégrée de la production et de la distribution.

Quelques modèles, basés sur la programmation multi-objectif, sont proposés dans la littérature comme Chern et Hsieh (2007) et Torabi et Hassini (2008).

La programmation non linéaire est moins présente dans la modélisation de ce problème (voir Lababidi *et al.* (2004)). Gebennini *et al.* (2009) proposent un programme non linéaire mixte (PNL mixte) pour la prise de décision d'ordre stratégique et tactique dans un environnement stochastique. C'est la première étude qui considère un réseau de production et de distribution multi-échelons (usine de production ou centre de distribution central, plusieurs centres de distribution régionaux et un ensemble de clients) et multi-périodes, intégrant la production, la gestion des stocks et les problèmes de transport, la détermination du stock de sécurité, les délais et les coûts associés. Les instances testées sont générées à partir d'un cas réel, constitué d'un centre de distribution central, de 5 centres de distribution régionaux et plus de 1100 clients. Les résultats montrent que le coût global peut être réduit d'au moins 10% grâce à une réduction significative des niveaux de stock de sécurité (environ de 20%) et du nombre de centres de distribution régionaux et également par l'optimisation des flux logistiques et du transport.

Afin d'intégrer l'imprécision des données issues du monde professionnel, certains auteurs utilisent la programmation floue. Liang (2011) propose une approche floue de programme linéaire multi-objectif basée sur la théorie des possibilités. Le réseau de production et de distribution étudié est constitué de plusieurs usines qui fournissent plusieurs types de produits à un ensemble de centres de distribution sur un horizon multi-périodes avec des objectifs et des prévisions de la demande imprécis.

Certains articles tels que celui de Ahumada *et al.* (2012) traitent un modèle stochastique. L'objectif de Ahumada *et al.* (2012) est d'élaborer des plans de plantation, de récolte et de distribution qui sont robustes face aux effets incertains des marchés et des conditions météorologiques, tout en maximisant le profit. Le modèle s'appuie sur les travaux de Ahumada et Villalobos (2011), en ajoutant des variables aléatoires, afin de mieux refléter la variabilité vécue par les producteurs. Les

résultats, fournis par le modèle stochastique, permettent d'obtenir des hausses de profit de plus de 50%. Cependant, le temps de calcul du modèle stochastique est 6 à 7 fois plus important que celui du modèle déterministe.

### 2.6.2 Résolution

Vu le nombre de variables de décision, le problème de planification intégrée de la production et de distribution est un problème très complexe. Les méthodes exactes sont utilisées pour tester des instances de petite ou moyenne taille qui en général ne correspondent pas à des cas réels.

Pour résoudre des instances plus proches de la réalité, des heuristiques et méta-heuristiques sont développées. L'avantage d'une heuristique est qu'elle fournit une solution réalisable, pour des problèmes très complexes en un temps acceptable. Par contre, l'optimalité de la solution obtenue n'est pas garantie. Souvent, les auteurs comparent les résultats fournis par l'heuristique et ceux d'une méthode exacte sur les instances de petite taille. Cette démarche permet de d'évaluer l'écart moyen, entre la solution de l'heuristique et l'optimum.

Lejeune (2006) étudie la planification de la production et la distribution en intégrant la gestion des stocks pour une chaîne d'approvisionnement à trois échelons (fournisseur, producteurs et distributeurs) sur un horizon multi-périodes. La chaîne d'approvisionnement est à capacité finie pour la production, le stockage et le transport. La distribution des produits finis et des matières premières est effectuée par une flotte de véhicules hétérogène, différents à travers la capacité de chargement maximale, la vitesse et la disponibilité. L'objectif est de minimiser la somme des coûts de stockage, de production et de distribution tout en satisfaisant la demande des clients. La méthodologie proposée repose sur la technique de décomposition, en relaxant les contraintes d'intégrité des variables entières. À chaque étape de la décomposition, la recherche à voisinage variable (*VNS- Variable Neighborhood Search*) est utilisée pour définir les voisinages à explorer. Ensuite, l'algorithme du *branch and bound* permet d'effectuer la recherche locale sur les voisinages. En comparant les résultats fournis par la méthode proposée par les auteurs avec ceux obtenus avec la méthode de *branch and bound* de CPLEX, le coût total de la solution peut diminuer de 19,27%.

Chern et Hsieh (2007) développent une heuristique pour plusieurs types de produit en prenant en compte l'ensemble des acteurs de la chaîne logistique : fournisseurs, producteurs, distributeurs et détaillants. L'objectif est de minimiser les pénalités de retard, l'utilisation de capacité externe et les coûts des matières, de production, de traitement, de transport et de stockage, tout en respectant les limites de capacité ainsi que les délais des demandes. En effet, chaque demande est caractérisée par des quantités, des délais et des pénalités de retard par période. Pour tous les scénarios évalués, le temps de résolution de l'heuristique varie entre 9 et 216 secondes alors que celui du programme linéaire varie entre 10 et 937 secondes ; lorsque celui-ci trouve une solution. Les résultats du programme linéaire et de l'heuristique sont en général les mêmes ou à moins de 0.15% de différence.

L'article de Boudia *et al.* (2007) traite la planification de la production et la distribution, pour

un seul produit, sur un horizon multi-périodes. Le réseau est constitué d'une usine qui fabrique un produit et le distribue à un ensemble de clients, à travers des tournées. Les auteurs prennent en compte les lancements de production, les niveaux des stocks et la distribution. Ils proposent un modèle linéaire en nombres entiers. Afin de résoudre des instances de grande taille, ils ont développé des méta-heuristiques. Il s'agit d'une procédure aléatoire gloutonne de recherche adaptative (GRASP) ainsi que deux versions améliorées de celle-ci. Le GRASP basique est moins performant que les variantes améliorées. Les temps de calcul restent raisonnables, pour un problème tactique : 2 min pour  $n = 50$ , 7 min pour  $n = 100$  et 30-35 min pour  $n = 200$ ,  $n$  étant le nombre de clients à servir.

Le réseau étudié par Lee *et al.* (2008) est composé de quatre niveaux, à savoir : les usines de production, les entrepôts, les centres de distribution et les clients finaux. Un seul type de produit est pris en compte avec plusieurs chemins possibles (alternatives) pour satisfaire la demande d'un client. L'objectif est de minimiser les coûts de réapprovisionnement, de stockage et de transport. Trois heuristiques ont été développées : décomposition (MCF-CS), horizon mobile (RH-CS) et recherche tabou. Les résultats obtenus par la recherche tabou sont meilleurs, mais le temps de calcul est très important.

Les auteurs Manzini et Bindi (2009) présentent une approche multi-périodique intégrant la production, la gestion des stocks et le transport, y compris le chargement et l'acheminement des véhicules. Les auteurs traitent la conception et l'optimisation d'un système de production et de distribution multi-échelons qui combine un programme linéaire mixte et des heuristiques.

Bashiri *et al.* (2012) présentent un modèle de programmation linéaire mixte développé pour des décisions d'ordre stratégique telles que la conception du réseau ainsi que le mode d'expansion, et d'ordre tactique pour planifier la production et la distribution. Les quatre échelons de ce réseau sont constitués par les fournisseurs, les usines, les entrepôts et les clients. La demande des clients est dynamique et déterministe sur l'horizon de planification. L'objectif est de maximiser le profit total du réseau. L'horizon de décision stratégique comprend 5 périodes et chacune inclut 4 périodes pour la planification tactique. La plus grande instance testée comprend 35 fournisseurs, 20 usines, 22 entrepôts, 120 clients, 32 matières premières pour 15 produits finis. Pour cette instance, le solveur CPLEX ne trouve pas de solution en 3 heures. Pour des applications réels le nombre de produits finis distribués dépasse souvent la centaine.

L'article de Cocco *et al.* (2013) propose également un modèle en programmation linéaire mixte (PL mixte) intégrant les activités de production et de transport. La formulation proposée repose sur une représentation en temps continu qui permet de prendre des décisions opérationnelles sur une base quotidienne. Le problème de la production se concentre sur la planification à court terme, de plusieurs produits, en une seule étape et de multiples unités parallèles avec des temps de *setup* dépendant de l'ordre et/ou des coûts. La production de chaque produit s'effectue en lot. Un réseau comprend des usines, des entrepôts, des centres de distribution (CD) et les clients finaux. Le

transport est assuré par une flotte de véhicules hétérogène. Une instance de petite taille est testée, avec 3 produits fabriqués dans 2 usines, pouvant transiter ou être stockés dans deux centres de distribution, pour être livrés à 16 clients finaux.

Récemment, Fahimnia *et al.* (2013) présentent un état de l'art des articles sur l'optimisation de la planification intégrée de la production et de la distribution. Les auteurs notent qu'à part certains secteurs particuliers, la plupart des premiers modèles sont seulement des simplifications des problèmes réels de planification de la chaîne logistique. La tendance est donc de développer des méthodes destinées à résoudre des instances de grande taille et pouvant être utilisées pour des applications industrielles. Le Tableau 2.3 présente une synthèse d'articles qui étudient les problèmes de planification intégrée de la production et de la distribution.

Article	Nombre éche-lons	Multi-produits	Péremption	Lot-sizing	Multi-sourcing	Coût stock	Livraison	Flotte hétérogène	Capacité transport	Méthode
Chandra et Fisher (1994)	2	X		X		X	tournée			heuristiques + recherche locale
Lejeune (2006)	3				X	X	directe	X	X	branch and bound + VNS
Boudia <i>et al.</i> (2007)	2			X		X	tournée		X	méta-heuristiques
Chern et Hsieh (2007)	4	X			X	X	entrepasage			heuristique
Lee <i>et al.</i> (2008)	4			X	X	X	entrepasage			heuristiques + recherche tabou
Torabi et Hassini (2008)	3	X		X		X	directe			MOLP (multi-objective linear programming)
Gebennini <i>et al.</i> (2009)	3					X	directe + entrepasage			PNL mixte
Manzini et Bindi (2009)	4				X		entrepasage	X		PL mixte + heuristiques
Ahumada <i>et al.</i> (2012)	6	X	X		X		directe + entrepasage	X		modèle stochastique
Bashiri <i>et al.</i> (2012)	4	X			X	X	entrepasage			PL mixte
Cóccola <i>et al.</i> (2013)	3	X		X	X		directe + entrepasage + cross-docking	X	X	PL mixte

TABLE 2.3 – Classification des articles sur la planification intégrée de la production et de la distribution

## 2.7 Conclusion

Dans cet état de l'art, nous présentons les généralités sur les méthodes d'optimisation utilisées pour les problèmes de planification dans des chaînes logistiques multi-échelons ainsi que les contraintes les plus rencontrées.

Les problèmes de planification dans des chaînes logistiques multi-échelons sont en général abordés dans la littérature par des méthodes de résolution approchées (Boudia *et al.* (2007), Chern et Hsieh (2007), Lejeune (2006), Basnet et Leung (2005)), voire des techniques de simulation (Kanchanasuntorn et Techanitisawad (2006), Sanchez *et al.* (2010)). La plupart des travaux considère des chaînes logistiques réduites à quelques étapes (Hindi *et al.* (1998), Basnet et Leung (2005)), traite certaines structures de réseaux très particulières comme Brecman *et al.* (1989), ou encore ne considère qu'un seul type de produit comme dans les articles de Boudia *et al.* (2007), Lee *et al.* (2008), Li *et al.* (2006), Brahimi *et al.* (2006) et Absi *et al.* (2011). À part de rares recherches comme Boudia *et al.* (2007) et Lejeune (2006), les étapes de transport sont très simplifiées, avec des véhicules identiques et des livraisons directes. Le plus souvent, comme dans Boudia *et al.* (2007), la demande peut être satisfaite en avance mais pas en retard. Certains auteurs comme Chern et Hsieh (2007) et Brecman *et al.* (1989) autorisent des retards, en les pénalisant. Enfin, quelques articles considèrent la péremption des produits, comme dans Kanchanasuntorn et Techanitisawad (2006).

La tendance est de développer des méthodes destinées à résoudre des instances de grande taille et pouvant être utilisées pour des applications industrielles.

Les articles étudiés nous ont permis de mieux cerner les contraintes liées au problème de planification dans un réseau multi-échelons avec une capacité finie. Aucun ne présente l'ensemble des spécificités de notre problématique à savoir : la gestion des alternatives d'approvisionnement, la multiplicité des produits, les limites de capacité pour le transport, la gestion des stocks, le transport en livraison directe (sans tournées de véhicules), la gestion des durées de vie des produits combinée à des contrats de fraîcheur entre un fournisseur et un client (obsolescence des stocks), la possibilité d'approvisionner en avance ou en retard et enfin un horizon fini multi-périodes.



## Chapitre 3

# Modèle mathématique et heuristique constructive

### 3.1 Introduction

Notre étude intègre le problème de *lot-sizing* dans le cadre de la distribution, dans un environnement multi-échelons et multi-produits. Dans la littérature, les problèmes de *lot-sizing* intégrant la gestion des stocks sont surtout étudiés dans le cadre de la production : peu d'articles traitent de ces problèmes pour la distribution. De plus, beaucoup d'auteurs ne considèrent qu'un seul type de produit, pour réduire la complexité. En effet, Chen *et al.* (1994) ont montré que le CSILSP (*Capacitated Single Item Lot-Sizing Problem*) est pseudo-polynomial, tandis que Chen et Thizy (1990) ont prouvé que le MCLS (*Multi item Capacitated Lot-Sizing Problem*) est NP-difficile au sens fort.

Le type de réseau étudié a été rencontré ces dernières années chez plusieurs clients de Futur-Master, notamment dans la distribution de produits lactés ou de boissons, et avec des cas pouvant atteindre de grandes tailles. Il a donc été décidé de développer un programme linéaire mixte spécifique pour ce type de réseau et d'inclure dans le logiciel une heuristique rapide, permettant d'obtenir des solutions quand le solveur n'arrive pas à trouver des solutions optimales dans des temps acceptables. Même pour des cas à portée du solveur, certains utilisateurs ne comprennent pas la construction du résultat et préfèrent une heuristique dont la logique peut être expliquée.

Nous considérons plusieurs types de produits à acheminer en lots depuis des nœuds appelés *entrepôts* vers des nœuds nommés *dépôts*. Un dépôt peut être fourni par plusieurs entrepôts. La demande-client de chaque produit au niveau de chaque dépôt est fournie par un module de prévision sur un horizon fini. Les produits ont des dates limites d'utilisation au-delà desquelles ils deviennent obsolètes. De plus, des contrats de fraîcheur sont établis entre les dépôts et leurs clients (durée de vie restante au minimum au moment de la livraison au client). Enfin, notre modèle pénalise les livraisons en retard, ce qui constitue une pratique courante, mais aussi les avances car elles peuvent

induire un surstock.

Le problème est décrit dans la section 3.2, puis modélisé sous forme de programme linéaire mixte dans la section 3.3. Il s'agit de déterminer un plan de distribution cohérent sur toutes les étapes du flux, respectant les différentes contraintes tout en étant satisfaisant en termes de niveau de service, qualité des stocks et coût (choix de l'alternative d'approvisionnement la moins coûteuse). La section 3.4 décrit une heuristique rapide pour des cas hors de portée des solveurs de programmation linéaire. Enfin, dans la section 3.5, les performances du modèle mathématique et de l'heuristique sont évaluées en termes de consommation-mémoire, de temps de calcul et de qualité des solutions. La dernière section 3.6 résume les principales contributions de ce chapitre ainsi que les perspectives.

## 3.2 Description du problème

Le réseau de distribution considéré comprend deux échelons, à savoir : des sites de production ou entrepôts qui fournissent des produits finis et périssables à plusieurs dépôts (voir Figure 3.1). Un dépôt peut être approvisionné par plusieurs entrepôts en livraison directe ; il n'y a pas de tournées de véhicules. Le choix de la source d'approvisionnement est déterminé en fonction de la disponibilité des produits dans les stocks des entrepôts, des moyens de transport (flotte homogène), mais aussi des coûts de transport sur les arcs. La rupture n'est permise qu'au niveau des dépôts, une demande non satisfaite peut être reportée mais est pénalisée par un coût. Les niveaux de stock sont pris en compte en fin de période. Le niveau de priorité de chaque entrepôt est défini par un coût de transport associé à chaque produit expédié sur l'arc.

Pour chaque couple produit-site (c'est-à-dire produit-entrepôt et produit-dépôt), une politique de stock est définie via trois éléments :

- un stock objectif variable dans le temps : cible à atteindre idéalement ;
- un stock maximum variable dans le temps : limite au-delà de laquelle il y a un surstock et le produit présente un risque d'obsolescence ;
- un stock minimum variable dans le temps : réserve permettant de faire face aux aléas de la demande.

Les produits sont constitués en lots à l'entrepôt. Pour différencier ce concept du dimensionnement en lots, le mot *batch* désigne le lot de produit caractérisé par une quantité d'un seul type de produit, une date d'entrée dans le stock et une date limite d'utilisation (DLU). Une demande d'approvisionnement d'un dépôt peut être satisfaite par plusieurs *batch* du même produit. Lorsqu'un *batch* n'est pas consommé avant sa DLU alors la quantité restante est perdue. Elle est pénalisée par un coût appelé coût de dégagement. Pour chaque entrepôt les *batch* de produits sont ordonnés dans l'ordre croissant de leurs DLU et leur consommation s'effectue en FIFO (*First-In First-Out*).

De plus, un contrat de fraîcheur du produit est établi entre les dépôts et leurs clients : durée de vie restante au minimum au moment de la livraison au client (correspond à la période de la

demande au dépôt). Le contrat de fraîcheur est exigé par certains hypermarchés. Par exemple, pour un lot de yaourts avec une date de péremption  $t$ , l'hypermarché peut exiger d'être livré 20 jours, au minimum, avant cette date  $t$ , pour avoir le temps de le commercialiser. Dans notre modèle, chaque couple produit-dépôt est associé à un contrat de fraîcheur.

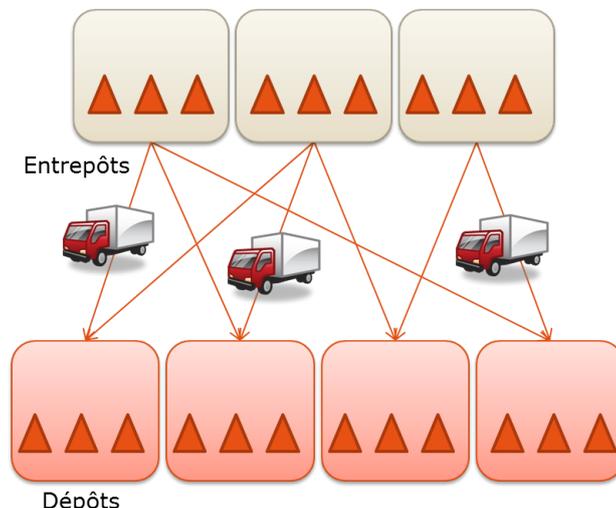


FIGURE 3.1 – Réseau de distribution à deux échelons avec alternatives

Au niveau de chacun des couples produit-site et produit-arc, une taille de lot d'approvisionnement et une quantité minimale (de commande ou de transport) sont définies. La quantité minimale exprime une quantité économique nécessaire pour rentabiliser les opérations de passation de commande et de réception. Quant à la taille de lot d'approvisionnement, elle correspond au dimensionnement fixé pour un lancement de production ou pour le transport. Elle exprime, dans notre cas, l'envoi de palettes complètes. La combinaison de ces deux exigences (taille de lot d'approvisionnement et quantité minimale) rend notre problème beaucoup plus complexe qu'un problème classique de gestion des stocks dans un environnement multi-échelons.

Pour chaque couple produit-site, des calendriers d'expédition pour les entrepôts et de réception pour les dépôts sont pris en compte. Un calendrier de transport (ouverture ou fermeture possible selon la période considérée) est aussi défini sur chaque arc de transport.

Le plan d'approvisionnement doit établir les quantités et les périodes pour livrer les entrepôts et les dépôts afin de satisfaire au mieux la demande des dépôts et atteindre le stock objectif des sites (entrepôt et dépôt). L'objectif consiste à minimiser les coûts de transport, de stock aux entrepôts et aux dépôts (pénalités appliquées aux écarts par rapport au stock objectif), et de la perte due à l'obsolescence des produits. Nous devons respecter la cohérence du flux, les contraintes de capacité, les contraintes d'approvisionnement (taille de lot d'approvisionnement, quantité minimale et dates) et chercher à optimiser l'équilibrage du stock multi-échelons.

Pour des raisons de flexibilité, l'utilisateur a la possibilité d'imposer des quantités pour l'approvisionnement des stocks des entrepôts, des dépôts, également pour le transport. Ce levier permet de

contourner les différentes contraintes dans une situation critique. Cependant, il peut engendrer des incohérences dans le calcul. Lors de l'exécution de l'heuristique, ces cas d'incohérence sont détectés et signalés sans interrompre le calcul du plan d'approvisionnement ; ce qui n'est pas possible avec le PL mixte.

### 3.3 Modèle linéaire mixte

Les entités sont les suivantes :

- Un ensemble  $K$  d'entrepôts, indicés par  $k$ ,
- Un ensemble  $L$  de dépôts, indicés par  $l$ ,
- Un ensemble  $P$  de produits, indicés par  $p$ ,
- Un horizon  $T$  de périodes de calcul, indicées par  $t$ ,
- Un horizon  $U$  des dates-limites d'utilisation, indicées par  $u$ ,  $U = 1, \dots, |T| + \max(DLU_{kp})$ .

Tous les indices commencent à 1 sauf pour le niveau de stock  $s_{ipt}$  où l'indice 0 est utilisé pour les stocks de départ. Certaines contraintes utilisent une grande constante positive  $M$ .

#### 3.3.1 Notations

Par convention, les noms des données sont en majuscule et ceux des variables en minuscule.

$D_{lpt} \in \mathbb{R}^+$ ,  $l \in L$ ,  $p \in P$ ,  $t \in T$  : demande-client pour le produit  $p$  au dépôt  $l$  à la période  $t$  ;

**Données concernant l'approvisionnement :**

$QT_{ipt}^{min} \in \mathbb{N}$ ,  $i \in K \cup L$ ,  $p \in P$ ,  $t \in T$  : quantité minimale de commande (*Quantity Minimum*) pour le produit  $p$  à l'entrepôt/dépôt  $i$  à la période  $t$  ;

$CE_{kpt} \in \{0, 1\}$ ,  $k \in K$ ,  $p \in P$ ,  $t \in T$  : calendrier d'expédition ; si  $CE_{kpt} = 1$  on peut expédier le produit  $p$  à partir de l'entrepôt  $k$  à la période  $t$ , sinon l'expédition est impossible ;

$CR_{lpt} \in \{0, 1\}$ ,  $l \in L$ ,  $p \in P$ ,  $t \in T$  : calendrier de réception ; si  $CR_{lpt} = 1$  si la réception du produit  $p$  est permise au dépôt  $l$  à la période  $t$ , sinon la réception est impossible ;

$DLU_{kp} \in \mathbb{N}$  : délai limite d'utilisation de la quantité de produits  $p$  à approvisionner à l'entrepôt  $k$ , ce délai s'applique à partir de l'arrivée à l'entrepôt ;

$MCL_{lpt} \in \mathbb{N}$  : contrat de fraîcheur (*Minimum Customer Life*) représente une durée de vie restante au minimum, au moment de la livraison au client du produit  $p$  depuis le dépôt  $l$  à la période  $t$  : elle est prise en compte à la période de la demande du dépôt ;

$FPO_{ipt} \in \mathbb{R}^+$  : quantité de produits  $p$  à approvisionner à l'entrepôt/dépôt  $i$  à la période  $t$ , imposée par l'utilisateur (*Firm planned order*) ;

$IsFPO_{ipt} \in \{0, 1\}$  :  $IsFPO_{ipt} = 1$  s'il y a une quantité imposée pour l'approvisionnement du produit  $p$  à l'entrepôt/dépôt  $i$  à la période  $t$ , donc on ne peut pas planifier une quantité

différente pour le même produit à cette période ( $FPO_{ipt}$  doit être pris en compte même si la valeur est nulle);

$QEID_{lpt} \in \mathbb{R}^+$  : quantité de produits  $p$  reçus au dépôt  $l$  à la période  $t$  à partir d'un fournisseur externe (*Quantity External In Distribution center*).

#### Données pour les stocks :

$S_{ip}^{init} \in \mathbb{R}^+$ ,  $i \in K \cup L$ ,  $p \in P$  : stock initial du produit  $p$  à l'entrepôt/dépôt  $i$ ;

$S_{ipt}^{obj} \in \mathbb{R}^+$ ,  $i \in K \cup L$ ,  $p \in P$ ,  $t \in T$  : niveau de stock objectif du produit  $p$  stocké à l'entrepôt/dépôt  $i$  à la période  $t$ ;

$S_{ipt}^{max} \in \mathbb{R}^+$ ,  $i \in K \cup L$ ,  $p \in P$ ,  $t \in T$  : niveau de stock maximum du produit  $p$  stocké à l'entrepôt/dépôt  $i$  à la période  $t$ ;

$S_{ipt}^{min} \in \mathbb{R}^+$ ,  $i \in K \cup L$ ,  $p \in P$ ,  $t \in T$  : niveau de stock minimum du produit  $p$  stocké à l'entrepôt/dépôt  $i$  à la période  $t$ ;

$P_{ipt}^{>max} \in \mathbb{R}^+$ ,  $i \in K \cup L$ ,  $p \in P$ ,  $t \in T$  : pénalité de surstock lorsqu'on dépasse le stock maximum du produit  $p$  stocké à l'entrepôt/dépôt  $i$  à la période  $t$  (par unité de stock excédentaire);

$P_{ipt}^{>obj} \in \mathbb{R}^+$ ,  $i \in K \cup L$ ,  $p \in P$ ,  $t \in T$  : pénalité d'avance lorsque le stock du produit  $p$  stocké à l'entrepôt/dépôt  $i$  à la période  $t$  est entre le stock objectif et le stock maximum (par unité de stock dépassant le niveau objectif);

$P_{ipt}^{<obj} \in \mathbb{R}^+$ ,  $i \in K \cup L$ ,  $p \in P$ ,  $t \in T$  : pénalité de retard lorsque le stock du produit  $p$  stocké à l'entrepôt/dépôt  $i$  à la période  $t$  est entre le stock minimum et le stock objectif (par unité de stock manquante pour atteindre le niveau objectif);

$P_{ipt}^{<min} \in \mathbb{R}^+$ ,  $i \in K \cup L$ ,  $p \in P$ ,  $t \in T$  : pénalité de risque de rupture lorsque le stock du produit  $p$  stocké à l'entrepôt/dépôt  $i$  à la période  $t$  est entre 0 et le stock minimum (par unité de stock manquante pour atteindre le niveau minimum);

$P_{lpt}^{rupt} \in \mathbb{R}^+$ ,  $l \in L$ ,  $p \in P$ ,  $t \in T$  : pénalité de rupture, lorsque le niveau du stock du produit  $p$  au dépôt  $l$  à la période  $t$  est en dessous de 0 (il n'est pas possible d'être en rupture de stock au niveau des entrepôts puisqu'on ne peut expédier plus que les quantités en stock);

$P_{kpt}^{obs} \in \mathbb{R}^+$ ,  $k \in K$ ,  $p \in P$ ,  $t \in T$  : pénalité d'obsolescence du produit  $p$  à l'entrepôt  $k$  à la période  $t$  (dépassement de la date limite d'utilisation).

#### Données pour le transport :

$QTA_{klpt}^{min} \in \mathbb{N}$  : quantité minimale de produits  $p$  si on transporte une quantité sur l'arc  $(k, l)$  pour un départ à la période  $t$ ;

$TL_{klp} \in \mathbb{N}$  : taille de lot de transport du produit  $p$  sur l'arc  $(k, l)$  en unités de produits, normalement c'est un multiple de palettes (pas de taille de lot si la valeur est 0);

$CT_{klpt} \in \{0, 1\}$  :  $CT_{klpt} = 1$  si le transport du produit  $p$  est permis sur l'arc  $(k, l)$  pour un départ à la période  $t$ ;

$CA_{klt} \in \mathbb{N}$ ,  $k \in K$ ,  $l \in L$ ,  $t \in T$  : capacité disponible sur l'arc  $(k, l)$  à la période  $t$ , exprimée en nombre de palettes ;

$CoPa_p \in \mathbb{R}^+$ ,  $p \in P$  : facteur de conversion quantité  $\rightarrow$  palettes du produit  $p$  ;

$HV_{kl} \in \mathbb{R}^+$ ,  $k \in K$ ,  $l \in L$  : coût d'utilisation d'un véhicule sur l'arc  $(k, l)$  (compté une seule fois même si le véhicule voyage plusieurs jours) ;

$HT_{klpt} \in \mathbb{R}^+$ ,  $k \in K$ ,  $l \in L$ ,  $p \in P$ ,  $t \in T$  : coût unitaire de transport du produit  $p$  sur l'arc  $(k, l)$  pour un départ à la période  $t$  ;

$LT_{klt} \in \mathbb{N}$ ,  $k \in K$ ,  $l \in L$ ,  $t \in T$  : délai de transport (*Lead Time*) sur l'arc  $(k, l)$  pour un départ à la période  $t$  en nombre de périodes, éventuellement nul ;

$NP^{max} \in \mathbb{N}$  : nombre maximum de palettes qu'un véhicule peut charger (identique pour tous les véhicules) ;

$PV^{max} \in \mathbb{R}^+$  : poids de chargement maximum d'un véhicule (identique pour tous les véhicules) ;

$PU_p \in \mathbb{R}^+$ ,  $p \in P$  : poids unitaire du produit  $p$  ;

$FPOA_{klpt} \in \mathbb{R}^+$  : quantité d'approvisionnement du produit  $p$  imposée à transporter sur l'arc  $(k, l)$  à la période  $t$  (utile pour imposer une origine et une destination) ;

$IsFPOA_{klpt} \in \{0, 1\}$  :  $IsFPOA_{klpt} = 1$  s'il y a une quantité imposée pour le produit  $p$  sur l'arc  $(k, l)$  à la période  $t$  ( $FPOA_{klpt}$  doit être pris en compte même si la valeur est nulle) et dans ce cas la taille de lot et la quantité minimale sont ignorées ( $IsFPOA_{klpt}$  a la priorité sur  $CT_{klpt}$ ).

#### Variables pour les ordres :

$q^{i_{ipt}}$ ,  $i \in K \cup L$ ,  $p \in P$ ,  $t \in T$  : quantité de produits  $p$  approvisionnés sur l'entrepôt/dépôt  $i$  à la période  $t$  ;

$isq^{i_{ipt}} \in \{0, 1\}$ ,  $i \in K \cup L$ ,  $p \in P$ ,  $t \in T$  :  $isq^{i_{ipt}} = 1$  ssi une quantité non nulle de produits  $p$  entre dans l'entrepôt/dépôt  $i$  à la période  $t$  ;

$qo_{kpt}$ ,  $k \in K$ ,  $p \in P$ ,  $t \in T$  : quantité de produits  $p$  expédiés par l'entrepôt  $k$  à la période  $t$  ;

$qp_{kpt}$ ,  $k \in K$ ,  $p \in P$ ,  $t \in T$  : quantité de produits  $p$  perdus, à cause de l'obsolescence, à l'entrepôt  $k$  à la période  $t$  ;

$n_{klpt} \in \mathbb{N}$ ,  $k \in K$ ,  $l \in L$ ,  $p \in P$ ,  $t \in T$  : nombre de lots qui constituent la quantité transportée sur l'arc  $(k, l)$  pour un départ à la période  $t$  ;

$qe_{klpt}$ ,  $k \in K$ ,  $l \in L$ ,  $p \in P$ ,  $t \in T$  : quantité de produits  $p$  expédiés sur l'arc  $(k, l)$  à la période  $t$  (la période de référence est la période d'expédition) ;

$isqe_{klpt} \in \{0, 1\}$ ,  $k \in K$ ,  $l \in L$ ,  $p \in P$ ,  $t \in T$  :  $isqe_{klpt} = 1$  ssi une quantité non nulle de produits  $p$  est transportée sur l'arc  $(k, l)$  à la période  $t$  ;

$qed_{klpu}$ ,  $k \in K$ ,  $l \in L$ ,  $p \in P$ ,  $u \in U$  : quantité de produits  $p$  expédiés sur l'arc  $(k, l)$  et dont la date limite d'utilisation correspond à  $u$  ;

$qedp_{klptu}$ ,  $k \in K$ ,  $l \in L$ ,  $p \in P$ ,  $t \in T$ ,  $u \in U$  : quantité de produits  $p$  expédiés sur l'arc  $(k, l)$  à la période  $t$  et dont la date limite d'utilisation correspond à  $u$  ;

$qoi_{klpt}$ ,  $k \in K$ ,  $l \in L$ ,  $p \in P$ ,  $t \in T$  : quantité de produits  $p$  expédiés par l'entrepôt  $k$ , réceptionnée au dépôt  $l$  à la période  $t$  (la période de référence est la période de réception) ;

$v_{klt}^{max}$ ,  $k \in K$ ,  $l \in L$ ,  $t \in T$  : nombre de véhicules utilisés sur l'arc  $(k, l)$  et ayant pris un départ à la période  $t$ .

#### Variables pour les stocks :

$sed_{kpu}$  : quantité de produits  $p$  dont la date limite d'utilisation est  $u$  à l'entrepôt  $k$  ;

$s_{ipt}$ ,  $i \in K \cup L$ ,  $p \in P$ ,  $t \in T$  : niveau de stock du produit  $p$  à l'entrepôt/dépôt  $i$  à la période  $t$  ;

$s_{lpt}^{rupt}$ ,  $l \in L$ ,  $p \in P$ ,  $t \in T$  : quantité manquante pour éviter la rupture (atteindre le niveau 0), lorsque le niveau de stock est négatif au dépôt  $l$  (la rupture n'est permise qu'au dépôt) ;

$s_{ipt}^{<min}$ ,  $i \in K \cup L$ ,  $p \in P$ ,  $t \in T$  : quantité manquante pour atteindre le stock minimum à partir du niveau 0, lorsque le niveau de stock est inférieur au stock minimum ;

$s_{ipt}^{<obj}$ ,  $i \in K \cup L$ ,  $p \in P$ ,  $t \in T$  : quantité manquante pour atteindre le stock objectif à partir du niveau du stock minimum, lorsque le niveau de stock est inférieur au stock objectif ;

$s_{ipt}^{>obj}$ ,  $i \in K \cup L$ ,  $p \in P$ ,  $t \in T$  : quantité d'écart entre le niveau de stock et le stock objectif lorsque le niveau de stock est supérieur au stock objectif (on ne dépasse pas le stock maximum) ;

$s_{ipt}^{>max}$ ,  $i \in K \cup L$ ,  $p \in P$ ,  $t \in T$  : quantité d'écart entre le niveau de stock et le stock maximum lorsque le niveau de stock est supérieur au stock maximum.

### 3.3.2 Fonctions pour les coûts de stock

Une fonction de coût dont l'allure est représentée par la Figure 3.2 permet d'évaluer les stocks des entrepôts et des dépôts.  $f(s_{ipt})$  désigne le coût généré par le stock du produit  $p$  dans l'entrepôt/dépôt  $i$  à la période  $t$ . Il s'agit d'une fonction convexe linéaire par morceaux. Elle a une valeur nulle lorsque le niveau de stock est égal au stock objectif. Elle correspond à un coût de stockage et éventuellement de surstock (niveau de stock supérieur au stock maximum) lorsque le niveau de stock est au-dessus du stock objectif. Lorsque le niveau du stock est inférieur au stock objectif, elle correspond à des coûts de retard et éventuellement de risque de rupture (niveau de stock inférieur au stock minimum). Pour les dépôts, on ajoute une composante pour représenter une pénalité supplémentaire en cas de stock négatif (coût de rupture). La fonction de coût de stock, au niveau des entrepôts et des dépôts, est exprimée comme suit :

$$f(s_{ipt}) = \begin{cases} \left[ P_{lpt}^{rupt} \cdot s_{lpt}^{rupt} \right] + P_{ipt}^{<min} \cdot s_{ipt}^{<min} + P_{ipt}^{<obj} \cdot s_{ipt}^{<obj} \\ + P_{ipt}^{>obj} \cdot s_{ipt}^{>obj} + P_{ipt}^{>max} \cdot s_{ipt}^{>max} \end{cases}$$

Les variables illustrées par la Figure 3.3 sont définies pour linéariser cette fonction de coût. Il s'agit en fait des quantités d'écart pour chaque morceau.

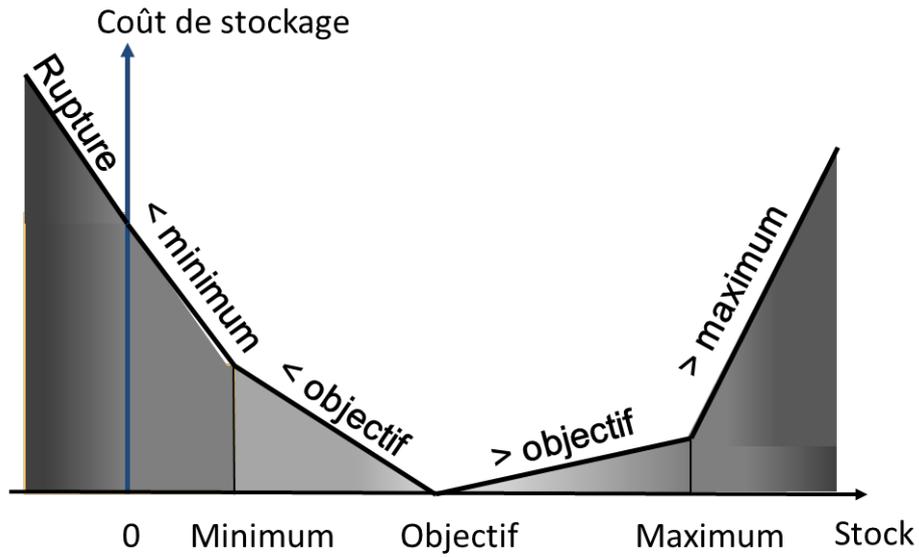


FIGURE 3.2 – Politique de stock et coûts associés (la rupture est ajoutée pour les dépôts)

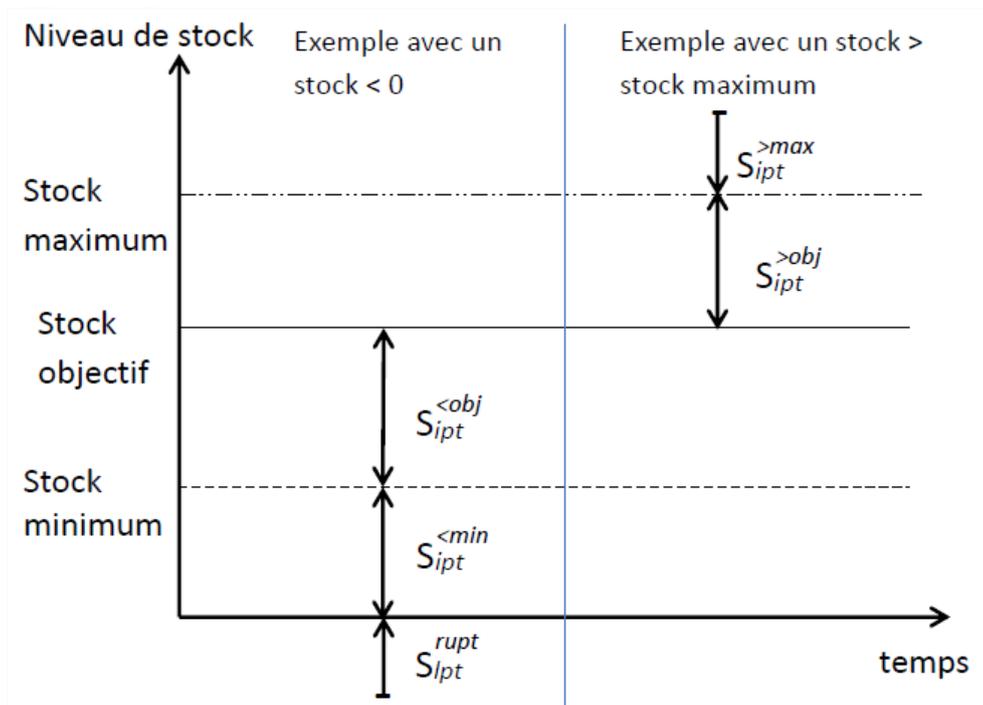


FIGURE 3.3 – Variables pour linéariser les fonctions de coût de stock

### 3.3.3 Fonction-objectif et contraintes

Dans les sommations, nous utilisons  $|K|$ ,  $|L|$ ,  $|P|$  et  $|T|$  pour indiquer le nombre d'éléments dans chaque ensemble.

Fonction-objectif :

$$\begin{aligned}
\text{Min } Z = & \underbrace{\sum_{k=1}^{|K|} \sum_{l=1}^{|L|} \sum_{t=1}^{|T|} \left[ (HV_{kl} \cdot v_{klt}^{max}) + \sum_{p=1}^{|P|} (qe_{klpt} \cdot HT_{klpt}) \right]}_{\text{coût de transport}} \\
& + \underbrace{\sum_{i=1}^{|K \cup L|} \sum_{p=1}^{|P|} \sum_{t=1}^{|T|} f(s_{ipt})}_{\text{coût de stock des entrepôts et des dépôts}} \\
& + \underbrace{\sum_{k=1}^{|K|} \sum_{p=1}^{|P|} \sum_{t=1}^{|T|} P_{kpt}^{obs} \cdot qp_{kpt}}_{\text{coût de perte due à l'obsolescence}}
\end{aligned} \tag{3.1}$$

Sous les contraintes suivantes :

$$qe_{klpt} = FPOA_{klpt} \text{ si } IsFPOA_{klpt} = 1 \quad \forall k \quad \forall l \quad \forall p \quad \forall t \tag{3.2}$$

$$qe_{klpt} = n_{klpt} \cdot TL_{klp} \text{ si } IsFPOA_{klpt} = 0, TL_{klp} \neq 0 \quad \forall k \quad \forall l \quad \forall p \quad \forall t \tag{3.3}$$

$$qe_{klpt} \geq isqe_{klpt} \cdot (1 - IsFPOA_{klpt}) \cdot QT_{klpt}^{min} \quad \forall k \quad \forall l \quad \forall p \quad \forall t \tag{3.4}$$

$$qe_{klpt} \leq isqe_{klpt} \cdot M \quad \forall k \quad \forall l \quad \forall p \quad \forall t \tag{3.5}$$

$$qe_{klpt} = qe_{klpt} \cdot CT_{klpt} \text{ si } IsFPOA_{klpt} = 0 \quad \forall k \quad \forall l \quad \forall p \quad \forall t \tag{3.6}$$

$$qoi_{klp,t+LT_{kit}} = qe_{klpt} \quad \forall k \quad \forall l \quad \forall p \quad \forall t \tag{3.7}$$

$$qi_{ipt} = FPO_{ipt} \text{ si } IsFPO_{ipt} = 1 \quad \forall i \quad \forall p \quad \forall t \tag{3.8}$$

$$qi_{ipt} \geq isqi_{ipt} \cdot (1 - IsFPO_{ipt}) \cdot QT_{ipt}^{min} \quad \forall i \quad \forall p \quad \forall t \tag{3.9}$$

$$qi_{ipt} \leq isqi_{ipt} \cdot M \quad \forall i \quad \forall p \quad \forall t \tag{3.10}$$

$$qi_{lpt} = CR_{lpt} \cdot \sum_{k=1}^{|K|} qoi_{klpt} \text{ si } IsFPO_{lpt} = 0 \quad \forall l \quad \forall p \quad \forall t \tag{3.11}$$

$$qo_{kpt} = CE_{kpt} \cdot \sum_{l=1}^{|L|} qe_{klpt} \quad \forall k \quad \forall p \quad \forall t \tag{3.12}$$

$$s_{ip0} = S_{ip}^{init} \quad \forall i \quad \forall p \tag{3.13}$$

$$s_{kpt} = s_{kpt-1} + qi_{kpt} - qo_{kpt} \quad \forall k \quad \forall p \quad \forall t > 0 \tag{3.14}$$

$$s_{lpt} = s_{lpt-1} + qi_{lpt} + QEID_{lpt} - D_{lpt} \quad \forall l \quad \forall p \quad \forall t > 0 \tag{3.15}$$

$$sed_{kpu} = 0 \quad \forall k \quad \forall p \quad \forall u < DLU_{kp} \tag{3.16}$$

$$sed_{kpu} = S_{kp}^{init} \quad \forall k \quad \forall p, \quad u = DLU_{kp} \tag{3.17}$$

$$sed_{kpu} = qi_{kpt} \quad \forall k \quad \forall p \quad \forall t, \quad u = t + DLU_{kp} \tag{3.18}$$

$$qe_{klpt} = \sum_{u=1}^{t+DLU_{kp}} qedp_{klptu} \quad \forall k \quad \forall l \quad \forall p \quad \forall t \quad (3.19)$$

$$qed_{klpu} = \sum_{t=1}^u qedp_{klptu} \quad \forall k \quad \forall l \quad \forall p \quad \forall u \quad (3.20)$$

$$\sum_{l=1}^{|L|} qed_{klpu} \leq sed_{kpu} \quad \forall k \quad \forall p \quad \forall u \quad (3.21)$$

$$\sum_{u=1}^{|T|} qed_{klpu} \leq \sum_{t=1}^{u-MCL_{lpt}} D_{lpt} \quad \forall k \quad \forall l \quad \forall p \quad (3.22)$$

$$qp_{kpt} = sed_{kpt} - \sum_{l=1}^{|L|} qed_{klpt} \quad \forall k \quad \forall p \quad \forall t \quad (3.23)$$

$$CA_{klt} \geq v_{klt}^{max} \cdot NP_{max} \quad \forall k \quad \forall l \quad \forall t \quad (3.24)$$

$$v_{klt}^{max} \geq \frac{\sum_{p=1}^{|P|} qe_{klpt} \cdot COPA_p}{NP_{max}} \quad \forall k \quad \forall l \quad \forall t \quad (3.25)$$

$$v_{klt}^{max} \geq \frac{\sum_{p=1}^{|P|} (qe_{klpt} \cdot PU_p)}{PV_{max}} \quad \forall k \quad \forall l \quad \forall t \quad (3.26)$$

$$s_{kpt}, sed_{kpu} \geq 0 \quad \forall k \quad \forall p \quad \forall t \quad \forall u \quad (3.27)$$

$$s_{lpt} \in \mathbb{R} \quad \forall l \quad \forall p \quad \forall t \quad (3.28)$$

$$n_{klpt} \in \mathbb{N} \quad \forall k \quad \forall l \quad \forall p \quad \forall t \quad (3.29)$$

$$qo_{kpt}, qp_{kpt}, q^i_{ipt}, qe_{klpt}, qoi_{klpt}, qed_{klpu}, qedp_{klptu} \geq 0 \quad \forall i \quad \forall k \quad \forall l \quad \forall p \quad \forall t \quad \forall u \quad (3.30)$$

$$v_{klt}^{max} \in \mathbb{N} \quad \forall k \quad \forall l \quad \forall t \quad (3.31)$$

$$isqe_{klpt}, isq^i_{ipt} \in \{0, 1\} \quad \forall k \quad \forall l \quad \forall p \quad \forall t \quad (3.32)$$

Commentaires :

1. La fonction-objectif inclut les coûts de transport, de stock des entrepôts et des dépôts et de perte due à l'obsolescence des produits.
2. S'il y a une quantité imposée sur l'arc  $(k, l)$  alors la quantité en transit est égale à la quantité imposée.
3. La quantité en transit correspond à  $n$  fois la taille de lot de transport, s'il n'y a pas de quantité imposée sur l'arc.
4. La quantité en transit doit être supérieure ou égale à la quantité minimale de l'arc, s'il n'y a pas de quantité imposée sur l'arc. Cette contrainte permet de respecter la contrainte de quantité minimale, s'il y a un transport.
5. La variable booléenne  $isqe_{klpt} = 1$ , s'il y a une quantité transportée sur l'arc  $(k, l)$  à la période  $t$ .
6. S'il n'y a pas de quantité imposée sur l'arc, la quantité de produits  $p$  est expédiée à la période  $t$  sur l'arc  $(k, l)$  que si le transport du produit est permis sur l'arc à cette période.

7. La quantité de produits  $p$  réceptionnés au dépôt  $l$  à la période  $t + LeadTime$ , est expédiée à  $t$  par l'entrepôt  $k$ .
8. La quantité de produits  $p$  approvisionnés à l'entrepôt/dépôt  $i$  à la période  $t$  est égale à la quantité imposée par l'utilisateur, s'il y en a.
9. La quantité de produits  $p$  approvisionnés à l'entrepôt/dépôt  $i$  à la période  $t$  doit être supérieure ou égale à la quantité minimale de commande, s'il n'y a pas de quantité imposée. Cette contrainte permet de respecter la contrainte de quantité minimale à l'entrepôt/dépôt.
10. La variable booléenne  $isqi_{ipt} = 1$ , s'il y a une quantité de produits  $p$  réceptionnés à l'entrepôt/dépôt  $i$  à la période  $t$ .
11. La quantité de produits  $p$  réceptionnés au dépôt  $l$  à la période  $t$  est égale à la somme des quantités en transit, s'il n'y a pas de quantité imposée et si la réception du produit  $p$  est permise au dépôt  $l$  à la période  $t$ .
12. La quantité de produits  $p$  sortant de l'entrepôt  $k$  à la période  $t$  est égale à la somme des quantités en transit sur les arcs tels que l'une des extrémités est égale à  $k$  et s'il est permis d'expédier le produit  $p$  à partir de l'entrepôt  $k$  à la période  $t$ .
13. À la période 0 le niveau de stock de chaque entrepôt/dépôt  $i$  est égale au stock initial.
14. L'équilibrage du flux au niveau du stock des entrepôts.
15. L'équilibrage du flux au niveau du stock des dépôts.
16. La quantité de produits  $p$  qui périssent avant son délai limite d'utilisation est nulle.
17. La quantité de produits  $p$  qui périssent à la période qui correspond au délai limite d'utilisation est égale au stock initial.
18. Si une quantité de produits  $p$  entrent dans le stock de l'entrepôt  $k$  à la période  $t$ , sa date limite d'utilisation est égale à  $t + DLU_{kp}$ .
19. La quantité de produits  $p$  expédiés sur l'arc  $(k, l)$  à la période  $t$  est égale à la somme des lots de produits expédiés à cette période avec différentes dates limites d'utilisation. Un lot de produits ne peut pas être expédié sur l'arc  $(k, l)$  avant sa date d'entrée dans l'entrepôt  $k$ .
20. La quantité de produits  $p$  expédiés sur l'arc  $(k, l)$  et qui périssent à la période  $u$  est égale à la somme des lots de produits expédiés sur cet arc et qui périssent à la période  $u$ .
21. La somme des lots de produits  $p$  expédiés à partir de l'entrepôt  $k$  et qui périssent à la période  $u$  est au plus égale au stock qui périssent à l'entrepôt  $k$  à la période  $u$ .
22. Le produit  $p$  acheminé doit être conforme à la date limite d'utilisation et au contrat de fraîcheur ; ceci permet au dépôt de disposer d'une certaine durée pour commercialiser le produit.
23. La quantité perdue à l'entrepôt  $k$  est égale à la différence entre la quantité de produits qui périssent à la période  $u$  dans le stock de l'entrepôt  $k$  et à la somme des quantités expédiées, qui périssent à la période  $u$ .

24. Le nombre de véhicules utilisés à la période  $t$  ne doit pas excéder la flotte disponible à cette période. La capacité de transport est exprimée en nombre de palettes.
25. Le nombre de véhicules utilisés sur l'arc  $(k, l)$  à la période  $t$  doit intégrer la capacité des véhicules en nombre de palettes de chargement.
26. Le nombre de véhicules utilisés sur l'arc  $(k, l)$  à la période  $t$  doit intégrer la capacité des véhicules en poids de chargement.
27. La rupture n'est pas permise au niveau des entrepôts.
28. La rupture est permise au niveau des dépôts.
29. Les contraintes (29), (30), (31) et (32) définissent le domaine de définition des variables.

La linéarisation des variables d'écart relatives au niveau de stock s'effectue en posant les contraintes suivantes pour les entrepôts et les dépôts (pour lesquels la rupture est ajoutée) :

$$s_{lpt}^{rupt} \geq 0 - s_{lpt} \quad \forall l \quad \forall p \quad \forall t \quad (3.33)$$

$$s_{ipt}^{<min} \left[ +s_{ipt}^{rupt} \text{ si } i \in L \right] \geq S_{ipt}^{min} - s_{ipt} \quad \forall i \quad \forall p \quad \forall t \quad (3.34)$$

$$s_{ipt}^{<obj} + s_{ipt}^{<min} \left[ +s_{ipt}^{rupt} \text{ si } i \in L \right] \geq S_{ipt}^{obj} - s_{ipt} \quad \forall i \quad \forall p \quad \forall t \quad (3.35)$$

$$s_{ipt}^{>obj} + s_{ipt}^{>max} \geq s_{ipt} - S_{ipt}^{obj} \quad \forall i \quad \forall p \quad \forall t \quad (3.36)$$

$$s_{ipt}^{>max} \geq s_{ipt} - S_{ipt}^{max} \quad \forall i \quad \forall p \quad \forall t \quad (3.37)$$

$$s_{lpt}^{rupt} \geq 0 \quad \forall l \quad \forall p \quad \forall t \quad (3.38)$$

$$s_{ipt}^{<min}, s_{ipt}^{<obj}, s_{ipt}^{>obj}, s_{ipt}^{>max} \geq 0 \quad \forall i \quad \forall p \quad \forall t \quad (3.39)$$

La politique de stock consiste à pénaliser les écarts à des paliers. Dans la plupart des cas industriels, l'idéal n'est pas d'avoir un stock nul. Le stock objectif permet d'assurer un taux de service convenable et de pallier les aléas de la production (dans le cas de la gestion de production) et de la demande. En terme financier, le coût de stockage est négligeable par rapport à celui des pertes engendrées par une mauvaise qualité de service et par les aléas. La valeur des différents paliers tels que le stock objectif, est paramétrable selon l'utilisateur. Si pour un industriel, l'idéal est d'avoir un stock nul en fin de période, la valeur du stock objectif est mise à 0.

### 3.4 Heuristique constructive

La grande taille de certaines instances industrielles, à traiter, empêche une résolution rapide avec un solveur de programmation linéaire. Nous proposons donc une heuristique comme méthode de résolution alternative. Le code détaillé de l'heuristique étant complexe, avec plusieurs cas particuliers, nous préférons donner un organigramme de principe plutôt qu'un algorithme touffu de plusieurs pages. L'organigramme correspond à la Figure 3.4 et commenté dans les sous-sections

suivantes. Signalons que certains utilisateurs n'acceptent pas les solutions obtenues par les solveurs de programmation linéaire car ils ne comprennent pas la logique de construction de la solution. L'heuristique, quant à elle, construit la solution étape par étape. Elle est guidée par des règles de priorité, ce qui rend le plan obtenu plus compréhensible.

L'heuristique est scindée en trois procédures principales pour mieux équilibrer les stocks. La demande est d'abord traitée sans tenir compte du stock objectif : ceci consiste à déterminer les quantités nécessaires afin d'éviter la rupture de stock des dépôts (procédure d'approvisionnement minimum). Ensuite, nous cherchons à atteindre le stock objectif. Enfin, le surplus des entrepôts est déployé vers les dépôts. Ce découpage en trois phases permet une meilleure gestion des cas de pénurie ou de capacité très limitée. On cherche à ce que tous les dépôts soient approvisionnés à hauteur de la demande (pour éviter les ruptures), avant d'essayer de constituer le stock objectif. Trois types d'approvisionnement sont possibles dans notre méthode :

- en juste à temps lorsque la période de réception au niveau du dépôt correspond à la période de la demande (pas de stockage du produit) ;
- en avance lorsque le produit est réceptionné au dépôt avant la période de la demande (stockage du produit jusqu'à la période de consommation) ;
- en retard lorsque le produit est réceptionné au dépôt après la période de la demande (le stock objectif n'est pas atteint ou encore on est en rupture de stock).

Lors de l'exécution de l'heuristique, il est possible de choisir un critère de tri pour les couples produit-dépôt (priorisation). Il s'agit de regrouper les couples produit-dépôt en ensembles selon (au choix) :

- la pénalité de rupture dans la procédure d'approvisionnement minimum ou la pénalité de retard dans la procédure d'approvisionnement pour atteindre le stock objectif ;
- la quantité relative de rupture (quantité nécessaire pour éviter la rupture/stock objectif) dans la procédure d'approvisionnement minimum ou la quantité relative au stock objectif (quantité nécessaire pour atteindre le stock objectif/stock objectif) dans la procédure d'approvisionnement pour atteindre le stock objectif ;
- le coût de stock du couple produit-dépôt qui correspond à la fonction de coût  $f(s_{ipt})$  (prend en compte à la fois la pénalité et la quantité associée).

Les couples produit-dépôt ayant la même valeur du critère appartiennent au même ensemble. Ces derniers sont triés dans l'ordre décroissant du critère. Pour chaque ensemble, la quantité de produits disponible pour l'approvisionnement est répartie au prorata de la demande des couples produit-dépôt qui le constituent.

### 3.4.1 Approvisionnement minimum pour éviter la rupture

L'acheminement de ces quantités prend en compte toutes les contraintes spécifiées dans la formulation du problème. À chaque période de l'horizon :

1. Parcourir la liste des couples produit-entrepôt.
2. Générer la liste des couples produit-dépôt bénéficiant du produit-entrepôt sélectionné.
3. Calculer la somme des quantités nécessaires pour éviter la rupture (ordre minimum) sur l'ensemble des couples produit-dépôt de la liste.
4. Répartir la quantité disponible :  
 si la quantité disponible à l'entrepôt est supérieure à la somme des ordres minima alors chaque dépôt est approvisionné à hauteur de son ordre minimum  
 sinon :
  - (a) regrouper en ensemble les couples produit-dépôt ayant la même valeur du critère ;
  - (b) trier les ensembles dans l'ordre décroissant de la valeur du critère ;
  - (c) répartir dans chaque ensemble la quantité de produits disponible à l'entrepôt au prorata des ordres minima respectifs.
5. Pour chaque produit-dépôt :
  - (a) approvisionner en juste à temps la quantité destinée à celui-ci ;
  - (b) si l'approvisionnement en juste à temps ne permet pas de fournir la totalité de la quantité destinée au dépôt sélectionné alors :
    - i. évaluer les coûts de stock générés par les approvisionnements en avance et en retard à partir de l'entrepôt sélectionné ;
    - ii. appliquer les approvisionnements en avance et en retard dans l'ordre croissant des coûts de stock tant que la totalité de la quantité n'est pas fournie au dépôt.

### 3.4.2 Approvisionnement pour atteindre le stock objectif

Cette étape consiste à déterminer et à acheminer les quantités manquantes aux dépôts afin d'atteindre le stock objectif. À chaque période de l'horizon :

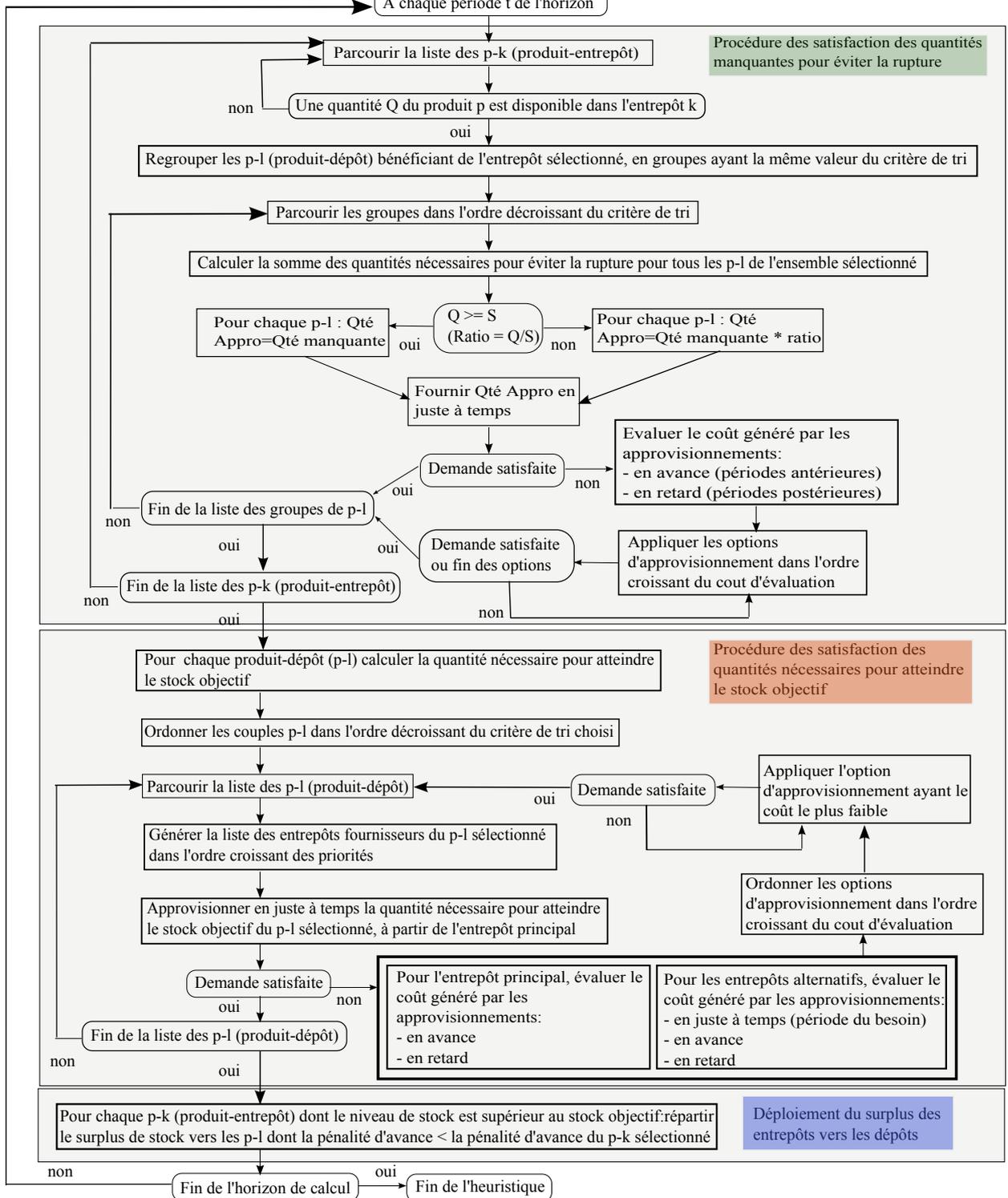
1. Parcourir la liste des couples produit-dépôt ayant un besoin d'approvisionnement, dans l'ordre décroissant du critère de tri choisi. Pour chaque couple produit-dépôt :
  - (a) générer la liste des entrepôts qui peuvent fournir le produit au dépôt sélectionné : le coût unitaire de transport définit la priorité (le coût le plus faible est associé à l'arc de transport de l'entrepôt principal) ;
  - (b) approvisionner le dépôt en juste à temps à partir de l'entrepôt principal ;
  - (c) si l'approvisionnement en juste à temps ne permet pas de fournir la totalité de la quantité destinée au dépôt sélectionné alors :
    - i. évaluer le coût de l'approvisionnement à partir de chaque entrepôt (liste précédemment générée) en juste à temps, en avance et en retard ;
    - ii. appliquer les approvisionnements en avance et en retard dans l'ordre croissant des coûts d'évaluation tant que la totalité de la quantité n'est pas fournie au dépôt.

### 3.4.3 Déploiement du surplus des entrepôts

Afin d'équilibrer les niveaux des stocks des entrepôts par rapport à ceux des dépôts, une étape est ajoutée pour effectuer le déploiement du stock en surplus. Il s'agit de déployer la quantité au-dessus du stock objectif des entrepôts vers les dépôts ayant une pénalité d'avance inférieure à celle de l'entrepôt fournisseur. Cette pénalité est appliquée lorsque le niveau de stock est entre le stock objectif et le stock maximum. Le déploiement de stock s'effectue dans la limite du stock maximum des dépôts. À chaque période de l'horizon, il s'agit de :

1. Sélectionner les couples produit-entrepôt qui disposent d'un stock supérieur à leur stock objectif.
2. Pour chaque produit-entrepôt sélectionné, comparer la pénalité d'avance à celle des couples produit-dépôt bénéficiaires : on ne déploie que vers les couples produit-dépôt ayant une pénalité d'avance strictement inférieure à celle du produit-entrepôt.

FIGURE 3.4 – Algorithme



## 3.5 Tests

L'heuristique est implémentée en C++ avec l'environnement de développement Visual Studio 2010 tandis que le PL mixte est résolu avec le solveur CPLEX en version 12.5. Les instances sont testées sur un serveur, avec un processeur Intel Xeon à 2,93 GHz et 48 GO de RAM.

### 3.5.1 Instances de test

Nous testons 20 instances (voir Tableau 3.1) avec un petit ou un grand réseau afin d'évaluer l'écart entre les résultats fournis par l'heuristique et ceux obtenus à partir de la résolution du PL mixte. Le premier réseau de distribution est constitué de 2 entrepôts et 2 dépôts. Les besoins des couples produit-dépôt sont générés aléatoirement. 10 instances issues de ce réseau sont construites en faisant varier le nombre de produits de 2 à 10 ainsi que le nombre de périodes de l'horizon de calcul (10 et 20).

Une base de données de grande volumétrie issue de l'industrie est utilisée pour le deuxième réseau de distribution, qui comprend 11 entrepôts et 145 dépôts. Une quantité minimale de transport est définie sur les combinaisons produit-route (produit x entrepôt x dépôt), ce qui rend d'après nos essais le programme linéaire mixte plus difficile. Dans le problème d'origine, il y a 495 produits mais 10 instances sont générées en faisant varier le nombre de produits (de 5 à 20), le nombre de périodes (de 10 à 20) et la quantité minimale de transport (0 et 100). Nous n'avons pas pu tester d'instances avec plus de 20 produits avec le solveur CPLEX à cause de la saturation de la mémoire.

Pour les 20 instances, les différents paliers de stock (stock maximum, stock objectif et stock minimum) sont définis par rapport à la couverture des besoins, à savoir :

- le stock maximum de chaque couple produit-dépôt à chaque période correspond à la somme des besoins des 4 périodes suivantes ;
- le stock objectif de chaque couple produit-dépôt à chaque période correspond à la somme des besoins des 2 périodes suivantes ;
- le stock minimum de chaque couple produit-dépôt à chaque période correspond au besoin de la période suivante.

Les pénalités associées au stock sont définies de la manière suivante :

- pénalité de surstock = 5 000 ;
- pénalité d'avance = 1 ;
- pénalité de retard = 100 pour les instances 1 à 10 et 500 pour les instances 11 à 20 ;
- pénalité de risque de rupture = 1 000 ;
- pénalité de rupture = 10 000 pour les instances 1 à 10 et 30 000 pour les instances 11 à 20 ;
- pénalité d'obsolescence = 10 000.

Les valeurs appliquées aux pénalités de stock sont inspirées de paramétrage réel, utilisé par des clients de FuturMaster.

Pour de mettre en évidence l'impact des critères de tri, un troisième réseau issu d'une autre base de données de l'industrie est testé. Il est composé d'un seul entrepôt central qui fournit 2766 produits à 151 dépôts. Cette base de données contient :

- 2 766 combinaisons de produit x entrepôt
- 15 349 combinaisons de produit x dépôt
- 15 349 combinaisons de produit x entrepôt x dépôt

En faisant varier le nombre de périodes de l'horizon de 10 à 100, 6 instances sont générées.

Test	$ P $	$ K $	$ L $	Couples produit-route	Nombre périodes	Qté min (transport)	Taille lot (transport)	DLU
1	2	2	2	8	10	1 000	10	8
2	3	2	2	12	10	1 000	10	8
3	4	2	2	16	10	1 000	10	8
4	5	2	2	20	10	1 000	10	6
5	6	2	2	24	10	1 000	10	6
6	7	2	2	28	10	1 000	10	6
7	8	2	2	32	10	1 000	10	6
8	9	2	2	36	10	1 000	10	6
9	10	2	2	40	10	1 000	10	6
10	10	2	2	40	20	1 000	10	6
11	5	11	145	154	10	100	0	6
12	5	11	145	154	20	100	0	6
13	10	11	145	536	10	100	0	6
14	10	11	145	536	10	100	0	6
15	10	11	145	404	10	100	0	6
16	15	11	145	690	10	100	0	6
17	15	11	145	520	10	100	0	6
18	20	11	145	932	10	100	0	6
19	10	11	145	536	15	0	0	6
20	15	11	145	690	15	0	0	6

TABLE 3.1 – Description des instances du premier et du deuxième réseau

### 3.5.2 Comparaison de l'heuristique avec le programme linéaire mixte

Les résultats du PL mixte et de l'heuristique sont donnés avec le temps de calcul en secondes et la consommation-mémoire (MO) dans le Tableau 3.2. Nous n'avons pas défini de limite de temps lors de la résolution du PL mixte. L'heuristique est exécutée sans priorisation des couples produit-dépôt. Sur les instances testées le temps de calcul de l'heuristique est toujours inférieur (0,01 à 1,58 secondes) à celui du PL mixte (0,5 à 16 707,21 secondes) (voir Table 3.2). Le temps de calcul devient très important dès qu'on augmente la taille des instances alors que celui de l'heuristique reste toujours inférieur à 2 secondes quelle que soit l'instance. L'heuristique consomme également

moins de mémoire pour le calcul : la consommation-mémoire de l'heuristique varie entre 0,1 et 3 MO, alors qu'avec le PL mixte elle varie entre 19,293 et 19 451,18 MO.

Pour certaines instances, le solveur n'atteint pas l'optimum à cause de la saturation de la mémoire. Le critère d'arrêt est le pourcentage d'écart entre la borne inférieure et la meilleure solution réalisable, qui est paramétré à 0,01%. Il nous a donc paru intéressant d'indiquer l'écart entre la meilleure solution entière (UB) et la meilleure borne inférieure (LB) lorsque le solveur n'atteint pas l'optimum : cet écart varie entre 0 et 0,265%.

Les deux méthodes sont comparées sur la base de la valeur de la fonction-objectif. Nous calculons l'écart entre le coût de la solution de l'heuristique (H) et celui de la solution optimale. Lorsque le solveur n'atteint pas l'optimum, l'écart est calculé entre le coût de la solution de l'heuristique (H) et celui de la meilleure borne inférieure (LB). Les formules sont les suivantes :

$$\% \text{ écart UB/LB} = \frac{\text{UB} - \text{LB}}{\text{LB}} \times 100$$

$$\% \text{ écart H/LB} = \frac{\text{H} - \text{LB}}{\text{LB}} \times 100$$

Pour les 10 premières instances, l'écart moyen entre la solution fournie par l'heuristique et la solution optimale est égal à 0,01%. Pour les instances de 11 à 18, l'une des raisons de l'augmentation de l'écart (10,636 à 28,886%) est la combinaison de la quantité minimale de transport et des dates limites d'utilisation des produits par rapport aux besoins des couples produit-dépôt. Un lot de produits (quantité + date limite d'utilisation) n'est expédié vers un dépôt que lorsque la somme des besoins sur les périodes constituant la durée de vie du lot de produits est suffisante pour éviter qu'une partie du lot ne devienne obsolète.

Nous remarquons une diminution importante de l'écart (0,586 et 0,098%) dès que la quantité minimale de transport est mise à 0, il s'agit des instances 19 et 20. Une autre raison est que l'heuristique n'effectue pas de mutualisation des quantités à expédier. En effet, la première option de la méthode est d'envoyer les quantités nécessaires en juste à temps. L'expédition en avance ou en retard n'est évaluée que si la totalité de la demande n'est pas satisfaite en juste à temps (à cause des différentes contraintes). L'écart moyen entre la solution fournie par l'heuristique par rapport à l'optimum est de 7,99%.

### 3.5.3 Impacts des critères de priorité

Lors du traitement des besoins (soit pour éviter la rupture, soit pour atteindre le stock objectif), les couples produit-dépôt en demande d'approvisionnement sont triés dans l'ordre défini par un critère de priorité, choisi par l'utilisateur. Cependant, il est aussi possible d'affecter le même poids aux dépôts. Dès lors, l'heuristique effectue un équilibrage et chaque dépôt reçoit une quantité, correspondant au prorata de sa demande, en fonction de la quantité disponible à l'entrepôt fournisseur.

Test	PL mixte				Heuristique (H)				
	LB	UB	% écart UB/LB	Durée	Mém.	% écart H/LB	Durée	Mém.	
1	1 274 418 317	1 274 418 317	0,000	6,82	32,86	0,001	0,01	0,10	
2	2 745 506 142	2 745 556 443	0,002	0,50	19,29	0,001	0,01	0,10	
3	3 411 486 703	3 411 532 891	0,001	1,45	30,40	0,000	0,01	0,10	
4	4 201 995 850	4 202 061 714	0,002	11,57	102,81	0,000	0,02	0,10	
5	5 462 645 255	5 462 672 078	0,000	0,80	26,68	0,001	0,02	0,10	
6	6 230 942 992	6 230 959 881	0,000	1,95	35,81	0,000	0,02	0,10	
7	7 219 747 716	7 219 751 124	0,000	3,65	45,86	0,000	0,02	0,10	
8	7 962 906 613	7 962 911 689	0,000	3,37	39,63	0,000	0,02	0,10	
9	8 926 061 700	8 926 088 031	0,000	5,16	65,10	0,002	0,02	0,10	
10	8 935 231 925	8 935 619 353	0,004	259,79	296,18	0,093	0,04	0,30	
Moyenne			0,001	29,51	69,46	0,010	0,02	0,12	
11	415 839 018	415 859 059	0,005	21,92	229,18	17,230	0,33	0,60	
12	1 445 225 147	1 445 369 566	0,010	172,70	277,50	10,636	1,58	1,00	
13	1 049 380 000	1 052 160 000	0,265	10 626,94	7 311,61	14,119	0,04	1,50	
14	78 056 651	78 064 202	0,010	17,73	138,07	28,886	0,08	2,60	
15	837 748 606	837 832 389	0,010	15 714,18	3 191,60	24,044	1,05	3,00	
16	493 871 524	493 920 692	0,010	61,20	325,31	19,068	0,55	1,40	
17	1 198 700 000	1 201 330 000	0,219	11 910,86	8 838,93	19,200	1,43	2,20	
18	965 281 833	965 378 369	0,010	3 580,71	1 464,00	25,842	1,49	2,40	
19	420 373 000	420 760 000	0,092	16 707,21	19 451,18	0,586	0,01	1,80	
20	497 180 000	497 645 000	0,094	15 457,87	18 997,70	0,098	0,02	1,90	
Moyenne			0,072	7427,13	6022,51	15,971	0,66	1,84	
Moyenne des bases 1 et 2			0,037	3728,32	3045,99	7,990	0,34	0,98	

TABLE 3.2 – Comparaison Heuristique-PL mixte

L'heuristique est évaluée avec différents critères de priorité en utilisant les six instances dérivées du troisième réseau. Chaque instance est définie par trois lettres : T correspond au nombre de périodes de l'horizon de calcul, Q représente la quantité minimale de transport appliquée sur chaque arc de transport et C est le critère de priorité, appliqué lors de l'exécution de l'heuristique. La version avec priorités notée HCT est comparée avec l'heuristique (H), la version sans priorités déjà décrite. Le Tableau 3.3 donne les résultats pour quatre critères de priorité C :

1. Sans priorité, on équilibre la répartition du stock disponible, pour les dépôts ayant un besoin d'approvisionnement (équivalent à l'heuristique H) ;
2. Tri des couples produit-dépôt dans l'ordre décroissant :
  - des pénalités de rupture lors du traitement de l'ordre minimum (quantité nécessaire pour éviter la rupture) ;
  - des pénalités de retard lors du traitement des quantités nécessaires pour atteindre le stock objectif ;
3. Tri des couples produit-dépôt dans l'ordre décroissant de :

- la quantité relative de rupture dans la procédure d'approvisionnement minimum ;
  - la quantité relative au stock objectif dans la procédure d'approvisionnement pour atteindre le stock objectif ;
4. Tri des couples produit-dépôt dans l'ordre décroissant du coût de stock global (fonction linéaire par morceaux).

Nous calculons le pourcentage d'écart entre les coûts trouvés par les versions à critère de priorité et l'heuristique sans priorité, selon la formule :

$$\% \text{ Écart HCT/H} = \frac{\text{HCT} - \text{H}}{\text{H}} \times 100$$

Le coût de la solution diminue systématiquement de 0,6 à 1,2% lorsqu'on applique le critère d'écart au stock objectif et celui du coût de stock global pour l'ensemble des 6 instances testées. En effet, la version sans priorité ne prend pas en compte l'état des stocks des dépôts, alors que les critères 3 et 4 considèrent les besoins futurs et permettent ainsi une anticipation sur les ruptures éventuelles.

T	Q	C	Coût solution	% Écart HCT/H
10	50	1	518 081 265	0,0
		2	518 080 628	0,0
		3	515 163 423	-0,6
		4	515 155 889	-0,6
20	50	1	1 049 590 668	0,0
		2	1 049 590 668	0,0
		3	1 040 972 734	-0,8
		4	1 040 971 150	-0,8
30	50	1	1 622 887 843	0,0
		2	1 622 979 583	0,0
		3	1 610 940 073	-0,7
		4	1 610 940 073	-0,7
40	50	1	1 849 078 316	0,0
		2	1 849 078 316	0,0
		3	1 832 474 644	-0,9
		4	1 832 476 930	-0,9
50	50	1	1 987 483 985	0,0
		2	1 987 476 182	0,0
		3	1 970 960 972	-0,8
		4	1 970 953 460	-0,8
100	50	1	2 453 612 386	0,0
		2	2 441 065 911	-0,5
		3	2 424 576 310	-1,2
		4	2 424 576 301	-1,2

TABLE 3.3 – Impact des critères de priorité

### 3.6 Conclusion

Un programme linéaire mixte et une heuristique sont développés pour résoudre un problème réel complexe de planification dans un réseau de distribution à deux échelons avec des capacités finies et des alternatives d'approvisionnement (*multi-sourcing*) en lots pour des produits périssables. L'heuristique est une alternative au programme linéaire mixte pour résoudre rapidement certaines instances de grande taille en prenant en compte des contraintes originales et difficiles. Pour certaines instances testées, l'écart entre les coûts de la solution de l'heuristique et la meilleure borne inférieure fournie par le solveur devient assez important. L'écart moyen entre la solution fournie par l'heuristique par rapport à l'optimum est de 7,99% pour l'ensemble des instances testées.

L'une des principales raisons est qu'une heuristique constructive prend des décisions définitives sans pouvoir revenir en arrière. Une autre raison est que l'heuristique proposée n'effectue pas de mutualisation des quantités à expédier. La première option de la méthode est d'envoyer les quantités nécessaires en juste à temps. L'expédition en avance ou en retard n'est évaluée que si la totalité de la demande n'est pas satisfaite en juste à temps. La combinaison, des contraintes de quantité minimale et de taille de lot de transport, entraîne une multiplication des effets de l'absence de mutualisation.

L'heuristique constructive peut être exécutée dans sa version de base (sans priorisation) ou avec des critères de priorité. L'utilisateur peut choisir un critère pour guider la construction de la solution dans les cas où la ressource (stock en entrepôt) n'est pas suffisante pour satisfaire la totalité des demandes des dépôts. Il s'agit, par exemple de donner la priorité à un type de produit avec une forte valeur ou à un dépôt pour lequel la qualité du service est prioritaire. Cette flexibilité permet de répondre à différents cas industriels.

Ces développements sont intégrés dans le logiciel APS de FuturMaster, pour traiter des applications rencontrées dans la distribution de boissons et de produits frais, certaines pouvant atteindre 20000 produits.

Les résultats de l'heuristique constructive, sur les 10 premières instances, ont été présentés à la conférence GOL 2012, Gestion Opérationnelle de la Logistique (conférence IEEE section France) à l'Université du Havre, en octobre 2012. Lors de la quatorzième conférence de la ROADEF (Société Française de Recherche Opérationnelle et d'Aide à la Décision) à l'Université de Troyes, en février 2013, les résultats du modèle mathématique et de l'heuristique constructive, sur l'ensemble des instances testées dans ce chapitre, ont été présentés.

Dans le chapitre suivant, nous analysons des versions randomisées de l'heuristique constructive, pour tenter d'améliorer ses résultats en moyenne. Il est aussi prévu de tester plus d'instances.

## Chapitre 4

# Heuristique réactive randomisée

### 4.1 Introduction

La performance d'une heuristique gloutonne est, en général, liée à l'instance et à l'ordre de traitement des demandes. Nous l'avons remarqué dans le chapitre précédent. Comme ce type de méthode est très rapide, une première idée est d'effectuer en boucle plusieurs constructions de solutions, en modifiant l'ordre d'examen des couples produit-entrepôt (sources d'approvisionnement) et des couples produit-dépôt (niveau de la demande). Ces constructions sont faites de façon aléatoire, en randomisant l'ordre de traitement des couples produit-entrepôt et produit-dépôt. À la fin de la boucle, nous retournons la meilleure solution trouvée. En général, ce procédé permet d'améliorer la qualité de la solution.

Cependant, il existe des cas limites pour lesquels la randomisation ne permet pas de corriger les faiblesses de l'heuristique, à cause de la structure de celle-ci. L'heuristique a donc été renforcée en la rendant réactive, c'est-à-dire qu'elle effectue des statistiques pour modifier les probabilités de choix des couples en fonction des résultats obtenus. Un générateur aléatoire d'instances est ensuite développé pour générer à volonté des instances de toutes tailles, au lieu de faire des extractions de bases de données réelles comme dans le chapitre précédent. Ce générateur est utilisé pour comparer la résolution par CPLEX avec l'heuristique déterministe du chapitre précédent et l'heuristique réactive randomisée. Les instances issues des bases de données du chapitre précédent, sont également testées avec l'heuristique réactive randomisée.

La section 4.2 décrit la méthode de randomisation utilisée pour le tri des listes des couples produit-entrepôt et produit-dépôt. L'heuristique réactive randomisée est expliquée dans la section 4.3. Les 40 instances issues du générateur aléatoire sont détaillées dans la section 4.4. Dans cette section, les résultats de l'heuristique réactive randomisée sont comparés à ceux de l'heuristique déterministe et à ceux de CPLEX. La section 4.5 résume les principales contributions de ce chapitre et donne les perspectives.

## 4.2 Méthode de randomisation

Rappelons que l’heuristique déterministe est scindée en trois procédures principales, pour mieux équilibrer les stocks. La demande est d’abord traitée sans tenir compte du stock objectif : ceci consiste à déterminer les quantités nécessaires afin d’éviter la rupture de stock des dépôts (procédure d’approvisionnement minimum). Ensuite, nous cherchons à atteindre le stock objectif. Enfin, le surplus des entrepôts est déployé vers les dépôts. Ce découpage en trois phases permet une meilleure gestion des cas de pénurie ou de capacité très limitée, puisqu’on cherche à ce que tous les dépôts soient approvisionnés à hauteur de la demande (pour éviter les ruptures), avant d’essayer de constituer le stock objectif. Deux types de randomisation sont intégrés dans l’heuristique. Il s’agit du tri de la liste des couples produit-entrepôt et celle des couples produit-dépôt.

### 4.2.1 Randomisation de la liste des couples produit-entrepôt

Rappelons que dans la procédure d’approvisionnement minimum, la liste des couples produit-entrepôt est triée puis parcourue dans l’ordre « naturel » (chaque couple produit-entrepôt est indexé par son adresse mémoire). Pour chaque couple produit-entrepôt, la liste des couples produit-dépôt bénéficiaires est générée. La répartition de la quantité disponible est effectuée, afin d’éviter la rupture de stock au niveau des couples produit-dépôt.

L’objectif de la première randomisation est de tester différents parcours de la liste des couples produit-entrepôt. On balaie la liste naturelle des couples produit-entrepôt avec un indice  $i$ . Pour chaque valeur de  $i$ , on tire au sort selon une loi uniforme le couple dans la RCL (restricted candidate list) formée des  $k$  couples d’indices  $i$  à  $i + k - 1$ ,  $k$  étant un petit entier à choisir. Le cas  $k = 1$  revient à l’heuristique déterministe.

### 4.2.2 Randomisation de la liste des couples produit-dépôt

Rappelons que dans la procédure d’approvisionnement, on affecte à chaque couple produit-dépôt un nombre réel positif, appelé criticité. On trie la liste en ordre décroissant de ces criticités. Par défaut, cette liste est triée dans l’ordre « naturel » (chaque couple produit-dépôt est indexé par son adresse mémoire).

Pour cette randomisation, on change aléatoirement l’ordre de la liste des couples produit-dépôt. On tire au sort selon une loi uniforme une position qu’on attribut à chaque couple produit-dépôt. Après exécution de l’heuristique avec cette liste, on augmente chaque criticité aléatoirement entre 0 et  $p\%$ , puis on trie avant d’appeler à nouveau l’heuristique. Le paramètre  $p$  correspond au pourcentage maximum d’augmentation de la criticité.

### 4.3 Heuristique Réactive Randomisée (HRR)

Nous nous sommes inspirés de l'article de Prais et Ribeiro (2000) et celui de Prins et Wolfer Calvo (2005). Prais et Ribeiro ont proposé une méthode réactive GRASP (*Greedy Randomized Search Procedure*) pour un problème en télécommunications. Prins et Wolfer Calvo ont également proposé un GRASP, dans laquelle on exécute plusieurs fois une heuristique gloutonne randomisée, qu'on améliore avec une recherche locale.

Nous appelons « réglage » un couple de paramètres  $(k, p)$ . Des essais préliminaires avec un seul réglage ayant été décevants, nous autorisons plusieurs réglages dont les probabilités de sélection sont uniformes au début. L'heuristique randomisée est appelée en choisissant au hasard un réglage puis le score moyen (somme des coûts obtenus par ce réglage, divisé par le nombre d'exécutions avec ce réglage) est mis à jour et les probabilités de sélection sont biaisées, en fonction des probabilités obtenues. Il s'agit donc d'un processus d'apprentissage qui conduit l'heuristique, après un certain nombre d'exécutions, à choisir le meilleur réglage. Les valeurs possibles pour  $k$  sont : 1, 10, 20, 30 et 50. Les valeurs possibles pour  $p$  sont : 5, 10, 15, 20 et 25%. Les valeurs de  $k$  et  $p$  sont choisies arbitrairement.

Un réglage correspond à la combinaison de 2 paramètres issus de ces listes. Les notations utilisées sont les suivantes :

- $\gamma$  correspond au nombre de réglages possibles,
- $i$  correspond au numéro du réglage,
- $Cost_{Best}$  est le coût de la meilleure solution trouvée,
- $\alpha_i$  correspond au réglage  $i$  : par exemple  $\alpha_1 = (1, 5)$ ,
- $p_i$  correspond à la probabilité d'utiliser le réglage  $\alpha_i$ ,
- $N_i$  est le nombre d'exécution utilisant le réglage  $\alpha_i$ ,
- $F_i$  est la somme des coûts des solutions obtenues avec le réglage  $\alpha_i$ ,
- $\beta$  correspond à la fréquence de mise à jour des probabilités,
- $j$  représente l'itération courante, il varie de 1 à  $max_{iterator}$ ,
- $q_i$  est une valeur permettant de mettre à jour les probabilités : la somme des  $q_i$  est notée  $\sigma$ ,
- $\delta$  est un facteur d'amplification qui sert à modifier plus fortement les probabilités.

Avec la combinaison des deux paramètres, nous obtenons 25 configurations possibles. La liste finale de candidats comprend  $\gamma$  couples  $(\alpha_1, \alpha_2, \dots, \alpha_\gamma)$  telles que  $\alpha_1 = (1, 5), \alpha_2 = (1, 10), \dots, \alpha_\gamma = (50, 25)$  (voir Tableau 4.1).

L'algorithme de l'heuristique réactive randomisée (voir Algorithme 4.1) se déroule de la manière suivante :

- initialisations du coût de la meilleure solution, des probabilités, du nombre maximum d'itérations et de la somme des coûts des solutions de chaque réglage,
- choix aléatoire d'un réglage, selon une loi uniforme,
- exécution de l'heuristique randomisée, avec le réglage choisi,

- archivage de la meilleure solution trouvée,
- mise à jour des probabilités des réglages, selon une fréquence  $\beta$  définie,
- arrêt si le critère d'arrêt est atteint : dans notre cas il s'agit d'un nombre maximum d'itérations.

---

**Algorithme 4.1** Heuristique réactive randomisée
 

---

**Paramètres**  $\alpha_i, max_{iterator}, \beta, \delta$ 

```

1:  $Cost_{Best} := +\infty$ 
2: Pour  $i := 1 \rightarrow \gamma$  Faire
3:    $p_i := 1/\gamma$ 
4:    $N_i := 0$ 
5:    $F_i := 0$ 
6: Fin Pour
7: Pour  $j := 1 \rightarrow max_{iterator}$  Faire
8:   Choisir aléatoirement le réglage  $\alpha_i$  en se basant sur la probabilité  $p_i$  (voir Algorithme 4.2)
9:   Exécuter l'heuristique randomisée avec le réglage  $\alpha_i$ 
10:  Si  $Cost(Sol) < Cost_{Best}$  Alors
11:     $Sol_{Best} := Sol$  : mise à jour de la meilleure solution
12:     $Cost_{Best} := Cost(Sol)$ 
13:  Fin Si
14:   $F_i := F_i + Cost(Sol)$ 
15:   $N_i := N_i + 1$ 
16:  Si  $j \bmod \beta = 0$  : modifier les probabilités chaque  $\beta$  itérations Alors
17:    Pour  $i := 1 \rightarrow \gamma$  Faire
18:       $q_i := (Cost_{Best}/(F_i/N_i))^\delta$ 
19:       $\sigma := \sigma + q_i$ 
20:    Fin Pour
21:    Pour  $i := 1 \rightarrow \gamma$  Faire
22:       $p_i := q_i/\sigma$  : Mise à jour des probabilités des réglages
23:    Fin Pour
24:  Fin Si
25: Fin Pour

```

---

## 4.4 Tests

L'heuristique réactive randomisée est implémentée en C++ avec l'environnement de développement Visual Studio 2010 tandis que le PL mixte est résolu avec le solveur CPLEX en version 12.5. Les instances sont testées sur un serveur avec un processeur Intel Xeon de 2.93 GHz et 48 GO de RAM.

**Algorithme 4.2** Choix aléatoire du réglage à utiliser

- 
- 1: Tirer au sort  $U$  dans  $[0,1[$  selon une loi uniforme
  - 2:  $somme := 0$
  - 3:  $i := 1$
  - 4: **Tant que**  $somme + p_i < U$  **Faire**
  - 5:    $somme := somme + p_i$
  - 6:    $i := i + 1$
  - 7: **Fin Tant que**
  - 8: **Retourne**  $i$  : le numéro du réglage à utiliser
- 

Réglage	Nombre d'éléments couples produit-entrepôt	% perturbation criticité couples produit-dépôt
$\alpha_1$	1	5
$\alpha_2$	1	10
$\alpha_3$	1	15
$\alpha_4$	1	20
$\alpha_5$	1	25
$\alpha_6$	10	5
$\alpha_7$	10	10
$\alpha_8$	10	15
$\alpha_9$	10	20
$\alpha_{10}$	10	25
$\alpha_{11}$	20	5
$\alpha_{12}$	20	10
$\alpha_{13}$	20	15
$\alpha_{14}$	20	20
$\alpha_{15}$	20	25
$\alpha_{16}$	30	5
$\alpha_{17}$	30	10
$\alpha_{18}$	30	15
$\alpha_{19}$	30	20
$\alpha_{20}$	30	25
$\alpha_{21}$	50	5
$\alpha_{22}$	50	10
$\alpha_{23}$	50	15
$\alpha_{24}$	50	20
$\alpha_{25}$	50	25

TABLE 4.1 – Liste des réglages utilisés pour l'heuristique réactive randomisée

**4.4.1 Instances de test**

L'heuristique déterministe (H) et l'heuristique réactive randomisée (HRR) ont été comparées sur les 20 instances du chapitre précédent (issues de bases de données de cas réels) et sur 40 instances obtenues par un générateur que nous avons spécialement conçu.

#### 4.4.1.1 Génération d'instances aléatoires

Les données telles que la taille de lot et la quantité minimale d'approvisionnement, le délai, le coût et le calendrier de transport, sont générées aléatoirement pour chaque couple produit-arc. Il en est de même pour la demande, le contrat de fraîcheur et le stock initial de chaque couple produit-dépôt. Pour chaque couple produit-entrepôt, la date limite d'utilisation et le stock initial sont générés aléatoirement. Les produits sont classés dans des catégories prédéfinies, selon la valeur du produit et l'état du cycle de vie (nouveau produit ou produit en fin de vie, remplacé par un autre). Pour les produits à grande valeur ajoutée, la rupture est fortement pénalisée et pour les produits en fin de vie le surstock est fortement pénalisé (voir Tableau 4.2).

La fonction de coût permet d'évaluer les stocks des entrepôts et des dépôts. La Figure 4.1 rappelle la fonction de coût des stocks. Il s'agit d'une fonction convexe linéaire par morceaux. Elle a une valeur nulle, lorsque le niveau de stock est égal au stock objectif. Lorsque le niveau de stock est au-dessus du stock objectif, elle correspond à un coût de stockage et éventuellement de surstock : niveau de stock supérieur au stock maximum. Lorsque le niveau du stock est inférieur au stock objectif, elle correspond à des coûts de retard et éventuellement de risque de rupture : niveau de stock inférieur au stock minimum. Pour les dépôts, on ajoute une composante pour représenter une pénalité supplémentaire en cas de stock négatif (coût de rupture). Lorsqu'un lot de produits n'est pas consommé avant sa date limite d'utilisation alors la quantité restante est perdue. Elle est pénalisée par un coût appelé pénalité de dégagement.

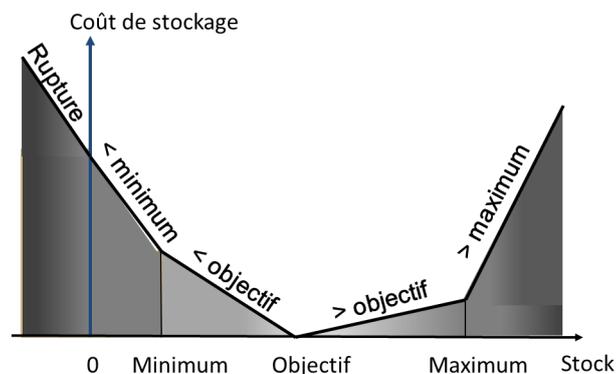


FIGURE 4.1 – Politique de stock et coûts associés (la rupture est ajoutée pour les dépôts)

Pour toutes les instances, le besoin (demande) de chaque couple produit-dépôt, à chaque période, est généré aléatoirement. Pour tous les couples produit-dépôt et produit-entrepôt :

- le stock maximum, à chaque période, correspond à la somme des besoins des 4 périodes suivantes,
- le stock objectif, à chaque période, correspond à la somme des besoins des 2 périodes suivantes,
- le stock minimum, à chaque période, correspond au besoin de la période suivante.

Les valeurs appliquées aux pénalités de stock et aux paliers de stock (stock maximum, stock

<b>Valeur du produit</b>	A (valeur très élevée)	B (valeur élevée)	C	D	E (peu important)
Pénalité rupture	30 000	20 000	10 000	5 000	2 000
Pénalité $0 < \text{niveau stock} < \text{minimum}$	5 000	3 000	2 000	1 000	500
Pénalité $\text{minimum} < \text{niveau stock} < \text{objectif}$	1 000	500	200	100	10
Pénalité dégageant	30 000	20 000	10 000	5 000	5 000
<b>Cycle de vie du produit</b>	normal		produit en fin de vie		
Pénalité $\text{objectif} < \text{niveau stock} < \text{maximum}$	1		10		
Pénalité $\text{maximum} < \text{niveau stock}$	1 000		10 000		

TABLE 4.2 – Classification des produits

objectif et stock minimum) sont inspirées de paramétrage réel, utilisé par des clients de FuturMaster.

Des améliorations sont apportées au générateur telles que :

- un paramètre qui permet de « doser » la contrainte de capacité de transport (capacité très limitée à capacité abondante) : la capacité de transport est générée aléatoirement, en prenant une borne inférieure égale au strict nécessaire pour répondre au besoin et une borne supérieure (on rajoute X%, au choix, de la borne inférieure) ;
- une contrainte de ressource, qui est rencontrée chez certains clients de FuturMaster, dont les entrées de stock au niveau des entrepôts sont figées sur un certain nombre de périodes en début d’horizon : la ressource disponible est générée aléatoirement en prenant comme borne supérieure X% (au choix) du stock objectif ;
- une interface graphique, pour faciliter l’utilisation du générateur aléatoire d’instances (voir Figure 4.2).

Des instances sont générées aléatoirement, en faisant varier plusieurs paramètres (voir Tableau 4.3), à savoir :

- le nombre de produits ( $|P|$ ),
- le nombre de dépôts ( $|L|$ ),
- le nombre d’entrepôts au choix ( $|K|$ ) : plus le choix est important, moins il y a de partage de la ressource,
- le nombre de source d’approvisionnement pour chaque couple produit-dépôt ( $|S|$ ),
- le nombre de périodes de l’horizon de calcul,
- le pourcentage maximum de capacité de transport, à chaque période de l’horizon,
- le nombre de périodes en début d’horizon, sur lesquelles les entrées en stock (entrepôt) sont figées,
- le pourcentage maximum de ressource, pour chaque couple produit-entrepôt, à chaque période de l’horizon.

En faisant varier les différents paramètres, nous avons généré 40 instances. Elles correspondent aux instances allant du numéro 21 à 60 (voir Tableau 4.3).

Parameter	Value
Name for file to create	GeneratedInstance.xls
Number products	10
Number distribution centers	20
Number warehouses	10
Number sourcing	2
Number time periods	30
Maximum Stock: number periods to cover	4
Target Stock: number periods to cover	2
Minimum Stock: number periods to cover	1
Max % Capacity added by period	30
Number periods warehouses supply constraint	5
Max % of target stock for warehouses supply	10

FIGURE 4.2 – Interface du générateur aléatoire d'instances

#### 4.4.2 Résultats de CPLEX et de l'heuristique déterministe pour les instances générées aléatoirement

Les résultats de CPLEX et de l'heuristique déterministe (sans randomisation) sont donnés, pour les 40 instances générées aléatoirement, dans le Tableau 4.4, avec le temps de calcul en secondes et la consommation-mémoire (MO). Les résultats des 20 premières instances sont fournis dans le chapitre précédent. Le temps de calcul est limité à 5400 secondes, soit 1 heure et 30 minutes, pour chaque instance, lors de la résolution du PL mixte (CPLEX). Plusieurs instances générées aléatoirement sont de grande taille. Le temps de calcul limite est défini, en nous basant sur l'expérience avec les utilisateurs de l'APS (*Advanced Planning System*) de FuturMaster. Le second critère d'arrêt est le pourcentage d'écart, entre la borne inférieure et la meilleure solution réalisable, qui est paramétré à 0,01%. L'écart, entre la meilleure solution réalisable (UB) et la meilleure borne inférieure (LB), est indiqué pour chaque instance.

Les deux méthodes sont comparées sur la base de la valeur de la fonction-objectif. Nous avons calculé l'écart, entre la valeur de la fonction-objectif de la solution de l'heuristique déterministe (H) par rapport à celle de la solution optimale, lorsque CPLEX atteint l'optimum, dans le cas contraire par rapport à celle de la meilleure borne inférieure (LB). Les formules de calcul des pourcentages

Test	$ P $	$ L $	$ K $	$ S $	Nombre périodes	Borne max (%) capacité transport	Nombre périodes ressource contrainte	Borne max (%) ressource
21	10	10	10	2	10	50	0	0
22	5	10	5	1	10	50	0	0
23	5	10	5	2	10	50	0	0
24	5	10	5	2	10	0	0	0
25	5	10	5	2	10	20	0	0
26	5	10	5	2	10	70	0	0
27	10	10	10	2	10	50	0	0
28	10	10	10	2	10	30	0	0
29	10	10	10	2	10	10	0	0
30	10	10	10	2	10	0	0	0
31	10	10	10	2	10	50	5	10
32	10	10	10	2	10	50	5	30
33	10	10	10	2	10	50	5	50
34	10	10	10	2	10	0	5	10
35	10	10	10	2	10	0	5	30
36	10	10	10	2	10	0	5	50
37	20	10	10	2	10	50	0	0
38	20	10	10	2	10	30	0	0
39	20	10	10	2	10	10	0	0
40	20	10	10	2	10	0	0	0
41	20	10	10	1	10	0	0	0
42	10	10	10	2	20	0	0	0
43	20	10	10	2	20	50	0	0
44	20	10	10	2	20	30	0	0
45	20	10	10	2	20	10	0	0
46	20	10	10	2	20	0	0	0
47	20	10	10	2	20	50	10	50
48	20	10	10	2	20	50	10	30
49	20	10	10	2	20	50	10	10
50	20	10	10	3	20	0	0	0
51	20	10	10	1	20	0	0	0
52	50	10	10	2	10	30	0	0
53	50	10	10	2	10	50	5	30
54	50	10	10	2	30	50	0	0
55	50	10	10	2	30	50	5	10
56	50	20	10	2	10	30	0	0
57	50	20	10	2	10	50	5	10
58	50	20	10	2	10	50	5	30
59	50	20	10	3	30	10	0	0
60	50	20	10	2	30	50	10	10

TABLE 4.3 – Paramètres de génération aléatoire d'instances

d'écart sont les suivantes :

$$\% \text{Écart UB/LB} = \frac{\text{UB} - \text{LB}}{\text{LB}} \times 100$$

$$\% \text{Écart H/LB} = \frac{H - LB}{LB} \times 100$$

Sur les 40 instances testées, le temps de calcul de l'heuristique déterministe est toujours inférieur (0,002 à 0,986 secondes) à celui de CPLEX (0,75 à 5 432,08 secondes) (voir Tableau 4.4). Le temps de calcul devient très important, dès qu'on augmente la taille des instances, alors que celui de l'heuristique déterministe reste toujours inférieur à 1 seconde, quelle que soit l'instance. L'heuristique déterministe consomme également moins de mémoire pour le calcul. la consommation-mémoire de l'heuristique déterministe varie entre 0,10 MO et 11,80 MO, alors qu'avec CPLEX elle varie entre 0,01 MO et 26 GO.

L'écart, entre la meilleure solution réalisable (UB) et la meilleure borne inférieure (LB), varie entre 0,002 et 16 173,312%. Nous n'avons pas comparé les résultats de l'heuristique à ceux de CPLEX pour les instances dont l'écart, entre la borne inférieure et la meilleure solution réalisable fournies par CPLEX, est trop important. Les valeurs moyennes fournies dans le Tableau 4.4 sont calculées pour toutes les instances, sauf celles-ci : 43, 44, 45, 46, 50, 52, 54, 55, 56, 59 et 60.

En moyenne, le temps de calcul de CPLEX est de 1 418,28 secondes et celui de l'heuristique déterministe est de 0,036 secondes. L'écart moyen, entre la solution fournie par l'heuristique par rapport à celle de CPLEX est de 31,829%. Les écarts importants sont analysés dans la sous-section 4.4.4.

#### 4.4.3 Résultats de l'heuristique réactive randomisée pour toutes les instances

L'heuristique réactive randomisée est exécutée avec 1000 itérations et une fréquence ( $\beta$ ) de mise à jour des probabilités des réglages de 100 ; c'est à dire après 100 itérations successives les probabilités sont recalculées. Les résultats de HRR sont fournis dans le Tableau 4.5, pour les 20 premières instances (chapitre précédent) et dans le Tableau 4.6, pour les 40 instances générées aléatoirement. Concernant le Tableau 4.5, rappelons que les 20 instances proviennent pour moitié de deux bases de données réelles différentes. Pour chaque instance, nous avons :

- le coût de la meilleure solution (Min),
- le coût de la solution avec le coût le plus élevé (Max),
- le coût moyen des solutions (Moyen),
- le pourcentage d'écart, entre le coût de la solution de HRR par rapport à la celle fournie par CPLEX (% Écart HRR/LB),
- le pourcentage d'écart, entre le coût de la solution de HRR par rapport à celle de l'heuristique déterministe (% Écart HRR/H),
- le temps de calcul en secondes (Durée).

Les formules de calcul des pourcentages d'écart sont les suivantes :

$$\% \text{ Écart HRR/LB} = \frac{\text{HRR} - \text{LB}}{\text{LB}} \times 100$$

$$\% \text{ Écart HRR}/H = \frac{\text{HRR} - H}{\text{HRR}} \times 100$$

Pour les 20 instances issues des bases de données, le temps de calcul de HRR varie entre 2 et 35 secondes pour 1000 itérations (voir Tableau 4.5). CPLEX consomme, en moyenne, 3 728,32 secondes pour ces instances alors que HRR dure, en moyenne, 11,05 secondes.

La variation de consommation-mémoire de HRR par rapport à celle de l'heuristique déterministe est négligeable.

L'écart moyen, entre le coût des solutions fournies par HRR par rapport à celles de CPLEX, est de 8,339%. Cet écart est de 7,990% avec l'heuristique déterministe. HRR améliore les résultats de certaines instances mais dégrade d'autres par rapport à H, pour les instances issues des bases de données (voir Figure 4.3).

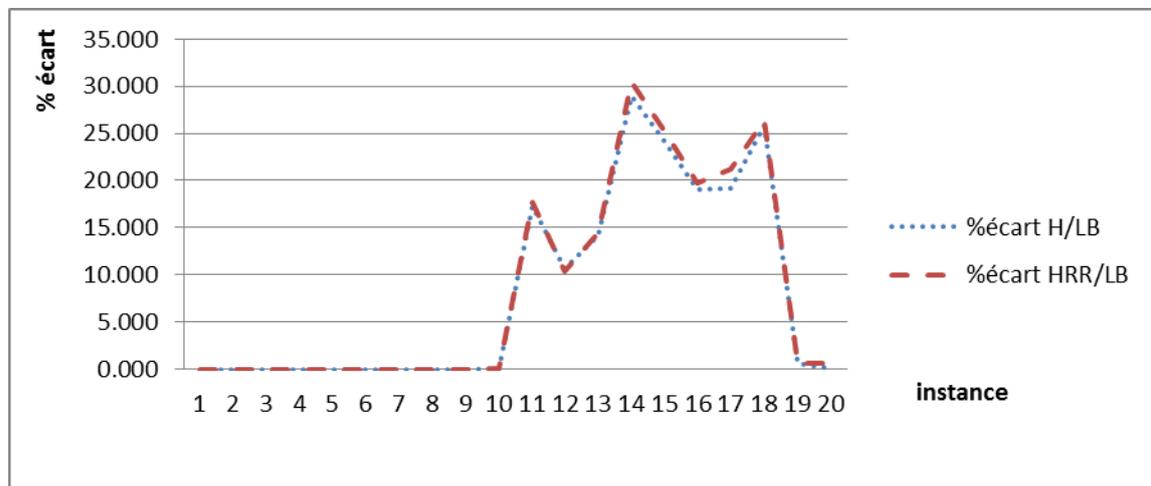


FIGURE 4.3 – Écarts de H et HRR par rapport à LB pour les 20 instances des bases de données

Pour les 40 instances générées aléatoirement, le temps de calcul de HRR varie entre 5 et 608 secondes, pour 1000 itérations (voir Tableau 4.6). La variation de consommation-mémoire de HRR par rapport à celle de l'heuristique déterministe est négligeable.

L'écart moyen, entre le coût des solutions fournies par HRR par rapport à celui des solutions de CPLEX (écart HRR/LB), est de 30,249%. Cet écart est de 31,829% lorsqu'on compare les solutions de l'heuristique déterministe et celles de CPLEX (écart H/LB). L'écart à l'optimum diminue de 1,58% avec HRR. Pour l'instance 33, par exemple, cet écart a diminué de 4,166%.

Le coût des solutions (des 40 instances générées aléatoirement) de HRR est inférieur à celui des solutions de l'heuristique déterministe, en moyenne, de 1,217% (voir Figure 4.4).

L'écart à l'optimum des solutions de HRR, reste assez important. Pour l'ensemble des instances testées, l'écart moyen à l'optimum est de 21,306% (voir Tableau 4.7). Le temps moyen de calcul étant faible, 19,35 secondes, la solution de HRR peut être utilisée comme solution initiale pour des méthodes basées sur la recherche locale.

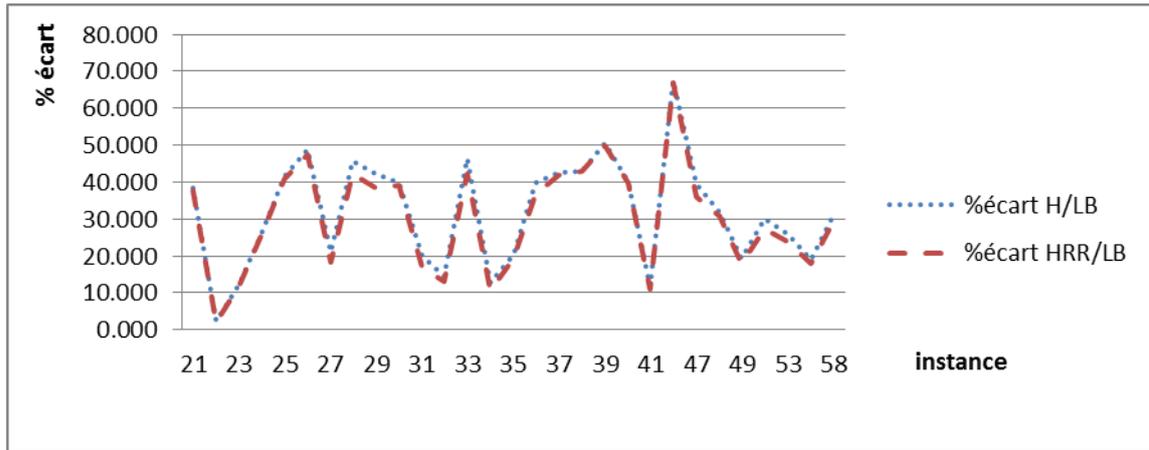


FIGURE 4.4 – Écarts de H et HRR par rapport à LB pour les 40 instances du générateur

Le réglage, qui donne la meilleure solution de HRR, varie selon l'instance. Le Tableau 4.8 fournit la part de meilleures solutions fournies, par réglage, pour les 40 instances générées aléatoirement. Pour les 20 premières instances, les solutions étant assez identiques d'un réglage à un autre, nous avons préféré établir les statistiques sur les 40 instances générées aléatoirement. Les réglages 17 et 25 sont celles qui donnent plus de meilleures solutions, soit 10% pour chacun des réglages. Les réglages 1, 7, 9, 13, 14 et 21 permettent chacun d'obtenir 7,5% des meilleures solutions (voir Figure 4.5). En somme, aucun réglage n'est prédominant.

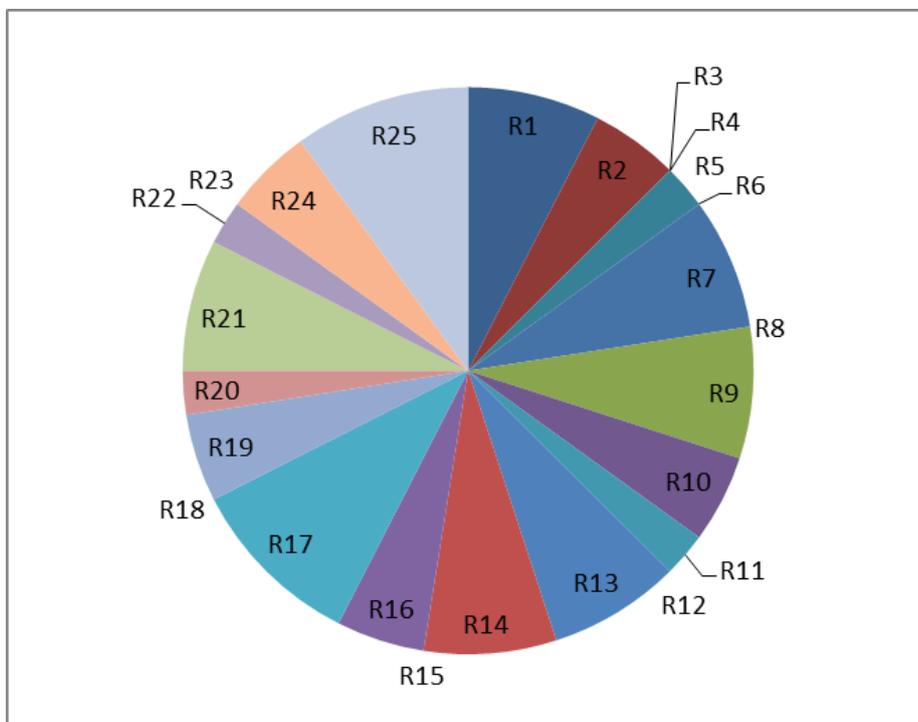


FIGURE 4.5 – Part de meilleures solutions fournies par réglage

#### 4.4.4 Analyse du grand écart moyen à l'optimum

Les pires cas, qui entraînent des écarts importants, entre les solutions des heuristiques et CPLEX, sont dus à la structure de ces heuristiques.

##### 4.4.4.1 Différenciation des périodes de référence

La procédure d'approvisionnement minimum s'effectue en se basant sur la disponibilité des produits dans les entrepôts. La période de référence correspond à la période d'expédition, à partir de l'entrepôt. Les délais de transport, pour acheminer les produits vers les dépôts, varient selon la route (arc). La rupture est donc évitée pour des périodes futures. La procédure d'atteinte du stock objectif, quant à elle, s'effectue en se basant sur la période de réception au dépôt. Dans certains cas (ressources non limitées et capacité de transport assez importante), la combinaison de l'anticipation des ruptures des dépôts et l'approvisionnement à hauteur du stock objectif conduit à un surstock, souvent constaté vers la fin de l'horizon.

##### 4.4.4.2 Option du juste à temps par défaut

L'idéal de la politique de stock des dépôts est d'atteindre le stock objectif à toutes les périodes de l'horizon. L'approvisionnement en juste à temps permet d'atteindre cet idéal lorsqu'il n'y a pas de contraintes d'approvisionnement telles que la quantité minimale et la taille de lot de transport. Dès que les contraintes d'approvisionnement sont prises en compte, l'efficacité de cette option (juste à temps) devient moins évidente. Il est parfois plus intéressant de mutualiser les quantités approvisionnées, même si cela entraîne des situations où le niveau de stock n'est pas idéal (niveau de stock inférieur ou supérieur au stock objectif).

##### 4.4.4.3 Atteinte systématique du stock objectif des entrepôts

L'idéal de la politique de stock des entrepôts est également d'atteindre le stock objectif à toutes les périodes de l'horizon. L'obsolescence des produits n'est analysée que lorsqu'on cherche à acheminer une quantité de produits vers un dépôt. Il peut arriver que l'approvisionnement de l'entrepôt à hauteur du stock objectif, à la période courante, entraîne une obsolescence à des périodes ultérieures.

## 4.5 Conclusion

Une heuristique réactive randomisée est proposée et testée sur les instances, issues des bases de données de clients de FuturMaster et évaluées dans le chapitre précédent. De nouvelles instances sont générées aléatoirement pour plus de diversité dans les tests. Les solutions fournies par l'heuristique

réactive randomisée (HRR) sont comparées à celles fournies par CPLEX et à celle de l'heuristique déterministe (H).

Pour l'ensemble des instances testées, le coût des solutions de HRR est inférieur, en moyenne, de 0,603% par rapport à celui des solutions de l'heuristique déterministe. Cependant, l'écart à l'optimum reste assez important : en moyenne 21,306%. Ce fait est dû aux cas limites pour lesquels la randomisation ne permet pas de corriger les faiblesses de l'heuristique, à cause de la structure de celle-ci. Le temps de calcul moyen est de 19,35 secondes, pour 1000 itérations de HRR. Les heuristiques H et HRR sont par contre très rapides et on peut donc les recycler pour initialiser des métaheuristiques. Les solutions des heuristiques peuvent être utilisées comme solutions initiales pour des méthodes basées sur la recherche locale. La randomisation de l'heuristique a été présentée à la quinzième conférence de la ROADEF (Société Française de Recherche Opérationnelle et d'Aide à la Décision) à l'Université de Bordeaux, en février 2014.

Dans le chapitre suivant, nous implémentons des métaheuristiques à base de voisinages, afin d'améliorer la qualité des solutions de l'heuristique. Il s'agit de la descente à voisinage variable (Variable neighborhood descent - VND), de la recherche locale itérative (Iterated local search - ILS) et de la recherche locale itérative à démarrages multiples (Multi-start iterated local search - MS-ILS).

Test	PL mixte					Heuristique (H)		
	LB	UB	% Écart UB/LB	Durée (s)	Mém. (MO)	% Écart H/LB	Durée (s)	Mém. (MO)
21	95 162 339 266	95 168 395 536	0,006	40,08	7,50	38,383	0,008	0,30
22	235 978 455 290	236 001 286 420	0,010	0,75	0,01	2,155	0,002	0,10
23	73 596 345 780	73 599 991 288	0,005	7,57	0,01	12,425	0,002	0,10
24	107 105 032 380	107 113 316 870	0,008	6,85	0,01	25,855	0,002	0,10
25	54 301 543 533	54 304 934 759	0,006	9,64	0,01	41,639	0,002	0,20
26	48 011 727 863	48 015 822 604	0,009	8,96	0,01	48,902	0,002	0,10
27	914 355 276 900	914 422 428 040	0,007	35,43	19,38	21,071	0,010	0,30
28	106 853 900 280	106 863 836 790	0,009	47,64	11,37	45,636	0,008	0,30
29	103 579 206 970	103 589 375 310	0,010	28,67	7,30	42,081	0,008	0,40
30	151 260 137 510	151 271 767 270	0,008	25,35	12,60	39,617	0,008	0,30
31	212 554 267 500	212 569 929 890	0,007	31,90	9,89	20,005	0,008	0,30
32	411 797 008 710	411 837 738 340	0,010	52,77	20,74	14,413	0,010	0,30
33	452 414 599 050	452 459 469 430	0,010	48,13	10,82	46,283	0,008	0,30
34	800 911 030 250	800 990 906 760	0,010	36,33	16,06	12,686	0,010	0,30
35	721 395 751 100	721 465 384 390	0,010	37,38	18,28	20,511	0,008	0,30
36	477 319 344 110	477 359 762 400	0,008	1 529,39	2 524,23	39,660	0,010	0,40
37	244 561 483 550	244 585 941 950	0,010	467,66	307,22	42,392	0,018	0,70
38	241 023 146 130	241 046 176 840	0,010	284,44	183,97	43,020	0,018	0,70
39	151 896 975 270	151 911 765 950	0,010	151,73	48,48	50,529	0,018	0,70
40	238 848 580 370	238 871 568 020	0,010	128,33	34,84	40,283	0,018	0,70
41	1 034 468 343 400	1 034 491 524 400	0,002	19,64	0,01	11,758	0,018	0,50
42	93 230 909 152	93 244 492 995	0,015	5 403,31	8 778,54	67,391	0,024	0,80
43	303 093 633 400	6 870 262 008 100	2 166,713	5 407,54	13 526,52	—	0,050	1,50
44	213 643 669 900	3 460 116 189 400	1 519,573	5 406,35	14 789,32	—	0,052	1,60
45	224 559 890 430	3 562 413 786 700	1 486,398	5 407,26	13 021,71	—	0,052	1,60
46	302 807 651 870	5 345 280 923 700	1 665,240	5 406,87	13 557,56	—	0,052	1,60
47	3 272 316 329 400	3 275 339 400 500	0,092	5 406,39	7 874,25	39,558	0,062	1,30
48	4 156 235 855 800	4 159 593 380 700	0,081	5 410,13	26 096,59	31,920	0,064	1,30
49	6 187 161 557 000	6 189 420 206 700	0,037	5 406,00	4 935,70	19,320	0,068	1,20
50	196 934 941 520	5 837 618 236 400	2 864,237	5 407,88	14 650,21	—	0,056	1,60
51	1 074 226 270 500	1 074 333 549 600	0,010	290,24	426,87	30,208	0,050	1,00
52	538 350 712 820	1 860 854 726 200	245,658	5 407,17	13 106,51	—	0,068	1,70
53	3 591 641 786 900	4 112 654 042 100	14,506	5 406,00	8 828,78	25,900	0,080	1,60
54	759 421 952 610	68 461 664 889 000	8 914,970	5 418,80	7 659,51	—	0,328	5,90
55	3 830 211 511 500	60 274 836 792 000	1 473,669	5 409,98	4 312,27	—	0,314	4,30
56	1 107 128 581 300	7 476 908 355 100	575,342	5 407,54	11 083,02	—	0,194	3,00
57	8 937 189 593 400	8 941 110 323 900	0,044	5 403,28	4 496,91	18,794	0,258	2,20
58	6 946 600 588 000	6 953 459 919 000	0,099	5 406,15	5 381,51	30,644	0,252	2,40
59	805 955 170 770	131 155 598 880 000	16 173,312	5 432,08	8 549,03	—	0,982	11,80
60	30 194 126 741 000	165 925 792 460 000	449,530	5 419,57	16 389,11	—	0,986	5,90
Moyenne			0,519	1 418,28	2 415,58	31,829	0,036	0,66

TABLE 4.4 – Résultats de CPLEX et de l'heuristique déterministe pour les instances générées aléatoirement

Test	Heuristique Réactive Randomisée (HRR)					
	Min	Max	Moyen	% Écart HRR/LB	% Écart HRR/H	Durée
1	1 274 426 047	1 274 426 047	1 274 426 047	0,001	0,000	2,00
2	2 745 520 513	2 745 520 513	2 745 520 513	0,001	0,000	2,00
3	3 411 502 169	3 411 502 169	3 411 502 169	0,000	0,000	2,00
4	4 202 009 953	4 202 009 953	4 202 009 953	0,000	0,000	2,00
5	5 462 674 460	5 462 674 460	5 462 674 460	0,001	0,000	2,00
6	6 230 973 230	6 230 973 230	6 230 973 230	0,000	0,000	2,00
7	7 219 782 625	7 219 782 625	7 219 782 625	0,000	0,000	2,00
8	7 962 941 082	7 963 081 392	7 963 031 278	0,000	0,000	3,00
9	8 926 097 872	8 926 570 012	8 926 360 428	0,000	-0,001	3,00
10	8 943 412 389	8 944 000 809	8 943 697 354	0,092	-0,001	6,00
Moyenne des instances de la base 1				0,010	0,000	2,60
11	489 417 057	494 913 786	492 004 234	17,694	0,394	7,00
12	1 594 989 490	1 639 786 777	1 613 323 674	10,363	-0,248	32,00
13	1 201 233 653	1 226 386 807	1 210 497 702	14,471	0,307	18,00
14	101 771 139	104 013 207	103 073 126	30,381	1,147	6,00
15	1 048 911 422	1 106 777 654	1 065 941 207	25,206	0,928	14,00
16	591 310 579	597 525 066	593 877 325	19,730	0,553	12,00
17	1 453 222 131	1 512 918 937	1 476 057 523	21,233	1,677	17,00
18	1 219 406 574	1 231 827 351	1 225 926 118	26,326	0,384	21,00
19	423 115 119	423 802 493	423 617 130	0,652	0,066	33,00
20	500 312 792	502 065 486	501 335 817	0,630	0,528	35,00
Moyenne des instances de la base 2				16,669	0,574	19,50
Moyenne des instances des bases 1 et 2				8,339	0,287	11,05

TABLE 4.5 – Résultats de HRR pour les 20 instances des bases de données

Test	Heuristique Réactive Randomisée (HRR)					
	Min	Max	Moyen	% Écart HRR/LB	% Écart HRR/H	Durée
21	130 981 860 205	134 337 574 886	132 051 372 095	37,640	-0,539	10,00
22	240 930 943 692	241 122 254 836	241 037 603 263	2,099	-0,055	5,00
23	82 459 861 815	82 781 920 658	82 665 762 361	12,043	-0,341	5,00
24	134 400 161 496	135 246 726 682	134 630 585 262	25,484	-0,295	5,00
25	76 473 859 859	77 521 827 650	77 027 675 436	40,832	-0,573	5,00
26	70 703 475 353	71 499 384 438	71 039 963 133	47,263	-1,113	6,00
27	1 082 567 164 301	1 130 053 536 879	1 105 778 575 350	18,397	-2,259	12,00
28	152 136 396 478	156 226 456 380	154 211 503 602	42,378	-2,288	10,00
29	143 290 410 626	147 560 202 035	145 442 213 551	38,339	-2,705	10,00
30	210 460 112 711	217 147 304 734	214 444 816 243	39,138	-0,345	10,00
31	248 782 189 154	263 703 378 616	254 672 327 155	17,044	-2,529	10,00
32	466 162 725 567	474 433 400 711	470 985 367 615	13,202	-1,070	10,00
33	642 959 446 919	689 483 055 979	662 861 929 750	42,117	-2,931	10,00
34	886 612 490 121	912 418 119 404	899 740 849 669	10,700	-1,794	12,00
35	859 700 422 099	883 047 745 865	874 004 743 392	19,172	-1,123	10,00
36	652 382 717 437	694 770 040 499	672 908 855 525	36,676	-2,183	11,00
37	346 955 033 166	352 377 597 115	347 926 529 111	41,868	-0,369	18,00
38	344 212 906 793	344 861 631 318	344 619 064 724	42,813	-0,145	19,00
39	227 536 277 480	229 333 180 464	228 268 636 130	49,796	-0,489	19,00
40	333 657 267 426	335 455 782 695	334 570 039 893	39,694	-0,421	19,00
41	1 147 798 173 757	1 162 893 954 215	1 154 211 907 286	10,955	-0,724	16,00
42	155 555 533 725	159 022 839 768	156 674 305 204	66,850	-0,324	27,00
43	621 318 192 155	627 306 849 917	623 870 471 535	—	-0,372	49,00
44	400 984 806 724	412 468 747 209	405 837 993 913	—	-0,621	50,00
45	429 965 035 294	438 679 283 611	432 496 739 056	—	-0,867	51,00
46	623 454 293 347	653 468 078 627	633 265 540 815	—	-1,337	49,00
47	4 450 727 530 752	4 626 138 764 089	4 530 731 087 365	36,012	-2,607	53,00
48	5 431 343 247 765	5 587 329 128 968	5 511 739 190 202	30,679	-0,949	53,00
49	7 288 611 080 497	7 507 809 946 565	7 374 946 975 862	17,802	-1,289	59,00
50	385 560 609 601	395 340 276 923	390 942 293 791	—	-2,136	57,00
51	1 368 863 151 969	1 412 967 392 300	1 397 344 806 480	27,428	-2,182	42,00
52	832 893 483 871	836 903 641 671	834 880 468 407	—	-0,238	50,00
53	4 444 367 388 309	4 549 860 342 613	4 507 124 437 607	23,742	-1,744	53,00
54	1 548 215 731 141	1 563 880 495 906	1 556 254 788 121	—	-0,665	256,00
55	5 952 577 144 138	6 131 278 239 727	6 036 130 192 549	—	-1,486	265,00
56	1 681 440 743 125	1 693 823 045 081	1 688 025 791 434	—	-0,881	103,00
57	10 542 129 177 058	10 664 557 063 361	10 599 797 669 682	17,958	-0,709	104,00
58	8 968 031 418 912	9 116 953 904 738	9 044 996 858 587	29,100	-1,196	104,00
59	1 868 728 509 949	1 896 249 940 453	1 883 481 934 711	—	-0,546	608,00
60	39 292 419 546 402	40 003 530 689 058	39 641 830 374 028	—	-0,976	534,00
Moyenne				30,249	-1,217	25,07

TABLE 4.6 – Résultats de HRR pour les 40 instances générées aléatoirement

Instances issues de	% Écart HRR/LB	% Écart HRR/H	Durée (s)
Base de données 1	0,010	0,000	2,60
Base de données 2	16,669	0,574	19,50
Générateur aléatoire	30,249	-1,217	25,07
Moyenne	21,306	-0,603	19,35

TABLE 4.7 – Synthèse des résultats de HRR

Réglage	Nombre d'éléments des couples produit-entrepôt	% perturbation criticité couples produit-dépôt	Nombre de meilleures solutions (HRR)	Part de meilleures solutions (en %)
R1	1	5	3	7,5
R2	1	10	2	5,0
R3	1	15	0	0,0
R4	1	20	0	0,0
R5	1	25	1	2,5
R6	10	5	0	0,0
R7	10	10	3	7,5
R8	10	15	0	0,0
R9	10	20	3	7,5
R10	10	25	2	5,0
R11	20	5	1	2,5
R12	20	10	0	0,0
R13	20	15	3	7,5
R14	20	20	3	7,5
R15	20	25	0	0,0
R16	30	5	2	5,0
R17	30	10	4	10,0
R18	30	15	0	0,0
R19	30	20	2	5,0
R20	30	25	1	2,5
R21	50	5	3	7,5
R22	50	10	1	2,5
R23	50	15	0	0,0
R24	50	20	2	5,0
R25	50	25	4	10,0
Total			40	100,0

TABLE 4.8 – Part de meilleures solutions de HRR fournies par réglage

## Chapitre 5

# Métaheuristiques à base de voisinages

### 5.1 Introduction

Dans les chapitres précédents, un programme linéaire mixte (PL mixte), une heuristique constructive (déterministe) et une heuristique réactive randomisée ont été implémentées pour la résolution du problème de planification étudié. Le solveur de PL mixte (CPLEX) consomme beaucoup de temps, surtout pour les instances de grande taille. Pour certaines instances, CPLEX ne fournit pas une bonne borne inférieure dans la limite de temps définie ; l'écart entre la meilleure borne inférieure et la solution réalisable est très important.

Les heuristiques développées sont très rapides. L'heuristique déterministe consomme moins d'une seconde en moyenne et l'heuristique réactive randomisée exécutée avec 1000 itérations dure 19,35 secondes en moyenne. Cependant, l'écart à la meilleure borne inférieure (LB) est assez important : environ 20%, en moyenne, sur les instances testées. Pour améliorer la qualité des solutions des heuristiques, trois métaheuristiques à base de voisinages sont développées. Il s'agit d'une descente à voisinage variable (Variable neighborhood descent - VND), d'une recherche locale itérative (Iterated local search - ILS) et d'une recherche locale itérative à démarrages multiples (Multi-start iterated local search - MS-ILS).

Une descente à voisinage variable (VND) est explicitée et testée dans la section 5.2. Nous détaillons l'implémentation et les tests d'une recherche locale itérative (ILS) dans la section 5.3. Une recherche locale itérative à démarrages multiples (MS-ILS) est développée et testée dans la section 5.4. Les principales contributions de ce chapitre et les perspectives sont résumées dans la dernière section (5.5).

## 5.2 Descente à voisinage variable

Dans notre étude, l'heuristique constructive est utilisée pour fournir la solution initiale qui est ensuite améliorée avec la VND : voir l'Algorithme 5.1. Une liste de mouvements est appliquée dans un ordre défini. Le mouvement n'est accepté que si le coût de la solution diminue, puis on passe au voisinage suivant. L'ensemble des voisinages est exploré pour chaque mouvement, avant de passer au mouvement suivant. La VND s'arrête si :

- aucun mouvement de la liste n'a permis d'améliorer la solution, ou
- le temps de calcul est atteint, ou
- le pourcentage d'amélioration (PA), entre 2 boucles successives d'application de la liste de mouvements, est inférieur à 0,01%.

La valeur du pourcentage d'amélioration est aussi paramétrée pour l'écart, entre la borne inférieure et la borne supérieure (solution réalisable), qui est un critère d'arrêt pour CPLEX. Le pourcentage d'amélioration se calcule de la manière suivante :

$$PA = \frac{\text{coût dernière solution} - \text{coût nouvelle solution}}{\text{coût nouvelle solution}} \times 100 \quad (5.1)$$

---

### Algorithme 5.1 Algorithme VND

---

- 1:  $Cost(Sol) := Cost_{\text{heuristique constructive}}$
  - 2: initialiser  $Number_{Move}$
  - 3: **Répéter**
  - 4:  $Cost(LastSol) := Cost(Sol)$
  - 5:  $i := 1$
  - 6: **Tant que**  $i \leq Number_{Move}$  **Faire**
  - 7:  $Sol' := Recherche\ Locale\ Voisinages(Sol, i)$  **Commentaires** l'ensemble des voisinages est exploré
  - 8: **Si**  $Cost(Sol') < Cost(Sol)$  **Alors**
  - 9:  $Sol := Sol'$
  - 10: **Fin Si**
  - 11: **Fin Tant que**
  - 12: Calculer PA
  - 13: **Jusqu'à**  $Cost(Sol) \geq Cost(LastSol)$  ou  $PA \leq 0,01$  ou *Temps limite atteint*.
- 

### 5.2.1 Mouvements appliqués

Les différents mouvements utilisés dans la VND sont plus ou moins complexes. Des mouvements simples permettent d'augmenter ou de diminuer les quantités sur le même dépôt ou sur le même entrepôt. Les mouvements complexes peuvent transférer des quantités d'une période de l'horizon

de calcul à une autre, d'un dépôt à un autre ou d'un entrepôt à un autre. Les mouvements sont les suivants :

1. diminution de quantité pour les couples produit-dépôt,
2. transfert de quantité inter périodes pour les couples produit-dépôt,
3. ajout de quantité pour les couples produit-dépôt,
4. transfert de quantités entre 2 couples produit-dépôt partageant la même route (capacité de transport partagée) : ce sont les produits stockés dans le même dépôt et qui sont fournis par le même entrepôt,
5. suppression des lots obsolètes pour les couples produit-entrepôt,
6. diminution de quantité pour les couples produit-entrepôt,
7. ajout de quantité pour les couples produit-entrepôt,
8. transfert de quantités pour entre 2 couples produit-dépôt partageant la même source d'approvisionnement (entrepôt fournisseur),
9. transfert de quantité inter périodes pour les couples produit-entrepôt.

La diminution de la quantité livrée (mouvements 1 et 6) est conçue pour éviter le sur-stock : lorsque le niveau de stock est supérieur au stock objectif au niveau des couples produit-site (produit-entrepôt et produit-dépôt). Lorsqu'il est appliqué sur un produit-dépôt la quantité est retirée sur tout le flux, sur la route et la source d'approvisionnement (produit-entrepôt).

L'augmentation de la quantité fournie (mouvements 3 et 7) peut compenser l'écart lorsque le niveau de stock est en dessous du niveau de stock objectif des couples produit-site.

Des quantités peuvent être transférées entre les différentes périodes de l'horizon pour tous les couples produit-site (mouvements 2 et 9). La quantité enlevée de la période de départ n'est pas nécessairement égale à celle ajoutée à la période d'arrivée.

Un transfert peut être effectué entre deux couples produit-dépôt partageant la même route de transport (mouvement 4). L'intérêt de ce mouvement est de trouver un équilibre entre les niveaux de stock lorsque la capacité de transport est saturée. La consommation de la capacité de transport n'est pas optimisée par l'heuristique constructive. La quantité enlevée au niveau du produit-dépôt d'origine peut être différente de la quantité ajoutée au produit-dépôt destinataire.

Les couples produit-dépôt qui se fournissent à partir du même produit-entrepôt font également l'objet d'un transfert de quantités (mouvement 8).

Un mouvement particulier (mouvement 5) est intégré pour réduire le coût de la perte due à l'obsolescence des produits dans les entrepôts. Lorsque la suppression est possible, le lot en question est échangé avec un ou plusieurs lots afin de répondre aux besoins des couples produit-dépôt. Les contraintes de dates telles que la date limite d'utilisation et le contrat de fraîcheur sont toujours respectées.

Selon le type de mouvement, plusieurs quantités sont évaluées :

- la quantité nécessaire pour éviter les ruptures de stock,
- la quantité nécessaire pour atteindre le stock minimum,
- la quantité nécessaire pour atteindre le stock objectif,
- la quantité à retirer pour être le stock objectif,
- la quantité à retirer pour être au stock maximum,
- la quantité minimale de transport,
- la quantité totale fournie à une période donnée,
- la quantité contenue dans un lot de produits.

Une des questions relatives à l'exécution de la VND est l'ordre d'application des mouvements. Des statistiques sont réalisées sur l'efficacité des mouvements, lorsqu'on applique un mouvement à la fois. Il s'agit d'un ratio entre le gain fourni par l'application d'un mouvement et le nombre de fois que le mouvement est appliqué. L'efficacité associée à chaque mouvement nous permet d'établir un ordre d'application pertinent (voir Tableau 5.1). Les valeurs données dans le tableau sont des moyennes. Les indicateurs sont calculés avec :

- le coût de la solution de la recherche locale avec le mouvement  $m$  seulement ( $RL_m$ ),
- le coût de la solution de l'heuristique constructive ( $H$ ), utilisée comme solution initiale de la recherche locale et
- la borne inférieure fournie par CPLEX ( $LB$ ).

Les formules de calcul des pourcentages d'écart sont les suivantes :

$$\% \text{ Écart } RL_m/H = \frac{RL_m - H}{RL_m} \times 100 \quad (5.2)$$

$$\% \text{ Écart } RL_m/LB = \frac{RL_m - LB}{LB} \times 100 \quad (5.3)$$

$$\text{Gain}_m = \frac{H - RL_m}{RL_m} \times 100 \quad (5.4)$$

$$\text{Degré d'efficacité}_m = \frac{\text{Gain}_m}{\text{nombre de fois que le mouvement } m \text{ est appliqué}} \times 100 \quad (5.5)$$

### 5.2.2 Réduction de l'espace d'exploration

Afin de réduire les temps de calcul et d'aborder les problèmes de grande taille, différentes procédures sont mises en place. Il s'agit d'injecter de l'intelligence dans l'exploration de l'espace de voisinages. Deux principes sont utilisées : des coupes de l'espace de voisinage et une orientation de l'exploration du voisinage.

Mouvement	% Écart moyen	% Écart moyen	Degré d'efficacité moyen
	RL/H	RL/LB	
3	-2,712	38,132	0,0253
8	-3,843	36,985	0,0073
4	-7,505	31,924	0,0070
1	-6,620	33,845	0,0037
2	-14,299	24,984	0,0013
5	-1,977	40,127	0,0011
6	-1,677	40,305	0,0010
9	-0,309	42,044	0,0000
7	0,000	42,451	0,0000

TABLE 5.1 – Ordre d'application des mouvements par degré d'efficacité

Nous avons introduit une notion de « coupe de voisinage » qui permet de n'explorer que les voisinages susceptibles d'apporter une amélioration. Lors du premier parcours de la liste des mouvements, l'ensemble des voisinages est exploré. Cependant, lors des parcours suivants (de la liste des mouvements) seuls les voisinages ayant subi au moins une modification ou qui sont susceptibles d'être impactés par une modification effectuée lors du parcours précédent, sont explorés.

Certains espaces de voisinage sont sans amélioration, selon le type de mouvement. C'est le cas des problématiques d'avance/retard et de l'exploration de l'horizon. Lorsqu'il y a du stock d'avance (niveau de stock supérieur au stock objectif), pour obtenir un gain sur le coût de stock, il est inutile d'ajouter de la quantité. Le même raisonnement peut être fait en situation de retard. La Figure 5.1 illustre ce principe.

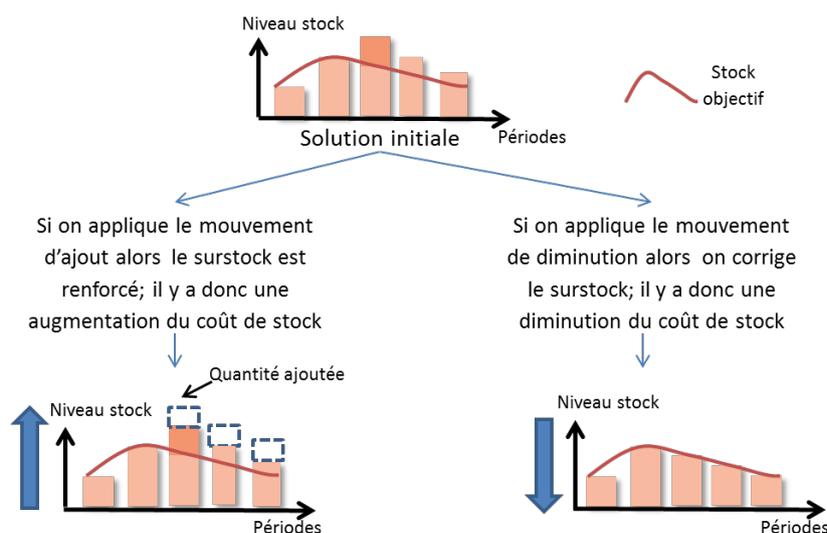


FIGURE 5.1 – Orientation de l'exploration du voisinage

L'autre espace sans amélioration concerne la manière dont l'heuristique explore les périodes de l'horizon de calcul. Après chaque amélioration, la période modifiée est sauvegardée pour servir de limite à l'intervalle de périodes à explorer lors du parcours (de la liste des mouvements) suivant. À la fin de chaque parcours, la plus petite période de modification est mémorisée. Un mouvement qui affecte les périodes  $t$  et  $t'$  par exemple ( $t < t'$ ) ne change pas la structure de la solution pour les périodes inférieures à  $t$ .

### 5.2.3 Tests

Les indicateurs sont l'écart (VND/LB) entre les coûts des solutions de la VND et la borne inférieure fournie par CPLEX, ainsi que l'écart (VND/H) entre les coûts des solutions de la VND et de l'heuristique (H) tels que :

$$\% \text{ Écart VND/LB} = \frac{\text{VND} - \text{LB}}{\text{LB}} \times 100 \quad (5.6)$$

$$\% \text{ Écart VND/H} = \frac{\text{VND} - \text{H}}{\text{VND}} \times 100 \quad (5.7)$$

#### 5.2.3.1 Évaluation des instances

Le détail des résultats de la VND est présenté dans le Tableau 5.2 pour les instances issues des bases de données 1 et 2, et le Tableau 5.3 pour les instances générées aléatoirement. Une synthèse des résultats est présentée dans le Tableau 5.4.

Le pourcentage d'écart (négatif), entre les coûts des solutions de l'heuristique et de la VND, est calculé pour évaluer l'impact de la VND sur la solution initiale. Les solutions des instances de la base de données 1 sont très légèrement améliorées, l'heuristique constructive permet d'obtenir de très bonnes solutions : l'écart H/LB est très faible. Pour les instances de la base de données 2, le coût de la solution de l'heuristique diminue en moyenne de 14,675%. L'écart entre les coûts de la solution de la VND et la borne inférieure de CPLEX (LB) est de 1,184%. Le temps de calcul moyen, lorsque la VND est appliquée aux instances de la base de données 2, est de 142,10 secondes alors que celui de CPLEX est de 7427,13 secondes.

Les mouvements ont considérablement corrigé les faiblesses de l'heuristique constructive. Le temps de calcul moyen de la VND sur les instances issues des bases de données (1 et 2) est de 72,46 secondes (Tableau 5.2). Il est beaucoup moins important que celui de CPLEX, qui est 3728,32 secondes en moyenne sur ces instances. Le pourcentage d'écart moyen entre les solutions de la VND et celles fournies par CPLEX, sur ces instances, est de 0,596% (voir Figure 5.2).

Le coût des solutions des instances générées aléatoirement diminue, en moyenne, de 23,396% par rapport aux solutions de l'heuristique constructive (voir Figure 5.3). L'écart moyen entre le coût des

VND avec avec réduction de l'espace d'exploration				
Test	Coût solution VND	% Écart VND/LB	% Écart VND/H	Durée (s)
1	1 274 426 047	0,001	0,000	0,01
2	2 745 520 513	0,001	0,000	0,01
3	3 411 502 169	0,000	0,000	0,01
4	4 202 009 953	0,000	0,000	0,02
5	5 462 674 460	0,001	0,000	0,02
6	6 230 973 230	0,000	0,000	0,02
7	7 219 782 625	0,000	0,000	0,02
8	7 962 941 082	0,000	0,000	0,02
9	8 926 202 452	0,002	0,000	0,02
10	8 942 877 033	0,086	-0,007	28,00
Moyenne		0,009	-0,001	2,82
11	418 489 606	0,637	-16,488	4,00
12	1 527 812 136	5,714	-4,655	12,00
13	1 061 664 733	1,171	-12,798	99,00
14	78 057 104	0,001	-28,885	3,00
15	843 771 965	0,719	-23,159	37,00
16	496 541 710	0,541	-18,427	23,00
17	1 209 297 076	0,884	-18,155	63,00
18	972 413 189	0,739	-24,919	95,00
19	423 514 488	0,747	0,160	495,00
20	500 571 884	0,682	0,580	590,00
Moyenne		1,184	-14,675	142,10
Moyenne des bases 1 et 2		0,596	-7,338	72,46

TABLE 5.2 – Résultats de la VND pour les instances issues des bases de données 1 et 2

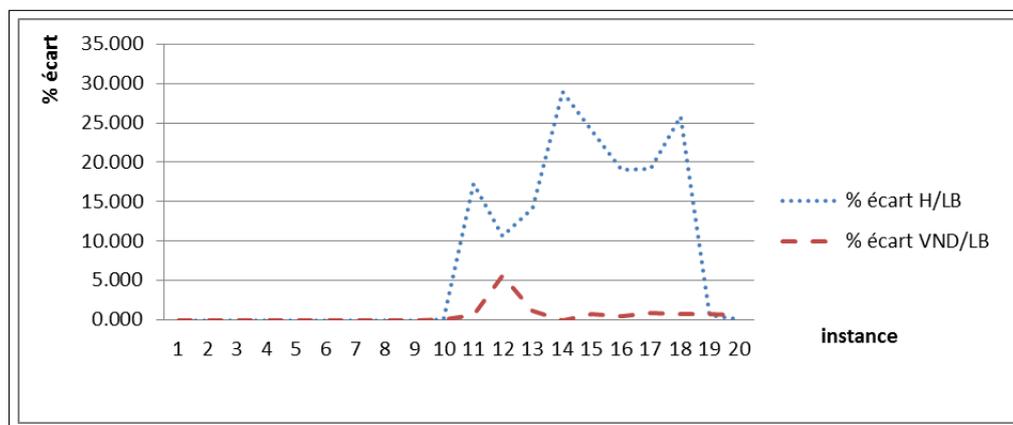


FIGURE 5.2 – Comparaison % Écart VND/LB et % Écart H/LB pour les instances issues des bases de données 1 et 2

solutions de la VND et celles fournies par CPLEX, est de 6,743% pour un temps de calcul moyen de 292,52 secondes (Tableau 5.3). Le calcul des valeurs moyennes est effectué avec les résultats des instances pour lesquelles CPLEX fournit une bonne borne inférieure dans la limite de temps que nous avons défini (une heure et demie). Les instances 43 à 46, 50, 52, 54, 55, 56, 59 et 60 sont celles pour lesquelles CPLEX ne fournit pas une bonne borne inférieure : l'écart entre la LB et la UB est très important. Nous ne pouvons pas comparer les solutions de la VND à celles de CPLEX.

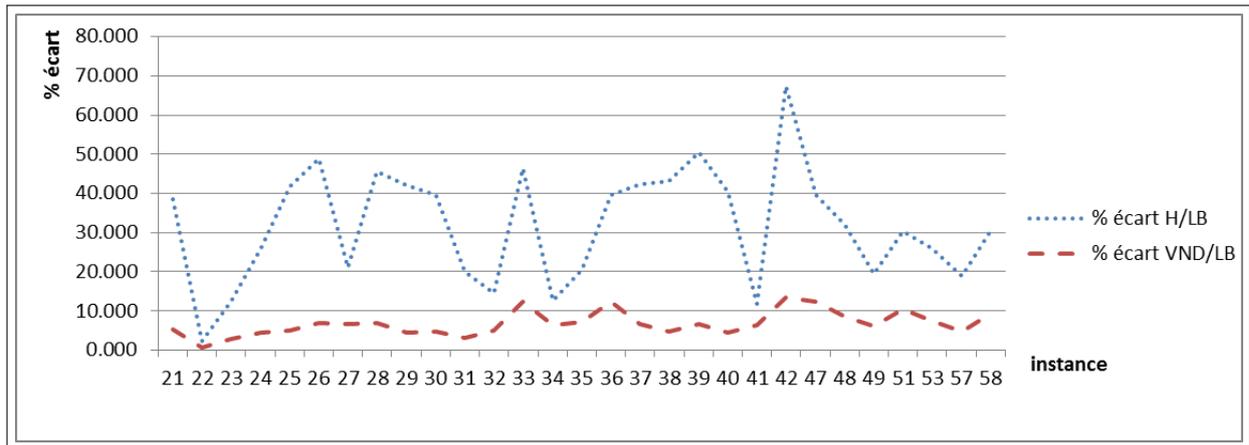


FIGURE 5.3 – Comparaison % Écart VND/LB et % Écart H/LB pour les instances générées aléatoirement

Pour toutes les instances évaluées, l'écart moyen entre la borne inférieure de CPLEX et la VND est de 4,234% (Tableau 5.4). Le coût des solutions initiales (heuristique constructive) diminue en moyenne de 16,842% lorsque la VND est appliquée. Le temps de calcul est beaucoup moins important que celui de CPLEX ; la VND (202,70 secondes en moyenne) est 11 fois plus rapide que CPLEX (2361,15 secondes en moyenne).

### 5.2.3.2 Impact de la réduction de l'espace d'exploration sur la méthode VND

Des tests supplémentaires sont effectués pour mettre en évidence l'apport de la réduction de l'espace d'exploration. La méthode VND exécutée sans cette phase est notée VND\*. Le gain en temps de calcul est assez important pour la méthode VND.

L'écart moyen entre les solutions de la VND (exécutée avec la phase de réduction de l'espace d'exploration : voir Tableau 5.4) et la LB est de 4,234%. Cet écart est de 4,142% (voir Tableau 5.5) lorsque l'espace d'exploration n'est pas réduit (VND\*). La différence entre ces écarts est de 0,092%. Le temps de calcul moyen de la VND est de 202,70 secondes et celui de la VND\* est de 306,71 secondes. La phase de réduction de l'espace d'exploration permet donc de réduire le temps de calcul, en moyenne, de 33,99%.

VND avec avec réduction de l'espace d'exploration						
Test	Coût solution			% Écart	% Écart	Durée
	VND	VND/LB	VND/H			(s)
21	100 106 522 078	5,196	-31,548			37,00
22	237 487 023 073	0,639	-1,506			1,00
23	75 598 502 792	2,720	-9,448			5,00
24	111 899 674 357	4,477	-20,462			4,00
25	57 057 444 417	5,075	-34,798			6,00
26	51 303 519 479	6,856	-39,348			6,00
27	975 185 588 898	6,653	-13,519			14,00
28	114 264 380 874	6,935	-36,191			25,00
29	108 298 735 816	4,556	-35,889			26,00
30	158 489 711 821	4,780	-33,249			25,00
31	219 215 997 651	3,134	-16,358			24,00
32	432 754 265 745	5,089	-8,872			13,00
33	508 798 944 994	12,463	-30,072			20,00
34	851 933 027 391	6,370	-5,938			9,00
35	772 912 691 488	7,141	-12,478			13,00
36	535 680 463 734	12,227	-24,445			19,00
37	260 693 926 941	6,596	-33,581			122,00
38	252 157 914 730	4,620	-36,705			108,00
39	162 081 924 541	6,705	-41,070			119,00
40	249 335 548 294	4,391	-34,383			119,00
41	1 101 422 987 040	6,472	-4,965			42,00
42	105 868 329 783	13,555	-47,409			249,00
43	353 223 813 003					1309,00
44	252 253 010 401					1193,00
45	261 065 643 314					1066,00
46	356 372 421 011					1164,00
47	3 673 210 978 822	12,251	-24,326			639,00
48	4 508 792 087 770	8,483	-21,605			569,00
49	6 560 274 600 748	6,030	-12,534			421,00
50	236 838 797 848					1626,00
51	1 187 528 564 447	10,547	-17,785			237,00
52	565 483 566 006					807,00
53	3 861 674 912 596	7,518	-17,096			748,00
54	1 064 040 135 607					5400,00
55	4 513 279 569 168					5400,00
56	1 155 323 148 187					3240,00
57	9 347 278 287 126	4,589	-13,582			1919,00
58	7 605 097 808 184	9,479	-19,332			2944,00
59	1 477 101 792 779					5400,00
60	33 672 132 915 545					5400,00
Moyenne		6,743	-23,396			292,52

TABLE 5.3 – Résultats de la VND pour les instances générées aléatoirement

Instances issues	% Écart VND/LB	% Écart VND/H	Durée (s)
Base de données 1	0,009	-0,001	2,82
Base de données 2	1,184	-14,675	142,10
Générateur aléatoire	6,743	-23,396	292,52
Moyenne	4,234	-16,842	202,70

TABLE 5.4 – Synthèse des résultats de la VND (avec réduction de l'espace d'exploration)

Instances issues	% Écart VND*/LB	% Écart VND*/H	Durée (s)
Base de données 1	0,009	-0,001	3,10
Base de données 2	1,183	-14,800	297,80
Générateur aléatoire	6,587	-23,396	414,48
Moyenne	4,142	-16,979	306,71

TABLE 5.5 – Synthèse des résultats de la VND\* (sans réduction de l'espace d'exploration)

### 5.2.3.3 Impact de l'ordre d'application des mouvements

Le degré d'efficacité des mouvements est calculé pour établir un ordre d'application des mouvements. Les tests précédents sont effectués avec cet ordre fixe d'application des mouvements pour toutes les instances. Nous avons ensuite exécuté la méthode VND avec un ordre choisi aléatoirement pour chaque instance : ces tests sont effectués deux fois. L'ordre est choisi aléatoirement au début de la VND. L'ordre des mouvements est appliqué tout au long de l'exécution. Lorsque la méthode VND est exécutée avec un ordre choisi aléatoirement pour l'application des mouvements, elle est notée VND<sub>oam</sub>. Une synthèse des résultats est présentée dans le Tableau 5.6 pour le premier test et dans le Tableau 5.7 pour le second test. Vous pouvez consulter les Tableaux 5.22 et 5.23 pour plus de détail sur les résultats des instances issues des bases de données 1 et 2. Les Tableaux 5.24 et 5.25 concernent les instances générées aléatoirement.

Sur les instances de la base de données 1, seul le temps de calcul moyen augmente très légèrement. Pour la méthode VND déterministe (avec un ordre fixe d'application des mouvements), il est de 2,82 secondes. Lors du premier et du second test de la VND<sub>oam</sub> (voir Tableaux 5.6 et 5.7), il est égale à 2,92 secondes. La qualité des solutions est la même entre la VND et les deux exécutions de la VND<sub>oam</sub>. Nous avons déjà remarqué que l'heuristique, utilisée comme solution initiale, fournit de très bonnes solutions pour ces instances. L'amélioration apportée par la VND est donc très faible.

Pour les instances de la base de données 2, les deux tests de la VND<sub>oam</sub> ont donné des résultats différents (voir Tableaux 5.6 et 5.7). L'ordre d'application des mouvements étant choisi aléatoirement, elle diffère d'une exécution à une autre. L'écart moyen, entre la VND<sub>oam</sub> et la meilleure borne inférieure (LB) fournie par CPLEX, pour les solutions du premier test est de 1,082% et 1,214% pour le second test (1,148% en moyenne pour la VND<sub>oam</sub>). Pour la VND déterministe, cet écart est de

1,184%. L'ordre fixe, que nous avons défini, est donc pertinent. Les temps de calcul restent, à peu près, équivalents : 143,60 secondes pour le test 1 de la VND<sub>oam</sub>, 145,40 secondes pour le test 2 de la VND<sub>oam</sub> et 142,10 secondes pour la VND déterministe.

Le même constat est fait concernant les instances générées aléatoirement (voir Tableaux 5.6 et 5.7). L'écart moyen, entre la VND<sub>oam</sub> et la meilleure borne inférieure (LB) fournie par CPLEX, pour les solutions du premier test est de 6,648% et 6,882% pour le second test (6,765% en moyenne pour la VND<sub>oam</sub>). L'écart est de 6,743% pour la VND déterministe ; l'ordre fixe établi est donc performant pour ces instances. Le temps de calcul moyen est de 289,52 secondes pour le test 1 de la VND<sub>oam</sub> et de 303,38 secondes pour le test 2 de la VND<sub>oam</sub> : en moyenne 296,45 secondes pour la VND<sub>oam</sub>. Lors de l'exécution de la VND déterministe, le temps de calcul moyen est égal à 292,52 secondes.

En résumé, l'ordre fixe d'application des mouvements est pertinent pour les instances évaluées. Les performances de la VND déterministe sont proches, en moyenne, de celles de la VND<sub>oam</sub>. Avec la randomisation, d'une exécution à une autre l'utilisateur peut avoir des plans de distribution différents. Même si tout le plan n'est pas modifié, l'acceptation de la modification de la quantité ou de la période d'approvisionnement n'est pas évidente pour un utilisateur. Il préférera la stabilité des résultats.

Instances issues	% Écart VND <sub>oam</sub> /LB	% Écart VND <sub>oam</sub> /H	Durée (s)
Base de données 1	0,009	-0,001	2,92
Base de données 2	1,082	-14,793	143,60
Générateur aléatoire	6,648	-23,499	289,52
Moyenne	4,157	-16,926	201,25

TABLE 5.6 – Synthèse des résultats VND<sub>oam</sub> avec un ordre aléatoire d'application des mouvements : test 1

Instances issues	% Écart VND <sub>oam</sub> /LB	% Écart VND <sub>oam</sub> /H	Durée (s)
Base de données 1	0,009	-0,001	2,92
Base de données 2	1,214	-14,630	145,40
Générateur aléatoire	6,882	-23,225	303,38
Moyenne	4,323	-16,731	209,82

TABLE 5.7 – Synthèse des résultats VND<sub>oam</sub> avec un ordre aléatoire d'application des mouvements : test 2

### 5.3 Recherche locale itérative (Iterated local search - ILS)

Pour Lourenço *et al.* (2003), la méthode ILS a beaucoup de caractéristiques souhaitables d'une méta-heuristique : elle est simple, facile à mettre en œuvre, robuste et très efficace. L'idée essentielle de l'ILS est de concentrer la recherche non pas sur l'espace complet de solutions mais sur un sous-espace plus petit défini par les solutions qui sont localement optimales pour une recherche locale donnée. À chaque itération, une copie de la solution courante est modifiée aléatoirement par un opérateur de perturbation puis une procédure de recherche locale est appliquée. La force de la méthode est que les solutions générées sont dépendantes et tracent une trajectoire d'optima locaux proches les uns des autres dans l'espace des solutions. La méthode ILS est aussi très rapide : si la perturbation est assez faible, la nouvelle solution ressemble beaucoup à la solution-parent et est ré-optimisée en quelques mouvements de recherche locale.

Nous proposons une ILS qui a l'originalité d'utiliser une VND pour la phase de recherche locale. Nous utilisons l'heuristique constructive, présentée dans les chapitres précédents, comme solution initiale et la VND permet d'améliorer cette solution. Nous obtenons un optimum local à la fin de la VND. À chaque itération, les mouvements de perturbation sont appliqués sur la copie de la solution. Une nouvelle solution est obtenue après la perturbation et la VND est appliquée sur celle-ci afin de trouver un autre optimum local (voir Algorithme 5.2). Après la perturbation la nouvelle solution est différente et, souvent, moins bonne que l'optimum local, mais la VND peut l'améliorer et donner une meilleure solution. ILS s'arrête si un nombre fixe d'itérations ou le temps de calcul sont atteints. Plus le nombre d'itérations est élevé, plus le temps de calcul est conséquent. La meilleure solution d'une ILS est toujours la solution actuelle, celle dont on prend des copies.

Les paramètres  $k_{min}$  et  $k_{max}$  (au choix de l'utilisateur) représentent les forces minimum et maximum de la perturbation, qu'on peut appliquer.  $k$  est la force de la perturbation courante. Le nombre maximum d'itérations est noté  $max_{iterator}$ ,  $j$  représente l'itération courante.

#### 5.3.1 Perturbation

La perturbation de la solution, issue de la VND, est effectuée par quatre mouvements. Le premier est le mouvement de suppression des lots obsolètes pour les couples produit-entrepôt. La suppression de lots obsolètes peut dégrader l'état du stock de l'entrepôt sur plusieurs périodes ; ce qui entraîne une augmentation du coût de stock même si le coût de l'obsolescence diminue. Lorsque ce mouvement est appliqué en perturbation la modification est acceptée même si le coût de la solution augmente. Le mouvement d'ajout de quantité pour les couples produit-entrepôt peut corriger cet effet lors de l'itération suivante.

Les deuxième et troisième mouvements de perturbation sont des mouvements de diminution de quantité pour les cas de surplus, respectivement pour les couples produit-dépôt et les couples produit-entrepôt. Le quatrième mouvement permet d'échanger des quantités entre les couples

---

**Algorithme 5.2** Algorithme ILS

---

```
1:  $H(BestSol)$ 
2:  $VND(BestSol)$ 
3: initialiser les paramètres de perturbation  $k_{min}, k_{max}$ 
4: initialiser le nombre d'itérations  $max_{iterator}$ 
5:  $k := k_{min}$ 
6:  $j := 1$ 
7: Répéter
8:    $Sol := BestSol$ 
9:    $Perturbation(Sol, k)$ 
10:  exécuter la  $VND(Sol)$ 
11:  Si  $Cost(Sol) < Cost(BestSol)$  Alors
12:     $BestSol := Sol$ 
13:     $k := k_{min}$ 
14:  Sinon
15:     $k := k + 1$ 
16:    Si  $k > k_{max}$  Alors
17:       $k := k_{min}$ 
18:    Fin Si
19:  Fin Si
20:   $j := j + 1$ 
21: Jusqu'à  $j > max_{iterator}$  ou Temps limite atteint.
```

---

produit-dépôt partageant la même source. Dans certains cas la ressource disponible à l'entrepôt n'est pas suffisante pour satisfaire toutes les demandes des dépôts. Quelques échanges de quantités sont effectués pour modifier la distribution effectuée par l'heuristique constructive et laisser la VND améliorer la solution.

Pour chaque mouvement de perturbation, la liste des couples produit-site est triée aléatoirement. La méthode ILS est implémentée avec un taux de perturbation constant. Chaque mouvement est appliqué sur 10% de la liste qui lui est associée afin que la VND répare facilement la perturbation.

### 5.3.2 Tests

Les résultats de l'ILS sont présentés en utilisant les indicateurs suivants :

- l'écart (ILS/LB) entre les coûts des solutions de l'ILS et la borne inférieure fournie par CPLEX,
- l'écart (ILS/H) entre les coûts des solutions de l'ILS et de l'heuristique (H),
- l'écart (ILS/VND) entre les coûts des solutions de l'ILS et de la descente à voisinage variable (VND).

$$\% \text{ Écart ILS/LB} = \frac{\text{ILS} - \text{LB}}{\text{LB}} \times 100 \quad (5.8)$$

$$\% \text{ Écart ILS/H} = \frac{\text{ILS} - \text{H}}{\text{ILS}} \times 100 \quad (5.9)$$

$$\% \text{ Écart ILS/VND} = \frac{\text{ILS} - \text{VND}}{\text{ILS}} \times 100 \quad (5.10)$$

#### 5.3.2.1 Évaluation des instances

Le détail des résultats de la méthode ILS est présenté dans le Tableau 5.8 pour les instances issues des bases de données 1 et 2, et le Tableau 5.9 pour les instances générées aléatoirement. Une synthèse des résultats est présentée dans le Tableau 5.10.

Avec l'ILS, les instances issues des bases de données 1 et 2 sont en moyenne à 0,567% de la LB (Tableau 5.8). Comparée à la VND, la méthode ILS améliore légèrement (0,029%) la qualité des solutions des instances issues des bases de données 1 et 2. Le temps de calcul moyen, de ces instances, est de 686,31 secondes pour l'ILS ; ce qui reste très inférieur au temps de CPLEX. Pour les instances de la base de données 1, ILS fournit la même qualité de solutions que la VND. L'heuristique constructive permet d'obtenir de très bonnes solutions. Les solutions des instances de la base de données 2 sont améliorées par ILS de 0,058%, par comparaison à la VND. Les coûts des solutions des instances sont assez importants, le gain apporté par l'ILS peut être significatif en coûts réels. Prenons l'instance 18 en exemple : 0,056% de diminution du coût de la solution par rapport à la solution de la VND représente 540 744 en coût réel.

ILS avec 10 itérations					
Test	coût solution ILS	% Écart ILS/LB	% Écart ILS/H	% Écart ILS/VND	Durée (s)
1	1 274 426 047	0,001	0,000	0,000	0,10
2	2 745 520 513	0,001	0,000	0,000	1,00
3	3 411 502 169	0,000	0,000	0,000	1,00
4	4 202 009 953	0,000	0,000	0,000	2,00
5	5 462 674 460	0,001	0,000	0,000	3,00
6	6 230 973 230	0,000	0,000	0,000	4,00
7	7 219 782 625	0,000	0,000	0,000	4,00
8	7 962 941 082	0,000	0,000	0,000	6,00
9	8 926 202 452	0,002	0,000	0,000	6,00
10	8 942 847 419	0,085	-0,007	0,000	450,00
Moyenne		0,009	-0,001	0,000	47,71
11	417 987 099	0,517	-16,628	-0,120	38,00
12	1 526 742 619	5,640	-4,729	-0,070	117,00
13	1 061 160 151	1,123	-12,852	-0,048	721,00
14	78 009 548	-0,060	-28,964	-0,061	29,00
15	843 768 998	0,719	-23,159	0,000	261,00
16	495 991 633	0,429	-18,559	-0,111	188,00
17	1 209 199 349	0,876	-18,165	-0,008	423,00
18	971 872 445	0,683	-24,989	-0,056	672,00
19	423 192 873	0,671	0,084	-0,076	5 400,00
20	500 413 838	0,650	0,548	-0,032	5 400,00
Moyenne		1,125	-14,741	-0,058	1 324,90
Moyenne des bases 1 et 2		0,567	-7,371	-0,029	686,31

TABLE 5.8 – Résultats ILS pour les instances issues des bases de données 1 et 2

Les instances générées aléatoirement (Tableau 5.9) présentent des contraintes plus difficiles telles que des capacités de transport assez limitées. Pour certaines instances, les quantités disponibles au niveau des entrepôts sont figées sur un certain nombre de périodes sans possibilité de réapprovisionnement. Il s'agit de cas rencontrés chez des clients de Futurmaster où le plan d'approvisionnement est figé en début d'horizon de calcul. La ressource disponible est générée de façon aléatoire en prenant comme limite supérieure un certain pourcentage du stock objectif.

Le coût des solutions de ces instances diminuent en moyenne de 24,734% (Tableau 5.9) par rapport aux solutions initiales (heuristique). Comparées aux solutions de la VND, le coût des solutions de l'ILS diminue en moyenne de 1,060% (voir Figure 5.4). L'écart par rapport à la LB est de 5,626% alors qu'il est 6,743% pour la VND. Le temps de calcul moyen de l'ILS est 964,97 secondes pour les instances du générateur aléatoire. Il est moins important que celui de CPLEX qui est de 1418,28 secondes.

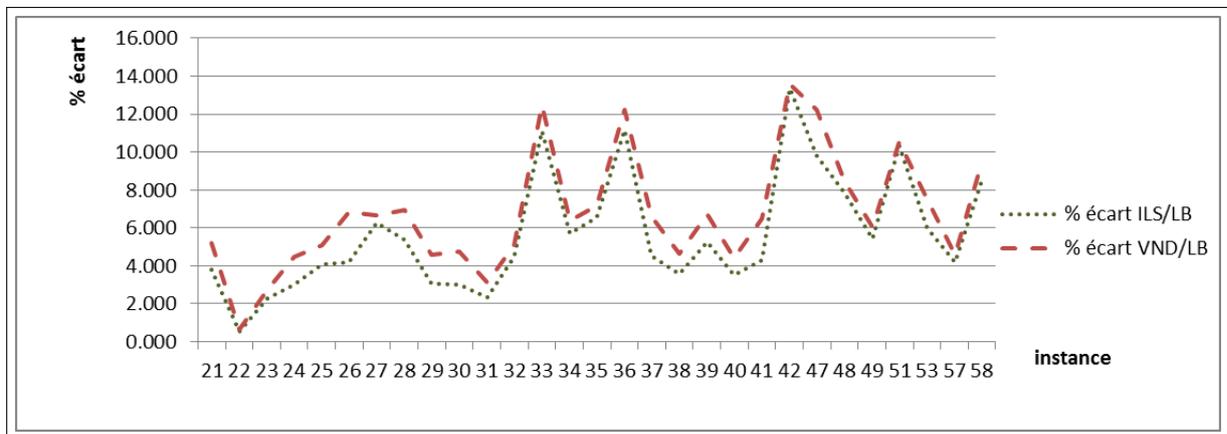


FIGURE 5.4 – Comparaison % Écart ILS/LB and % Écart VND/LB pour les instances générées aléatoirement

Pour toutes les instances testées, l'écart moyen entre la LB et les solutions de l'ILS est de 3,567% (Tableau 5.10). Elle permet de réduire le coût des solutions initiales, issues de l'heuristique, de 17,640%. Le coût des solutions diminue en moyenne de 0,633% par rapport à la VND. Le temps de calcul de l'ILS (851,23 secondes) est moins important que celui de CPLEX (2361,15 secondes). La méthode ILS utilise 36% du temps nécessaire à CPLEX.

ILS avec 10 itérations								
Test	coût solution				% Écart ILS/LB	% Écart ILS/H	% Écart ILS/VND	Durée (s)
	ILS	ILS/LB	ILS/H	ILS/VND				
21	98 771 571 163	3,793	-33,326	-1,352	205,00			
22	237 181 792 408	0,510	-1,637	-0,129	12,00			
23	75 271 332 727	2,276	-9,924	-0,435	36,00			
24	110 305 140 368	2,988	-22,204	-1,446	35,00			
25	56 520 818 875	4,087	-36,078	-0,949	42,00			
26	50 023 366 795	4,190	-42,914	-2,559	45,00			
27	971 434 077 599	6,243	-13,957	-0,386	53,00			
28	112 618 993 626	5,395	-38,181	-1,461	171,00			
29	106 734 766 574	3,047	-37,880	-1,465	180,00			
30	155 814 384 428	3,011	-35,537	-1,717	158,00			
31	217 504 081 035	2,329	-17,274	-0,787	169,00			
32	430 137 559 870	4,454	-9,534	-0,608	55,00			
33	502 641 316 445	11,102	-31,665	-1,225	79,00			
34	846 590 753 580	5,703	-6,606	-0,631	47,00			
35	768 564 791 333	6,539	-13,115	-0,566	58,00			
36	530 507 786 423	11,143	-25,658	-0,975	81,00			
37	255 615 001 276	4,520	-36,235	-1,987	835,00			
38	249 607 177 154	3,561	-38,102	-1,022	757,00			
39	159 925 368 885	5,285	-42,972	-1,348	783,00			
40	247 256 466 873	3,520	-35,513	-0,841	855,00			
41	1 078 756 012 042	4,281	-7,170	-2,101	216,00			
42	105 696 515 963	13,371	-47,649	-0,163	1323,00			
43	350 017 886 766				5400,00			
44	249 574 166 766				5400,00			
45	257 313 765 664				5400,00			
46	353 265 180 907				5400,00			
47	3 592 976 190 367	9,799	-27,103	-2,233	3097,00			
48	4 481 625 718 308	7,829	-22,342	-0,606	2213,00			
49	6 522 881 651 500	5,426	-13,179	-0,573	1696,00			
50	233 618 236 773				5400,00			
51	1 185 112 769 279	10,322	-18,025	-0,204	1309,00			
52	561 888 266 228				5400,00			
53	3 807 100 711 092	5,999	-18,775	-1,433	2674,00			
54	1 064 040 135 607				5400,00			
55	4 513 279 569 168				5400,00			
56	1 152 675 612 668				5400,00			
57	9 311 551 858 467	4,189	-14,018	-0,384	5400,00			
58	7 539 606 809 569	8,537	-20,368	-0,869	5400,00			
59	1 477 101 792 779				5400,00			
60	33 672 132 915 545				5400,00			
Moyenne		5,636	-24,722	-1,050	964,97			

TABLE 5.9 – Résultats ILS pour les instances générées aléatoirement

Instances issues	% Écart	% Écart	% Écart	Durée (s)
	ILS/LB	ILS/H	ILS/VND	
Base de données 1	0,009	-0,001	0,000	47,71
Base de données 2	1,125	-14,741	-0,058	1 324,90
Générateur aléatoire	5,636	-24,722	-1,050	964,97
Moyenne	3,567	-17,640	-0,633	851,23

TABLE 5.10 – Synthèse des résultats de l’ILS avec 10 itérations (avec réduction de l’espace d’exploration)

### 5.3.2.2 Impact de la réduction de l’espace d’exploration

La réduction de l’espace d’exploration est encore plus pertinente pour l’ILS en termes de gain sur le temps de calcul et la qualité des solutions. Les résultats présentés ci-dessous, le montrent. La méthode ILS exécutée sans la phase de réduction de l’espace d’exploration est notée ILS\*.

Instances issues	% Écart	% Écart	% Écart	Durée (s)
	ILS*/LB	ILS*/H	ILS*/VND*	
Base de données 1	0,009	-0,001	0,000	59,80
Base de données 2	1,137	-14,852	-0,046	1 602,90
Générateur aléatoire	5,686	-24,674	-0,855	1 485,62
Moyenne	3,599	-17,634	-0,516	1 218,57

TABLE 5.11 – Synthèse des résultats de l’ILS\* avec 10 itérations (sans réduction de l’espace d’exploration)

Lorsque la méthode ILS est exécutée avec la phase de réduction de l’espace d’exploration (Tableau 5.10), l’écart moyen, pour toutes les instances, entre les solutions de l’ILS et de CPLEX, est de 3,567%. Le temps de calcul moyen de l’ILS est de 851,23 secondes. Lorsque tout l’espace des voisinages est exploré (ILS\* : voir Tableau 5.11), l’écart moyen par rapport à la LB est de 3,599% et le temps de calcul moyen de 1218,57 secondes. La réduction de l’espace d’exploration ne dégrade pas la qualité des solutions, au contraire on gagne 0,032%. Cette phase réduit en moyenne le temps de calcul de 30,145% ; ce qui permet d’effectuer plus d’itérations de l’ILS pour les instances de grande taille.

Tous les tests qui suivent sont effectués avec la réduction de l’espace d’exploration.

### 5.3.2.3 Impact de l’ordre d’application des mouvements

Le degré d’efficacité des mouvements est calculé pour établir un ordre d’application des mouvements. Les tests précédents sont effectués avec cet ordre d’application pour chaque instance. Nous avons exécuté la méthode ILS avec un ordre aléatoire pour chaque instance. Ces tests sont effectués de deux manières. L’ordre aléatoire est défini :

- soit avant chaque appel de la méthode VND : l'ordre est différent pour chaque exécution de la VND,
- soit une seule fois au début de l'ILS : l'ordre est le même pour toutes les exécutions de la VND.

Lorsque la méthode ILS est exécutée avec un ordre choisi aléatoirement pour l'application des mouvements, elle est notée :

- $ILS_{VND_{oam}}$  (voir Tableau 5.12), lorsque l'ordre est différent pour chaque exécution de la VND
- $ILS_{oam}$  (voir Tableau 5.13), lorsque l'ordre est le même pour toutes les exécutions de la VND.

Pour plus de détail sur les résultats de chaque instance, vous pouvez consulter l'annexe. La méthode  $ILS_{VND_{oam}}$  est exécutée sur les instances issues des bases de données 1 et 2 dans le Tableau 5.26 et sur les instances générées aléatoirement dans le Tableau 5.27. Les résultats de la méthode  $ILS_{oam}$  sont présentés dans le Tableau 5.28 pour les instances issues des bases de données 1 et 2 et dans le Tableau 5.29 pour les instances générées aléatoirement.

Sur les instances de la base de données 1, le temps de calcul moyen augmente pour l' $ILS_{VND_{oam}}$  (voir Tableau 5.12) et l' $ILS_{oam}$  (voir Tableau 5.13). Pour la méthode ILS classique (avec un ordre fixe d'application des mouvements), il est de 47,71 secondes. Lorsque l'ordre est différent pour chaque exécution de la VND, le temps de calcul moyen de l' $ILS_{VND_{oam}}$  est égal à 50,70 secondes. Pour l' $ILS_{oam}$  (l'ordre est le même pour toutes les exécutions de la VND), il est de 53,90 secondes. Nous avons déjà remarqué que l'heuristique constructive, utilisée comme solution initiale, fournit de très bonnes solutions pour ces instances. La VND apporte une légère amélioration et la méthode ILS ne modifie en rien la qualité des solutions.

Pour les instances de la base de données 2, les deux méthodes  $ILS_{VND_{oam}}$  (voir Tableau 5.12) et  $ILS_{oam}$  (voir Tableau 5.13) ont donné des résultats différents. L'écart moyen, entre les solutions de ces méthodes et la meilleure borne inférieure (LB) fournie par CPLEX, est de 0,968% pour l' $ILS_{VND_{oam}}$  et de 1,142% pour l' $ILS_{oam}$ . L'exécution avec l'ordre aléatoire différent pour chaque exécution de la VND ( $ILS_{VND_{oam}}$ ) est meilleure en terme de qualité des solutions. Le fait d'expérimenter plusieurs types de parcours de la liste des mouvements évite d'être pénalisé par un ordre non performant pour l'instance courante. C'est ce qui s'est passé avec la  $ILS_{oam}$ , pour certaines instances l'ordre choisi aléatoirement n'est pas pertinent et il est appliqué pour toutes les exécutions de la VND. De plus, le temps de calcul moyen de  $ILS_{VND_{oam}}$  (1336,40 secondes) est inférieur à celui de la  $ILS_{oam}$  (1351,80 secondes). Comparée à la méthode ILS classique (avec un ordre fixe d'application des mouvements), qui donne un écart moyen par rapport à la LB de 1,125%, l' $ILS_{VND_{oam}}$  fournit en moyenne des solutions de meilleure qualité. Le temps de calcul moyen est de 1324,90 secondes pour l'ILS ; c'est 11,5 secondes de moins que l' $ILS_{VND_{oam}}$ .

Le même constat est fait concernant les instances générées aléatoirement. L'écart moyen, entre les solutions de ces méthodes et la meilleure borne inférieure (LB) fournie par CPLEX, est de 5,510% pour l' $ILS_{VND_{oam}}$  (voir Tableau 5.12) et de 5,725% pour l' $ILS_{oam}$  (voir Tableau 5.13). L'exécution avec l'ordre aléatoire différent pour chaque exécution de la VND ( $ILS_{VND_{oam}}$ ) est meilleure en

terme de qualité des solutions. Par contre, le temps de calcul moyen de l'ILS<sub>VND<sub>oam</sub></sub> (1005,24 secondes) est supérieur, de 20,52 secondes, à celui de l'ILS<sub>oam</sub> (984,72 secondes). Comparée à la méthode ILS classique (avec un ordre fixe d'application des mouvements), qui donne un écart moyen par rapport à la LB de 5,636%, l'ILS<sub>VND<sub>oam</sub></sub> fournit en moyenne des solutions de meilleure qualité. Le temps de calcul moyen est de 964,97 secondes pour l'ILS; c'est 40,27 secondes de moins que l'ILS<sub>VND<sub>oam</sub></sub>.

En résumé, la randomisation du choix de l'ordre d'application des mouvements est plus intéressante pour la méthode ILS, lorsque chaque appel de la VND est exécuté avec un ordre différent. Pour en moyenne 26,79 secondes de plus que l'ILS classique (avec un ordre fixe d'application des mouvements), ILS<sub>VND<sub>oam</sub></sub> permet d'obtenir des solutions à 3,461% de la LB. Pour l'ILS classique, cet écart est de 3,567%; d'où un gain de 3% sur la qualité des solutions avec l'ILS<sub>VND<sub>oam</sub></sub>. Avec la méthode ILS classique, la phase de perturbation est randomisée. L'ajout de la randomisation, sur le choix de l'ordre d'application des mouvements, ne pose pas de difficulté à l'utilisateur.

Instances issues	% Écart			Durée (s)
	ILS <sub>VND<sub>oam</sub></sub> /LB	ILS <sub>VND<sub>oam</sub></sub> /H	ILS <sub>VND<sub>oam</sub></sub> /VND	
Base de données 1	0,009	-0,001	0,000	50,70
Base de données 2	0,968	-14,911	-0,110	1336,40
Générateur aléatoire	5,510	-24,862	-1,076	1005,24
Moyenne	3,461	-17,757	-0,659	878,02

TABLE 5.12 – Synthèse des résultats de l'ILS<sub>VND<sub>oam</sub></sub> (10 itérations) avec un ordre aléatoire d'application des mouvements pour chaque exécution de la VND

Instances issues	% Écart			Durée (s)
	ILS <sub>oam</sub> /LB	ILS <sub>oam</sub> /H	ILS <sub>oam</sub> /VND	
Base de données 1	0,009	-0,001	0,000	53,90
Base de données 2	1,142	-14,710	-0,071	1351,80
Générateur aléatoire	5,725	-24,596	-1,091	984,72
Moyenne	3,623	-17,559	-0,660	869,67

TABLE 5.13 – Synthèse des résultats de l'ILS<sub>oam</sub> (10 itérations) avec un seul ordre aléatoire d'application des mouvements pour toutes les exécutions de la VND

### 5.3.2.4 Effet du nombre d'itérations

La méthode ILS est testée en augmentant le nombre d'itérations à 20 et 30. Lorsqu'elle est exécutée avec 10, 20 ou 30 itérations, nous la notons respectivement  $ILS_{10}$ ,  $ILS_{20}$ ,  $ILS_{30}$ . Les résultats sont présentés dans les Tableaux 5.14 pour les instances issues des bases de données 1 et 2 et 5.15 pour les instances générées aléatoirement. Les indicateurs sont les suivants :

- les écarts  $(ILS_{20}/LB)$  et  $(ILS_{30}/LB)$  entre les coûts des solutions de l'ILS exécutée avec 20 itérations (respectivement 30 itérations), et la borne inférieure fournie par CPLEX,
- l'écart  $(ILS_{20}/ILS_{10})$  et  $(ILS_{30}/ILS_{10})$  entre les coûts des solutions de l'ILS exécutée avec 20 itérations (respectivement 30 itérations) par rapport à l'ILS avec 10 itérations.

$$\% \text{ Écart } ILS_{20}/LB = \frac{ILS_{20} - LB}{LB} \times 100 \quad (5.11)$$

$$\% \text{ Écart } ILS_{20}/ILS_{10} = \frac{ILS_{20} - ILS_{10}}{ILS_{10}} \times 100 \quad (5.12)$$

Pour les instances issues des bases de données 1 et 2, lorsque le nombre d'itérations passe de 10 à 20 le coût des solutions ne diminue, en moyenne, que de 0,005% (Tableau 5.14). Le temps de calcul moyen passe de 686,31 à 817 secondes, soit une augmentation de 19%. Il en est de même pour les tests avec 30 itérations : avec une diminution moyenne du coût des solutions de 0,006% par rapport à l'ILS avec 10 itérations et une augmentation du temps moyen de calcul de 37,49%. Pour ces instances il n'est donc pas nécessaire d'effectuer plus de 10 itérations pour obtenir de bonnes solutions.

Pour les instances générées aléatoirement, lorsque le nombre d'itérations passe de 10 à 20 le coût des solutions ne diminue, en moyenne, que de 0,293% (Tableau 5.16). Le temps de calcul moyen passe de 964,97 à 1372,76 secondes, soit une augmentation de 42,26%. Il en est de même pour les tests avec 30 itérations : avec une diminution moyenne du coût des solutions de 0,437% par rapport à l'ILS avec 10 itérations et une augmentation du temps moyen de calcul de 76,71% (de 964,97 à 1705,21 secondes). L'augmentation du temps de calcul est assez importante par rapport au pourcentage d'amélioration de la qualité des solutions.

Les instances 43 à 46, 50, 52, 54, 55, 56, 59 et 60 (Tableau 5.15) sont celles pour lesquelles CPLEX ne fournit pas une bonne solution réalisable : l'écart entre la LB et la UB est très important. Cela montre la difficulté de résolution de ces instances. Nous ne pouvons pas comparer les solutions de l'ILS à celles de CPLEX. L'ILS exécutée avec 10 itérations consomme la totalité du temps de calcul défini (1 heure et demi). Il n'est donc pas possible de tester plus d'itérations pour ces instances, dans le temps fixé.

Tests	ILS avec 20 itérations			ILS avec 30 itérations		
	% Écart	% Écart	Durée	% Écart	% Écart	Durée
	ILS <sub>20</sub> /LB	ILS <sub>20</sub> /ILS <sub>10</sub>	(s)	ILS <sub>30</sub> /LB	ILS <sub>30</sub> /ILS <sub>10</sub>	(s)
1	0,001	0,000	1,00	0,001	0,000	2,00
2	0,001	0,000	2,00	0,001	0,000	3,00
3	0,000	0,000	3,00	0,000	0,000	5,00
4	0,000	0,000	5,00	0,000	0,000	7,00
5	0,001	0,000	6,00	0,001	0,000	9,00
6	0,000	0,000	8,00	0,000	0,000	11,00
7	0,000	0,000	9,00	0,000	0,000	14,00
8	0,000	0,000	12,00	0,000	0,000	18,00
9	0,002	0,000	12,00	0,002	0,000	19,00
10	0,085	0,000	969,00	0,085	0,000	1 410,00
Moyenne	0,009	0,000	102,70	0,009	0,000	149,80
11	0,517	0,000	72,00	0,517	0,000	106,00
12	5,640	0,000	223,00	5,640	0,000	329,00
13	1,069	-0,053	1 312,00	1,067	-0,055	1 902,00
14	-0,060	0,000	56,00	-0,060	0,000	82,00
15	0,718	0,000	481,00	0,717	-0,001	694,00
16	0,429	0,000	351,00	0,429	0,000	516,00
17	0,825	-0,050	787,00	0,824	-0,052	1 146,00
18	0,683	0,000	1 231,00	0,665	-0,018	1 799,00
19	0,671	0,000	5 400,00	0,671	0,000	5 400,00
20	0,650	0,000	5 400,00	0,650	0,000	5 400,00
Moyenne	1,114	-0,010	1 531,30	1,112	-0,013	1 737,40
Moyenne des bases 1 et 2	0,562	-0,005	817,00	0,561	-0,006	943,60

TABLE 5.14 – Résultats de l'ILS avec 20 et 30 itérations pour les instances issues des bases de données 1 et 2

Tests	ILS avec 20 itérations			ILS avec 30 itérations		
	% Écart	% Écart	Durée	% Écart	% Écart	Durée
	ILS <sub>20</sub> /LB	ILS <sub>20</sub> /ILS <sub>10</sub>	(s)	ILS <sub>30</sub> /LB	ILS <sub>30</sub> /ILS <sub>10</sub>	(s)
21	3,793	0,000	365,00	3,623	-0,164	514,00
22	0,507	-0,003	21,00	0,496	-0,014	31,00
23	1,771	-0,496	63,00	1,750	-0,517	90,00
24	2,840	-0,143	61,00	2,831	-0,152	89,00
25	4,043	-0,043	76,00	4,009	-0,075	109,00
26	3,576	-0,592	79,00	3,280	-0,881	113,00
27	6,175	-0,063	95,00	5,506	-0,698	135,00
28	4,958	-0,417	311,00	4,770	-0,597	460,00
29	2,907	-0,135	319,00	2,760	-0,279	457,00
30	2,791	-0,214	279,00	2,765	-0,239	395,00
31	2,047	-0,276	309,00	2,026	-0,297	435,00
32	4,212	-0,232	92,00	4,015	-0,422	130,00
33	10,364	-0,669	127,00	9,974	-1,026	176,00
34	5,546	-0,150	83,00	5,348	-0,337	119,00
35	5,975	-0,532	99,00	5,642	-0,849	137,00
36	10,955	-0,170	130,00	10,775	-0,333	179,00
37	4,011	-0,489	1472,00	3,894	-0,602	2090,00
38	3,289	-0,264	1382,00	3,182	-0,367	2025,00
39	4,819	-0,445	1377,00	4,585	-0,669	1943,00
40	3,150	-0,359	1514,00	3,037	-0,469	2147,00
41	3,651	-0,608	388,00	3,604	-0,653	559,00
42	12,729	-0,569	2285,00	12,617	-0,669	3292,00
43		0,000	5400,000		0,000	5400,00
44		0,000	5400,000		0,000	5400,00
45		0,000	5400,000		0,000	5400,00
46		0,000	5400,000		0,000	5400,00
47	9,639	-0,146	5005,00	9,490	-0,282	5400,00
48	7,507	-0,299	3551,00	7,052	-0,725	4990,00
49	5,147	-0,266	2800,00	5,012	-0,395	3853,00
50		0,000	5400,000		0,000	5400,00
51	10,322	0,000	2345,00	10,322	0,000	3383,00
52		0,000	5400,000		0,000	5400,00
53	5,029	-0,924	4382,00	4,977	-0,974	5400,00
54		0,000	5400,000		0,000	5400,00
55		0,000	5400,000		0,000	5400,00
56		0,000	5400,000		0,000	5400,00
57	4,189	0,000	5400,00	4,189	0,000	5400,00
58	8,537	0,000	5400,00	8,537	0,000	5400,00
59		0,000	5400,000		0,000	5400,00
60		0,000	5400,000		0,000	5400,00
Moyenne	5,327	-0,293	1 372,76	5,175	-0,437	1 705,21

TABLE 5.15 – Résultats de l'ILS avec 20 et 30 itérations pour les instances générées aléatoirement

Tests	ILS avec 20 itérations			ILS avec 30 itérations		
	% Écart	% Écart	Durée	% Écart	% Écart	Durée
	ILS <sub>20</sub> /LB	ILS <sub>20</sub> /ILS <sub>10</sub>	(s)	ILS <sub>30</sub> /LB	ILS <sub>30</sub> /ILS <sub>10</sub>	(s)
Base de données 1	0,009	0,000	102,70	0,009	0,00	149,80
Base de données 2	1,114	-0,010	1 531,30	1,112	-0,01	1 737,40
Générateur aléatoire	5,327	-0,293	1 372,76	5,175	-0,44	1 705,21
Moyenne	3,382	-0,143	1 145,92	3,291	-0,21	1 394,35

TABLE 5.16 – Synthèse des résultats de l’ILS avec 20 et 30 itérations

### 5.3.2.5 Impact du taux de perturbation

L’ensemble des tests précédents sont effectués avec un taux de perturbation fixé à 10%. Ce choix découle des expérimentations présentées ci-après. Les résultats de la méthode ILS exécutée avec 5% (Tableau 5.17) et 20% de taux de perturbation (Tableau 5.18) sont comparés avec ceux obtenus en fixant le taux de perturbation à 10%. Les tests sont effectués avec 10 itérations et en intégrant la phase de réduction de l’espace d’exploration. Les tests, sur les instances issues des bases de données 1 et 2, pour un taux de perturbation fixé à 5% sont présentés dans le Tableau 5.30 et pour un taux de perturbation fixé à 20% dans le Tableau 5.31. Le détail des résultats des instances générées aléatoirement est présenté, pour un taux de perturbation fixé à 5% dans le Tableau 5.32 et pour un taux de perturbation fixé à 20% dans le Tableau 5.33.

Lorsque la méthode ILS est exécutée avec un taux de perturbation fixé à 10% (Tableau 5.10), les solutions des instances issues de la base de données 1 sont à 0,009% de la LB pour un temps de calcul moyen de 47,71 secondes. Avec un taux de perturbation à 5% (Tableau 5.17) ou à 20% (Tableau 5.18), la qualité des solutions est la même. Mais le temps de calcul moyen diminue très faiblement pour un taux de perturbation à 5% (42,40 secondes) et augmente pour un taux de perturbation à 20% (59,40 secondes).

Pour les instances issues de la base de données 2, les solutions de l’ILS avec un taux de perturbation fixé à 10% sont à 1,125% de la LB pour un temps de calcul moyen de 1 324,90 secondes. Avec un taux de perturbation à 5% (Tableau 5.17), l’écart à la LB est de 1,371% avec un temps de calcul moyen de 1 262,60 secondes. Ce taux de perturbation est trop faible pour ces instances. En fixant le taux de perturbation à 20% (Tableau 5.18), l’écart à la LB est de 1,274% avec un temps de calcul moyen de 1 351,40 secondes. Le temps de calcul est presque le même que la perturbation à 10% mais la qualité des solutions est moins bonne.

Pour les instances générées aléatoirement, les solutions de l’ILS avec un taux de perturbation fixé à 10% sont à 5,636% de la LB pour un temps de calcul moyen de 964,97 secondes. Avec un taux de perturbation à 5% (Tableau 5.17), l’écart à la LB est de 5,8% avec un temps de calcul moyen de 889,03 secondes. C’est plus rapide que la perturbation à 10% mais ce taux de perturbation (5%) est moins performant pour ces instances. En fixant le taux de perturbation à 20% (Tableau 5.18), l’écart à la LB est de 5,317% avec un temps de calcul moyen de 1 063,41 secondes. Le temps de

calcul est plus important (98 secondes) par rapport à la perturbation de 10% mais la qualité des solutions est meilleure.

Pour toutes les instances testées, l'écart à la LB des solutions de l'ILS, avec un taux de perturbation fixé à 10%, est de 3,567% en moyenne (Tableau 5.10). Le temps de calcul de l'ILS avec 10% de perturbation (851,23 secondes en moyenne) est plus important que celui de l'ILS avec 5% de perturbation (792,49 secondes en moyenne). Cependant, la perturbation à 5% (Tableau 5.17) fournit de moins bonnes solutions (à 3,714% de la LB en moyenne). La perturbation à 20% (Tableau 5.18) prend plus de temps (917,29 secondes en moyenne) mais permet de réduire, un peu, l'écart à la LB (à 3,408% de la LB en moyenne). Selon le temps alloué au calcul, l'utilisateur peut utiliser un taux de perturbation de 10 ou 20%. Des tests statistiques peuvent être effectués afin de déterminer le meilleur paramétrage.

Instances issues	% Écart	% Écart	% Écart	Durée (s)
	ILS/LB	ILS/H	ILS/VND	
Base de données 1	0,009	-0,001	0,000	42,40
Base de données 2	1,371	-14,435	0,186	1 262,60
Générateur aléatoire	5,800	-24,513	-0,893	889,03
Moyenne	3,714	-17,454	-0,491	792,49

TABLE 5.17 – Synthèse des résultats ILS avec 10 itérations et 5% de taux de perturbation

Instances issues	% Écart	% Écart	% Écart	Durée (s)
	ILS/LB	ILS/H	ILS/VND	
Base de données 1	0,009	-0,001	0,000	59,40
Base de données 2	1,274	-14,536	0,092	1 351,40
Générateur aléatoire	5,317	-25,081	-1,354	1 063,41
Moyenne	3,408	-17,810	-0,783	917,29

TABLE 5.18 – Synthèse des résultats ILS avec 10 itérations et 20% de taux de perturbation

## 5.4 Recherche locale itérative à démarrages multiples (Multi-start iterated local search - MS-ILS)

La méthode MS-ILS est une méthode de recherche locale itérative à démarrages multiples. Elle utilise plusieurs exécutions de la méthode ILS, avec une heuristique réactive randomisée (RRH) utilisée comme solution initiale et la méthode de descente à voisinage variable (VND).

Nous nous sommes inspirés de l'article de Prais et Ribeiro (2000) et celui de Prins et Wolfer Calvo (2005). Prais et Ribeiro ont proposé une méthode réactive GRASP (Greedy Randomized Search Procedure) pour un problème en télécommunication. La structure général de l'algorithme du GRASP est présenté avec l'Algorithme 5.3. Prins et Wolfer Calvo ont également proposé un GRASP réactif pour un problème de tournées sur arcs.

---

### Algorithme 5.3 Structure général de l'algorithme du GRASP

---

- 1: initialiser le générateur aléatoire
  - 2:  $Cost(S^*) := +\infty$
  - 3: **Pour**  $i := 1 \rightarrow max_{iterator}$  **Faire**
  - 4:   Heuristique Randomisée( $S$ )
  - 5:   Recherche Locale( $S$ )
  - 6:   **Si**  $Cost(S) < Cost(S^*)$  **Alors**
  - 7:      $Cost(S^*) := Cost(S)$
  - 8:      $S^* := S$
  - 9:   **Fin Si**
  - 10: **Fin Pour**
- 

#### 5.4.1 Implémentation

La MS-ILS correspond à un GRASP, dans lequel on remplace la recherche locale par une ILS. La méthode commence par l'exécution de l'heuristique réactive randomisée ( $HRR$ ) pour laquelle nous avons 25 paramétrages possibles (chapitre précédent). Chaque paramétrage est testé 4 fois ; ce qui fait au total 100 exécutions de  $HRR$ . Cette phase permet de faire des statistiques afin de mettre en évidence les meilleurs paramétrages pour chaque instance. Pour plus de précision sur l'heuristique réactive randomisée, le lecteur peut se référer au chapitre précédent. À chaque lancement de la méthode MS-ILS,  $HRR$  est utilisée pour générer la solution initiale. Cette dernière est ensuite améliorée par la méthode ILS, explicitée dans la section 5.3. Les critères d'arrêt de la MS-ILS sont le nombre de lancements et le temps de calcul définis par l'utilisateur.

**Algorithme 5.4** Algorithme MS-ILS

---

```

1: exécuter 100 fois HRR et renvoyer la meilleure solution  $Global_{Best}$ 
2:  $j := 1$ 
3: Répéter
4:    $ILS(BestSol)$ 
5:   Si  $Cost(BestSol) < Cost(Global_{Best})$  Alors
6:      $Global_{Best} := BestSol$ 
7:   Fin Si
8:    $j := j + 1$ 
9: Jusqu'à  $j > Nombre_{iteration}$  ou Temps limite atteint.

```

---

**5.4.2 Tests**

La méthode MS-ILS est comparée à CPLEX, à l'heuristique, à la VND et à l'ILS. Dans le Tableau 5.19, nous présentons les résultats de la méthode MS-ILS pour les instances issues des bases de données 1 et 2. Les résultats des instances du générateur aléatoire sont présentées dans le Tableau 5.20. Les indicateurs sont les suivants :

- l'écart (MS-ILS/LB) entre les coûts des solutions de la MS-ILS et la borne inférieure fournie par CPLEX,
- l'écart (MS-ILS/H) entre les coûts des solutions de la MS-ILS et de l'heuristique (H),
- l'écart (MS-ILS/VND) entre les coûts des solutions de la MS-ILS et de la descente à voisinage variable (VND),
- l'écart (MS-ILS/ILS) entre les coûts des solutions de la MS-ILS et de la recherche locale itérative (ILS).

$$\% \text{ Écart MS-ILS/LB} = \frac{MS-ILS - LB}{LB} \times 100 \quad (5.13)$$

$$\% \text{ Écart MS-ILS/H} = \frac{MS-ILS - H}{MS-ILS} \times 100 \quad (5.14)$$

$$\% \text{ Écart MS-ILS/VND} = \frac{MS-ILS - VND}{MS-ILS} \times 100 \quad (5.15)$$

$$\% \text{ Écart MS-ILS/ILS} = \frac{MS-ILS - ILS}{MS-ILS} \times 100 \quad (5.16)$$

La méthode MS-ILS a un effet négligeable sur les instances issues de la base de données 1 car l'heuristique fournit de très bonnes solutions. Pour les instances issues de la base de données 2, la MS-ILS est légèrement moins efficace que la VND : on perd 0,086% en qualité des solutions (Tableau 5.19). Il en est de même lorsqu'on compare les méthodes MS-ILS et ILS : l'ILS est un peu plus performante de 0,115%. Le temps de calcul moyen pour ces instances (bases de données 1 et 2)

pour la VND est de 72,46 secondes, de 686,31 secondes pour l'ILS (avec 10 itérations) et de 1375,30 secondes pour la MS-ILS (avec 3 lancements d'ILS de 10 itérations). La méthode MS-ILS n'est pas intéressante pour ces instances.

MS-ILS avec 3 lancements d'ILS de 10 itérations						
Test	Coût solution MS-ILS	% Écart MS-ILS/LB	% Écart MS-ILS/H	% Écart MS-ILS/VND	% Écart MS-ILS/ILS	Durée (s)
1	1 274 426 047	0,001	0,000	0,000	0,000	3,00
2	2 745 520 513	0,001	0,000	0,000	0,000	5,00
3	3 411 502 169	0,000	0,000	0,000	0,000	7,00
4	4 202 009 953	0,000	0,000	0,000	0,000	10,00
5	5 462 674 460	0,001	0,000	0,000	0,000	13,00
6	6 230 973 230	0,000	0,000	0,000	0,000	17,00
7	7 219 782 625	0,000	0,000	0,000	0,000	21,00
8	7 962 941 082	0,000	0,000	0,000	0,000	25,00
9	8 926 095 172	0,000	-0,001	-0,001	-0,001	30,00
10	8 942 950 921	0,086	-0,006	0,001	0,001	1 469,00
Moyenne		0,009	-0,001	0,000	0,000	160,00
11	417 987 110	0,517	-16,628	-0,120	0,000	198,00
12	1 530 158 587	5,877	-4,495	0,153	0,223	523,00
13	1 060 279 620	1,039	-12,946	-0,131	-0,083	3 835,00
14	78 009 548	-0,060	-28,964	-0,061	0,000	182,00
15	846 739 507	1,073	-22,727	0,350	0,351	1 354,00
16	496 994 166	0,632	-18,320	0,091	0,202	1 328,00
17	1 225 223 781	2,213	-16,619	1,300	1,308	2 286,00
18	975 566 451	1,065	-24,515	0,323	0,379	5 400,00
19	423 115 485	0,652	0,066	-0,094	-0,018	5 400,00
20	500 072 961	0,582	0,481	-0,100	-0,068	5 400,00
Moyenne		1,359	-14,467	0,171	0,229	2 590,60
Moyenne des bases 1 et 2		0,684	-7,234	0,086	0,115	1 375,30

TABLE 5.19 – Résultats MS-ILS pour les instances issues des bases de données 1 et 2

L'application de la MS-ILS donne des solutions à 5,349% de la LB, pour les instances générées aléatoirement (Tableau 5.20). La méthode MS-ILS entraîne une diminution du coût des solutions de l'heuristique (déterministe) de 25,067%, mieux que la VND et l'ILS. La MS-ILS fournit de meilleurs solutions, comparée à la VND et à l'ILS, pour les instances générées aléatoirement. L'amélioration de la qualité des solutions est de 1,325% par rapport à la VND et de 0,272% par rapport à l'ILS. Dans le chapitre précédent, sur la randomisation de l'heuristique, nous avons déjà remarqué que l'effet de la randomisation était assez faible vue la structure de l'heuristique. Le temps de calcul moyen (2245,76 secondes) est, par contre, largement supérieur à ceux de la VND (292,52 secondes) et de l'ILS (964,97 secondes).

Les instances 43 à 46, 50, 52, 54, 55, 56, 59 et 60 (Tableau 5.20) sont celles pour lesquelles CPLEX ne fournit pas une bonne solution réalisable : l'écart entre la LB et la UB est très important. Nous ne pouvons pas comparer les solutions de la MS-ILS à celles de CPLEX.

Lorsqu'on observe la moyenne pour toutes les instances testées (voir Tableau 5.21), les solutions de l'heuristique constructive et celles de la méthode VND sont améliorées par la méthode MS-ILS exécutée avec 3 lancements dont chacun utilise l'ILS avec 10 itérations. L'écart moyen entre la borne inférieure fournie par CPLEX (LB) et la solution de la MS-ILS est de 3,445%. L'écart moyen entre les LB et les solutions de l'heuristique constructive est de 22%. Il a diminué de 17,788% avec la MS-ILS. Le temps de calcul moyen de la MS-ILS est de 1890,47 secondes. Cette méthode consomme plus de temps que l'ILS exécutée avec 30 itérations qui prend 1394,35 secondes (voir Tableau 5.16). En moyenne, l'ILS avec 30 itérations est meilleure que le MS-ILS (3 lancements et ILS de 10 itérations) de 0,154%.

MS-ILS avec 3 lancements d'ILS de 10 itérations						
Test	Coût solution	% Écart	% Écart	% Écart	% Écart	Durée
	MS-ILS	MS-ILS/LB	MS-ILS/H	MS-ILS/VND	MS-ILS/ILS	(s)
21	99 199 447 104	4,242	-32,751	-0,914	0,431	1 031,00
22	236 951 686 463	0,412	-1,735	-0,226	-0,097	53,00
23	74 907 496 080	1,782	-10,457	-0,922	-0,486	161,00
24	109 864 905 715	2,577	-22,693	-1,852	-0,401	157,00
25	56 163 005 710	3,428	-36,945	-1,593	-0,637	159,00
26	49 530 768 239	3,164	-44,336	-3,579	-0,995	187,00
27	967 247 266 214	5,785	-14,450	-0,821	-0,433	260,00
28	112 765 033 625	5,532	-38,002	-1,330	0,130	759,00
29	107 204 922 932	3,500	-37,276	-1,020	0,439	765,00
30	155 908 232 565	3,073	-35,455	-1,656	0,060	712,00
31	217 331 536 072	2,248	-17,367	-0,867	-0,079	779,00
32	430 683 267 463	4,586	-9,396	-0,481	0,127	241,00
33	491 985 097 437	8,747	-34,517	-3,418	-2,166	334,00
34	837 636 191 848	4,585	-7,746	-1,707	-1,069	207,00
35	758 096 871 206	5,088	-14,677	-1,954	-1,381	262,00
36	528 079 148 384	10,634	-26,236	-1,439	-0,460	363,00
37	255 113 216 312	4,315	-36,503	-2,188	-0,197	3 765,00
38	249 358 228 422	3,458	-38,239	-1,123	-0,100	3 564,00
39	159 042 651 846	4,704	-43,766	-1,911	-0,555	3 694,00
40	246 886 426 440	3,365	-35,716	-0,992	-0,150	3 876,00
41	1 074 978 988 172	3,916	-7,547	-2,460	-0,351	1 059,00
42	104 678 203 959	12,278	-49,085	-1,137	-0,973	4 939,00
43	350 814 187 273					5 400,00
44	247 320 637 964					5 400,00
45	258 500 523 159					5 400,00
46	353 766 810 606					5 400,00
47	3 612 341 475 073	10,391	-26,421	-1,685	0,536	5 400,00
48	4 459 720 703 629	7,302	-22,943	-1,100	-0,491	5 400,00
49	6 509 714 676 761	5,213	-13,408	-0,777	-0,202	5 400,00
50	234 865 034 818					5 400,00
51	1 187 945 820 688	10,586	-17,743	0,035	0,238	5 400,00
52	564 135 807 189					5 400,00
53	3 818 664 357 644	6,321	-18,415	-1,126	0,303	5 400,00
54	1 228 442 503 004					5 401,00
55	4 578 418 984 539					5 400,00
56	1 155 819 889 523					5 400,00
57	9 307 628 341 889	4,145	-14,066	-0,426	-0,042	5 400,00
58	7 623 802 224 740	9,749	-19,039	0,245	1,104	5 400,00
59	1 499 084 906 192					5 402,00
60	34 657 210 579 797					5 401,00
Moyenne		5,349	-25,067	-1,325	-0,272	2 245,76

TABLE 5.20 – Résultats MS-ILS pour les instances générées aléatoirement

Instances issues	% Écart	% Écart	% Écart	% Écart	Durée (s)
	MS-ILS/LB	MS-ILS/H	MS-ILS/VND	MS-ILS/ILS	
Base de données 1	0,009	-0,001	0,000	0,000	160,00
Base de données 2	1,359	-14,467	0,171	0,229	2 590,60
Générateur aléatoire	5,349	-25,067	-1,325	-0,272	2 245,76
Moyenne	3,445	-17,788	-0,749	-0,114	1 890,47

TABLE 5.21 – Synthèse résultats MS-ILS

## 5.5 Conclusion

Dans ce chapitre, des implémentations de la descente à voisinage variable (Variable neighborhood descent - VND), de la recherche locale itérative (Iterated local search - ILS) et de la recherche locale itérative à démarrages multiples (Multi-start iterated local search - MS-ILS) sont testées. Les instances, utilisées dans les chapitres précédents, sont évaluées avec ces trois métaheuristiques. Les résultats de ces tests sont comparés à ceux fournis par CPLEX et ceux de notre heuristique constructive.

La méthode VND améliore significativement la qualité des solutions de l'heuristique constructive. Le coût des solutions initiales (heuristique constructive) diminue en moyenne de 16,868%, lorsque la VND est appliquée. Les solutions de la VND sont à 4,234% de LB. La VND est 11 fois plus rapide que CPLEX. La méthode ILS (avec 10 itérations) améliore davantage la qualité des solutions (3,567% de la LB). Le temps de calcul moyen augmente, mais reste très inférieur au temps de calcul de CPLEX. La méthode MS-ILS consomme beaucoup de temps, même si la distance à la LB des solutions est de 3,445%. Le temps de calcul moyen de la MS-ILS exécutée avec 3 lancements (chaque lancement s'effectue avec la méthode ILS de 10 itérations) représente plus du double de celui de l'ILS (avec 10 itérations) : 1890,47 secondes pour la MS-ILS et 851,23 secondes pour l'ILS.

Un résumé a été présenté sur les résultats de la méthode VND, appliquée à notre problème, lors de la conférence META 2014 (*International conference on Metaheuristics and Nature Inspired Computing*), en octobre 2014. Un article sur la méthode ILS a été publié dans les actes de la conférence IEEE SSCI 2014 (*Symposium Series on Computational Intelligence*). La méthode MS-ILS a également fait l'objet d'un article publié dans les actes de la conférence ICORES 2015 (*International Conference on Operations Research and Enterprise Systems*). Une synthèse de ce chapitre est en cours de publication dans un livre pour la série des CCIS (*Communications in Computer and Information Science*), publié par Springer-Verlag.

Pour certaines instances, CPLEX ne fournit pas une bonne solution réalisable dans le temps imparti. Nous n'avons pas pu évaluer la distance à l'optimum des solutions des métaheuristiques développées, pour ces instances. Dans le chapitre suivant, une relaxation Lagrangienne est implémentée pour évaluer ces instances avec les autres méthodes : heuristique constructive, VND, ILS et MS-ILS. Le problème lagrangien est résolu par une méthode de sous-gradient, qui change les

multiplicateurs à chaque itération et fournit une suite de bornes inférieures. Les solutions correspondantes subissent une heuristique de réparation, ce qui donne en parallèle une suite de solutions, dont le coût a tendance à baisser au cours des itérations.

Variable Neighborhood Descent (VND) avec un ordre aléatoire des mouvements				
Test	Coût solution	% Écart VND <sub>oam</sub> /LB	% Écart VND <sub>oam</sub> /H	Durée (s)
1	1 274 426 047	0,001	0,000	0,01
2	2 745 520 513	0,001	0,000	0,01
3	3 411 502 169	0,000	0,000	0,01
4	4 202 009 953	0,000	0,000	0,02
5	5 462 674 460	0,001	0,000	0,02
6	6 230 973 230	0,000	0,000	0,02
7	7 219 782 625	0,000	0,000	0,02
8	7 962 941 082	0,000	0,000	0,02
9	8 926 202 452	0,002	0,000	0,02
10	8 942 880 798	0,086	-0,007	29,00
Moyenne		0,009	-0,001	2,92
11	418 489 631	0,637	-16,488	4,00
12	1 525 928 708	5,584	-4,785	12,00
13	1 060 159 926	1,027	-12,958	102,00
14	78 057 104	0,001	-28,885	3,00
15	841 819 760	0,486	-23,445	38,00
16	494 723 325	0,172	-18,863	26,00
17	1 207 380 645	0,724	-18,343	64,00
18	972 565 509	0,755	-24,900	98,00
19	423 514 488	0,747	0,160	496,00
20	500 569 890	0,682	0,579	593,00
Moyenne		1,082	-14,793	143,60
Moyenne des bases 1 et 2		0,545	-7,397	73,26

TABLE 5.22 – Résultats VND<sub>oam</sub> pour les instances issues des bases de données 1 et 2 : test 1

Variable Neighborhood Descent (VND) avec un ordre aléatoire des mouvements					
Test	Coût solution	% Écart VND <sub>oam</sub> /LB	% Écart VND <sub>oam</sub> /H	Durée (s)	
1	1 274 426 047	0,001	0,000	0,01	
2	2 745 520 513	0,001	0,000	0,01	
3	3 411 502 169	0,000	0,000	0,01	
4	4 202 009 953	0,000	0,000	0,02	
5	5 462 674 460	0,001	0,000	0,02	
6	6 230 973 230	0,000	0,000	0,02	
7	7 219 782 625	0,000	0,000	0,02	
8	7 962 941 082	0,000	0,000	0,02	
9	8 926 202 452	0,002	0,000	0,02	
10	8 942 835 514	0,085	-0,008	29,00	
Moyenne		0,009	-0,001	2,92	
11	418 489 606	0,637	-16,488	4,00	
12	1 525 927 971	5,584	-4,785	12,00	
13	1 061 673 581	1,172	-12,797	100,00	
14	78 633 747	0,739	-27,940	3,00	
15	843 890 726	0,733	-23,142	40,00	
16	495 877 497	0,406	-18,586	25,00	
17	1 208 944 131	0,855	-18,190	64,00	
18	970 915 765	0,584	-25,112	97,00	
19	423 513 518	0,747	0,160	491,00	
20	500 571 458	0,682	0,580	618,00	
Moyenne		1,214	-14,630	145,40	
Moyenne des bases 1 et 2		0,611	-7,315	74,16	

TABLE 5.23 – Résultats VND<sub>oam</sub> pour les instances issues des bases de données 1 et 2 : test 2

Variable Neighborhood Descent (VND) avec un ordre aléatoire des mouvements						
Test	Coût solution	% Écart VND <sub>oam</sub> /LB	% Écart VND <sub>oam</sub> /H	Durée (s)		
21	101 024 257 058	6,160	-30,353	36,00		
22	237 495 519 161	0,643	-1,502	1,00		
23	75 371 018 423	2,411	-9,778	5,00		
24	111 094 078 597	3,724	-21,336	5,00		
25	57 074 477 968	5,107	-34,758	7,00		
26	50 877 452 100	5,969	-40,515	6,00		
27	974 575 957 428	6,586	-13,590	16,00		
28	113 907 756 429	6,601	-36,617	25,00		
29	108 447 620 250	4,700	-35,703	28,00		
30	158 682 812 823	4,907	-33,087	25,00		
31	218 108 441 456	2,613	-16,949	26,00		
32	431 807 754 913	4,859	-9,111	15,00		
33	508 233 105 482	12,338	-30,217	20,00		
34	845 586 483 873	5,578	-6,733	10,00		
35	770 311 738 680	6,781	-12,858	18,00		
36	535 587 293 340	12,207	-24,466	22,00		
37	257 995 850 284	5,493	-34,978	120,00		
38	252 409 028 853	4,724	-36,569	115,00		
39	161 686 346 149	6,445	-41,415	119,00		
40	250 187 874 639	4,747	-33,925	119,00		
41	1 102 008 187 335	6,529	-4,909	44,00		
42	107 088 691 703	14,864	-45,730	244,00		
43	357 938 191 940			1119,00		
44	250 391 097 075			1146,00		
45	263 242 793 153			952,00		
46	357 397 520 097			1139,00		
47	3 674 084 596 270	12,278	-24,297	681,00		
48	4 521 997 235 550	8,800	-21,250	553,00		
49	6 579 928 239 585	6,348	-12,198	429,00		
50	237 777 368 444			2033,00		
51	1 188 910 744 677	10,676	-17,648	264,00		
52	566 516 320 519			895,00		
53	3 858 915 094 266	7,442	-17,180	476,00		
54	1 075 765 684 433			5400,00		
55	4 537 921 744 574			5400,00		
56	1 157 608 778 655			3239,00		
57	9 317 935 953 609	4,260	-13,940	2065,00		
58	7 571 977 108 322	9,003	-19,854	2902,00		
59	1 453 536 004 717			5401,00		
60	34 423 592 698 354			5400,00		
Moyenne		6,648	-23,499	289,52		

TABLE 5.24 – Résultats VND<sub>oam</sub> pour les instances générées aléatoirement : test 1

Variable Neighborhood Descent (VND) avec un ordre aléatoire des mouvements						
Test	Coût solution	% Écart VND <sub>oam</sub> /LB	% Écart VND <sub>oam</sub> /H	Durée (s)		
21	100 754 886 384	5,877	-30,702	35,00		
22	237 378 266 344	0,593	-1,553	1,00		
23	75 626 201 960	2,758	-9,408	6,00		
24	111 614 627 364	4,210	-20,770	5,00		
25	57 623 526 474	6,118	-33,474	5,00		
26	51 235 694 481	6,715	-39,533	6,00		
27	972 955 911 901	6,409	-13,779	14,00		
28	114 286 189 929	6,956	-36,165	22,00		
29	109 049 034 523	5,281	-34,954	27,00		
30	158 369 310 000	4,700	-33,350	22,00		
31	218 322 748 086	2,714	-16,834	23,00		
32	435 484 319 352	5,752	-8,190	13,00		
33	512 587 822 736	13,300	-29,110	17,00		
34	851 140 310 559	6,272	-6,036	11,00		
35	771 841 635 629	6,993	-12,634	15,00		
36	536 825 811 144	12,467	-24,179	20,00		
37	258 148 512 688	5,556	-34,898	139,00		
38	252 725 012 925	4,855	-36,398	117,00		
39	162 813 885 428	7,187	-40,436	110,00		
40	248 774 296 266	4,156	-34,686	135,00		
41	1 103 065 661 088	6,631	-4,808	36,00		
42	107 111 414 203	14,888	-45,699	215,00		
43	353 672 352 599			1120,00		
44	253 710 426 356			1136,00		
45	259 402 896 474			1079,00		
46	355 348 565 653			1191,00		
47	3 698 794 243 391	13,033	-23,466	660,00		
48	4 500 658 727 421	8,287	-21,824	694,00		
49	6 544 956 368 150	5,783	-12,797	431,00		
50	234 793 798 676			1687,00		
51	1 188 453 853 199	10,633	-17,693	290,00		
52	566 010 555 096			1090,00		
53	3 857 171 989 726	7,393	-17,233	718,00		
54	1 081 220 934 263			5400,00		
55	4 597 236 466 611			5400,00		
56	1 154 591 607 850			3208,00		
57	9 329 018 915 389	4,384	-13,805	2381,00		
58	7 618 697 967 643	9,675	-19,119	2630,00		
59	1 360 460 515 968			5400,00		
60	34 703 539 825 755			5400,00		
Moyenne		6,882	-23,225	303,38		

TABLE 5.25 – Résultats VND<sub>oam</sub> pour les instances générées aléatoirement : test 2

Iterated Local Search (ILS) avec 10 itérations et un ordre aléatoire des mouvements à chaque VND						
Test	Coût solution	% Écart		% Écart		Durée (s)
	$ILS_{VND_{oam}}$	$ILS_{VND_{oam}}/LB$	$ILS_{VND_{oam}}/H$	$ILS_{VND_{oam}}/VND$	$ILS_{VND_{oam}}/VND$	
1	1 274 426 047	0,001	0,000	0,000	0,000	0,01
2	2 745 520 513	0,001	0,000	0,000	0,000	1,00
3	3 411 502 169	0,000	0,000	0,000	0,000	1,00
4	4 202 009 953	0,000	0,000	0,000	0,000	2,00
5	5 462 674 460	0,001	0,000	0,000	0,000	3,00
6	6 230 973 230	0,000	0,000	0,000	0,000	4,00
7	7 219 782 625	0,000	0,000	0,000	0,000	4,00
8	7 962 941 082	0,000	0,000	0,000	0,000	6,00
9	8 926 202 452	0,002	0,000	0,000	0,000	6,00
10	8 942 859 613	0,085	-0,007	0,000	0,000	480,00
Moyenne		0,009	-0,001	0,000	0,000	50,70
11	417 978 333	0,514	-16,630	-0,122	0,000	38,00
12	1 515 490 487	4,862	-5,506	-0,689	0,000	115,00
13	1 059 963 870	1,009	-12,979	-0,018	0,000	768,00
14	78 057 104	0,001	-28,885	0,000	0,000	29,00
15	841 818 581	0,486	-23,445	0,000	0,000	276,00
16	494 673 491	0,162	-18,875	-0,010	0,000	198,00
17	1 206 024 985	0,611	-18,476	-0,112	0,000	445,00
18	972 404 394	0,738	-24,920	-0,017	0,000	695,00
19	423 086 837	0,646	0,059	-0,101	0,000	5 400,00
20	500 430 572	0,654	0,552	-0,028	0,000	5 400,00
Moyenne		0,968	-14,911	-0,110	0,000	1 336,40
Moyenne des bases 1 et 2		0,489	-7,456	-0,055	0,000	693,55

TABLE 5.26 – Résultats  $ILS_{VND_{oam}}$  pour les instances issues des bases de données 1 et 2

Iterated Local Search (ILS) avec 10 itérations et un ordre aléatoire des mouvements à chaque VND						
Test	Coût solution	% Écart		% Écart		Durée (s)
	$ILS_{VND_{oam}}$	$ILS_{VND_{oam}}/LB$	$ILS_{VND_{oam}}/H$	$ILS_{VND_{oam}}/VND$	$ILS_{VND_{oam}}/VND$	
21	98 915 641 897	3,944	-33,132	-2,132	221,00	
22	237 041 853 462	0,451	-1,697	-0,191	11,00	
23	74 962 259 099	1,856	-10,377	-0,545	36,00	
24	108 683 689 851	1,474	-24,027	-2,218	35,00	
25	56 574 468 417	4,186	-35,949	-0,884	41,00	
26	49 799 589 263	3,724	-43,557	-2,164	44,00	
27	966 866 987 690	5,743	-14,495	-0,797	64,00	
28	112 872 853 422	5,633	-37,870	-0,917	181,00	
29	106 882 660 407	3,189	-37,689	-1,464	172,00	
30	157 173 438 528	3,909	-34,365	-0,960	165,00	
31	217 061 359 269	2,120	-17,513	-0,482	165,00	
32	429 105 584 857	4,203	-9,798	-0,630	58,00	
33	504 719 784 091	11,561	-31,123	-0,696	80,00	
34	841 441 208 294	5,061	-7,258	-0,493	52,00	
35	767 474 289 643	6,387	-13,275	-0,370	70,00	
36	527 605 980 243	10,535	-26,349	-1,513	90,00	
37	255 354 928 999	4,413	-36,374	-1,034	822,00	
38	249 755 346 649	3,623	-38,020	-1,063	760,00	
39	159 955 852 063	5,305	-42,945	-1,082	764,00	
40	247 604 681 178	3,666	-35,322	-1,043	852,00	
41	1 093 449 777 389	5,702	-5,730	-0,783	218,00	
42	105 496 225 863	13,156	-47,929	-1,510	1358,00	
43	352 245 834 466				5400,00	
44	246 437 125 024				5400,00	
45	258 461 062 472				5400,00	
46	356 939 688 698				5400,00	
47	3 583 763 719 870	9,518	-27,429	-2,520	3467,00	
48	4 470 839 780 749	7,569	-22,637	-1,144	2667,00	
49	6 519 076 579 427	5,365	-13,245	-0,933	1968,00	
50	235 272 584 302				5400,00	
51	1 184 331 092 632	10,250	-18,103	-0,387	1360,00	
52	561 840 223 328				5400,00	
53	3 788 498 398 835	5,481	-19,358	-1,859	2631,00	
54	1 075 765 684 433				5400,00	
55	4 537 921 744 574				5400,00	
56	1 153 896 156 660				5400,00	
57	9 260 818 335 135	3,621	-14,643	-0,617	5400,00	
58	7 512 977 053 339	8,153	-20,795	-0,785	5400,00	
59	1 453 536 004 717				5401,00	
60	34 423 592 698 354				5400,00	
Moyenne		5,510	-24,862	-1,076	1 005,24	

TABLE 5.27 – Résultats  $ILS_{VND_{oam}}$  pour les instances générées aléatoirement

Iterated Local Search (ILS) avec 10 itérations et un seul ordre aléatoire d'application des mouvements pour toutes les exécutions de la VND					
Test	Coût solution	% Écart ILS <sub>oam</sub> /LB	% Écart ILS <sub>oam</sub> /H	% Écart ILS <sub>oam</sub> /VND	Durée (s)
1	1 274 426 047	0,001	0,000	0,000	0,01
2	2 745 520 513	0,001	0,000	0,000	1,00
3	3 411 502 169	0,000	0,000	0,000	1,00
4	4 202 009 953	0,000	0,000	0,000	2,00
5	5 462 674 460	0,001	0,000	0,000	3,00
6	6 230 973 230	0,000	0,000	0,000	4,00
7	7 219 782 625	0,000	0,000	0,000	4,00
8	7 962 941 082	0,000	0,000	0,000	6,00
9	8 926 202 452	0,002	0,000	0,000	6,00
10	8 942 835 514	0,085	-0,008	0,000	512,00
Moyenne		0,009	-0,001	0,000	53,90
11	418 164 119	0,559	-16,579	-0,078	38,00
12	1 524 145 614	5,461	-4,907	-0,117	120,00
13	1 061 673 581	1,172	-12,797	0,000	828,00
14	78 579 473	0,670	-28,028	-0,069	30,00
15	843 886 359	0,733	-23,142	-0,001	281,00
16	494 987 026	0,226	-18,799	-0,180	195,00
17	1 207 486 390	0,733	-18,332	-0,121	462,00
18	970 915 765	0,584	-25,112	0,000	764,00
19	423 242 471	0,683	0,096	-0,064	5 400,00
20	500 164 550	0,600	0,499	-0,081	5 400,00
Moyenne		1,142	-14,710	-0,071	1 351,80
Moyenne des bases 1 et 2		0,575	-7,355	-0,036	702,85

TABLE 5.28 – Résultats ILS<sub>oam</sub> pour les instances issues des bases de données 1 et 2

Iterated Local Search (ILS) avec 10 itérations et un seul ordre aléatoire d'application des mouvements pour toutes les exécutions de la VND						
Test	Coût solution	% Écart ILS <sub>oam</sub> /LB	% Écart ILS <sub>oam</sub> /H	% Écart ILS <sub>oam</sub> /VND	Durée (s)	
21	99 561 207 001	4,622	-32,269	-1,199	205,00	
22	237 266 565 107	0,546	-1,600	-0,047	11,00	
23	74 974 169 411	1,872	-10,359	-0,870	36,00	
24	109 159 502 536	1,918	-23,486	-2,249	45,00	
25	57 218 895 285	5,373	-34,418	-0,707	38,00	
26	50 225 177 214	4,610	-42,340	-2,012	38,00	
27	965 951 141 885	5,643	-14,604	-0,725	59,00	
28	112 540 409 782	5,322	-38,277	-1,551	185,00	
29	107 545 387 244	3,829	-36,841	-1,398	193,00	
30	157 196 587 026	3,925	-34,345	-0,746	145,00	
31	217 941 233 899	2,534	-17,038	-0,175	167,00	
32	431 204 351 133	4,713	-9,263	-0,993	57,00	
33	499 501 322 981	10,408	-32,493	-2,620	78,00	
34	847 029 626 935	5,758	-6,551	-0,485	50,00	
35	763 452 123 176	5,830	-13,872	-1,099	63,00	
36	527 812 044 273	10,578	-26,300	-1,708	88,00	
37	255 622 835 541	4,523	-36,231	-0,988	803,00	
38	249 311 637 474	3,439	-38,265	-1,369	735,00	
39	159 680 794 835	5,124	-43,191	-1,962	688,00	
40	247 338 864 278	3,555	-35,467	-0,580	807,00	
41	1 084 214 145 060	4,809	-6,631	-1,739	220,00	
42	106 600 957 081	14,341	-46,396	-0,479	1226,00	
43	350 925 675 929				5400,00	
44	252 011 574 418				5400,00	
45	259 222 135 759				5400,00	
46	354 696 518 510				5400,00	
47	3 615 666 015 046	10,493	-26,305	-2,299	2738,00	
48	4 470 956 797 004	7,572	-22,634	-0,664	2763,00	
49	6 503 193 387 588	5,108	-13,522	-0,642	1993,00	
50	234 240 115 673				5400,00	
51	1 185 103 414 280	10,322	-18,026	-0,283	1610,00	
52	562 406 548 059				5400,00	
53	3 812 057 234 692	6,137	-18,620	-1,183	2716,00	
54	1 081 220 934 263				5400,00	
55	4 597 236 466 611				5400,00	
56	1 152 429 111 055				5400,00	
57	9 297 755 548 900	4,034	-14,187	-0,336	5400,00	
58	7 578 189 326 680	9,092	-19,755	-0,535	5400,00	
59	1 360 460 515 968				5400,00	
60	34 703 539 825 755				5400,00	
Moyenne		5,725	-24,596	-1,091	984,72	

TABLE 5.29 – Résultats ILS<sub>oam</sub> pour les instances générées aléatoirement

ILS avec 10 itérations et 5% de taux de perturbation					
Test	coût solution	% Écart	% Écart	% Écart	Durée
	ILS	ILS/LB	ILS/H	ILS/VND	(s)
1	1 274 426 047	0,001	0,000	0,000	0,00
2	2 745 520 513	0,001	0,000	0,000	1,00
3	3 411 502 169	0,000	0,000	0,000	1,00
4	4 202 009 953	0,000	0,000	0,000	2,00
5	5 462 674 460	0,001	0,000	0,000	3,00
6	6 230 973 230	0,000	0,000	0,000	4,00
7	7 219 782 625	0,000	0,000	0,000	4,00
8	7 962 941 082	0,000	0,000	0,000	5,00
9	8 926 202 452	0,002	0,000	0,000	6,00
10	8 942 845 621	0,085	-0,007	0,000	398,00
Moyenne		0,009	-0,001	0,000	42,40
11	419 021 964	0,765	-16,340	0,127	36,00
12	1 526 814 681	5,645	-4,724	-0,065	110,00
13	1 061 140 276	1,121	-12,854	-0,049	631,00
14	79 057 588	1,282	-27,254	1,266	29,00
15	844 678 854	0,827	-23,027	0,107	238,00
16	497 940 770	0,824	-18,095	0,281	186,00
17	1 210 326 777	0,970	-18,055	0,085	391,00
18	975 032 076	1,010	-24,584	0,269	609,00
19	423 137 522	0,658	0,071	-0,089	4 996,00
20	500 196 749	0,607	0,505	-0,075	5 400,00
Moyenne		1,371	-14,435	0,186	1 262,60
Moyenne des bases 1 et 2		0,690	-7,218	0,093	652,50

TABLE 5.30 – Résultats ILS avec 5% de taux de perturbation pour les instances issues des bases de données 1 et 2

ILS avec 10 itérations et 20% de taux de perturbation					
Test	coût solution ILS	% Écart ILS/LB	% Écart ILS/H	% Écart ILS/VND	Durée (s)
1	1 274 426 047	0,001	0,000	0,000	0,00
2	2 745 520 513	0,001	0,000	0,000	1,00
3	3 411 502 169	0,000	0,000	0,000	1,00
4	4 202 009 953	0,000	0,000	0,000	2,00
5	5 462 674 460	0,001	0,000	0,000	3,00
6	6 230 973 230	0,000	0,000	0,000	4,00
7	7 219 782 625	0,000	0,000	0,000	5,00
8	7 962 941 082	0,000	0,000	0,000	6,00
9	8 926 202 452	0,002	0,000	0,000	6,00
10	8 942 853 389	0,085	-0,007	0,000	566,00
Moyenne		0,009	-0,001	0,000	59,40
11	418 674 873	0,682	-16,436	0,044	41,00
12	1 515 768 247	4,881	-5,487	-0,795	120,00
13	1 060 508 193	1,060	-12,921	-0,109	803,00
14	79 057 588	1,282	-27,254	1,266	31,00
15	844 683 178	0,828	-23,026	0,108	289,00
16	497 775 003	0,790	-18,134	0,248	209,00
17	1 209 603 098	0,910	-18,125	0,025	481,00
18	975 125 302	1,020	-24,572	0,278	740,00
19	423 320 647	0,701	0,114	-0,046	5 400,00
20	500 067 636	0,581	0,480	-0,101	5 400,00
Moyenne		1,274	-14,536	0,092	1 351,40
Moyenne des bases 1 et 2		0,641	-7,268	0,046	705,40

TABLE 5.31 – Résultats ILS avec 20% de taux de perturbation pour les instances issues des bases de données 1 et 2

ILS avec 10 itérations et 5% de taux de perturbation						
Test	coût solution	% Écart	% Écart	% Écart	Durée	
	ILS	ILS/LB	ILS/H	ILS/VND	(s)	
21	99 419 845 038	4,474	-32,457	-0,691	198,00	
22	237 165 984 273	0,503	-1,644	-0,135	10,00	
23	75 054 748 810	1,982	-10,241	-0,724	32,00	
24	110 620 549 503	3,282	-21,855	-1,156	27,00	
25	56 644 984 090	4,316	-35,780	-0,728	31,00	
26	50 310 632 384	4,788	-42,098	-1,974	37,00	
27	971 486 750 169	6,248	-13,951	-0,381	45,00	
28	112 616 908 255	5,393	-38,183	-1,463	142,00	
29	107 514 308 247	3,799	-36,881	-0,730	128,00	
30	156 126 017 435	3,217	-35,266	-1,514	131,00	
31	218 542 610 144	2,817	-16,716	-0,308	133,00	
32	428 737 473 221	4,114	-9,892	-0,937	49,00	
33	509 850 858 502	12,695	-29,804	0,206	63,00	
34	844 982 429 489	5,503	-6,809	-0,823	41,00	
35	764 306 342 735	5,948	-13,745	-1,126	52,00	
36	530 750 195 357	11,194	-25,601	-0,929	67,00	
37	256 390 652 698	4,837	-35,823	-1,678	704,00	
38	250 231 573 889	3,821	-37,757	-0,770	627,00	
39	160 748 987 463	5,828	-42,240	-0,829	654,00	
40	247 144 469 949	3,473	-35,574	-0,887	699,00	
41	1 091 057 911 914	5,470	-5,962	-0,950	181,00	
42	105 411 209 534	13,065	-48,049	-0,434	1046,00	
43	351 363 690 856				5400,00	
44	249 204 136 142				5400,00	
45	257 929 676 218				5400,00	
46	354 183 726 116				5383,00	
47	3 609 774 205 242	10,313	-26,511	-1,757	2674,00	
48	4 459 445 496 689	7,295	-22,950	-1,107	2219,00	
49	6 524 445 370 226	5,451	-13,152	-0,549	1653,00	
50	234 380 517 646				5400,00	
51	1 185 202 954 339	10,331	-18,016	-0,196	1060,00	
52	562 185 099 178				5400,00	
53	3 810 636 979 103	6,097	-18,664	-1,339	2279,00	
54	1 063 153 095 169				5400,00	
55	4 520 240 434 137				5400,00	
56	1 154 365 280 531				5400,00	
57	9 264 056 974 134	3,657	-14,603	-0,898	5400,00	
58	7 522 415 117 624	8,289	-20,643	-1,099	5400,00	
59	1 476 275 066 114				5400,00	
60	33 608 700 802 168				5400,00	
	Moyenne	5,800	-24,513	-0,893	889,03	

TABLE 5.32 – Résultats ILS avec 5% de taux de perturbation pour les instances générées aléatoirement

ILS avec 10 itérations et 20% de taux de perturbation						
Test	coût solution	% Écart	% Écart	% Écart	Durée	
	ILS	ILS/LB	ILS/H	ILS/VND	(s)	
21	99 127 692 929	4,167	-32,847	-0,987	233,00	
22	237 061 233 636	0,459	-1,688	-0,180	13,00	
23	74 865 203 846	1,724	-10,520	-0,979	42,00	
24	110 053 690 681	2,753	-22,483	-1,677	44,00	
25	56 641 854 031	4,310	-35,787	-0,734	41,00	
26	49 739 234 616	3,598	-43,731	-3,145	52,00	
27	962 966 330 013	5,316	-14,959	-1,269	63,00	
28	112 787 702 577	5,553	-37,974	-1,309	192,00	
29	106 850 260 540	3,158	-37,731	-1,356	207,00	
30	156 493 849 512	3,460	-34,948	-1,275	188,00	
31	217 269 153 596	2,218	-17,400	-0,896	216,00	
32	427 797 082 341	3,885	-10,134	-1,159	60,00	
33	504 398 484 361	11,490	-31,207	-0,872	84,00	
34	843 096 654 074	5,267	-7,048	-1,048	55,00	
35	762 711 990 583	5,727	-13,983	-1,337	65,00	
36	526 575 132 652	10,319	-26,596	-1,729	95,00	
37	255 344 609 275	4,409	-36,379	-2,095	967,00	
38	249 478 877 234	3,508	-38,173	-1,074	887,00	
39	159 565 669 235	5,049	-43,294	-1,577	894,00	
40	246 580 532 096	3,237	-35,884	-1,117	966,00	
41	1 078 027 497 994	4,211	-7,243	-2,170	258,00	
42	105 257 329 816	12,900	-48,265	-0,580	1395,00	
43	348 114 314 761				5400,00	
44	249 122 118 143				5400,00	
45	257 220 432 409				5400,00	
46	354 257 694 326				5400,00	
47	3 562 587 430 185	8,871	-28,187	-3,105	3724,00	
48	4 432 815 719 270	6,655	-23,689	-1,714	2661,00	
49	6 514 099 205 000	5,284	-13,332	-0,709	1921,00	
50	232 226 196 234				5400,00	
51	1 183 861 244 787	10,206	-18,150	-0,310	1518,00	
52	559 840 906 586				5400,00	
53	3 763 652 702 210	4,789	-20,146	-2,604	3198,00	
54	1 062 600 553 735				5400,00	
55	4 522 177 966 747				5400,00	
56	1 152 490 781 339				5400,00	
57	9 255 911 999 097	3,566	-14,703	-0,987	5400,00	
58	7 508 944 370 958	8,095	-20,860	-1,281	5400,00	
59	1 476 275 066 114				5401,00	
60	33 608 805 522 085				5400,00	
	Moyenne	5,317	-25,081	-1,354	1 063,41	

TABLE 5.33 – Résultats ILS avec 20% de taux de perturbation pour les instances générées aléatoirement



## Chapitre 6

# Relaxation Lagrangienne

### 6.1 Introduction

Dans le chapitre 3, nous proposons un modèle mathématique sous la forme d'un programme linéaire mixte. Des instances sont testées avec le solveur CPLEX. Pour certaines instances, CPLEX ne fournit pas une bonne solution réalisable, dans le temps imparti (1 heure et demie). Nous n'avons pas pu évaluer la distance à l'optimum, des solutions de l'heuristique constructive et des métaheuristicques (VND, ILS et MS-ILS) présentées dans les chapitres précédents, pour ces instances.

Une relaxation Lagrangienne est implémentée pour évaluer ces instances et comparer les résultats avec ceux des autres méthodes : heuristique constructive, VND, ILS et MS-ILS. Une heuristique Lagrangienne est également développée. Le problème lagrangien est résolu par une méthode de sous-gradient, qui change les multiplicateurs à chaque itération et fournit une suite de bornes inférieures. Les solutions (non réalisables) correspondantes subissent une heuristique de réparation, ce qui donne en parallèle une suite de solutions, dont le coût a tendance à baisser au cours des itérations.

Des tests préliminaires sont effectués dans la section 6.2 sur les contraintes qui rendent le problème difficile. Nous détaillons ensuite le principe de la relaxation Lagrangienne dans la section 6.3. Une heuristique Lagrangienne est implémentée dans la section 6.4. Les résultats des tests préliminaires ont montré que la relaxation de la contrainte de taille de lot est la plus pertinente. La dualisation de cette contrainte est expliquée dans la section 6.5 et ensuite testée dans la section 6.6. Nous résumons les contributions de ce chapitre et donnons des perspectives dans la section 6.7.

### 6.2 Tests préliminaires

Dans les chapitres précédents, nous remarquons que les instances générées aléatoirement présentent plus de difficultés. Les instances, pour lesquelles CPLEX n'a pas pu fournir une bonne

solution réalisable, au bout de 1 heure et demie de calcul, font partie des instances générées aléatoirement. La relaxation des contraintes est testée sur ces instances.

Les contraintes relaxées sont respectivement la capacité de transport, la quantité minimale de transport, l'ordre imposé par l'utilisateur, le contrat de fraîcheur et la taille de lot de transport. Ces choix découlent de la connaissance du problème traité. La fonction-objectif est la même que celle du programme linéaire mixte (PL mixte) non relaxé. La contrainte relaxée n'a pas été ajoutée à la fonction-objectif ; ce qui équivaut à des pénalités nulles. Les notations utilisées sont les suivantes :

- UBr est le coût de la solution du problème relaxé,
- LBr est la meilleure borne inférieure du problème relaxé,
- UB est le coût de la solution du problème primal (non relaxé),
- LB est la meilleure borne inférieure du problème primal (non relaxé).

$$\% \text{ Écart UB/LB} = \frac{\text{UB} - \text{LB}}{\text{LB}} \times 100$$

$$\% \text{ Écart UBr/LBr} = \frac{\text{UBr} - \text{LBr}}{\text{LBr}} \times 100$$

$$\% \text{ Ratio LBr/LB} = \frac{\text{LBr}}{\text{LB}} \times 100$$

$$\% \text{ Ratio UBr/UB} = \frac{\text{UBr}}{\text{UB}} \times 100$$

$$\% \text{ Ratio durée (R/NR)} = \frac{\text{Durée de CPLEX (PL mixte relaxé)}}{\text{Durée de CPLEX (PL mixte non relaxé)}} \times 100$$

Une synthèse des résultats de la relaxation des contraintes est présentée dans le Tableau 6.1. Les valeurs moyennes sont calculées avec les instances pour lesquelles CPLEX fournit une bonne solution réalisable, pour le problème primal (PL mixte non relaxé). Le temps de calcul défini pour CPLEX, pour le problème relaxé (PL mixte relaxé), est le même que celui du problème primal ; c'est à dire 1 heure et demie (5400 secondes). L'autre critère d'arrêt est le pourcentage d'écart (% Écart UBr/LBr), entre la borne inférieure et la borne supérieure. Il est paramétré à 0,01% ; c'est la même valeur pour le problème primal et le problème relaxé.

La relaxation de la contrainte de quantité minimale de transport et celle de la taille de lot de transport, donnent les meilleurs résultats (voir Tableau 6.1). Lorsque ces deux contraintes sont, respectivement, relaxées, le PL mixte est résolu à l'optimum. Les écarts, entre la borne inférieure et la borne supérieure sont, en moyenne, de 0,006%. En arrondissant cette valeur, elle correspond au critère d'arrêt.

Le temps de calcul moyen le plus faible est celui de la relaxation de la contrainte de taille de lot (32,97 secondes). Lorsque CPLEX résout le PL mixte avec la contrainte de taille de lot relaxée,

il consomme 6,07% de la durée utilisée pour le problème primal. La relaxation de la contrainte de taille de lot permet également d'obtenir un très bon ratio, entre le coût des solutions du problème primal et celui des solutions du problème relaxé (% Ratio UBr/UB). Les coûts des solutions du PL mixte, avec la contrainte de taille de lot relaxée, sont à 99,186% de ceux du problème primal. Il en est de même (environ) pour les bornes inférieures (99,635%).

Lorsque les contraintes de capacité transport, de contrat de fraîcheur et de l'ordre imposé sont, respectivement, relaxées, CPLEX ne fournit pas une bonne solution réalisable pour toutes les instances.

Pour plus de détail sur les résultats de chaque instance, le lecteur peut consulter l'annexe. Le détail des résultats des instances générées aléatoirement est présenté :

- pour la relaxation de la contrainte de capacité de transport, dans le Tableau 6.5,
- pour la relaxation de la contrainte de quantité minimale de transport, dans le Tableau 6.6,
- pour la relaxation de la contrainte de contrat de fraîcheur, dans le Tableau 6.7,
- pour la relaxation de la contrainte d'ordre imposé, dans le Tableau 6.8,
- pour la relaxation de la contrainte de taille de lot de transport, dans le Tableau 6.9.

Les résultats des tests préliminaires ont montré que la relaxation de la contrainte de taille de lot est la plus pertinente pour les instances générées aléatoirement.

PL mixte non relaxé		PL mixte relaxé			% Ratio	% Ratio	% Ratio
% Écart UB/LB	Durée (s)	Contrainte relaxée	(UBr/LBr)	Durée (s)	LBr/LB	UBr/UB	durée (R/NR)
0,519	1418,28	capacité transport	2,295	1581,23	89,157	90,905	220,55
		quantité minimale	0,006	305,73	99,484	99,040	38,43
		contrat de fraîcheur	2,378	1417,357	99,818	101,738	137,240
		ordre imposé	40,175	1380,70	63,904	69,025	118,60
		taille de lot	0,006	32,97	99,635	99,186	6,07

TABLE 6.1 – Synthèse des résultats de la relaxation des contraintes difficiles

## 6.3 Principe de la relaxation Lagrangienne

La relaxation Lagrangienne permet de calculer une suite de bornes inférieures avec la méthode de sous-gradient.

### 6.3.1 Fonction de Lagrange

Considérons un problème (P) de programmation mathématique discrète (PMD) très général, où  $f$  et les  $m$   $g_i$  sont des fonctions quelconques de  $n$  variables, et  $S$  un sous-ensemble fini de  $\mathbb{R}^n$ , défini en pratique par d'autres contraintes. Par exemple, dans un programme linéaire en nombres entiers (PLNE),  $f$  et les  $g_i$  sont linéaires, et  $S = \mathbb{N}^n$ . On dira que (P) est le problème primal, par opposition au dual qui sera introduit ensuite. On notera  $x^*$  une solution optimale de (P) et  $z^*$  son coût.

$$\begin{aligned} & \text{Min } z = f(x) \\ (P) \quad & g_i(x) \leq 0, \quad i = 1..m \\ & x \in S \end{aligned}$$

Supposons que les contraintes  $g_i(x) \leq 0$  rendent le problème particulièrement difficile. La relaxation Lagrangienne (RL) consiste à les inclure sous forme de pénalités dans la fonction économique : on dit qu'on dualise ces contraintes. Pour cela, on associe à chaque contrainte un multiplicateur de Lagrange ou variable duale  $\Pi_i$ , réel positif. En notant  $g(x)$  le vecteur des fonctions  $g_i$ , on peut définir la fonction de Lagrange  $L(x, \Pi) = f(x) + \Pi.g(x)$ .

Notez qu'un PMD peut toujours se mettre sous la forme canonique (P) pour avoir des multiplicateurs positifs. Si on a une contrainte  $g_i(x) \geq 0$ , on peut la remplacer par  $g'_i(x) = -g_i(x) \leq 0$ . On peut aussi remplacer une égalité  $g_i(x) = 0$  par deux contraintes d'inégalité, mais cela donne des formulations lourdes : il vaut mieux dans ce cas conserver l'égalité et lui associer un multiplicateur non contraint en signe.

### 6.3.2 Méthode de sous-gradient

**Définition 1.** La fonction duale  $L(\Pi)$  est concave et linéaire par morceaux.

**Preuve 1.** Pour une fonction à une variable, l'interprétation géométrique de la concavité est la suivante : si on choisit deux points A et B sur la courbe de la fonction, le segment [A,B] est situé sous la courbe. C'est le cas par exemple pour  $f(x) = -x^2$ .

Dans le cas général, une fonction  $f$  est concave ssi pour tout couple de points  $(x_1, x_2)$  et tout  $k$  dans  $[0, 1]$ , on a  $f(k.x_1 + (1 - k).x_2) \geq k.f(x_1) + (1 - k).f(x_2)$ . Soit deux vecteurs de multiplicateurs

$\Pi^1, \Pi^2, k$  dans  $[0, 1]$ , et  $\Pi = k.\Pi^1 + (1 - k).\Pi^2$ . Par définition de  $L(\Pi)$ , il existe  $x'$  de  $S$  avec  $L(\Pi) = f(x') + \Pi.g(x')$ . Par définition de  $L(\Pi^1)$  et  $L(\Pi^2)$ , on a :

$$\begin{aligned} L(\Pi^1) &\leq f(x') + \Pi^1.g(x') \\ L(\Pi^2) &\leq f(x') + \Pi^2.g(x') \end{aligned}$$

Multiplions la première expression par  $k \geq 0$ , la deuxième par  $(1 - k) \geq 0$  et additionnons. On obtient :

$$\begin{aligned} k.L(\Pi^1) + (1 - k).L(\Pi^2) &\leq f(x') + k.\sum(\Pi_i^1.g_i(x')) + (1 - k).\sum(\Pi_i^2.g_i(x')) \\ &\leq f(x') + \sum(k.\Pi_i^1 + (1 - k).\Pi_i^2).g_i(x') = f(x') + \sum\Pi_i.g_i(x') = L(\Pi) \end{aligned}$$

La fonction duale est donc concave. Elle est aussi linéaire par morceaux, car elle est l'enveloppe inférieure des hyperplans définis par tout point  $x_k$  de  $S$ , d'équation :

$$t = f(x_k) + \Pi.g(x_k)$$

La concavité de la fonction duale est intéressante : tout optimum local sera aussi global, ce qui explique que le problème dual est en général plus facile que le primal. De plus, elle ne suppose rien sur la convexité ou concavité de  $f$  et des  $g_i$ . La linéarité par morceaux empêche cependant les techniques classiques de gradient, puisque  $L$  n'admet pas de gradients aux points anguleux. On utilise alors une méthode d'augmentations successives basée sur des sous-gradients, qui généralisent la notion de gradient aux fonctions non partout différentiables.

**Définition 1.** Soit une fonction concave  $f$  de  $\mathbb{R}^n$  dans  $\mathbb{R}$  et un point  $x_0$  de  $\mathbb{R}^n$ . Un sous-gradient de  $f$  en  $x_0$  est une direction d'augmentation de  $f$ , c'est-à-dire un vecteur  $\gamma$  tel que :  $f(x) \leq f(x_0) + \gamma^T.(x - x_0)$ .

**Propriété 1.** Soit  $y$  la solution du problème lagrangien en  $\Pi^0$ , c'est-à-dire  $L(\Pi^0) = f(y) + \Pi^0.g(y) = \text{Min}f(x) + \Pi^0.g(x), x \in S$ . Alors  $g(y)$  est un sous-gradient de la fonction duale en  $\Pi^0$ .

**Preuve 1.** Pour tout  $\Pi$ ,  $L(\Pi) \leq f(y) + \Pi.g(y) = f(y) + \Pi.g(y) + \Pi^0.g(y) - \Pi^0.g(y) = L(\Pi^0) + g(y).( \Pi - \Pi^0)$ .

Malheureusement, la fonction peut ne pas augmenter pour certains sous-gradients. C'est pourquoi, il est inutile de chercher à maximiser  $L$  dans la direction d'un seul sous-gradient. Il vaut mieux calculer un seul déplacement, puis rechercher un nouveau sous-gradient au point obtenu.

### 6.3.2.1 Algorithme du sous-gradient

L'algorithme du sous-gradient consiste à calculer, pour un certain nombre d'itérations, le minimum (pour un problème de minimisation) de la fonction duale pour un vecteur de multiplicateurs

fixé. À chaque itération, les multiplicateurs sont mis à jour. La solution issue de la résolution de la fonction duale, est une borne inférieure du problème primal. L'objectif est d'obtenir la meilleure borne inférieure (pour un problème de minimisation) ou, mieux encore, la solution optimale du problème primal.

L'algorithme de résolution du problème duale doit être rapide pour effectuer plusieurs itérations. La valeur de la fonction-objectif du problème dual peut parfois osciller, il faut donc mémoriser la meilleure borne inférieure ( $LB_{primal}$ ) trouvée au cours des itérations. La projection de  $\Pi$  consiste à forcer à 0 les composantes négatives. Elle n'est nécessaire que lorsque la contrainte dualisée correspond à une inégalité ( $g_i(x) \leq 0$ ), pour conserver des multiplicateurs positifs.

L'algorithme 6.1 représente la structure générale d'un algorithme de sous-gradient. L'algorithme s'arrête si :

- le nombre maximum d'itérations est atteint ou
- la norme du vecteur des contraintes dualisées devient nulle ou
- la borne inférieure et la borne supérieure deviennent égales.

Les notations utilisées sont les suivantes :

- $\Pi$  : le vecteur de multiplicateurs de Lagrange,
- $\lambda$  : le pas de mise à jour,
- $LB_{primal}$  : la borne inférieure du problème primal ; elle correspond au coût de la solution du problème relaxé,
- $j_{max}$  : le nombre maximum d'itérations avec  $j$  comme l'itération courante,
- $L(\Pi)$  : valeur de la fonction-objectif du problème dual au point obtenu ( $y$ ),
- $f(x)$  : fonction objectif du problème initial (non relaxé),
- $g(x)$  : représente les contraintes dualisées,
- $\gamma(\Pi)$  : vecteur des contraintes dualisées au point obtenu ( $y$ ),
- $\|\gamma(\Pi)\|$  : norme du vecteur des contraintes dualisées au point obtenu ( $y$ ),
- $UB$  : la borne supérieure fournie par une heuristique.

**Algorithme 6.1** Algorithme général de sous-gradient

- 
- 1: initialiser le vecteur de multiplicateurs  $\Pi$  et le pas de mise à jour  $\lambda$
  - 2:  $LB_{\text{primal}} := 0$
  - 3:  $j := 0$
  - 4: **Répéter**
  - 5:    $j := j + 1$
  - 6:   calculer le minimum de la fonction duale en  $\Pi$
  - 7:    $L(\Pi) = f(y) + \Pi \cdot g(y) = \text{Min} f(x) + \Pi \cdot g(x) : x \in S$
  - 8:    $\gamma(\Pi) = g(y)$  est un sous-gradient de  $L$  en  $\Pi$  d'après la propriété
  - 9:   **Si**  $L(\Pi) > LB_{\text{primal}}$  **Alors**
  - 10:      $LB_{\text{primal}} := L(\Pi)$
  - 11:   **Fin Si**
  - 12:   **Si**  $\|\gamma(\Pi)\|$  est non nul **Alors**
  - 13:     calculer le point suivant  $\Pi = \Pi + \lambda \cdot \gamma(\Pi) / \|\gamma(\Pi)\|$
  - 14:     **Si**  $\Pi \in \mathbb{R}^{m+}$  **Alors**
  - 15:       mettre à jour le pas  $\lambda$
  - 16:     **Fin Si**
  - 17:   **Fin Si**
  - 18: **Jusqu'à** ( $\|\gamma(\Pi)\| = 0$ ) ou ( $j = j_{\text{max}}$ ) ou ( $LB_{\text{primal}} = UB$ ).
- 

Il existe plusieurs règles pour choisir le pas  $\lambda$ .

**6.3.2.2 Règle 1 : Méthode à pas constant (R1)**

C'est la méthode la plus simple : les  $\lambda_j$  sont tous égaux à une constante positive  $\lambda_0$ . Held et Karp (1971) ont résolu pour la première fois des problèmes de voyageur de commerce à 64 villes avec une relaxation Lagrangienne, résolue par une méthode de sous-gradient à pas constant  $\lambda_0 = 1$ .

**6.3.2.3 Règle 2 : Règle de la série divergente (R2)**

Polyak (1967) a montré que  $L(\Pi)$  converge vers l'optimum  $L(\Pi^*)$  sous les seules conditions pour la suite des  $\lambda$  (règle de la série divergente) :

$$\lambda_j \rightarrow 0 \ (j \rightarrow +\infty) \text{ et } \sum_{j=1}^{\infty} \lambda_j = +\infty$$

Un exemple simple est  $\lambda_j = 1/j$  avec  $j$  comme l'itération courante. La convergence avec cette règle peut être très lente.

Les deux méthodes suivantes marchent bien en pratique, même si elles font parfois converger  $L(\Pi)$  vers un point non maximal.

### 6.3.2.4 Règle 3 : Règle de la série convergente (R3) (Shor (1968) et Goffin (1977))

On a alors  $\lambda_j = \sigma \cdot \alpha^j$ , avec  $\sigma$  une constante positive et  $0 < \alpha < 1$ . Pour éviter une convergence vers un point non-optimal, on a intérêt à prendre  $\alpha$  assez proche de 1 (0,9 par exemple) et  $\sigma$  assez grand. Gondran et Minoux (1979) signale que  $\alpha = 0,87$  donne de bons résultats sur de nombreux problèmes.

### 6.3.2.5 Règle 4 : Méthode de relaxation (R4) (Polyak (1969) et Held *et al.* (1974))

$$\lambda_j = \rho_j \frac{L(\Pi^*) - L(\Pi)}{\|\gamma(\Pi)\|} \text{ avec } \rho_j \leq 2 \text{ (appelé coefficient de relaxation)}$$

C'est la technique la plus utilisée. Polyak (1969) a montré que sa convergence est géométrique mais en pratique on ne connaît pas  $L(\Pi^*)$ . En fait, elle marche bien en remplaçant  $L(\Pi^*)$  par une borne supérieure UB, obtenue par une heuristique par exemple. Held *et al.* (1974) ont également montré empiriquement sur plusieurs problèmes qu'on a de bons résultats en partant de  $\rho_0 = 2$  et en divisant ce coefficient de relaxation par 2 après un certain nombre d'itérations, par exemple 5 itérations. Certains auteurs font plusieurs cycles de ce type, par exemple en remettant  $\rho$  à 2 toutes les 50 itérations.

## 6.4 Heuristique Lagrangienne

Nous proposons une heuristique Lagrangienne qui utilise la relaxation Lagrangienne, à laquelle une réparation est ajoutée pour générer une suite de solutions réalisables, de coûts décroissants.

### 6.4.1 Algorithme de l'heuristique Lagrangienne

L'Algorithme 6.2 décrit l'heuristique Lagrangienne, avec les notations suivantes :

- $\Pi$  : le vecteur de multiplicateurs de Lagrange,
- $\lambda$  : le pas de mise à jour,
- $LB_{primal}$  : la meilleure borne inférieure du problème primal ; elle correspond à la valeur la plus élevée de la fonction-objectif du problème dual,
- $j_{max}$  : le nombre maximum d'itérations avec  $j$  comme étant le numéro de l'itération courante,
- $L(\Pi)$  : valeur de la fonction-objectif du problème dual,
- $f(x)$  : fonction objectif du problème initial (non relaxé),
- $g(x)$  : représente les contraintes dualisées,
- $\gamma(\Pi)$  : vecteur des contraintes dualisées au point obtenu ( $y$ ),
- $\|\gamma(\Pi)\|$  : norme du vecteur des contraintes dualisées au point obtenu ( $y$ ),
- $Tc$  : le temps de calcul défini par l'utilisateur,
- $Te$  : le temps d'exécution écoulé depuis le début de la procédure,

- *Pre* : la précision correspond à l'écart entre les bornes inférieure et supérieure,
- $HRe(\Pi)$  : coût de la solution après l'exécution de l'heuristique de réparation,  $HRe(\Pi^*)$  représente le coût de la meilleure solution trouvée,
- *PasViolation* : est à vrai si aucune violation des contraintes dualisées n'est détectée.

$$\text{Écart}(HRe/LB_{\text{primal}}) = \frac{HRe(\Pi^*) - LB_{\text{primal}}}{LB_{\text{primal}}} \times 100$$

L'heuristique Lagrangienne commence par la construction du modèle en initialisant les variables et les multiplicateurs de Lagrange (mis à 0) et en construisant les contraintes non relaxées et la fonction objectif. Les critères d'arrêt et les paramètres, tels que le nombre maximum d'itérations, le temps alloué au calcul et la précision, sont initialisés. À chaque itération, le modèle est résolu avec CPLEX et on sauvegarde le coût de la solution la plus élevée, qui correspond à la meilleure borne inférieure du problème primal. S'il y a des contraintes violées alors l'heuristique de réparation est exécutée pour obtenir une solution réalisable. On sauvegarde la meilleure solution issue de l'heuristique de réparation (coût le plus faible). Les multiplicateurs de Lagrange sont mis à jour en utilisant le pas, qui est également mis à jour avec la règle choisie par l'utilisateur (pas constant, règle de la série divergente, règle de la série convergente ou méthode de relaxation).

Quatre critères d'arrêt sont utilisés pour l'algorithme de l'heuristique Lagrangienne. Le premier est le nombre maximum d'itérations. Le deuxième critère d'arrêt correspond au temps de calcul défini par l'utilisateur, qui ne doit pas être dépassé. Une valeur booléenne (*PasViolation*) est utilisée comme troisième critère d'arrêt. Elle prend la valeur VRAI, si aucune violation des contraintes dualisées n'est détectée; dans ce cas l'algorithme s'arrête. Le pourcentage d'écart ( $\text{Écart}(HRe/LB_{\text{primal}})$ ), entre la meilleure solution trouvée par l'heuristique de réparation et la meilleure solution du PL mixte relaxé, est le dernier critère d'arrêt. L'algorithme s'arrête, si cet écart est inférieur ou égal à la précision, qui correspond à l'écart entre les bornes inférieure et supérieure. La précision, paramétrée à 0,01%, est utilisée comme critère d'arrêt pour CPLEX.

## 6.4.2 Heuristique de réparation

À chaque itération de l'heuristique Lagrangienne, les valeurs des variables de la solution fournie par CPLEX sont extraites. Une vérification est faite sur la violation des contraintes dualisées. Lorsqu'une violation est détectée, le noeud (produit-entrepôt ou produit-dépôt ou produit-entrepôt-dépôt) et les périodes concernées, sont relevés. Un coût est calculé, selon le type de violation, pour chaque noeud présentant une violation. Une heuristique permet de corriger les violations détectées.

La phase la plus délicate est la copie des valeurs des variables, pour la construction des listes des lots associés à chaque produit-entrepôt ou produit-dépôt. Il faut aussi s'assurer que toutes les contraintes soient respectées, après l'application de l'heuristique de réparation. Au cas contraire, la solution fournie n'est pas réalisable.

---

**Algorithme 6.2** Algorithme de l'heuristique Lagrangienne
 

---

- 1: Initialiser les variables du modèle
  - 2: Initialiser les multiplicateurs de Lagrange à 0
  - 3: Construire les contraintes non relaxées et la fonction objectif
  - 4: Renseigner les paramètres :  $j_{max}, Tc, Pre$
  - 5:  $HRe(\Pi^*) = \infty$
  - 6:  $LB_{primal} := 0$
  - 7:  $j := 1$
  - 8:  $Te := 0$
  - 9:  $PasViolation := faux$
  - 10:  $\acute{E}cart(HRe/LB_{primal}) := \infty$
  - 11: **Tant que** ( $j \leq j_{max}$  et  $Te < Tc$  et  $PasViolation == faux$  et  $\acute{E}cart(HRe/LB_{primal}) > Pre$ )
    - Faire**
    - 12: Calculer le minimum de la fonction duale en  $\Pi$
    - 13:  $L(\Pi) = f(y) + \Pi.g(y) = Min f(x) + \Pi.g(x) : x \in S$
    - 14: **Si**  $L(\Pi) > LB_{primal}$  **Alors**
    - 15:      $LB_{primal} := L(\Pi)$
    - 16: **Fin Si**
    - 17: Récupérer la valeur des variables de la solution fournie par CPLEX
    - 18:  $PasViolation :=$  détecter des violations
    - 19: **Si** Il y a une violation des contraintes dualisées **Alors**
    - 20:     Exécuter l'heuristique de réparation
    - 21:     **Si**  $HRe(\Pi) < HRe(\Pi^*)$  **Alors**
    - 22:          $HRe(\Pi^*) := HRe(\Pi)$
    - 23:     **Fin Si**
    - 24:     Calculer  $\acute{E}cart(HRe/LB_{primal})$
    - 25: **Fin Si**
    - 26: Mettre à jour le temps d'exécution  $Te$
    - 27:  $j := j + 1$
    - 28: **Si**  $j < j_{max}$  et  $Te < Tc$  et  $PasViolation == faux$  **Alors**
    - 29:     Mettre à jour les multiplicateurs de Lagrange
    - 30:     Mettre à jour le pas avec la règle choisie
    - 31: **Fin Si**
  - 32: **Fin Tant que.**
-

L'algorithme 6.3 décrit l'heuristique de réparation. L'heuristique de réparation se déroule en deux phases, si nécessaire. Lors de la première phase, on évalue deux mouvements : la diminution et l'augmentation de la quantité (en excès ou en manque) pour éviter la violation de la contrainte, qui est relaxée. Dans la plupart des cas, on arrive à la corriger. Mais il est parfois impossible d'effectuer ces deux mouvements ; la deuxième phase concerne ces cas. On enlève alors la totalité de la quantité proposée à la période concernée, par CPLEX.

## 6.5 Dualisation de la contrainte de taille de lot

Les résultats des tests préliminaires ont montré que la relaxation de la contrainte de taille de lot est la plus pertinente pour les instances générées aléatoirement. Lorsque CPLEX résout le PL mixte (à l'optimum) avec la contrainte de taille de lot relaxée, il consomme 6,07% de la durée utilisée pour le problème primal.

La contrainte de taille de lot assure que la quantité en transit correspond à  $n$  fois la taille de lot de transport ; s'il n'y a pas de quantité imposée. Elle est modélisée sous la forme d'une égalité :

$$qtr_{klpt} = n_{klpt} \cdot TL_{klp} \text{ si } A_{klpt} = 0, TL_{klp} \neq 0 \quad \forall k \quad \forall l \quad \forall p \quad \forall t$$

Les notations sont les suivantes :

$qtr_{klpt}$ ,  $k \in K$ ,  $l \in L$ ,  $p \in P$ ,  $t \in T$  : quantité de produits  $p$  expédiée sur l'arc  $(k, l)$  à la période  $t$  (la période de référence est la période d'expédition),

$n_{klpt} \in \mathbb{N}$ ,  $k \in K$ ,  $l \in L$ ,  $p \in P$ ,  $t \in T$  : nombre de lots qui constituent la quantité transportée sur l'arc  $(k, l)$  pour un départ à la période  $t$ ,

$TL_{klp} \in \mathbb{N}$  : taille de lot de transport du produit  $p$  sur l'arc  $(k, l)$  en unités de produits, normalement c'est un multiple de palettes (pas de taille de lot si la valeur est 0),

$A_{klpt} \in \{0, 1\}$  :  $A_{klpt} = 1$  s'il y a une quantité de produits  $p$  imposée sur l'arc  $(k, l)$  à la période  $t$  ( $FPOA_{klpt}$  doit être pris en compte même si la valeur est nulle) et dans ce cas la taille de lot et la quantité minimale sont ignorées ( $A_{klpt}$  a la priorité sur  $CTA_{klpt}$ ),

$FPOA_{klpt} \in \mathbb{R}^+$  : quantité d'approvisionnement du produit  $p$  imposée à transporter sur l'arc  $(k, l)$  à la période  $t$  (utile pour imposer une origine et une destination).

Pour dualiser une contrainte d'égalité, deux méthodes existent. La première consiste à conserver l'égalité, en pénalisant la contrainte dans la fonction-objectif avec des multiplicateurs non contraints en signe. La deuxième méthode consiste à transformer l'égalité en deux inégalités. L'inégalité qui est sous la forme  $gi(x) \leq 0$ , est pénalisée dans la fonction-objectif avec des multiplicateurs positifs. L'autre inégalité est intégrer dans les contraintes du modèle. Nous testons ces deux types de dualisation pour la contrainte de taille de lot.

Le multiplicateur de Lagrange est noté  $LMTR_{klpt}$ . Il est associée à la contrainte de taille de lot de transport du produit  $p$  expédié sur l'arc  $(k, l)$  pour un départ à la période  $t$ . La fonction de

---

**Algorithme 6.3** Algorithme de l'heuristique de réparation
 

---

```

1:  $Sol_{Meilleure} := Sol_{CPLX}$ 
2: Détecter les noeuds et les périodes pour lesquels il y a une violation
3: Pour Tous les noeuds détectés avec une violation Faire
4:   Pour Toutes les périodes associées à une violation pour le noeud courant Faire
5:      $Coût(Sol_{Diminution}) := +\infty$ 
6:      $Coût(Sol_{Augmentation}) := +\infty$ 
7:      $Sol_{Diminution} :=$  mouvement de diminution quantité en excès
8:      $Sol_{Augmentation} :=$  mouvement d'augmentation de quantité manquante
9:     Si Le mouvement de diminution de quantité est possible Alors
10:       $Coût(Sol_{Diminution}) :=$  mouvement diminution quantité( $Sol_{CPLX}$ )
11:     Fin Si
12:     Si Le mouvement d'augmentation de quantité est possible Alors
13:       $Coût(Sol_{Augmentation}) :=$  mouvement augmentation quantité( $Sol_{CPLX}$ )
14:     Fin Si
15:     Si  $Coût(Sol_{Diminution}) < Coût(Sol_{Augmentation})$  Alors
16:       $Sol_{Meilleure} := Sol_{Diminution}$ 
17:     Sinon
18:       $Sol_{Meilleure} := Sol_{Augmentation}$ 
19:     Fin Si
20:   Fin Pour
21: Fin Pour
22: Détecter les noeuds et les périodes pour lesquels il reste une violation non corrigée
23: Pour Tous les noeuds détectés avec une violation Faire
24:   Pour Toutes les périodes associées à une violation pour le noeud courant Faire
25:     Si Le mouvement de diminution de la totalité de la quantité proposée à la période à laquelle
        la violation est détectée, est possible Alors
26:       $Sol_{Meilleure} := Sol_{Diminution}$ 
27:     Fin Si
28:   Fin Pour
29: Fin Pour

```

---

Lagrange s'écrit de la manière suivante :

$$L = \underbrace{\sum_{k=1}^K \sum_{l=1}^L \sum_{p=1}^P \sum_{t=1}^T LMTR_{klpt} \cdot (n_{klpt} \cdot TL_{klp} - qtr_{klpt})}_{\text{dualisation de la contrainte de taille de lot}}$$

### 6.5.1 Conservation de l'égalité

Le multiplicateur  $LMTR_{klpt}$  n'est pas contraint en signe : il peut être positif ou négatif ( $LMTR_{klpt} \in \mathbb{R}$ ). À la première itération de la procédure de relaxation Lagrangienne, les multiplicateurs sont initialisés à zéro. Lors des itérations suivantes, les composants du vecteur du multiplicateur sont mis à jour.

Des tests sont effectués avec plusieurs itérations de la relaxation Lagrangienne. Dès qu'un composant du multiplicateur devient négatif, CPLEX renvoie le message *MIP infeasible or unbounded* puis *Current MIP best bound is infinite*. La fonction-objectif devient infinie, ce qui signifie que le polyèdre n'est pas borné dans la direction de la fonction-objectif. La variable  $n_{klpt}$  pouvant être infinie, la fonction-objectif devient infinie. Cette variable n'est utilisée que dans la contrainte de taille de lot ( $n_{klpt} \in \mathbb{N}$ ). Nous avons testé les 4 règles de mise à jour et il y a toujours ce problème sur la borne : le pas constant (règle 1), la série divergente (règle 2), la série convergente (règle 3) et la méthode de relaxation (règle 4). La dualisation de la contrainte de taille de lot en conservant l'égalité est abandonnée pour une transformation de l'égalité en deux inégalités

### 6.5.2 Transformation en deux inégalités

La contrainte de taille de lot est transformée en deux inégalités :

$$qtr_{klpt} - n_{klpt} \cdot TL_{klp} \leq 0 \text{ si } A_{klpt} = 0, TL_{klp} \neq 0 \quad \forall k \quad \forall l \quad \forall p \quad \forall t \quad (6.1)$$

$$qtr_{klpt} - n_{klpt} \cdot TL_{klp} \geq 0 \text{ si } A_{klpt} = 0, TL_{klp} \neq 0 \quad \forall k \quad \forall l \quad \forall p \quad \forall t \quad (6.2)$$

La contrainte (1) est ajoutée à la fonction-objectif, avec une pénalisation des multiplicateurs de Lagrange. Les composants du vecteur  $LMTR_{klpt}$  sont toujours positifs ( $LMTR_{klpt} \in \mathbb{R}^+$ ). La contrainte (2) est introduite dans le modèle. Les tests avec cette manière de dualiser ont mieux fonctionné.

## 6.6 Tests de dualisation de la contrainte de taille de lot

La relaxation Lagrangienne est testée avec la dualisation de la contrainte de taille de lot sur les instances les plus difficiles. Ce sont celles pour lesquelles CPLEX n'a pas pu fournir une bonne solution réalisable au bout de 1 heure et demie de temps de calcul. L'heuristique Lagrangienne est

testée sur l'ensemble des instances générées aléatoirement, avec la dualisation de la contrainte de taille de lot.

### 6.6.1 Évaluation des instances non résolues pour le problème primal

Dans les chapitres précédents, un PL mixte est proposé et les instances sont évaluées avec CPLEX. La limite de temps est définie à 1 heure et 30 minutes. L'écart, entre la meilleure borne inférieure (LB) et la solution réalisable (UB), est paramétré à 0,01% comme critère d'arrêt pour CPLEX. Parmi les instances générées aléatoirement, il y en a 11 pour lesquelles CPLEX ne fournit pas une solution acceptable pour le problème primal. L'écart entre la LB et la solution réalisable est très élevé (voir Tableau 6.2). Dans les chapitres précédents, les solutions fournies par l'heuristique et les métaheuristicues implémentées, sont comparées à celles de CPLEX en écartant ces 11 instances.

L'un des objectifs de l'implémentation de l'heuristique Lagrangienne est de pouvoir comparer ces méthodes à une résolution exacte pour ces 11 instances. Elles sont testées en relaxant la contrainte de taille de lot et en fixant les multiplicateurs à 0 ; tous les composants du vecteur  $LMT R_{klpt}$  sont mis à 0. Une seule itération est exécutée pour chacune des instances. Nous cherchons à comparer les bornes fournies par CPLEX pour le problème primal (non relaxé) et le problème relaxé ainsi que leurs temps de calcul. Les notations utilisées sont les mêmes que celles de la section 6.2.

$$\% \text{ Écart UB/LB} = \frac{\text{UB} - \text{LB}}{\text{LB}} \times 100$$

$$\% \text{ Écart UBr/LBr} = \frac{\text{UBr} - \text{LBr}}{\text{LBr}} \times 100$$

$$\% \text{ Ratio LBr/LB} = \frac{\text{LBr}}{\text{LB}} \times 100$$

$$\% \text{ Ratio UBr/UB} = \frac{\text{UBr}}{\text{UB}} \times 100$$

$$\% \text{ Ratio durée (R/NR)} = \frac{\text{Durée problème relaxé}}{\text{Durée problème non relaxé}} \times 100$$

La relaxation de la contrainte de taille de lot permet à CPLEX de résoudre, de manière optimale, les instances. Les écarts entre la meilleure borne inférieure (LBr) et la borne supérieure (UBr) du problème relaxé sont très faibles (voir Tableau 6.3). Ils sont inférieurs ou (presque) égaux au critère d'arrêt défini à 0,01%. La borne inférieure du problème relaxé (LBr) est, en moyenne, à 99,91% de celle du problème d'origine (LB). Les solutions, issues de la relaxation de la contrainte de taille de lot, sont très proches de celles du problème non relaxé (LB). Pour ces 11 instances, très difficiles,

Test	PL mixte				% Écart UB/LB	Durée
	LB		UB			
43	303 093 633 400	6 870 262 008 100			2166,713	5407,54
44	213 643 669 900	3 460 116 189 400			1519,573	5406,35
45	224 559 890 430	3 562 413 786 700			1486,398	5407,26
46	302 807 651 870	5 345 280 923 700			1665,240	5406,87
50	196 934 941 520	5 837 618 236 400			2864,237	5407,88
52	538 350 712 820	1 860 854 726 200			245,658	5407,17
54	759 421 952 610	68 461 664 889 000			8914,970	5418,80
55	3 830 211 511 500	60 274 836 792 000			1473,669	5409,98
56	1 107 128 581 300	7 476 908 355 100			575,342	5407,54
59	805 955 170 770	131 155 598 880 000			16173,312	5432,08
60	30 194 126 741 000	165 925 792 460 000			449,530	5419,57
Moyenne					3412,240	5411,91

TABLE 6.2 – Résultats de CPLEX pour le problème primal

Test	Relaxation de la contrainte de taille de lot (1 itération)				% Ratio LBr/LB	% Ratio UBr/UB	% Ratio durée (R/NR)
	LBr	UBr	% Écart (UBr/LBr)	Durée (s)			
43	302 860 972 862	302 879 827 079	0,006	17,00	99,92	4,41	0,31
44	213 472 472 339	213 493 559 930	0,010	39,00	99,92	6,17	0,72
45	224 377 125 043	224 391 269 783	0,006	17,00	99,92	6,30	0,31
46	302 459 460 877	302 475 160 231	0,005	38,00	99,89	5,66	0,70
50	196 802 059 849	196 817 472 510	0,008	49,00	99,93	3,37	0,91
52	538 156 217 294	538 185 755 243	0,005	41,00	99,96	28,92	0,76
54	758 918 245 802	758 932 091 851	0,002	382,00	99,93	1,11	7,05
55	3 819 805 174 802	3 820 069 985 962	0,007	585,00	99,73	6,34	10,81
56	1 106 675 575 649	1 106 685 658 474	0,001	95,00	99,96	14,80	1,76
59	805 366 175 629	805 452 959 879	0,011	5497,00	99,93	0,61	101,20
60	30 171 860 724 879	30 175 070 964 407	0,011	5443,00	99,93	18,19	100,43
Moyenne					99,91	8,72	20,45

TABLE 6.3 – Résultats de la relaxation de la contrainte de taille de lot

le temps de calcul du problème relaxé consomme, en moyenne, 20,45% de celui de la résolution du problème non relaxé.

L'heuristique de réparation est appliquée sur la solution, du problème relaxé, fournie par CPLEX. Les valeurs du Tableau 6.4 sont obtenues à partir d'une résolution du problème relaxé par CPLEX, suivie de l'heuristique de réparation. Le coût de la solution issue de ces deux phases est noté HRe. Nous nous assurons que ces solutions soient toujours réalisables afin de pouvoir les comparer aux méthodes de résolution, déjà implémentées. Les notations utilisées sont les suivantes :

$$\% \text{ Écart HRe/UBr} = \frac{\text{HRe} - \text{UBr}}{\text{UBr}} \times 100$$

Les coûts des solutions de l'heuristique de réparation sont en moyenne supérieurs à ceux des solutions issues de la résolution du problème relaxé, de seulement 0,6%. Avec une seule itération, la solution réalisable est de bonne qualité. En exécutant plusieurs itérations avec la méthode de sous-gradient, les solutions seront probablement de meilleure qualité. Le temps de calcul moyen sur ces 11 instances, très difficiles, est de 1443,27 secondes pour résoudre le problème relaxé et réparer les violations. La résolution du problème non relaxé a duré en moyenne 5411,91 secondes, en aboutissant à des solutions de très mauvaise qualité.

Les métaheuristiques présentées dans le chapitre précédent, sont comparées à la relaxation de la contrainte de taille de lot suivie de l'heuristique de réparation (HRe). Il s'agit de la descente à voisinage variable (Variable neighborhood descent - VND), de la recherche locale itérative (Iterated local search - ILS) et de la recherche locale itérative à démarrages multiples (Multi-start iterated local search - MS-ILS). Les notations utilisées sont les suivantes :

$$\% \text{ Écart VND/HRe} = \frac{\text{VND} - \text{HRe}}{\text{HRe}} \times 100$$

$$\% \text{ Ratio ILS/HRe} = \frac{\text{ILS} - \text{HRe}}{\text{HRe}} \times 100$$

$$\% \text{ Ratio MS-ILS/HRe} = \frac{\text{MS-ILS} - \text{HRe}}{\text{HRe}} \times 100$$

Sur les 11 instances, aucune des métaheuristiques ne fournit de bonnes solutions. L'écart moyen entre les coûts des solutions de la HRe et ceux des solutions de la VND est de 22,150%. Cet écart est de 21,467% pour l'ILS et de 24,232% pour la MS-ILS. Ces méthodes sont très efficaces sur les autres instances; cela démontre, encore une fois, toute la difficulté de ces 11 instances. La durée moyenne de la HRe (1443,27 secondes) est toujours inférieure à celle de la VND (2909,54 secondes), de l'ILS (5400,00 secondes) et de la MS-ILS (5400,36 secondes).

L'heuristique Lagrangienne avec la relaxation de la contrainte de taille de lot fournit les meilleures solutions pour les instances les plus difficiles. Elle est performante en termes de qualité des solutions et de temps de calcul. Avec la réparation des violations par l'heuristique, les solutions sont toujours réalisables ; ce qui n'est pas négligeable pour l'utilisateur.

Test	Heuristique réparation (HRe)			VND		ILS		MS-ILS		
	Coût solution	Écart HRe/UBr	Durée (s)	Écart VND/HRe	Durée (s)	Écart ILS/HRe	Durée (s)	Écart MS-ILS/HRe	Durée (s)	
43	304 234 681 901	0,447	46	16,102	1 309	15,049	5 400	15,310	5 400	
44	215 723 219 647	1,044	78	16,934	1 193	15,692	5 400	14,647	5 400	
45	225 496 334 300	0,492	52	15,774	1 066	14,110	5 400	14,636	5 400	
46	304 232 806 656	0,581	101	17,138	1 164	16,117	5 400	16,282	5 400	
50	198 042 926 201	0,623	165	19,590	1 626	17,963	5 400	18,593	5 400	
52	538 930 714 746	0,138	193	4,927	807	4,260	5 400	4,677	5 400	
54	763 571 340 916	0,611	920	39,350	5 400	39,350	5 400	60,881	5 401	
55	3 856 679 412 796	0,958	2673	17,025	5 400	17,025	5 400	18,714	5 400	
56	1 108 153 136 682	0,133	485	4,257	3 240	4,018	5 400	4,301	5 400	
59	813 121 778 395	0,952	5629	81,658	5 400	81,658	5 400	84,362	5 402	
60	30 362 512 930 169	0,621	5534	10,900	5 400	10,900	5 400	14,145	5 401	
Moyenne			0,600	1443,27	22,150	2909,55	21,467	5400,00	24,232	5400,36

TABLE 6.4 – Comparaison résultats heuristique réparation, VND, ILS et MS-ILS

### 6.6.2 Évaluation de l'ensemble des instances générées aléatoirement

Les tests sont effectués en utilisant, respectivement, les 4 règles de mise à jour du pas. Les instances générées aléatoirement présentent plus de difficultés, notamment à cause de la contrainte de taille de lot. La dualisation de cette contrainte a donc été évaluée sur ces instances. Le nombre d'itérations est fixé à 30, le temps alloué au calcul de l'heuristique Lagrangienne est de 1 heure et demie (5400 secondes).

Le coût de la solution issue de l'heuristique de réparation est noté  $HRe$ . La borne inférieure du problème primal est notée  $LB_{primal}$ ; elle correspond au coût de la solution du problème relaxé. Le pourcentage d'écart (% Écart  $HRe/LB_{primal}$ ), entre la meilleure solution trouvée par l'heuristique de réparation et la meilleure borne inférieure du problème primal, est calculé à chaque itération. Le ratio ( $LB_{primal}/LB$ ), entre le coût de la solution du problème dual et la borne inférieure fournie par CPLEX pour la résolution du problème primal, est également calculé. La borne inférieure du problème primal est prise en compte à la place de borne supérieure à cause des 11 instances pour lesquelles CPLEX ne fournit pas une solution acceptable pour le problème primal.

$$\% \text{ Écart } HRe/LB_{primal} = \frac{HRe - LB_{primal}}{LB_{primal}} \times 100$$

$$\% \text{ Ratio } LB_{primal}/LB = \frac{LB_{primal}}{LB} \times 100$$

Lorsque la limite de temps ne permet pas à CPLEX de fournir l'optimum, le résultat de cette itération n'est pas pris en compte. Lorsque le calcul s'arrête sans qu'on ait effectué 30 itérations alors la valeur de la dernière itération, qui a abouti, est projetée pour les itérations restantes.

#### 6.6.2.1 Résultats avec la règle 1 : pas constant

$\lambda_j = \lambda_0 = 0,1$  avec  $j$  comme l'itération courante.

Pour les 40 instances testées, aucune n'a abouti à une solution optimale. Par contre les solutions du problème dual (PL mixte relaxé) sont très proches de la borne inférieure de la résolution du problème primal (PL mixte non relaxé). Le ratio moyen le plus bas est de 99,848% et le meilleur est de 99,867% (voir Figure 6.1). L'écart moyen  $HRe/LB_{primal}$  a tendance à décroître tout au long des itérations. Dans la Figure 6.2, il varie entre 0,60% (itération 1) et 0,48% (28ème itération avec 58,2% du temps consommé). On consomme 50% du temps limite défini (5400 secondes) après l'itération 18 (voir Figure 6.3).

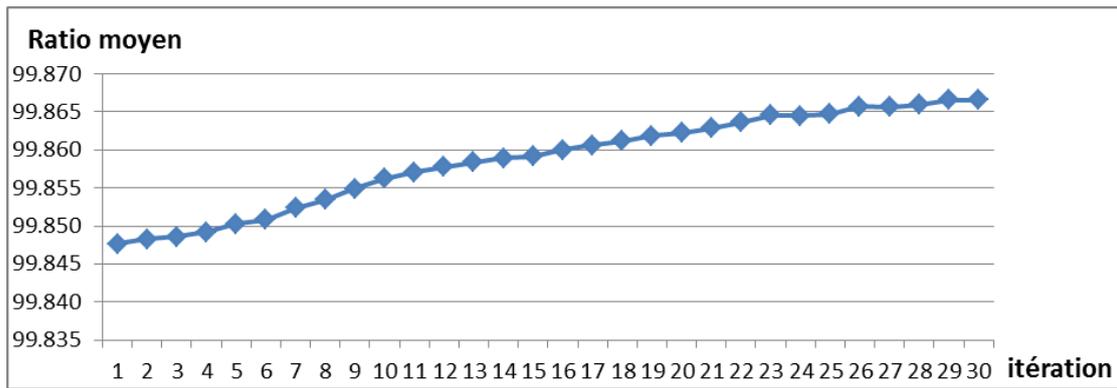
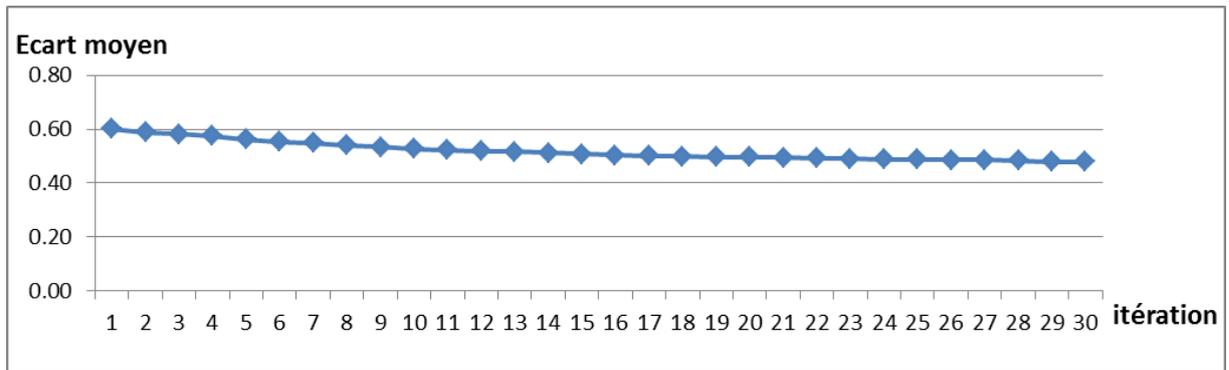
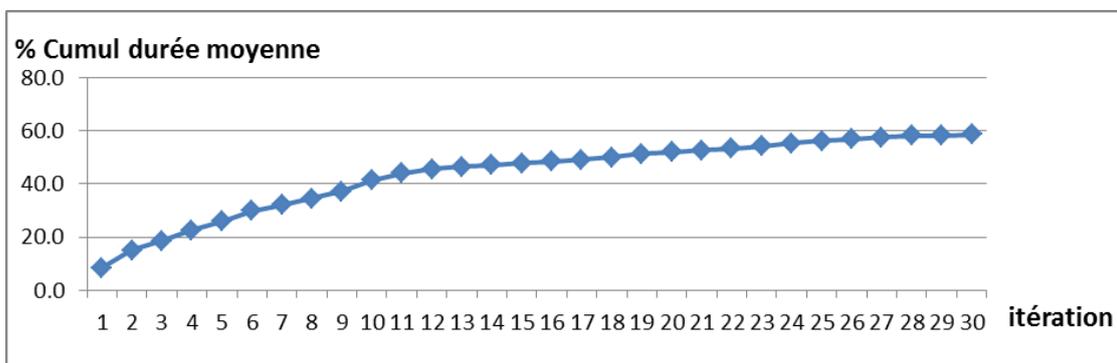
FIGURE 6.1 – Ratio moyen  $LB_{primal}/LB$  : règle 1 utilisée pour la mise à jour du pasFIGURE 6.2 – Écart moyen  $HRe/LB_{primal}$  : règle 1 utilisée pour la mise à jour du pas

FIGURE 6.3 – Pourcentage du cumul de la durée moyenne : règle 1 utilisée pour la mise à jour du pas

### 6.6.2.2 Résultats avec la règle 2 : série divergente

$\lambda_j = 1/j$  avec  $j$  comme l'itération courante.

Pour les 40 instances testées, aucune n'a abouti à une solution optimale. Par contre, les solutions du problème dual sont très proches de celles du problème primal. Le ratio moyen le plus bas est de 99,848% et le meilleur est de 99,870% (voir Figure 6.4). L'écart moyen  $HRe/LB_{primal}$  a tendance à décroître tout au long des itérations (voir Figure 6.5). Il varie entre 0,60% (1ère itération) et 0,46% (25ème itération avec 59,5% du temps consommé). On consomme 50% du temps limite défini (voir Figure 6.6) après la 14ème itération et l'écart moyen  $HRe/LB_{primal}$  est de 0,5%.

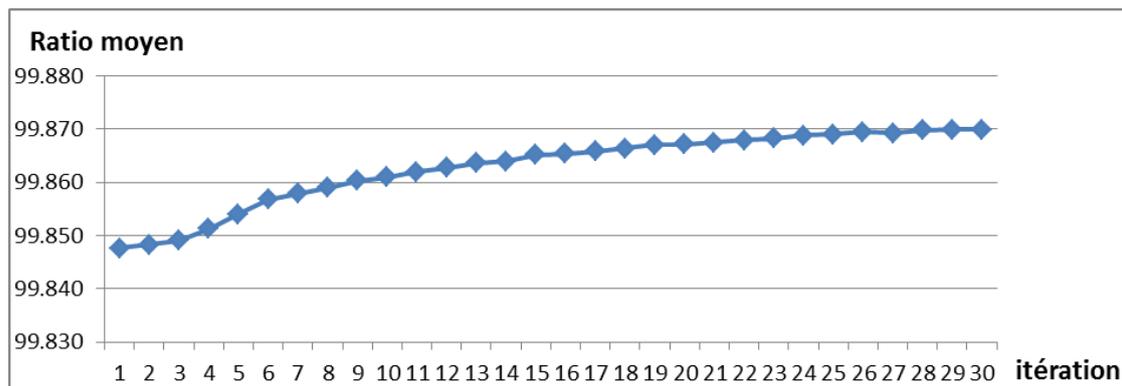


FIGURE 6.4 – Ratio moyen  $LB_{primal}/LB$  : règle 2 utilisée pour la mise à jour du pas

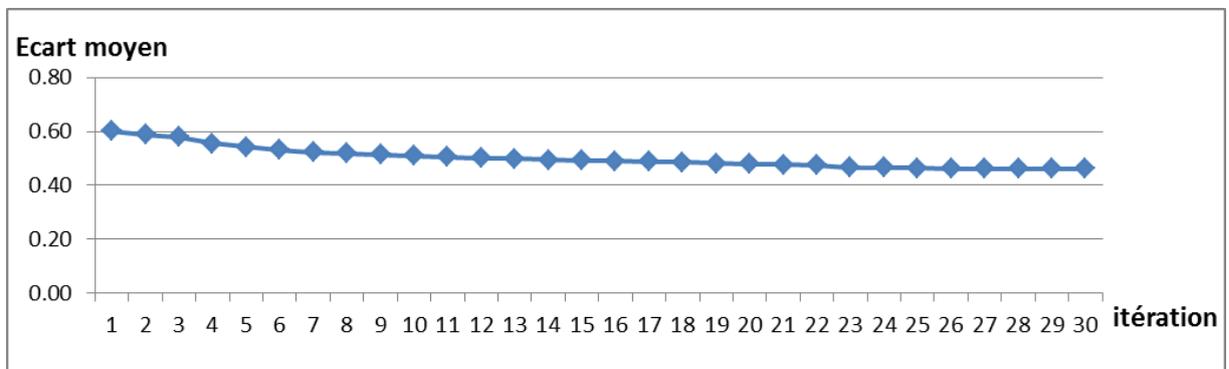


FIGURE 6.5 – Écart moyen  $HRe/LB_{primal}$  : règle 2 utilisée pour la mise à jour du pas

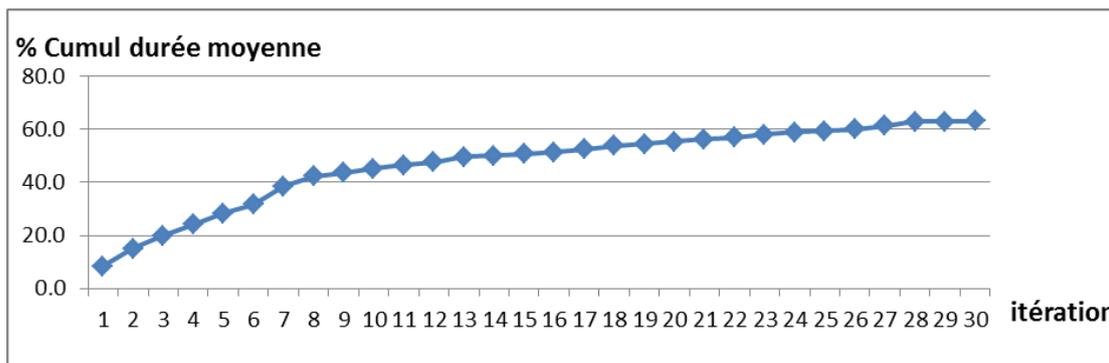


FIGURE 6.6 – Pourcentage du cumul de la durée moyenne : règle 2 utilisée pour la mise à jour du pas

### 6.6.2.3 Résultats avec la règle 3 : série convergente

$$\lambda_j = \sigma \cdot \alpha^j \text{ avec } \sigma = 10\,000 \text{ et } \alpha = 0,87.$$

Pour les 40 instances issues du générateur aléatoire, avec la règle 3 nous obtenons :

- 23 instances pour lesquelles, la solution optimale est trouvée à la troisième itération,
- 3 instances pour lesquelles, la solution optimale est trouvée à la quatrième itération.

Nous avons donc 57,5% des instances pour lesquelles CPLEX trouve la solution optimale au bout de la troisième itération et 65% au bout de la quatrième itération. Au cours des tests, aucune instance n'a tourné au-delà de 4 itérations, soit parce que la solution optimale est trouvée, soit parce que le temps de calcul limite est atteint.

Le ratio moyen le plus bas est de 99,848% et le meilleur est de 100% (voir Figure 6.7). L'écart moyen  $HRe/LB_{primal}$  a tendance à décroître tout au long des itérations (voir Figure 6.8). Il varie entre 0,60% (première itération) et 0,16% (quatrième itération). Par contre la troisième itération consomme beaucoup de temps, soit 73,4% du temps limite défini à 5400 secondes (voir Figure 6.9). Des tests supplémentaires seront effectués en modifiant les paramètres associés à la règle de mise à jour du pas. Par exemple  $\sigma = 100$  ou  $\sigma = 1\,000\,000$  et  $\alpha = 0,5$  ou  $\alpha = 0,95$ , plusieurs combinaisons des valeurs de  $\sigma$  et  $\alpha$  vont être testées.

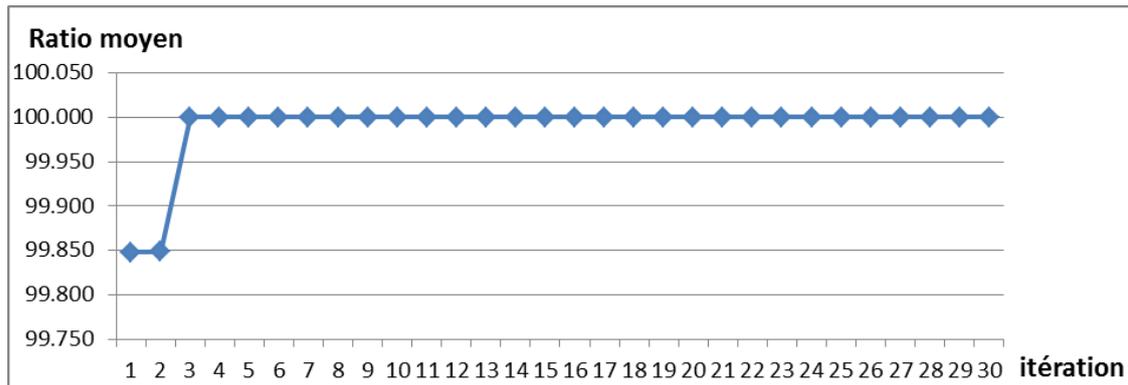


FIGURE 6.7 – Ratio moyen  $LB_{primal}/LB$  : règle 3 utilisée pour la mise à jour du pas

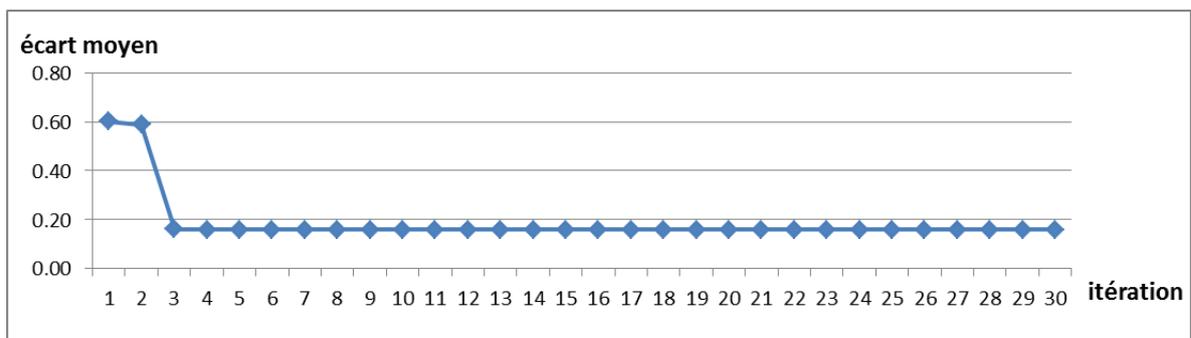


FIGURE 6.8 – Écart moyen  $HRe/LB_{primal}$  : règle 3 utilisée pour la mise à jour du pas

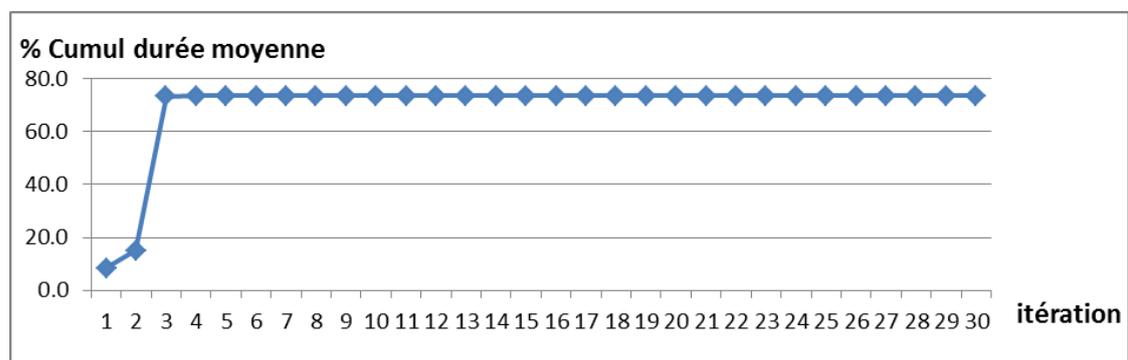


FIGURE 6.9 – Pourcentage du cumul de la durée moyenne : règle 3 utilisée pour la mise à jour du pas

#### 6.6.2.4 Résultats avec la règle 4 : méthode de relaxation

$\lambda_j = \rho_j \frac{L(\Pi^*) - L(\Pi)}{\|\gamma(\Pi)\|}$  avec  $\rho_j \leq 2$  (appelé coefficient de relaxation).

$L(\Pi^*)$  correspond à la meilleure solution trouvée par l'heuristique de réparation.

$\rho_0 = 2$  et après 5 itérations consécutives ce coefficient est divisé par 2.

Avec la règle 4, sur les 40 instances issues du générateur aléatoire, il y a :

- 19 instances pour lesquelles, la solution optimale est trouvée à la troisième itération,
- 8 instances pour lesquelles, la solution optimale est trouvée à la quatrième itération.

Nous avons donc 47,5% des instances pour lesquelles CPLEX trouve la solution optimale au bout de la troisième itération et 67,5% au bout de la quatrième itération. Au cours des tests, aucune instance n'a tourné au-delà de 4 itérations, soit parce que la solution optimale est trouvée, soit parce que le temps de calcul limite est atteint.

Le ratio moyen le plus bas est de 99,848% et le meilleur est de 99,991% (voir Figure 6.10). L'écart moyen  $HRe/LB_{primal}$  varie entre 0,60% pour la première itération et 0,21% dès la quatrième itération (voir Figure 6.11). On remarque le même phénomène sur la consommation du temps que pour la règle 3. La troisième itération consomme beaucoup de temps, soit 75,4% du temps limite défini à 5400 secondes (voir Figure 6.12). Des tests supplémentaires seront effectués en modifiant le coefficient de relaxation initial ( $\rho_0$ ) : par exemple  $\rho_0 = 1$  ou  $\rho_0 = 0,5$ .

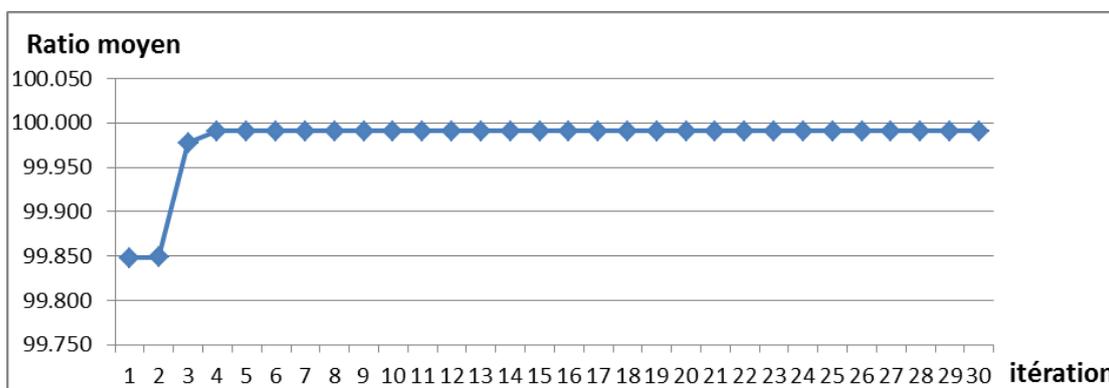


FIGURE 6.10 – Ratio moyen  $LB_{primal}/LB$  : règle 4 utilisée pour la mise à jour du pas

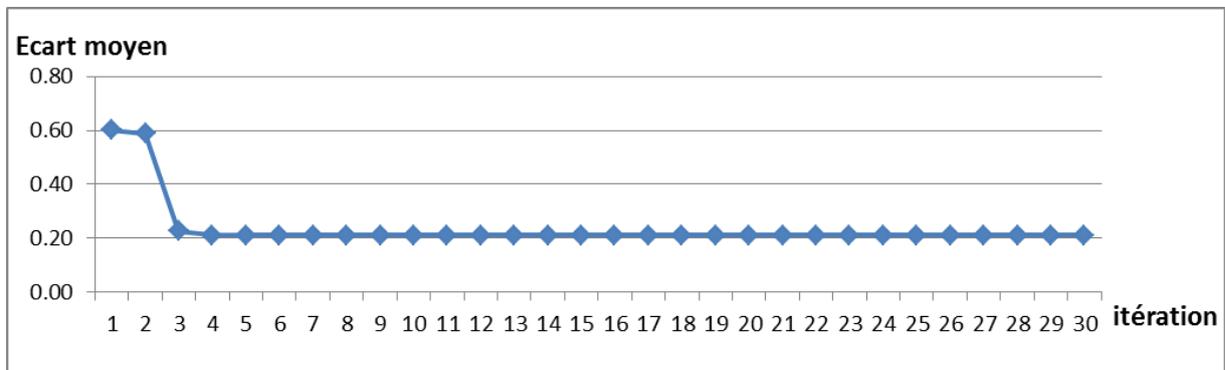


FIGURE 6.11 – Écart moyen  $HRe/LB_{\text{primal}}$  : règle 4 utilisée pour la mise à jour du pas

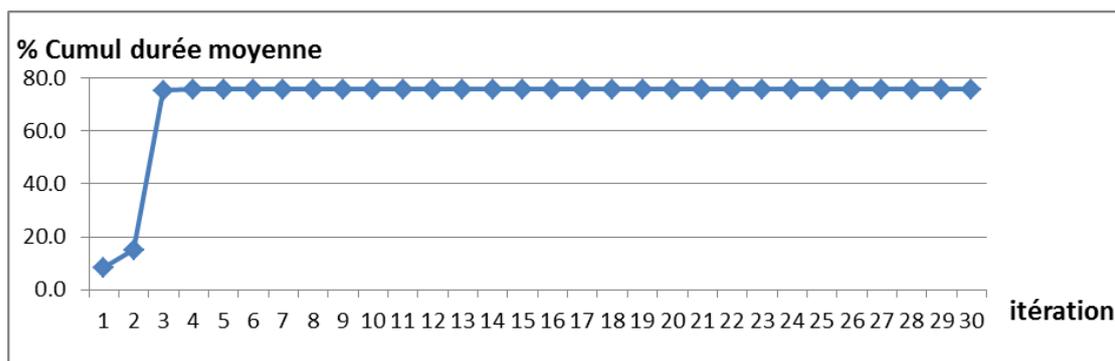


FIGURE 6.12 – Pourcentage du cumul de la durée moyenne : règle 4 utilisée pour la mise à jour du pas

## 6.7 Conclusion

Dans ce chapitre une heuristique Lagrangienne (HRe) est proposée. Elle utilise la relaxation Lagrangienne, à laquelle une réparation est ajoutée pour générer une suite de solutions réalisables, de coûts décroissants.

Les 11 instances pour lesquelles nous n'avons pas pu comparer les solutions des autres méthodes implémentées dans les chapitres précédents, sont évaluées. Sur les 11 instances, aucune des méta-heuristiques ne fournit de bonnes solutions. L'écart moyen entre les coûts des solutions de la HRe et ceux des solutions de la VND est de 22,150%. Cet écart est de 21,467% pour l'ILS et de 24,232% pour la MS-ILS. Ces méthodes sont très efficaces sur les autres instances; cela démontre, encore une fois, toute la difficulté de ces 11 instances. La durée moyenne de la HRe (1443,27 secondes) est toujours inférieure à celle de la VND (2909,54 secondes), de l'ILS (5400,00 secondes) et de la MS-ILS (5400,36 secondes). L'heuristique Lagrangienne avec la relaxation de la contrainte de taille de lot fournit les meilleures solutions pour les instances les plus difficiles.

L'ensemble des instances générées aléatoirement sont testées, avec chacune des 4 règles de mise à jour du pas. Les règles 3 (série convergente) et 4 (méthode de relaxation) ont permis d'obtenir les meilleurs résultats en peu d'itérations. Dès la quatrième itération, pour 65% des instances la solution optimale du problème primal est trouvée, en utilisant la règle de série convergente. Avec la méthode de relaxation, pour 67,5% des instances nous obtenons la solution optimale du problème primal, au bout de la quatrième itération.

Les résultats de ce chapitre ont été présentés à la seizième conférence de la ROADEF (Société Française de Recherche Opérationnelle et d'Aide à la Décision) à l'Université d'Aix-Marseille, en février 2015.

La solution issue de l'heuristique Lagrangienne peut être améliorée en exécutant la méthode VND sur la meilleure solution fournie par l'heuristique de réparation. Le temps de calcul peut également être amélioré en arrêtant l'heuristique de réparation dès que le coût de la solution courante dépasse celui de la meilleure solution trouvée.

Test	Relaxation de la contrainte de capacité de transport (1 itération)				% Ratio LBr/LB	% Ratio UBr/UB	% Ratio durée (R/NR)	
	LBr	UBr	% Écart (UBr/LBr)	Durée (s)				
21	90 322 562 583	90 331 382 713	0,010	65,75	94,91	94,92	164,05	
22	166 940 067 140	166 955 993 420	0,010	8,83	70,74	70,74	1177,33	
23	62 606 551 064	62 608 186 390	0,003	9,89	85,07	85,07	130,65	
24	90 973 205 912	90 976 124 828	0,003	7,80	84,94	84,93	113,87	
25	53 362 065 470	53 365 809 004	0,007	8,35	98,27	98,27	86,62	
26	43 243 297 026	43 247 282 236	0,009	13,53	90,07	90,07	151,00	
27	849 336 066 160	849 413 849 390	0,009	50,16	92,89	92,89	141,57	
28	94 916 361 519	94 925 788 934	0,010	26,30	88,83	88,83	55,21	
29	98 781 404 674	98 790 001 723	0,009	63,85	95,37	95,37	222,71	
30	126 164 419 680	126 175 028 950	0,008	32,06	83,41	83,41	126,47	
31	193 775 173 890	193 791 176 970	0,008	30,05	91,17	91,17	94,20	
32	380 575 688 560	380 601 025 430	0,007	72,76	92,42	92,42	137,88	
33	420 951 419 810	420 986 226 660	0,008	78,39	93,05	93,04	162,87	
34	691 626 040 250	691 694 599 590	0,010	40,42	86,35	86,35	111,26	
35	664 162 655 390	664 225 679 400	0,009	53,68	92,07	92,07	143,61	
36	442 752 234 050	442 796 513 110	0,010	243,61	92,76	92,76	15,93	
37	231 457 804 810	231 480 952 250	0,010	709,66	94,64	94,64	151,75	
38	240 072 822 180	240 096 831 710	0,010	1172,22	99,61	99,61	412,12	
39	149 114 443 940	149 128 953 380	0,010	201,19	98,17	98,17	132,60	
40	204 878 383 300	204 895 853 470	0,009	192,29	85,78	85,78	149,84	
41	665 654 813 500	665 666 391 610	0,002	25,07	64,35	64,35	127,65	
42	79 265 788 871	79 304 943 602	0,049	5403,02	85,02	85,05	99,99	
43	299 333 871 720	299 496 053 340	0,054	5409,99	98,76	4,36	100,05	
44	210 217 290 110	210 339 024 530	0,058	5407,68	98,40	6,08	100,02	
45	220 805 718 580	220 941 716 700	0,062	5407,71	98,33	6,20	100,01	
46	280 497 838 930	280 643 138 620	0,052	5406,10	92,63	5,25	99,99	
47	3 217 604 424 600	3 219 076 211 100	0,046	5405,90	98,33	98,28	99,99	
48	3 990 139 158 400	6 620 297 598 600	65,916	5409,21	96,00	159,16	99,98	
49	5 823 102 843 800	5 828 114 946 100	0,086	5408,91	94,12	94,16	100,05	
50	196 604 534 845	2 584 800 037 812	1214,720	5410,00	99,83	44,28	100,04	
51	411 565 434 070	411 599 255 740	0,008	4894,89	38,31	38,31	1686,50	
52	537 243 933 260	537 441 264 930	0,037	5407,60	99,79	28,88	100,01	
53	3 580 974 458 400	3 581 379 779 700	0,011	5407,31	99,70	87,08	100,02	
54	759 351 586 848	61 377 053 277 592	7982,824	5434,00	99,99	89,65	100,28	
55	3 808 654 852 390	55 929 057 036 617	1368,473	5429,00	99,44	92,79	100,35	
56	1 106 291 674 516	4 624 812 223 301	318,046	5414,00	99,92	61,85	100,12	
57	8 887 309 086 900	8 900 431 295 300	0,148	5407,43	99,44	99,55	100,08	
58	6 931 729 340 726	6 940 576 118 115	0,128	5413,00	99,79	99,81	100,13	
59	805 991 849 824	163 738 337 417 520	20215,136	5461,00	100,00	124,84	100,53	
60	29 572 608 990 510	188 702 867 171 777	538,100	5456,00	97,94	113,73	100,67	
Moyenne des instances pour lesquelles le problème non relaxé est résolu de manière optimale				2,295	1581,225	89,157	90,905	220,549

TABLE 6.5 – Tests sur la relaxation de la contrainte de capacité de transport

Test	Relaxation de la contrainte de quantité minimale de transport (1 itération)				% Ratio LBr/LB	% Ratio UBr/UB	% Ratio durée (R/NR)
	LBr	UBr	% Écart (UBr/LBr)	Durée (s)			
21	94 907 108 699	94 907 865 107	0,001	21,20	99,73	99,73	52,89
22	235 953 045 220	235 967 378 990	0,006	0,65	99,99	99,99	86,67
23	73 569 733 909	73 576 649 014	0,009	3,65	99,96	99,97	48,22
24	107 092 042 140	107 100 662 450	0,008	2,08	99,99	99,99	30,36
25	54 239 339 729	54 240 509 731	0,002	4,84	99,89	99,88	50,21
26	47 992 871 050	47 996 423 550	0,007	4,85	99,96	99,96	54,13
27	909 432 893 240	909 485 839 150	0,006	16,01	99,46	99,46	45,19
28	105 950 311 210	105 950 750 910	0,000	18,39	99,15	99,15	38,60
29	102 865 424 440	102 867 020 620	0,002	15,85	99,31	99,30	55,28
30	149 918 611 220	149 930 068 000	0,008	14,24	99,11	99,11	56,17
31	212 492 799 560	212 503 154 780	0,005	22,70	99,97	99,97	71,16
32	408 897 179 630	408 923 807 460	0,007	5,88	99,30	99,29	11,14
33	444 735 687 120	444 776 451 830	0,009	18,56	98,30	98,30	38,56
34	790 554 897 920	790 607 864 290	0,007	19,67	98,71	98,70	54,14
35	718 827 010 130	718 892 561 470	0,009	14,82	99,64	99,64	39,65
36	473 328 901 370	473 375 974 830	0,010	44,32	99,16	99,17	2,90
37	244 218 930 020	244 221 114 280	0,001	52,74	99,86	99,85	11,28
38	240 875 230 990	240 880 022 830	0,002	58,13	99,94	99,93	20,44
39	151 055 081 950	151 057 125 170	0,001	52,29	99,45	99,44	34,46
40	237 842 415 960	237 847 394 180	0,002	57,92	99,58	99,57	45,13
41	1 033 399 069 000	1 033 470 098 000	0,007	6,47	99,90	99,90	32,94
42	93 137 771 430	93 143 714 545	0,006	326,45	99,90	99,89	6,04
43	302 485 437 720	302 515 417 640	0,010	5130,22	99,80	4,40	94,87
44	213 444 142 660	213 468 073 610	0,011	5410,07	99,91	6,17	100,07
45	224 397 161 880	224 416 083 710	0,008	2897,45	99,93	6,30	53,58
46	302 560 070 540	302 589 041 220	0,010	4957,59	99,92	5,66	91,69
47	3 256 755 552 600	3 257 175 885 900	0,013	5405,70	99,52	99,45	99,99
48	4 130 836 753 100	4 131 040 436 100	0,005	365,99	99,39	99,31	6,76
49	6 124 654 650 700	6 125 009 619 500	0,006	312,31	98,99	98,96	5,78
50	196 679 053 130	196 706 283 410	0,014	5407,42	99,87	3,37	99,99
51	1 072 596 163 100	1 072 636 359 200	0,004	243,32	99,85	99,84	83,83
52	537 192 166 840	537 209 170 240	0,003	238,07	99,78	28,87	4,40
53	3 546 067 280 800	3 546 393 701 000	0,009	255,50	98,73	86,23	4,73
54	758 048 173 950	67 144 068 223 000	8757,494	5426,49	99,82	98,08	100,14
55	3 790 564 882 000	55 714 382 093 000	1369,817	5420,36	98,96	92,43	100,19
56	1 106 586 075 700	1 106 604 009 900	0,002	911,76	99,95	14,80	16,86
57	8 836 728 387 500	8 837 577 220 600	0,010	620,00	98,88	98,84	11,47
58	6 906 456 310 100	6 907 086 274 400	0,009	881,75	99,42	99,33	16,31
59	805 586 271 110	79 963 397 877 000	9826,112	5441,88	99,95	60,97	100,18
60	29 605 746 170 000	127 027 639 850 000	329,064	5428,16	98,05	76,56	100,16
Moyenne des instances pour lesquelles le problème non relaxé est résolu de manière optimale			0,006	305,73	99,48	99,04	38,43

TABLE 6.6 – Tests sur la relaxation de la contrainte de quantité minimale de transport

Test	Relaxation de la contrainte de contrat de fraîcheur (1 itération)				% Ratio LBr/LB	% Ratio UBr/UB	% Ratio durée (R/NR)
	LBr	UBr	% Écart (UBr/LBr)	Durée (s)			
21	94 926 330 865	94 932 671 678	0,007	58,13	99,75	99,75	145,03
22	235 959 742 050	235 977 916 000	0,008	1,28	99,99	99,99	170,67
23	73 596 442 360	73 603 359 702	0,009	10,25	100,00	100,00	135,40
24	107 098 968 150	107 108 671 050	0,009	4,49	99,99	100,00	65,55
25	54 288 653 332	54 293 418 620	0,009	7,08	99,98	99,98	73,44
26	48 009 667 671	48 014 403 859	0,010	17,13	100,00	100,00	191,18
27	914 308 016 270	914 371 853 770	0,007	62,88	99,99	99,99	177,48
28	106 010 455 080	106 020 560 220	0,010	55,72	99,21	99,21	116,96
29	102 898 531 470	102 907 198 980	0,008	41,37	99,34	99,34	144,30
30	150 011 106 730	150 023 891 770	0,009	35,79	99,17	99,18	141,18
31	212 535 343 690	212 545 525 220	0,005	29,09	99,99	99,99	91,19
32	411 785 556 500	411 826 398 820	0,010	219,09	100,00	100,00	415,18
33	449 107 781 810	449 152 639 830	0,010	79,58	99,27	99,27	165,34
34	800 902 950 940	800 982 892 130	0,010	54,79	100,00	100,00	150,81
35	721 391 006 890	721 441 910 460	0,007	61,29	100,00	100,00	163,96
36	474 347 234 210	474 390 504 180	0,009	195,42	99,38	99,38	12,78
37	244 301 462 340	244 325 572 980	0,010	1013,01	99,89	99,89	216,61
38	240 947 930 710	240 972 026 730	0,010	645,47	99,97	99,97	226,93
39	151 125 301 540	151 140 378 300	0,010	201,88	99,49	99,49	133,05
40	237 897 308 770	237 915 056 650	0,007	140,62	99,60	99,60	109,58
41	1 033 630 418 400	1 033 637 447 700	0,001	29,69	99,92	99,92	151,17
42	93 226 536 660	93 241 142 211	0,016	5404,33	100,00	100,00	100,02
43	302 661 818 330	302 826 354 620	0,054	5408,17	99,86	4,41	100,01
44	213 508 392 640	213 752 366 500	0,114	5406,12	99,94	6,18	100,00
45	224 438 561 870	224 566 318 270	0,057	5406,54	99,95	6,30	99,99
46	302 711 415 670	302 869 849 440	0,052	5410,4	99,97	5,67	100,07
47	3 272 098 551 700	5 510 121 121 600	68,397	5409,94	99,99	168,23	100,07
48	4 155 570 923 100	4 159 462 623 300	0,094	5407,95	99,98	100,00	99,96
49	6 186 241 619 700	6 191 069 687 400	0,078	5452,98	99,99	100,03	100,87
50	196 934 970 752	197 671 741 588	0,374	5410	100,00	3,39	100,04
51	1 072 860 340 000	1 072 929 302 500	0,006	235,09	99,87	99,87	81,00
52	537 302 730 690	537 442 266 410	0,026	5406,89	99,81	28,88	99,99
53	3 590 269 423 700	3 590 793 164 200	0,015	5407,26	99,96	87,31	100,02
54	758 988 172 772	62 225 752 580 603	8098,514	5428	99,94	90,89	100,17
55	3 830 464 738 395	54 007 727 296 624	1309,952	5429	100,01	89,60	100,35
56	1 107 143 306 441	5 321 356 743 329	380,638	5414	100,00	71,17	100,12
57	8 935 885 131 100	8 943 663 136 000	0,087	5406,74	99,99	100,03	100,06
58	6 946 457 202 633	6 953 446 826 630	0,101	5415	100,00	100,00	100,16
59	805 844 437 144	101 759 617 514 393	12527,700	5461	99,99	77,59	100,53
60	30 191 148 242 699	153 330 398 293 444	407,865	5461	99,99	92,41	100,76
Moyenne des instances pour lesquelles le problème non relaxé est résolu de manière optimale				2,378 1417,357	99,818	101,738	137,240

TABLE 6.7 – Tests sur la relaxation de la contrainte de contrat de fraîcheur

Test	Relaxation de la contrainte d'ordre imposé (1 itération)				% Ratio LBr/LB	% Ratio UBr/UB	% Ratio durée (R/NR)	
	LBr	UBr	% Écart (UBr/LBr)	Durée (s)				
21	94 926 784 024	94 931 478 440	0,005	57,58	99,75	99,75	143,66	
22	235 959 736 050	235 970 989 450	0,005	1,89	99,99	99,99	252,00	
23	73 595 523 322	73 601 108 233	0,008	23,63	100,00	100,00	312,15	
24	107 104 296 140	107 114 895 750	0,010	6,08	100,00	100,00	88,76	
25	54 288 504 361	54 293 486 411	0,009	6,93	99,98	99,98	71,89	
26	48 009 731 765	48 012 198 273	0,005	14,23	100,00	99,99	158,82	
27	157 910 277 770	157 925 921 360	0,010	56,57	17,27	17,27	159,67	
28	106 012 211 170	106 022 473 840	0,010	51,65	99,21	99,21	108,42	
29	102 897 904 250	102 904 140 840	0,006	37,86	99,34	99,34	132,05	
30	150 010 249 380	150 017 091 770	0,005	33,04	99,17	99,17	130,34	
31	212 535 437 870	212 547 439 450	0,006	29,59	99,99	99,99	92,76	
32	52 844 734 700	52 847 022 294	0,004	26,72	12,83	12,83	50,63	
33	84 526 761 810	84 533 607 448	0,008	28,44	18,68	18,68	59,09	
34	114 641 501 250	114 649 861 680	0,007	25,29	14,31	14,31	69,61	
35	121 608 411 280	121 617 402 450	0,007	41,32	16,86	16,86	110,54	
36	99 543 339 406	99 553 234 958	0,010	38,19	20,85	20,85	2,50	
37	244 298 860 380	244 323 284 250	0,010	622,83	99,89	99,89	133,18	
38	240 947 784 910	240 971 494 260	0,010	418,8	99,97	99,97	147,24	
39	151 125 453 600	151 140 563 250	0,010	221,71	99,49	99,49	146,12	
40	237 898 291 300	237 918 813 670	0,009	175,77	99,60	99,60	136,97	
41	1 033 630 495 200	1 033 637 359 200	0,001	30,05	99,92	99,92	153,00	
42	93 227 449 951	93 241 163 783	0,015	5404,03	100,00	100,00	100,01	
43	302 676 088 910	302 862 841 330	0,062	5407,24	99,86	4,41	99,99	
44	213 515 151 160	213 644 577 020	0,061	5412,78	99,94	6,17	100,12	
45	224 490 140 650	224 640 482 360	0,067	5406,74	99,97	6,31	99,99	
46	302 706 914 600	302 893 714 650	0,062	5407,07	99,97	5,67	100,00	
47	210 498 793 380	210 731 788 700	0,111	5405,92	6,43	6,43	99,99	
48	178 212 088 280	178 371 381 580	0,089	5411,41	4,29	4,29	100,02	
49	264 969 323 340	265 093 811 330	0,047	5407,43	4,28	4,28	100,03	
50	196 943 960 741	2 992 867 147 383	1419,654	5412	100,00	51,27	100,08	
51	1 072 866 856 600	1 072 942 135 400	0,007	231,4	99,87	99,87	79,73	
52	537 296 376 210	537 452 482 820	0,029	5406,59	99,80	28,88	99,99	
53	501 513 353 190	501 579 707 180	0,013	5408,02	13,96	12,20	100,04	
54	759 462 645 156	64 004 371 387 231	8327,587	5432	100,01	93,49	100,24	
55	600 319 083 466	54 044 640 571 722	8902,652	5428	15,67	89,66	100,33	
56	1 107 154 520 394	7 651 849 175 968	591,127	5413	100,00	102,34	100,10	
57	1 015 377 590 100	8 782 677 140 600	764,967	5408,96	11,36	98,23	100,11	
58	1 103 708 079 904	5 514 990 072 399	399,678	5415	15,89	79,31	100,16	
59	805 998 735 568	132 281 286 751 962	16312,096	5462	100,01	100,86	100,55	
60	1 214 153 685 428	152 073 987 694 106	12425,102	5444	4,02	91,65	100,45	
Moyenne des instances pour lesquelles le problème non relaxé est résolu de manière optimale				40,175	1380,701	63,904	69,025	118,603

TABLE 6.8 – Tests sur la relaxation de la contrainte d'ordre imposé

Test	Relaxation de la contrainte de taille de lot (1 itération)				% Ratio LBr/LB	% Ratio UBr/UB	% Ratio durée (R/NR)
	LBr	UBr	% Écart (UBr/LBr)	Durée (s)			
21	94 894 463 992	94 900 489 437	0,006	1,67	99,72	99,72	4,17
22	235 608 014 250	235 608 014 500	0,000	0,22	99,84	99,83	29,33
23	73 493 932 626	73 497 441 849	0,005	0,42	99,86	99,86	5,55
24	107 082 793 240	107 083 812 390	0,001	0,44	99,98	99,97	6,42
25	54 267 693 775	54 272 242 021	0,008	0,50	99,94	99,94	5,19
26	47 976 298 568	47 976 301 611	0,000	0,66	99,93	99,92	7,37
27	912 015 939 670	912 028 394 360	0,001	2,26	99,74	99,74	6,38
28	105 937 146 900	105 945 688 160	0,008	2,17	99,14	99,14	4,55
29	102 834 454 720	102 839 073 080	0,004	1,58	99,28	99,28	5,51
30	149 902 558 080	149 915 889 600	0,009	1,75	99,10	99,10	6,90
31	212 393 844 090	212 404 794 860	0,005	1,37	99,92	99,92	4,29
32	410 579 286 960	410 605 016 000	0,006	10,50	99,70	99,70	19,90
33	446 519 847 710	446 562 392 520	0,010	5,19	98,70	98,70	10,78
34	798 592 969 360	798 599 575 060	0,001	3,76	99,71	99,70	10,35
35	719 218 898 510	719 253 628 650	0,005	3,62	99,70	99,69	9,68
36	472 298 353 920	472 315 055 300	0,004	5,54	98,95	98,94	0,36
37	244 121 901 990	244 142 945 500	0,009	4,85	99,82	99,82	1,04
38	240 852 030 610	240 854 254 130	0,001	4,20	99,93	99,92	1,48
39	151 018 381 780	151 032 840 940	0,010	4,06	99,42	99,42	2,68
40	237 808 426 700	237 813 679 170	0,002	4,15	99,56	99,56	3,23
41	1 030 449 740 900	1 030 531 676 900	0,008	2,28	99,61	99,62	11,61
42	92 932 564 516	92 938 585 458	0,006	5,96	99,68	99,67	0,11
43	302 431 748 170	302 455 723 540	0,008	20,23	99,78	4,40	0,37
44	213 328 122 680	213 346 296 170	0,009	14,35	99,85	6,17	0,27
45	224 296 041 980	224 302 283 520	0,003	66,61	99,88	6,30	1,23
46	302 354 228 320	302 383 740 950	0,010	54,27	99,85	5,66	1,00
47	3 264 111 436 600	3 264 426 389 100	0,010	128,47	99,75	99,67	2,38
48	4 144 301 674 500	4 144 705 783 400	0,010	90,20	99,71	99,64	1,67
49	6 176 268 614 600	6 176 825 664 300	0,009	202,26	99,82	99,80	3,74
50	196 631 724 440	196 644 903 580	0,007	68,08	99,85	3,37	1,26
51	1 070 641 559 400	1 070 657 503 000	0,001	7,91	99,67	99,66	2,73
52	537 105 750 420	537 144 043 460	0,007	15,21	99,77	28,87	0,28
53	3 575 396 221 700	3 575 614 546 900	0,006	72,74	99,55	86,94	1,35
54	757 853 098 420	757 928 762 390	0,010	553,73	99,79	1,11	10,22
55	3 817 424 198 700	3 817 789 902 900	0,010	1569,90	99,67	6,33	29,02
56	1 106 416 224 800	1 106 436 477 700	0,002	27,18	99,94	14,80	0,50
57	8 923 833 712 000	8 924 710 877 300	0,010	175,02	99,85	99,82	3,24
58	6 933 517 841 500	6 933 981 217 400	0,007	212,33	99,81	99,72	3,93
59	805 320 965 190	805 453 916 730	0,017	5484,14	99,92	0,61	100,96
60	30 171 343 003 000	30 174 319 658 000	0,010	5111,23	99,92	18,19	94,31
Moyenne des instances pour lesquelles le problème non relaxé est résolu de manière optimale			0,006	32,97	99,63	99,19	6,07

TABLE 6.9 – Tests sur la relaxation de la contrainte de taille de lot de transport

## Chapitre 7

# Cas d'application : planification de la distribution de produits cosmétiques

### 7.1 Introduction

Cette thèse a été effectuée en collaboration avec l'entreprise FuturMaster. C'est un éditeur de logiciel qui propose un outil APS (*Advanced Planning System*), nommé FuturMaster. Plusieurs modules constituent l'offre de l'entreprise, à savoir : la gestion de budget, la prévision de ventes, le lancement et la promotion de produits, la planification de la production, la planification de la distribution et l'ordonnancement.

Ce chapitre est dédié à l'étude d'un cas d'application industrielle, à laquelle nous avons participé. Il s'agit d'un projet de mise en place de l'APS de FuturMaster. La solution proposée au client comprend un module de prévision de ventes et un autre dédié à la planification de la distribution de produits cosmétiques. Ce dernier module utilise l'heuristique constructive développée dans la thèse. La volumétrie des instances traitées dans ce projet, est l'une des raisons de ce choix. Le réseau de distribution est à l'échelle internationale avec 130 pays à approvisionner. La planification de la distribution est faite sur la base des prévisions hebdomadaires, sur un horizon de 52 semaines.

Le contexte du projet est présenté dans la section 7.2, à travers le périmètre, les objectifs et les enjeux du projet. L'offre de l'APS de FuturMaster est décrite dans la section 7.3. Les principales difficultés rencontrées lors de la mise en œuvre de l'outil, sont détaillées dans la section 7.4. Elles sont liées au recueil du besoin, à l'appropriation de l'outil et l'accompagnement au changement, et à la fiabilité des données. La section 7.5 est dédiée aux apports de l'heuristique constructive. La dernière section 7.6 présente le bilan du projet et des pistes de recherche, pour améliorer les outils d'optimisation destinés à des applications industrielles.

## 7.2 Présentation du contexte

Le projet, nommé DRP Monde (*Distribution Resource Planning*), s'effectue avec une entreprise dans le secteur pharmaceutique et cosmétique. La planification de la distribution concerne, pour le moment, les produits cosmétiques. Les produits pharmaceutiques sont plus contraints, en termes de réglementations et de durée de vie des produits. Le projet DRP Monde inclut les modules de prévision de ventes et de planification de la distribution.

### 7.2.1 Le projet DRP Monde

Les prévisions de ventes sont calculées sur une maille mensuelle sur 18 mois. Les résultats de ces prévisions sont scindés sur une maille hebdomadaire selon des règles de pondération spécifiques au client. La planification de la distribution est faite sur la base des prévisions hebdomadaires, sur un horizon de 52 semaines.

Le réseau de distribution est à l'échelle internationale avec 130 pays à approvisionner. Le réseau de distribution globale est constitué d'entrepôts locaux qui sont approvisionnés par un seul entrepôt central situé en France. Ce dernier assure la réception, le stockage, la préparation, l'expédition en France et à l'international et la gestion des retours de produits. En période de forte activité, l'entrepôt central peut expédier plus de 20 000 colis par jour. Chaque entrepôt local est géré par une filiale ou un partenaire, qui est en charge de la distribution sur son marché local. Par exemple, l'entrepôt local en Espagne mutualise les demandes venant de l'Espagne et du Portugal (voir Figure 7.1).

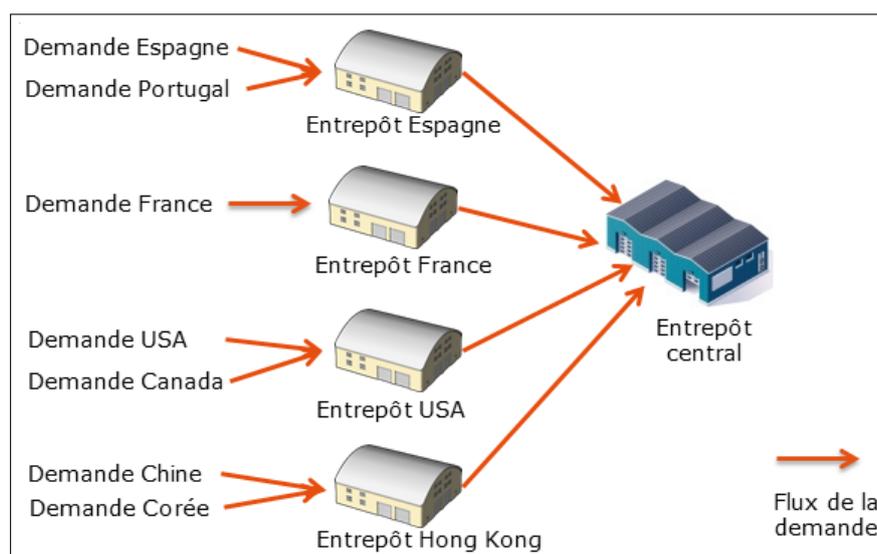


FIGURE 7.1 – Périmètre du projet DRP Monde

## 7.2.2 Les objectifs et les enjeux du projet

La réduction des niveaux des stocks et l'amélioration du taux de service font partie des principaux objectifs de la mise en place de l'APS. Il est donc nécessaire de réduire le biais des prévisions, en responsabilisant et en gratifiant des bons prévisionnistes. La visualisation des ruptures prévisionnelles sur les marchés dans le monde permet de les anticiper. Elle permet aussi d'équilibrer les ruptures entre les différents centres de distribution par un pilotage central (entrepôt central) du déploiement des stocks.

La réactivité, dans l'ajustement aux tendances, engendre une meilleure gestion des lancements de nouveaux produits.

La vision globale du réseau de distribution permet de réduire les coûts. Le lissage (avance et retard) de la charge en distribution est effectué de manière à optimiser les coûts.

Certaines décisions des planificateurs sont assez arbitraires, surtout pour des situations de pénurie. Des entretiens avec les planificateurs nous ont permis de constater des différences de traitement, pour une même situation. La mise en place de l'outil de planification permet de construire des processus de diffusion des *Best Practices* (bonnes pratiques) et de mutualisation des pratiques des filiales.

## 7.3 Présentation de FuturMaster

### 7.3.1 Définition et rôle d'un APS

Les APS (*Advanced Planning System*) sont des applications, qui ont pour objet l'optimisation de la chaîne logistique, depuis la prévision de la demande jusqu'au transport final, en passant par la planification de la production et des approvisionnements (voir Figure 7.2).

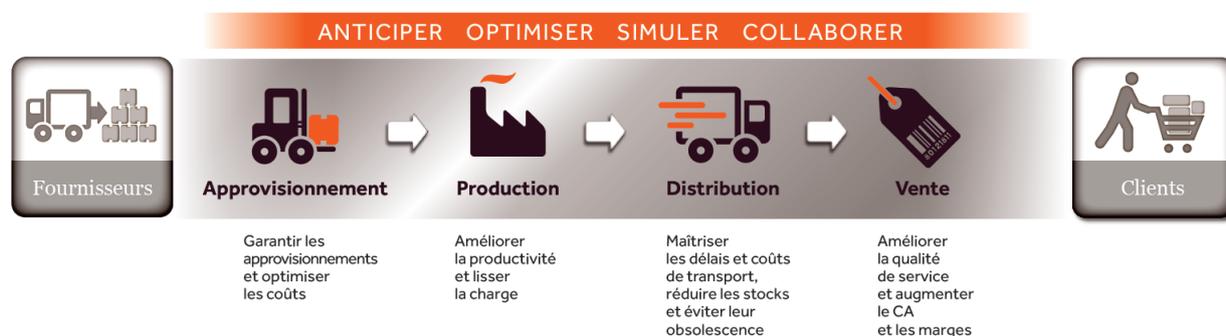


FIGURE 7.2 – Couverture fonctionnelle des APS

Les méthodes traditionnelles de planification fonctionnent par étape, pour affecter les moyens et les ressources. L'inconvénient majeur de ces méthodes est leur rigidité face aux évolutions de la

demande, des capacités et de la disponibilité des ressources. Elles ne prennent pas en compte les contraintes de type : limite de capacité et indisponibilités des produits.

Les APS organisent la production et/ou la distribution en fonction de ces contraintes de disponibilité et de capacité, pour adapter l'utilisation des ressources et définir des plans de production et/ou de distribution. L'optimisation est donc globale, en jouant sur l'ensemble de ces domaines, plutôt que de chercher à optimiser chaque processus de manière indépendante. Les APS mettent en œuvre des techniques de simulation (modèles statistiques) et des méthodes d'optimisation issues de la recherche opérationnelle (modèles mathématiques résolus avec un solveur, théorie des graphes, programmation sous contraintes, heuristiques, etc).

### 7.3.2 Quelques mots sur la société

FuturMaster SAS est un éditeur de logiciel spécialisé dans les solutions de gestion de la chaîne logistique, à destination des industriels et des acteurs de la distribution, pour des produits de grande consommation, de l'agroalimentaire, de beauté et de santé.



FIGURE 7.3 – Les activités de FuturMaster

FuturMaster est un acteur dominant dans le domaine de la *Supply Chain* depuis plus de 20 ans. Sa vocation est de relever les enjeux des organisations, en fournissant à leurs départements de *Supply Chain*, des solutions métier performantes. Les enjeux sont multiples :

- forte croissance du chiffre d'affaire et développement à l'international,
- réduction de coûts et des investissements,
- rentabilité.

C'est dans ce contexte que FuturMaster aide ses clients à relever ces multiples défis, leur permettant d'obtenir des retours sur investissement (ROI-*Return On Investment*) conséquents, notamment,

grâce à une réduction des niveaux des stocks, une baisse des pénalités et des dégagements (perte due à l'obsolescence des produits), un meilleur lissage des activités des usines permettant une optimisation des coûts du personnel et dans certains cas, le report de construction d'une usine.

Les solutions FuturMaster sont aujourd'hui déployées au sein de plus de 500 groupes clients, présents dans 50 pays, représentant 3 500 licences déployées et environ 10 000 utilisateurs.

### 7.3.3 Les solutions standards FuturMaster

FuturMaster est un outil flexible, entièrement paramétrable, configuré pour répondre aux besoins opérationnels des clients. L'offre modulaire (voir Figure 7.4) permet de construire une solution sur-mesure, afin de mettre en œuvre des solutions simples d'utilisation, personnalisées pour les utilisateurs, et à même de fournir des niveaux d'efficacité et de performance optimaux.



FIGURE 7.4 – L'offre FuturMaster

Le module de gestion de la demande (*Demand Management*) permet de construire une prévision de ventes, basée sur les données historiques et utilisant des modèles statistiques tels que la régression linéaire, la régression polynomiale et le double lissage exponentielle. On décompose dans ce processus la tendance et la saisonnalité de la vente des produits. Les événements exceptionnels, tels que les ruptures de stock et les promotions, sont également intégrés. Deux modules spécifiques permettent la gestion des promotions et des lancements de produits.

Les modules de gestion de l'approvisionnement (*Procurement*), de planification de la production (*Production Planning*) et de planification de la distribution (*Distribution Planning*) permettent de calculer, à partir de la prévision de ventes, des plans d'approvisionnement, de production et de distribution. Pour traiter ces problématiques, FuturMaster dispose d'un algorithme de « juste à temps » et d'un modèle de programmation linéaire mixte résolu avec le solveur CPLEX.

L'algorithme de « juste à temps » permet de calculer des ordres de réapprovisionnement, le long de la chaîne logistique. Ce calcul prend en compte toutes les contraintes locales telles que les délais d'approvisionnement, la taille de lot d'approvisionnement, la quantité minimale d'approvisionnement et les calendriers. Les besoins sont remontés sur les niveaux en amont du flux, en appliquant

les coefficients de nomenclature. Cette approche ne permet pas de gérer des contraintes de capacité et ne vérifie pas la disponibilité des produits sur les niveaux en amont du flux.

Le modèle de programmation linéaire mixte est un modèle générique. Il permet de représenter toutes les contraintes et détermine un plan cohérent sur l'ensemble du réseau logistique : prise en compte des délais à tous les niveaux, de la disponibilité des produits, des capacités, des contraintes d'approvisionnement ou de production. La fonction-objectif à minimiser correspond à la somme des coûts associés au plan : pénalités lors de la non réponse à la demande ou retard sur le niveau de stock objectif, coûts de production, coûts de stockage, coûts de transports.

## 7.4 Mise en œuvre d'un outil d'optimisation : de la théorie à la pratique

La mise en œuvre d'un outil d'optimisation comme FuturMaster, représente toujours un véritable challenge. Outre la complexité intrinsèque des problématiques de planification que l'outil doit résoudre en un temps raisonnable, nous rencontrons de nombreuses difficultés, tout au long des phases du projet.

### 7.4.1 Le recueil du besoin

La première difficulté réside dans la modélisation de la problématique. La première phase d'un projet de mise en œuvre d'un outil d'optimisation, consiste en un audit du besoin auprès du client. Le rôle du consultant et/ou de l'expert est primordial, car il est toujours très difficile de repérer les besoins essentiels, et de traduire des expressions métier en règles ou formules mathématiques.

Certains clients ont recours à des méthodes simplificatrices, pour appréhender et résoudre leurs problématiques de planification. Ils mettent en place des règles de gestion ou des stratégies, permettant de déterminer un plan de façon efficace. Ces règles sont basées sur l'expérience des planificateurs, et fournissent en général de bons résultats. Cependant, elles correspondent souvent à une stratégie valable à un certain moment, pour une situation particulière. Elles peuvent se révéler caduques ou inadaptées à l'évolution du contexte. De plus, ces règles qui sont en général sous-optimales, peuvent présenter une difficulté de modélisation ou rendre le problème plus difficile à résoudre.

Par exemple, un fabricant de yaourt explique au consultant, lors de l'audit, que le plan doit toujours proposer les yaourts à la fraise, le mardi. Le consultant doit alors être capable de comprendre l'origine de cette demande. S'agit-il d'une véritable contrainte (réglementation, contrainte industrielle, etc.) ou d'une habitude permettant de simplifier le travail de planification et de production ?

Comme second exemple, le fabricant de yaourt demande que le plan calculé par l'outil, soit en palettes complètes. La modélisation de ce type de contraintes nécessite des variables en nombres entiers et peut donc rendre le calcul du plan très difficile. En réalité, en demandant l'historique des

données de production, on s'aperçoit que les yaourts ne peuvent jamais être produits en palettes complètes ; puisque la ligne de conditionnement ne peut pas être arrêtée après un nombre de yaourt exact. Cette contrainte correspond donc d'avantage à un besoin de lisibilité/confort de visualisation du plan (possible par un simple arrondi), qu'à une véritable contrainte indispensable à l'obtention d'un plan de production réalisable.

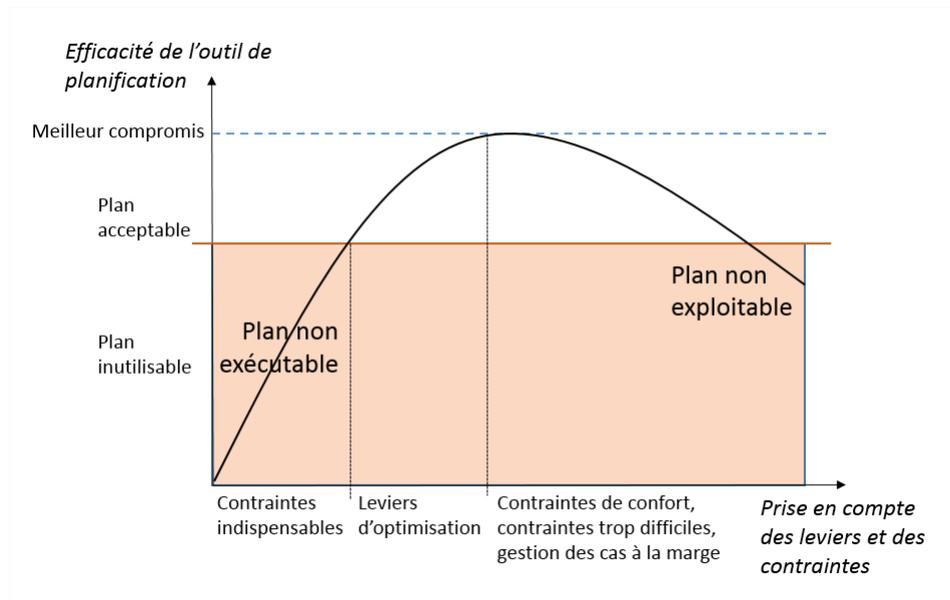


FIGURE 7.5 – Efficacité de l'outil de planification

Par ces exemples, on comprend l'importance de cette phase d'audit. Le consultant doit être capable de distinguer les contraintes (indispensables à l'obtention d'un plan exécutable), les leviers d'optimisation (permettant de rendre le plan meilleur), et les règles ou stratégies moins essentielles pouvant ou devant être abandonnées. Sans cela, on risque de modéliser une problématique trop lourde, qui à la fois pourra s'avérer trop difficile à résoudre, mais qui en plus n'aboutira pas forcément à un plan satisfaisant. Si l'on cherche à représenter trop de choses dans l'outil, les règles de gestion et stratégies principales risquent d'être noyées. L'outil deviendra finalement inexploitable avec trop de leviers à manipuler, des résultats difficiles à comprendre et un temps de calcul trop important.

La recommandation est donc d'appliquer le vieil adage « le mieux est l'ennemi du bien » : il faut trouver le bon compromis entre la complexité de la problématique et du pilotage de l'outil, et la qualité « ressentie » du plan (voir Figure 7.5).

#### 7.4.2 Appropriation de l'outil et accompagnement au changement

L'abandon ou la modification des règles ou des processus en place est une source importante de stress pour les futurs utilisateurs de l'outil. La mise en place d'un outil d'optimisation basé sur des approches complexes, de type modèles mathématiques résolus par un solveur ou métaheuristiques, représente souvent un changement conséquent. Il est donc indispensable d'accompagner les

planificateurs, tout au long du projet, afin qu'ils s'approprient au mieux ce nouvel outil.

Il s'agit, tout d'abord, d'initier les futurs utilisateurs à la complexité et à la modélisation mathématique (notions de flux, de contraintes et de coûts). Les planificateurs n'étant généralement pas des spécialistes en recherche opérationnelle, on ne cherche pas à donner des éléments trop théoriques mais simplement à faire sentir le comportement de l'outil face à des situations bien concrètes.

Lors de la phase de conception générale, les experts et consultants font participer les futurs utilisateurs à la construction du modèle de données, et à la définition des contraintes et des règles de gestion (coûts, pénalités). On commence par considérer une problématique simple, qu'on rendra de plus en plus complexe au fur et à mesure des ateliers de travail, en travaillant directement avec l'outil sur des exemples. Il est ainsi possible de faire comprendre de façon assez intuitive le comportement de la méthode de résolution et les moyens de pilotage du plan, et on évite l'effet « boîte noire » souvent redouté avec ce type de méthodes.

Une fois la partie de mise en œuvre, plus technique de l'outil, effectuée, on entame une phase de recette. Elle permet de s'assurer que l'outil donne entièrement satisfaction. On constate, en général, au démarrage de l'utilisation opérationnelle de l'outil, que les utilisateurs passent beaucoup de temps à vérifier que la méthode fournit bien une solution cohérente (cohérence du flux entre les niveaux, volumes de stocks projetés, quantités produites). Une fois cette phase de mise en confiance passée, les planificateurs peuvent se concentrer uniquement sur les cas problématiques (alertes, gestion par exception). On considère que le projet est pleinement abouti lorsque les utilisateurs savent analyser ces cas et maîtrisent les leviers de pilotage, permettant d'orienter la méthode de calcul vers l'obtention d'une solution plus adaptée.

### 7.4.3 Les difficultés liées aux données

Une des difficultés pouvant parfois être négligée, lors de la mise en œuvre d'outils d'optimisation, concerne l'exactitude et la stabilité des données d'entrée.

Il est fréquent que plusieurs versions des mêmes données existent dans le système d'information du client, ou que les différents utilisateurs de l'outil ne soient pas en accord sur ces données. Il est donc nécessaire de faire un nettoyage des données et de se mettre d'accord en amont du projet. En outre, les APS doivent fournir des plans à une fréquence régulière (quotidienne ou hebdomadaire), à partir de la dernière mise à jour des données. Si la mise à jour de ces données n'est pas effectuée de façon rigoureuse, selon un processus bien défini, cela peut aboutir à des instances infaisables, ou générer des comportements du calcul très instables. Enfin, l'APS étant un outil d'aide à la décision ouvert au paramétrage des utilisateurs, ceux-ci peuvent à tout moment reprendre la main sur les données de l'instance. Si ces modifications manuelles ne sont pas cohérentes, on peut aussi être confronté à des problèmes de résolution.

Les conséquences de données incohérentes peuvent, évidemment, être catastrophiques pour la

pertinence du plan calculé et le temps de calcul. Il est parfois difficile de faire comprendre aux utilisateurs la sensibilité d'un outil d'optimisation.

## 7.5 Apports de l'heuristique constructive

L'heuristique constructive représente une alternative au programme linéaire mixte. Ce dernier a l'avantage d'être générique, mais présente aussi quelques inconvénients. Notamment, lorsque certaines données du problème sont incompatibles, rendant le problème infaisable, la résolution échoue et il est extrêmement difficile de localiser les données en cause. L'utilisateur perd du temps pour le lancement d'un calcul, qui ne lui donne aucun résultat, et n'a aucun moyen de connaître la cause du blocage. L'heuristique constructive permet de contourner cette difficulté : la détection d'une incompatibilité au cours du déroulement de l'algorithme génère un message d'alerte, et les données sont éventuellement corrigées localement, mais le calcul ne doit en aucun cas être bloqué. Cette gestion et détection des données incohérentes amène une énorme valeur ajoutée pour l'utilisateur.

Même pour des cas à portée du solveur, certains utilisateurs ne comprennent pas la construction du résultat et préfèrent une heuristique dont la logique peut être expliquée.

Avec l'heuristique, il est aussi plus aisé d'intégrer des règles de priorité, pour guider la construction de la solution.

## 7.6 Conclusion

Dans ce chapitre, nous présentons l'étude d'un cas d'application industrielle. L'APS (*Advanced Planning System*) de FuturMaster est mis en place dans le cadre d'un projet comprenant la gestion des prévisions et la planification de la distribution de produits cosmétiques. L'heuristique constructive, que nous avons développée, est utilisée pour la résolution du problème de planification de la distribution.

La prochaine étape consiste à mettre en place les métaheuristiques afin d'améliorer la qualité des solutions.

Pour faire face aux difficultés liées aux données, plusieurs pistes sont envisageables. Tout d'abord, il est possible d'intégrer une phase de vérification et validation des données en amont des calculs d'optimisation. Cette vérification peut porter à la fois sur :

- les données structurelles telles que la configuration du réseau et les délais,
- les données dynamiques, par exemple les niveaux des stocks disponibles ou les ordres imposés par le planificateur, et
- les paramétrages des contraintes et des stratégies, par exemple les capacités, les tailles de lot ou les niveaux de stock objectif.

Cette vérification peut être effectuée de manière assez basique (blocage de certains types de données à certains niveaux du flux ou vérification des valeurs des données par rapport à des bornes prédéfinies) ou plus élaborée (vérifications « croisées » des données ou algorithmes de prétraitement permettant de détecter certaines incohérences). Cette étape est très importante car elle permet de faire gagner un temps précieux aux utilisateurs en ciblant directement les données aberrantes sans avoir besoin de lancer un calcul d'optimisation.

Le deuxième axe de recherche concerne le développement d'outils d'analyses des résultats. Ces outils doivent permettre de diagnostiquer les causes des problèmes d'infaisabilité et guider l'utilisateur lors de l'analyse des résultats. Lorsque la méthode de calcul est basée sur un solveur, il est possible par exemple d'exploiter le *minimal conflict set* fourni par le solveur en faisant le lien avec les données en cause. On peut aussi créer différentes versions de la problématique en relâchant certaines contraintes. Dans le cas où la résolution du problème initial pose problème, on lance la résolution d'un problème « dégradé ». On peut ainsi définir une procédure de résolutions en cascade. L'intérêt de cette approche est double. Tout d'abord cela permet de fournir quoiqu'il arrive une solution. Ensuite cela permet de cibler les contraintes posant problème lors de la résolution initiale du problème.

Ces différentes approches doivent permettre de limiter l'effet « boîte noire » parfois ressenti, et à terme de sensibiliser les utilisateurs sur l'importance de la cohérence des données. Il est parfois difficile de faire comprendre que si le calcul est infaisable, ce n'est pas la méthode de résolution qui est en cause mais simplement les données de l'instance.

Une piste d'étude très prometteuse pour les outils d'optimisation concerne la robustesse des méthodes utilisées. Pour des systèmes devant tourner à fréquence régulière et aussi critiques pour l'entreprise que des outils de planification, il est indispensable de pouvoir garantir l'obtention d'une solution de qualité satisfaisante en un temps prédictible. Il peut en effet être très compliqué de concilier l'inconnue liée à la résolution de problèmes NP-difficiles, avec les contraintes d'un processus opérationnel :

- traitements à heure fixe,
- travail des planificateur sur l'outil sur des plages horaires bien définies, et
- criticité de l'obtention d'un plan à une maille quotidienne ou hebdomadaire.

Il est donc très intéressant d'améliorer la robustesse des méthodes d'optimisation utilisées, et aussi d'envisager des études d'analyse de sensibilité des problématiques. Se pose aussi la question de la réutilisation des anciens résultats permettant d'avoir une solution plus rapidement et de garantir une certaine stabilité du plan. Ces sujets sont des problématiques de recherche opérationnelle très complexes, et il reste aujourd'hui difficile pour des éditeurs d'outils de mettre en œuvre ce type d'étude.

# Chapitre 8

## Conclusion générale

Dans cette thèse, nous étudions un problème réel et complexe de planification dans un réseau de distribution à deux échelons, avec des capacités finies et des alternatives d’approvisionnement (*multi-sourcing*) en lots pour des produits périssables.

### 8.1 Contributions

Dans le chapitre 2 un l’état de l’art est présenté sur la planification de la chaîne logistique. Un rappel est proposé sur les méthodes d’optimisation utilisées pour les problèmes de planification dans des chaînes logistiques multi-échelons et sur les types de contraintes les plus significatifs.

Un programme linéaire mixte est proposé et testé avec le solveur CPLEX dans le chapitre 3. Pour certaines instances de grande taille, CPLEX consomme beaucoup de temps. Une heuristique constructive (déterministe) est développée dans ce même chapitre, comme une alternative à CPLEX. Les instances testées sont extraites de bases de données de clients de FuturMaster. L’heuristique est très rapide (0,34 secondes en moyenne) même sur des grandes instances. Pour certains cas, l’écart entre les coûts de la solution de l’heuristique et la meilleure borne inférieure fournie par CPLEX, devient assez important. L’écart moyen entre la solution fournie par l’heuristique par rapport à l’optimum est de 7,99% pour l’ensemble des instances testées.

L’heuristique constructive est randomisée dans le chapitre 4 et un générateur aléatoire d’instances est développé, pour plus de diversité dans les tests. Les tests sont effectués sur les instances extraites des bases de données de clients du chapitre 3 et du générateur aléatoire. Pour l’ensemble des instances, le coût des solutions de l’heuristique réactive randomisée est inférieur, en moyenne, de 0.603% par rapport à celui des solutions de l’heuristique déterministe. Cependant, l’écart à l’optimum reste assez important : en moyenne 21,306%.

Les instances générées aléatoirement sont plus difficiles car elles combinent plusieurs contraintes difficiles. Dans les cas réels, il est rare que toutes les contraintes soient prises en compte en même

temps. L'un des objectifs de cette thèse est de proposer un modèle générique qui puisse répondre à différents cas. C'est la raison pour laquelle nous avons implémenté un générateur aléatoire d'instances. La limite du temps de calcul est fixée à 5400 secondes, soit 1 heure et 30 minutes pour chaque instance résolue avec CPLEX. Le temps de calcul est défini, en nous basant sur l'expérience avec les utilisateurs de l'APS (*Advanced Planning System*) de FuturMaster.

Dans le chapitre 5, nous implémentons des métaheuristiques à base de voisinages afin d'améliorer la qualité des solutions de l'heuristique constructive et de l'heuristique réactive randomisée.

La méthode VND (202,70 secondes en moyenne) améliore significativement la qualité des solutions en étant 11 fois plus rapide que CPLEX. Le coût des solutions initiales (heuristique constructive) diminue en moyenne de 16,868% lorsque la VND est appliquée. Les solutions de la VND sont à 4,234% de l'optimum. La méthode ILS (avec 10 itérations) améliore davantage la qualité des solutions (3,567% de l'optimum). Le temps de calcul moyen augmente, mais reste très inférieur au temps de calcul de CPLEX. La méthode MS-ILS, quant à elle, consomme beaucoup de temps, même si elle fournit en moyenne une qualité de solutions à 3,445% de l'optimum. Le temps de calcul moyen de la MS-ILS exécutée avec 3 lancements (chaque lancement s'effectue avec la méthode ILS de 10 itérations) représente plus du double de celui de l'ILS (avec 10 itérations) : 1890,47 secondes pour la MS-ILS et 851,23 secondes pour l'ILS.

Selon le temps que l'utilisateur alloue au calcul, nous pourrions lui proposer les méthodes VND ou ILS pour résoudre efficacement ce type de problème.

Pour certaines instances issues du générateur aléatoire, CPLEX ne fournit pas une bonne solution réalisable dans le temps imparti. L'heuristique Lagrangienne, proposée dans le chapitre 6, nous permet de comparer les solutions de ces instances à celles de l'heuristique constructive et des métaheuristiques.

L'heuristique Lagrangienne, utilisant la relaxation de la contrainte de taille de lot, fournit les meilleures solutions pour les instances les plus difficiles. Les règles de mise à jour du pas de la série convergente et de la méthode de relaxation permettent d'obtenir les meilleurs résultats, en peu d'itérations, pour l'ensemble des instances générées aléatoirement. Dès la quatrième itération, pour 65% des instances, la solution optimale du problème primal est trouvée, en utilisant la règle de série convergente. Avec la méthode de relaxation, pour 67,5% des instances, nous obtenons la solution optimale du problème primal, au bout de la quatrième itération. Ces résultats sont remarquables car en peu d'itérations, nous trouvons la solution optimale du problème primal.

Dans le chapitre 7, l'étude d'un cas d'application industrielle est présentée. L'APS (*Advanced Planning System*) de FuturMaster est mis en place dans le cadre d'un projet comprenant la gestion des prévisions de ventes et la planification de la distribution de produits cosmétiques. L'heuristique constructive, développée dans cette thèse, est utilisée pour la résolution du problème de planification de la distribution.

L'heuristique constructive représente une alternative au module de FuturMaster reposant sur

la programmation linéaire. Ce module a l'avantage d'être très puissant et très générique, mais présente aussi quelques inconvénients. Notamment, lorsque certaines données du problème sont incompatibles, rendant le problème infaisable, la résolution échoue et il est extrêmement difficile de localiser les données en cause. L'utilisateur perd du temps pour le lancement d'un calcul, qui ne lui donne aucun résultat, et n'a aucun moyen de connaître la cause du blocage. L'heuristique constructive permet de contourner cette difficulté : la détection d'une incompatibilité au cours du déroulement de l'algorithme génère un message d'alerte, et les données sont éventuellement corrigées localement, mais le calcul ne doit en aucun cas être bloqué. Cette gestion des données incohérentes amène une énorme valeur ajoutée pour l'utilisateur.

Les travaux de cette thèse ont fait l'objet d'un article accepté Kande *et al.* (2015a) pour une revue et un autre en révision Kande *et al.* (2015b) pour un livre. Deux articles (Kande *et al.* (2015c) et Kande *et al.* (2014b)) sont publiés dans les actes de conférences internationales. Cinq communications ont été faites lors de conférences nationales (Kande *et al.* (2015d), Kande *et al.* (2014a), Kande *et al.* (2013) et Kande *et al.* (2012)) et internationale (Kande *et al.* (2014c)).

## 8.2 Perspectives

Concernant les perspectives, nous envisageons d'intégrer une flotte hétérogène de véhicules de transport, afin de mieux refléter certains cas industriels. Un centre de distribution peut être approvisionné à partir d'un même entrepôt selon plusieurs modes de transport. Ces derniers diffèrent à travers leur capacité, leur délai de livraison et leur coût.

Les réseaux de distribution réels peuvent comporter plusieurs dizaines de sites logistiques et industriels, avec plusieurs niveaux de profondeur de flux : avec deux, trois ou quatre niveaux de stockage. Dans notre étude, nous considérons des entrepôts qui approvisionnent des centres de distribution. Une extension possible du problème concerne l'ajout d'étapes de distribution. Les stocks d'usines en amont, fournissant les entrepôts, peuvent être pris en compte. En aval, les stocks de magasins approvisionnés par les centres de distribution, peuvent également être intégrés dans le réseau de distribution.

Un autre développement serait d'intégrer les activités de production. Le réseau global d'une chaîne logistique est souvent composé d'usines, d'entrepôts ou/et de centres de distribution, et de clients finaux ou des détaillants. L'optimisation de la planification intégrée de la production et de la distribution consiste à optimiser simultanément les décisions concernant la production et la distribution dont les périmètres sont traditionnellement considérés séparément.

Ce problème présente un réel intérêt pour les chercheurs et les industriels au cours des dernières années. Plusieurs raisons peuvent expliquer cette tendance : l'effet positif sur la rentabilité de la chaîne logistique à travers l'intégration globale des activités de production et de distribution, la réduction des délais, la réactivité face aux changements du marché et donc la réduction de la

propagation des imprévus et des événements indésirables à travers le réseau. Les objectifs sont souvent la maximisation du profit ou la minimisation des coûts liés à la production, au transport et à la gestion des stocks.

Chandra et Fisher (1994) sont parmi les premiers à proposer une analyse détaillée du gain obtenu par la coordination de la planification de la production et de la distribution. Fahimnia *et al.* (2013) présentent un état de l'art des articles sur l'optimisation de la planification intégrée de la production et de la distribution. Les auteurs notent qu'à part certains secteurs particuliers, la plupart des premiers modèles sont seulement des simplifications des problèmes réels de planification de la chaîne logistique. La tendance est de développer des méthodes destinées à résoudre des instances de grande taille et pouvant être utilisées pour des applications industrielles.

Il serait intéressant d'effectuer une étude sur la robustesse des méthodes de résolution proposées dans cette thèse. Le terme robustesse possède ici un double sens. Tout d'abord, les méthodes élaborées doivent être stables en termes de qualité et de temps de calcul. Il n'est pas concevable d'expliquer à un client que d'un lancement du calcul à l'autre, on obtienne une solution beaucoup moins bonne ou que le calcul prenne deux fois plus de temps. Ensuite la variabilité des demandes (prévisions de ventes), ne doit pas dégrader la qualité des solutions.

# Bibliographie

- N. ABSI, B. DETIENNE et S. DAUZÈRE-PÉRÈS : Heuristics for the multi-item capacitated lot-sizing problem with lost sales. *Computers & Operations Research*, 40:264–272, 2013.
- N. ABSI, S. KEDAD-SIDHOUM et S. DAUZÈRE-PÉRÈS : Uncapacitated lot-sizing problem with production time windows, early productions, backlogs and lost sales. *International Journal of Production Research*, 49(9):2551–2566, 2011.
- A. AGGARWAL et J.K. PARK : Improved algorithms for economic lot size problems. *Operations Research*, 41:549–571, 1993.
- O. AHUMADA et J. R. VILLALOBOS : Operational model for planning the harvest and distribution of perishable agricultural products. *International Journal Production Economics*, 133:677–687, 2011.
- O. AHUMADA, J. R. VILLALOBOS et A. N. MASON : Tactical planning of the production and distribution of fresh agricultural products under uncertainty. *Agricultural Systems*, 112:17–26, 2012.
- R. AKKERMAN, A. RONG et M. GRUNOW : An optimization approach for managing fresh food quality throughout the supply chain. *International Journal of Production Economics*, 131:421–429, 2011.
- D. AKSEN, K. ALTINKEMER et S. CHAND : The single-item lot-sizing problem with immediate lost sales. *European Journal of Operational Research*, 147(3):558–566, 2003.
- P. AMORIM, H.O. GUNTHER et B. ALMADA-LOBO : Multi-objective integrated production and distribution planning of perishable products. *International Journal Production Economics*, 138:89–101, 2012.
- U.M. APTE et S. VISWANATHAN : Effective cross docking for improving distribution efficiencies. *International Journal of Logistics : Research and Applications*, 3(3):291–302, 2000.
- K.J. ARROW, S. KARLIN et H. SCARF : Studies in the mathematical theory of inventory and production. *Stanford University Press, Stanford, California*, 1958.
- M. BAKKER, J. RIEZEBOS et R. H. TEUNTER : Inventory lot-sizing with supplier selection. *Review of inventory systems with deterioration since 2001*, 221:275–284, 2012.

- M. BASHIRI, H. BADRI et J. TALEBI : A new approach to tactical and strategic planning in production-distribution networks. *Applied Mathematical Modelling*, 36:1703–1717, 2012.
- C. BASNET et J.M.Y. LEUNG : Inventory lot-sizing with supplier selection. *Computers & Operations Research*, 32:1–14, 2005.
- G. BELVAUX et L. A. WOLSEY : Modelling practical lot-sizing problems as mixed-integer programs. *Management Science*, 47(7):993–1007, 2001.
- G. BITRAN et H.H. YANASSE : Computational complexity of the capacitated lot size problem. *Management Science*, 28(10):1174–1186, 1982.
- F. BOCK : An algorithm for solving "traveling-salesman" and related network optimization problems : abstract. *Bulletin Fourteenth National Meeting of the Operations Research Society of America*, page 897, 1958.
- M. BOUDIA, M.A.O. LOULY et C. PRINS : A reactive GRASP and path relinking for a combined production-distribution problem. *Computers & Operations Research*, 34:3402–3419, 2007.
- N. BRAHIMI, S. DAUZÈRE-PÉRÈS, N. M. NAJID et A. NORDLI : A lagrangian relaxation heuristic for the capacitated single item lot sizing problem with time windows, in : Sixth workshop on : Models and algorithms for planning and scheduling problems. *Aussois*, pages 105–106, 2003.
- N. BRAHIMI, S. DAUZÈRE-PÉRÈS, N. M. NAJID et A. NORDLI : Single item lot sizing problems. *European Journal of Operational Research*, 168:1–16, 2006.
- R. L. BRECMAN, L. P. RITZMAN et L. J. KRAJEWSKI : A heuristic algorithm for managing inventory in a multi-echelon environment. *Journal of Operations Management*, 8 (13):186–208, 1989.
- E.S. BUFFA et R.K. SARIN : *Modern production/operations management*. Wiley, New York, 1987.
- P. CHANDRA et M. L. FISHER : Coordination of production and distribution planning. *European Journal of Operational Research*, 72:503–517, 1994.
- H. CHEN, D. HEARN et C. LEE : A new dynamic programming method for the single item capacitated dynamic lot size model. *Journal of Global Optimization*, 4:285–300, 1994.
- W.H. CHEN et J.M. THIZY : Analysis of relaxations for the multi-item capacitated lot-sizing problem. *Annals of Operations Research*, 26:29–72, 1990.
- C.C. CHERN et J.S. HSIEH : A heuristic algorithm for master planning that satisfies multiple objectives. *Computers & Operations Research*, 34:3491–3513, 2007.
- M.E. CÓCCOLA, M. ZAMARRIPA, C.A. MÉNDEZA et A. ESPUÑA : Toward integrated production and distribution management in multi-echelon supply chains. *Computers and Chemical Engineering*, 2013.

- G.A. CROES : A method for solving traveling salesman problems. *Operations Research*, 6:791–812, 1958.
- D.P. CUERVO, P. GOOS, K. SÖRENSEN et E. ARRÀIZ : An iterated local search algorithm for the vehicle routing problem with backhauls. *European Journal of Operational Research*, 237:454–464, 2014.
- C. DUHAMEL, P. LACOMME, A. QUILLIOT et H.TOUSSAINT : A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Computers & Operations Research*, 38:617–640, 2011.
- B. FAHIMNIA, R. Z. FARAHANI, R. MARIAN et L. LUONG : A review and critique on integrated production-distribution planning models and techniques. *Journal of Manufacturing Systems*, 32:1–19, 2013.
- A. FEDERGRUEN et M. TZUR : The dynamic lot-sizing model with backlogging : A simple  $o(n \log n)$  algorithm and minimal forecast horizon procedure. *Naval Research Logistics*, 40:459–478, 1993.
- M.R. FELLOWS, F.V. FOMIN, D. LOKSHTANOV, F. ROSAMOND, S. SAURABH et Y. VILLANGER : Local search : Is brute-force avoidable? *Journal of Computer and System Sciences*, 78:707–719, 2012.
- T.A. FEO et J. BARD : Flight scheduling and maintenance base planning. *Management Science*, 35(12):1415–1432, 1989.
- T.A. FEO et M.G.C. RESENDE : Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.
- M. L. FISHER : The lagrangian relaxation method for solving integer programming problems. *Management science*, 27(1):1–18, 1981.
- M. FLORIAN, J.K. LENSTRA et A.H.G. Rinnoy KAN : Deterministic production planning : algorithms and complexity. *Management Science*, 26(7):669–679, 1980.
- E. GEBENNINI, R. GAMBERINI et R. MANZINI : An integrated production-distribution model for the dynamic location and allocation problem with safety stock optimization. *International Journal Production Economics*, 122:286–304, 2009.
- A.M. GEOFFRION : Lagrangian Relaxation for Integer Programming. *Mathematical Programming Study*, 2:82–114, 1974.
- J.L. GOFFIN : On the convergence rates of subgradient optimization methods. *Mathematical Programming*, 13:329–347, 1977.
- M. GONDRAN et M. MINOUX : *Graphes et algorithmes*. Eyrolles, 1979.
- S.K. GOYAL et B.C. GIRI : Recent trends in modeling of deteriorating inventory. *European Journal of Operational Research*, 134:1–16, 2001.

- P. HANSEN et N. MLADENOVIC : Variable neighborhood search : Principles and applications. *European Journal of Operational Research*, 130(3):449–467, 2001.
- M. HELD et R.M. KARP : The traveling salesman problem and minimum spanning trees. *Operations Research*, 18:1138–1162, 1970.
- M. HELD et R.M. KARP : The traveling salesman problem and minimum spanning trees : Part II. *Mathematical Programming*, 1:6–25, 1971.
- M. HELD, P. WOLFE et H.P. CROWDER : Validation of subgradient optimization. *Mathematical Programming*, 6(1):62–88, 1974.
- K.S. HINDI, T. BASTA et K. K. PIENÂ : Efficient solution of a multi-commodity, two-stage distribution problem with constraints on assignment of customers to distribution centres. *International Transactions in Operational Research*, 5(16):519–527, 1998.
- R. JANS et Z. DEGRAEVE : Meta-heuristics for dynamic lot sizing : A review and comparison of solution approaches. *European Journal of Operational Research*, 177:1855–1875, 2007.
- K. KANCHANASUNTORN et A. TECHANITISAWAD : An approximate periodic model for fixed-life perishable products in a two-echelon inventory-distribution system. *International Journal of Production Economics*, 100:101–115, 2006.
- S. KANDE, C. PRINS et L. BELGACEM : Heuristique pour un réseau de distribution à deux échelons avec flotte homogène limitée et alternatives d’approvisionnement pour des produits périssables. 1ère édition de GOL 2012, Gestion Opérationnelle de la Logistique (GOL - conférence IEEE section France) à l’Université du Havre, France, Octobre 2012.
- S. KANDE, C. PRINS et L. BELGACEM : Approvisionnement de produits périssables en lot dans un réseau à deux échelons avec capacité de transport. 14ème conférence ROADEF à l’Université de Technologie de Troyes, France, Février 2013.
- S. KANDE, C. PRINS et L. BELGACEM : Modèle linéaire mixte et heuristique randomisée pour un réseau de distribution à deux échelons pour des produits périssables. 15ème conférence ROADEF à l’Université de Bordeaux, France, Février 2014a.
- S. KANDE, C. PRINS et L. BELGACEM : Modèle linéaire mixte et heuristiques pour un réseau de distribution à deux échelons pour des produits périssables. *Information Systems and Operational Research (INFOR)*, 52(2):73–93, 2015a.
- S. KANDE, C. PRINS, L. BELGACEM et B. REDON : Iterated local search with neighborhood space reduction for two-echelon distribution network for perishable products. In IEEE Computational Intelligence SOCIETY, éditeur : *IEEE SSCI 2014 : Symposium Series on Computational Intelligence, Orlando, United State, 09-12/12/2014*, pages 67–74, 2014b.

- S. KANDE, C. PRINS, L. BELGACEM et B. REDON : Variable neighborhood descent for two-echelon distribution network with capacity constraints, multi-sourcing and lot-sizing for perishable products. International conference on Metaheuristics and Nature Inspired Computing, Marrakech, Maroc, Octobre 2014c.
- S. KANDE, C. PRINS, L. BELGACEM et B. REDON : *In : Communications in Computer and Information Science*, chapitre Local Search Based Metaheuristics for Two-Echelon Distribution Network with Perishable Products. Springer-Verlag, 2015b.
- S. KANDE, C. PRINS, L. BELGACEM et B. REDON : Multi-start iterated local search for two-echelon distribution network for perishable products. *In Begoña VITORIANO et Greg H. PARLIER, éditeurs : ICORES 2015 : 4th International Conference on Operations Research and Enterprise Systems, Lisbonne, Portugal, 10-12/01/2015*, pages 294–303. SCITEPRESS-Science and Technology Publications, 2015c.
- S. KANDE, C. PRINS, L. BELGACEM et B. REDON : Relaxation lagrangienne et recherche locale itérative pour un problème de distribution à deux échelons pour des produits périssables. 16ème conférence ROADEF à l'Université d'Aix-Marseille, France, Février 2015d.
- Y.J. KWON, Y.J. CHOI et D.H. LEE : Heterogeneous fixed fleet vehicle routing considering carbon emission. *Transportation Research, Part D* 23:81–89, 2013.
- H.M.S. LABABIDI, M.A. AHMED, I.M. ALATIQUI et A.F. AL-ENZI : Optimizing the supply chain of a petrochemical company under uncertain operating and economic conditions. *Industrial and Engineering Chemistry Research*, 43:63–73, 2004.
- S. ÓLAFSSON : *In : Handbook in OR & MS*, chapitre Metaheuristics, pages 633–654. Elsevier, 2006.
- B.K. LEE, K. H. KANG et Y. H. LEE : Decomposition heuristic to minimize total cost in a multi-level supply chain network. *Computers & Industrial Engineering*, 54:945–959, 2008.
- M. LEJEUNE : A variable neighborhood decomposition search method for supply chain management planning problems. *European Journal of Operational Research*, 175:959–976, 2006.
- J. LI, F. CHU, C. PRINS et Z. ZHU : Lower and upper bounds for a two-stage capacitated facility location problem with handling costs. *European Journal of Operational Research*, 236:957–967, 2014.
- X. LI, A. L. LIM, Z. MIA et B. RODRIGUES : Reducing transportation costs in distribution network. *M. Ali and R. Dapoigny (Eds.) : IEA/AIE, Springer-Verlag Berlin Heidelberg*, LNAI 4031:1138–1148, 2006.
- T.F. LIANG : Application of fuzzy sets to manufacturing/distribution planning decisions in supply chains. *Information Sciences*, 181:842–854, 2011.

- S.C. LIU et C.C. LIN : A heuristic method for the combined location routing and inventory problem. *International Journal Advanced Manufacturing Technology*, 26:372–381, 2005.
- H.R. LOURENÇO, O.C. MARTIN et T. STÜTZLE : In : *Handbook of Metaheuristics*, chapitre Iterated local search, pages 321–353. Kluwer and Dordrecht, 2003.
- H.R. LOURENÇO, O.C. MARTIN et T. STÜTZLE : In : *Handbook of Metaheuristics*, volume 146, chapitre Iterated local search : Framework and applications, pages 363–397. Springer, New York, 2010.
- S.F. LOVE : *Inventory Control*. McGraw-Hill, 1979.
- R. MANZINI et F. BINDI : Strategic design and operational management optimization of a multi stage physical distribution system. *Transportation Research Part E*, 45:915–936, 2009.
- K.F. MATTA et D. SINHA : Policy and cost approximations of two-echelon distribution systems with a procurement cost at the higher echelon. *IIE Transactions*, 27:638–645, 1995.
- J. MICHALLET, C. PRINS, L. AMODEO, F. YALAOUI et G. VITRY : Multi-start iterated local search for the periodic vehicle routing problem with time windows and time spread constraints on services. *Computers & Operations Research*, 41:196–207, 2014.
- R.B. MISRA : Optimum production lot size model for a system with deteriorating inventory. *International Journal of Production Research*, 13(5):495–505, 1975.
- N. MLADENOVIĆ et P. HANSEN : Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997.
- J. MULA, D. PEIDRO, M. DÍAZ-MADROÑERO et E. VICENS : Mathematical programming models for supply chain production and transport planning. *European Journal of Operational Research*, 204:377–390, 2010.
- A.M. NEZHAD, H. MANZOUR et S. SALHI : Lagrangian relaxation heuristics for the uncapacitated single source multi-product facility location problem. *International Journal Production Economics*, page <http://dx.doi.org/10.1016/j.ijpe.2013.06.001>, 2013.
- V.P. NGUYEN, C. PRINS et C. PRODHON : A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem. *Engineering Applications of Artificial Intelligence*, 25:56–71, 2012.
- K.S. PARK : An integrated production-inventory model for decaying raw materials. *International Journal of Systems Science*, 14(7):801–806, 1983.
- B.T. POLYAK : A general method for solving extremal problems. *Soviet Math Doklady*, 8:593–597, 1967.
- B.T. POLYAK : Minimization of unsmooth functionals. *USSR Computational mathematics and Mathematical Physics*, 9:509–521, 1969.

- M. PRAIS et C.C. RIBEIRO : Reactive grasp : an application to a matrix decomposition problem in tdma assignment. *INFORMS Journal on Computing*, 12(3):164–176, 2000.
- C. PRINS, V. P. NGUYEN et C. PRODHON : A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem. *Engineering Applications of Artificial Intelligence*, 25:56–71, 2012.
- C. PRINS et R. WOLFLER CALVO : A fast grasp with path relinking for the capacitated arc routing problem. In L. Gouveia et C. MOURÃO, éditeur : *INOC 2005 : 3rd International Network Optimization Conference, Lisbonne, Portugal, 20-23/03/2005*, pages 289–295. Université de Lisbonne, 2005.
- F. RAAFAT, P.M. WOLFE et H.K. EL-DIN : An inventory model for deteriorating items. *Computers and Industrial Engineering*, 20(1):89–94, 1991.
- D. SANCHEZ, L. AMODEO et C. PRINS : *Artificial intelligence techniques for networked manufacturing enterprises management*, Springer Series in Advanced Manufacturing, chapitre Metaheuristic approaches for multi-objective simulation-based optimization in supply chain inventory management, pages 249–269. Springer, 2010.
- N.Z. SHOR : On the rate of convergence of the generalized gradient method. *Kibernetika*, 4:3, 1968.
- A.J. STENGER et J.L. CAVINATO : Adapting mrp to the outbound side-distribution requirements planning. *Production and Inventory Management, Fourth Quarter*, pages 1–13, 1979.
- S.A. TORABI et E. HASSINI : An interactive possibilistic programming approach for multiple objective supply chain master planning. *Fuzzy Sets and Systems*, 159:193–214, 2008.
- R.J.M. VAESSENS, E.H.L. AARTS et J.K. LENSTRA : A local search template. *Computers & Operations Research*, 25(11):969–979, 1998.
- J. van BELLE, P. VALCKENAERS et D. CATTRYSSSE : Cross-docking : State of the art. *Omega*, 40:827–846, 2012.
- K. van DONSELAAR, T. van WOENSEL, R. BROEKMEULEN et J. FRANSOO : Inventory control of perishables in supermarkets. *International Journal of Production Economics*, 104:462–472, 2006.
- C.P.M. van HOESEL : *Models and algorithms for single-item lot sizing problems*. Thèse de doctorat, Erasmus Universiteit, 1991.
- T. VIDAL, T. G. CRAINIC, M. GENDREAU et C. PRINS : Heuristics for multi-attribute vehicle routing problems : A survey and synthesis. *European Journal of Operational Research*, pages –, 2013.
- J. G. VILLEGAS, C. PRINS, C. PRODHON, A. L. MEDAGLIA et N. VELASCO : Grasp/vnd and multi-start evolutionary local search for the single truck and trailer routing problem with

- satellite depots. *Engineering Applications of Artificial Intelligence*, 23:780–794, 2010.
- A. WAGELMANS, C.P.M. Van HOESEL et A. KOLEN : Economic lot sizing : an  $o(n \log n)$  that runs in linear time in the wagner-whitin case. *Operations Research*, 40:S145–S156, 1992.
- H. M. WAGNER et T. M. WHITIN : A dynamic version of the economic lot size model. *Management Science*, 5(1):89–96, 1958.
- D.C. WHYBARK : Mrp : A profitable concept for distribution. *In Proceedings of the Fifth Annual Transportation and Logistics Educators Conference*, pages 82–93, 1975.
- S. WOLF et P. MERZ : *Hybrid Metaheuristics*, chapitre Evolutionary Local Search for the Super-Peer Selection Problem and the p-Hub Median Problem, pages 1–15. Springer Berlin Heidelberg, 2007.
- Y. XIAO, Q. ZHAO, I. KAKU et Y. XU : Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers & Operations Research*, 39:1419–1431, 2012.
- Y. YU, Z. WANG et L. LIANG : A vendor managed inventory supply chain with deteriorating raw materials and products. *International Journal Production Economics*, 136:266–274, 2012.
- W.I. ZANGWILL : A backlogging model and a multi-echelon model of a dynamic economic lot size production system - a network approach. *Management Science*, 15(9):506–527, 1969.
- S. ZANONI et L. ZAVANELLA : Single-vendor single-buyer with integrated transport-inventory system : Models and heuristics in the case of perishable goods. *Computers & Industrial Engineering*, 52:107–123, 2007.
- W.Q. ZHOU, L. CHEN et H.M. GE : A multi-product multi-echelon inventory control model with joint replenishment strategy. *Applied Mathematical Modelling*, 37:2039–2050, 2013.

# Sona KANDE

## Doctorat : Optimisation et Sûreté des Systèmes

### Année 2015

#### Etude et résolution de problèmes de planification dans des réseaux logistiques multi-échelons

Les travaux de cette thèse concernent la résolution d'un problème de planification dans un réseau de distribution à deux échelons intégrant la gestion de stocks de produits périssables, le dimensionnement de lots, des alternatives d'approvisionnement. La livraison s'effectue directement entre un fournisseur et son client, sans tournée avec une flotte homogène de véhicules. Nous proposons un programme linéaire mixte, une heuristique constructive (déterministe) et une heuristique réactive randomisée. Pour certaines instances, le solveur de programme linéaire mixte ne fournit pas une bonne solution réalisable dans la limite de temps définie ou prend beaucoup de temps. Les heuristiques proposées sont rapides mais ne donnent pas de bonnes solutions pour certaines instances. Pour améliorer la qualité des solutions des heuristiques, la descente à voisinage variable (VND), la recherche locale itérative (ILS) et la recherche locale itérative à démarrages multiples (MS-ILS) sont développées.

Toutes ces méthodes ont été incluses dans un APS (Advanced Planning System) et sont comparées avec CPLEX sur des instances extraites de bases de données réelles. Un générateur aléatoire d'instances est conçu pour plus de diversité pour les tests. Une relaxation lagrangienne est implémentée pour comparer les solutions des instances, pour lesquelles CPLEX ne fournit pas une bonne solution réalisable dans le temps imparti, avec les autres méthodes. Une heuristique lagrangienne, utilisant la relaxation lagrangienne et une heuristique de réparation, est également développée.

Mots clés : logistique (organisation) - planification - optimisation combinatoire - algorithmes optimaux - algorithmes d'approximation.

#### Study and Solving Planning Problems in Multi-echelon Supply Networks

This work presents a planning problem in a distribution network incorporating two levels inventory management of perishable products, lot-sizing, multi-sourcing and transport capacity with a homogeneous fleet of vehicles. A mixed integer linear programming (MILP) a greedy heuristic and a reactive randomized heuristic are developed to solve this real planning problem. There are some instances for which the solver CPLEX cannot give a good upper bound within the limited time and for other instances it takes a lot of time to solve MILP. The heuristics are alternatives to the mixed integer linear program to quickly solve some large instances taking into account original and difficult constraints. For some instances the gap between the solutions of the solver (MILP) and the heuristics becomes quite significant. The variable neighborhood descent (VND), the iterated local search (ILS) and the multi-start iterated local search (MS-ILS) are implemented. These methods are included in an APS (Advanced Planning System) and compared with a MILP solver. The instances are derived from actual data or built using a random generator of instances to have wider diversity for computational evaluation. A lagrangian relaxation is developed to compare the solutions of the instances, for which CPLEX cannot give a good upper bound within the limited time, with the other methods (greedy heuristic, VND, ILS and MS-ILS). A lagrangian heuristic is proposed; the solution of lagrangian relaxation is used to build a feasible solution with a repair heuristic.

Keywords: business logistics - planning - combinatorial optimization - exact methods – approximation algorithms.

Thèse réalisée en partenariat entre :

