



HAL
open science

Optimisation des flux : application aux problèmes de distribution en nutrition animale

Cadet David Joseph

► **To cite this version:**

Cadet David Joseph. Optimisation des flux : application aux problèmes de distribution en nutrition animale. Recherche opérationnelle [math.OC]. Université de Technologie de Troyes, 2013. Français. NNT : 2013TROY0035 . tel-03355959

HAL Id: tel-03355959

<https://theses.hal.science/tel-03355959>

Submitted on 27 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

Cadet David JOSEPH

**Optimisation des flux :
application aux problèmes
de distribution en nutrition animale**

**Spécialité :
Optimisation et Sécurité des Systèmes**

2013TROY0035

Année 2013

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

Cadet David JOSEPH

le 18 décembre 2013

Optimisation des flux : application aux problèmes de distribution en nutrition animale

JURY

M. L. AMODEO	PROFESSEUR DES UNIVERSITES	Président
M. R. AGGOUNE	DOCTEUR	Examineur
M. A. HAMZAOUI	PROFESSEUR DES UNIVERSITES	Rapporteur
M. C. PRINS	PROFESSEUR DES UNIVERSITES	Directeur de thèse
Mme W. RAMDANE-CHERIF	MAITRE DE CONFERENCES ENSM NANCY	Examineur
M. Z. SARI	PROFESSEUR	Rapporteur
M. F. YALAOUI	PROFESSEUR DES UNIVERSITES	Directeur de thèse

Résumé

Cette thèse porte sur l'étude du problème de tournées de véhicules compartimentés (Multi-Compartment Vehicle Routing Problem ou MC-VRP) dans le contexte de l'industrie de la nutrition animale. Les travaux de recherche et d'application sont concentrés sur la distribution dans le domaine agroalimentaire.

En dépit d'une large application industrielle, les problèmes de MC-VRP ont été peu étudiés dans la littérature scientifique. Trois variantes du MC-VRP sont traitées dans cette thèse. D'une part, nous proposons les algorithmes "Greedy Randomized Adaptive Search Procedure" (GRASP) et "Iterated Local Search procedure" (ILS) pour résoudre le MC-VRP avec un compartiment de taille fixe dédié à chaque produit. Une extension de ce problème à des compartiments de tailles variables (Flexible Compartments Vehicle Routing Problem ou FC-VRP) est également résolue. D'autre part, nous proposons un GRASP et un Multi Start ILS pour la résolution du MC-VRP avec décisions d'affectation de chaque compartiment à un client et un produit.

En dernier lieu, des travaux d'application industrielle sont présentés. Des tests d'évaluation de performances ont été réalisés dans un contexte de distribution d'aliments au bétail chez la société Nestlé. Des outils d'aide à la décision ont été développés et mis en place dans cette société.

Mots clés : recherche opérationnelle, optimisation combinatoire, logistique (organisation), transport, méta-heuristiques, industries agroalimentaires, animaux - nutrition.

Abstract

This research concerns solving the Multi-Compartment Vehicle Routing Problem (MC-VRP) in the context of animal nutrition industry. Research and application work focuses on distribution in food industry.

Despite its vast application in industry, little attention has been paid to the MC-VRP. We address three classes of MC-VRP in this research. Firstly, we propose two metaheuristics, "Greedy Randomized Adaptive Search Procedure" (GRASP) et "Iterated Local Search procedure" (ILS), in order to solve a MC-VRP with a fixed-sized compartment dedicated to each product. Also, an extension of this problem to variable-sized compartments which we call Flexible Compartments Vehicle Routing Problem (FC-VRP) is studied. Further, we propose a GRASP and a Multi Start ILS to solve a MC-VRP problem with assignment decisions of each compartment to one client and one product.

Finally, some application work is presented. Experiments intended to measure performance in the context of food distribution to cattle were conducted for Nestlé company. Decision support tools had been developed and implemented for this company.

Keywords : operations research, combinatorial optimization, business logistics, transportation, food industry and trade, animal nutrition.

Remerciements

Je tiens à remercier mes directeurs de thèse MM. les professeurs Christian PRINS et Farouk YALAOUI pour leur encadrement. Je remercie M. le professeur Lionel AMODEO également pour sa participation à l'encadrement de cette thèse.

La réalisation de cette thèse n'aurait pas été possible sans le soutien de quelques collègues de laboratoire (Badia DANDACH, Mathieu LE BERRE, Julien MICHALLET), de l'école doctorale, en particulier, M. Régis LENGELLE et de Mme Pascale DENIS. J'ai également pu compter sur le soutien des secrétaires du pôle ROSAS, Mmes Véronique BANSE et Marie-José ROUSSELET.

Je remercie du fond du cœur mes parents qui sont et ont toujours été tout pour moi. Sans leurs sacrifices, rien n'aurait été possible pour moi. Je leur rend hommage à travers cet accomplissement personnel que représente cette thèse pour moi.

Table des matières

Remerciements	3
Notations	13
1 Introduction et contexte	15
1.1 La logistique	15
1.1.1 Définition	15
1.1.2 Structures de la chaîne logistique	18
1.1.3 Niveaux de décision	18
1.1.4 Gestion de la chaîne logistique	20
1.2 La logistique du transport	21
1.3 Contexte industriel	22
1.3.1 Le groupe Nestal	22
1.4 Objectifs de la thèse	25
2 État de l’art	27
2.1 Introduction	27
2.2 Problèmes de tournées de véhicules	27
2.2.1 Les différents paramètres	28
2.2.2 Les principaux problèmes de tournées de véhicules	29
2.3 Le MC-VRP	39
2.4 Méthodes de résolution	41
2.4.1 Méthodes exactes	42

2.4.2	Les heuristiques simples	43
2.4.3	Les métaheuristiques	46
2.5	Conclusion	49
3	MC-VRP avec un compartiment de taille fixe dédié à chaque produit	51
3.1	Introduction	51
3.2	Travaux antérieurs	51
3.3	Modèle mathématique	52
3.4	Résolution heuristique	54
3.4.1	Résolution à l'aide d'un GRASP	54
3.4.2	Résolution à l'aide d'un ILS	59
3.5	Résultats numériques	66
3.5.1	Génération des instances	66
3.5.2	Paramétrage	69
3.5.3	Application au MC-VRP	71
3.6	Conclusion	77
4	Extension à des compartiments de tailles flexibles	79
4.1	Introduction	79
4.2	Travaux antérieurs	79
4.3	Modèle mathématique	80
4.4	Résolution exacte	82
4.4.1	Protocole	83
4.4.2	Application	84
4.5	Adaptation du GRASP et de l'ILS	84
4.6	Résultats numériques	85
4.6.1	Génération des instances	85
4.6.2	Paramétrage	86
4.6.3	Application au FC-VRP	87
4.7	Conclusion	91

5	Le MC-VRP avec décisions d'affectation de chaque compartiment à un client et un produit	93
5.1	Introduction	93
5.2	Travaux antérieurs	94
5.3	Modèle mathématique	94
5.4	Résolution heuristique	96
5.4.1	Description du MS-ILS	97
5.4.2	Optimisation du chargement des véhicules compartimentés	97
5.5	Résultats numériques	106
5.5.1	Génération des instances	106
5.5.2	Paramétrage	106
5.5.3	Application au MC-VRP	107
5.6	Conclusion	110
6	Application industrielle	111
6.1	Introduction	111
6.2	L'activité de Nestal	111
6.3	Etude de lissage de la distribution journalière des produits conditionnés	113
6.3.1	Contexte	113
6.3.2	Mode opératoire	114
6.3.3	Application	115
6.4	Evaluation des performances de l'outil d'optimisation des tournées chez Nestal	117
6.4.1	Contexte	117
6.4.2	Mode opératoire	118
6.4.3	Résultats industriels	118
6.5	Etudes et travaux divers	120
6.6	Conclusion	122
7	Conclusion et perspectives	123

Bibliographie

127

Table des figures

1.1	Différentes structures de la chaîne logistique (Huang <i>et al.</i> [49])	18
1.2	Description des niveaux de décision d'après Huang <i>et al.</i> [49]	19
1.3	Schéma de la chaîne logistique de Nestal	23
1.4	Plan des problématiques traitées dans le document	26
2.1	Graphe d'un réseau routier reliant des clients à un dépôt	31
3.1	Fusion de tournées par la méthode de Clarke and Wright	57
3.2	Mouvements de relocalisation illustrés sur une seule et deux routes	61
3.3	Mouvements d'échange illustrés sur une seule et deux routes	62
3.4	Mouvements 2-Opt illustrés sur une seule et deux routes	63
4.1	Illustration du dimensionnement des compartiments d'un véhicule	82
5.1	Exemple de chargement de compartiments	94
6.1	Comparaison des fluctuations journalières des charges de livraisons (en tonnes) avant et après lissage en périodes de pics en 2009	116
6.2	Cartographie de l'affectation journalière des zones de livraisons avant	116
6.3	Cartographie de l'affectation journalière des zones de livraisons après	117

Liste des tableaux

3.1	Moyenne des solutions GRASP pour différentes valeurs du paramètre de randomisation	70
3.2	Moyenne des solutions et des temps de calcul pour différentes valeurs du nombre d'itérations GRASP	70
3.3	Moyenne des solutions ILS pour différentes valeurs du paramètre concernant le niveau de perturbation	71
3.4	Moyenne des solutions et des temps de calcul pour différents nombres d'itérations de recherche locale	72
3.5	Comparaison de l'ILS et du GRASP sur les types d'instances MC1 et MC2 .	73
3.6	Comparaison de l'ILS et du GRASP sur les types d'instances MC3 et MC4 .	74
3.7	Comparaison de l'ILS et du GRASP sur les types d'instances MC5 et MC6 .	75
3.8	Comparaison de l'ILS et du GRASP sur les types d'instances MC7 et MC8 .	76
4.1	Demande des clients par produit	83
4.2	Résultats de la résolution exacte des problèmes de FC-VRP	84
4.3	Comparaison de l'ILS et du GRASP sur les types d'instances FC1	89
4.4	Comparaison de l'ILS et du GRASP sur les types d'instances FC2	89
4.5	Comparaison de l'ILS et du GRASP sur les types d'instances FC3	90
4.6	Comparaison de l'ILS et du GRASP sur les types d'instances FC4	90
5.1	Comparaison du MS-ILS et du GRASP sur les instances de type MC'1 . . .	108
5.2	Comparaison du MS-ILS et du GRASP sur les instances de type MC'2 . . .	109

6.1	Données statistiques sur les fluctuations de la demande journalière exprimée en tonnes pendant les périodes chargées de 2009	114
6.2	Données statistiques après lissage sur les fluctuations journalières des charges de livraisons en tonnes pendant les périodes de pics en 2009	115
6.3	Résultats industriels	119

Notations

Δ	pas ou intervalle de variation minimale de la capacité d'un compartiment de taille flexible
A	ensemble des arcs (i, j)
C	matrice qui définit pour chaque arête $[i, j]$ un coût de traversée
c_{ij}	coût de traversée de l'arc (i, j)
d_i	demande d'un client i
d_{ip}	quantité d'un produit p demandée par un client i
E	ensemble des arêtes $[i, j]$ qui relient les différents nœuds
G	graphe représentant un réseau routier
K	ensemble désignant la flotte de véhicules utilisées pour des tournées
L	longueur maximale d'une tournée de livraison
$MaxGRASPIter$	nombre maximal d'itérations de la recherche locale dans un GRASP
$MaxIterNonAméliorant$	nombre maximal d'applications consécutives de la recherche locale sur une solution sans amélioration dans un <i>MS-ILS</i>
$MaxLSIter$	nombre maximal d'itérations de la recherche locale dans un <i>ILS</i>
$MaxPerturbIter$	niveau de perturbation défini pour une solution donnée
n	nombre de clients à livrer
p	ensemble des produits livrés aux clients
Q	capacité d'un véhicule, les véhicules d'une flotte étant identiques
q_{ip}^{kt}	quantité de la demande d_{ip} transportée dans le compartiment t du véhicule k
S	ensemble de sommets ou nœuds dans un graphe
s_{ij}	économie obtenue après la fusion de 2 tournées contenant respectivement les nœuds i et j
V	ensemble des sommets ou nœuds d'un graphe y compris le nœud dépôt

V'	ensemble des sommets d'un graphe sauf le nœud dépôt
w_{kp}	taille choisie pour le compartiment du véhicule k affecté au produit p
w_{max}	capacité maximale définie pour un compartiment de taille flexible
w_{min}	capacité minimale définie pour un compartiment de taille flexible
w_p	capacité d'un compartiment d'un véhicule affecté au produit p
w_t	capacité du compartiment t d'un véhicule, les véhicules de la flotte étant considérés identiques
x_{ij}	variable désignant si l'arc (i, j) est traversé ou pas
x_{ijk}	variable indiquant si l'arc (i, j) est traversé par le véhicule k
y_{ik}	variable binaire indiquant si le véhicule k visite le client i
z_{ip}^{kt}	variable binaire indiquant si le compartiment t du véhicule k est affecté à la demande d_{ip}

Chapitre 1

Introduction et contexte

1.1 La logistique

1.1.1 Définition

La logistique est un domaine très vaste qui recouvre diverses fonctions et activités dans le monde de l'entreprise. D'amont en aval, elle englobe l'achat et l'approvisionnement, la production, la gestion des stocks, le transport et la distribution. Le terme « logistique » trouve ses origines dans l'armée en France au XVIIIe siècle. Il désigne l'ensemble des activités de soutien à la stratégie et aux opérations militaires. Il s'agit, en l'occurrence, d'activités de réapprovisionnement en armes, munitions, vivres, chevaux, uniformes, chaussures....

Par la suite, ce terme s'est vu approprié par le milieu industriel pour faire référence à la maintenance et au transport de marchandises. Depuis, la logistique s'est longtemps limitée aux activités d'entrepôts c'est-à-dire la réception, le conditionnement et l'expédition des marchandises. Cependant, vers les années 90, dans un contexte de globalisation des marchés, la logistique a pris un nouvel essor. Une concurrence féroce sur les marchés, le raccourcissement du cycle de vie des produits, les exigences sans cesse croissantes des clients sont autant de facteurs qui ont contribué à complexifier les flux logistiques et ont poussé les entreprises à développer et à étendre le périmètre de la fonction logistique. Ainsi, la logistique a évolué et regroupe plusieurs fonctions dans l'industrie, qui sont principalement :

- **La logistique d'approvisionnement** qui permet d'apporter, d'une part, dans les usines les matières premières, composants et sous-ensembles nécessaires à la production et,

d'autre part, dans les entreprises de service ou des administrations les produits requis pour leurs activités ;

- **La logistique de production (logistique interne)** qui consiste à rendre disponibles à proximité des lignes de production les matières et composants nécessaires à la production et à planifier la production ; cette logistique peut s'étendre jusqu'à la gestion de production toute entière ;

- **La logistique de distribution ou de transport** qui consiste à amener au consommateur final les produits dont il a besoin ;

- **La logistique de soutien** présente dans les secteurs militaire, aéronautique, énergétique, industriel, etc., qui consiste à organiser tout ce qui est nécessaire pour maintenir en opération un système complexe (à travers des activités de maintenance, principalement) ;

- **La logistique des retours** qui consiste à traiter les retours des produits par les clients pour des raisons diverses comme (demandes de réparation, produits non conformes).

De nos jours, le terme logistique laisse sa place à l'expression *chaîne logistique* ou *supply chain* en anglais. La plupart des structures logistiques n'étant pas linéaires, certains préfèrent même utiliser l'expression *réseau logistique*. Car, au-delà des périmètres organisationnels qui se rattachent aux différentes fonctions relevant de la logistique, le concept de *chaîne logistique* repose sur une approche transversale et intégrée du pilotage des flux dans l'entreprise.

Dans la littérature, les définitions abondent en ce qui a trait à la chaîne logistique. Cela dit, il n'en existe pas une définition universelle. Ci-dessous, nous recensons quelques définitions à travers le tableau suivant :

Christopher [20]	Réseau d'entreprises qui sont impliquées, en amont et en aval, dans les différents processus et activités qui créent de la valeur sous forme de produits et de services destinés au consommateur final.
Lee et Billington [62]	Réseaux d'entreprises de production et de distribution qui fournissent des matières premières qui les transforment en produits finis et semi finis et distribuent les produits finis aux consommateurs.
Kopczak [56]	L'ensemble des organisations comprenant les fournisseurs, les prestataires de services logistiques, les fabricants, les distributeurs, les revendeurs par lesquelles transitent les flux de matières, de produits et d'informations.
Tayur <i>et al.</i> [96]	Un système de sous-traitants, de producteurs, de distributeurs, de détaillants et de clients entre lesquels s'échangent les flux matériels dans le sens des fournisseurs vers les clients et des flux d'information dans les deux sens.
Lummus et Vokurka [67]	Toutes les activités impliquées dans la livraison d'un produit depuis le stade de matières premières jusqu'au client en incluant l'approvisionnement en matières premières et produits semi-finis, la fabrication et l'assemblage, l'entreposage et le suivi des stocks, la saisie et la gestion des ordres de fabrication, la distribution via tous les canaux, la livraison au client et le système d'information permettant le suivi de toutes ces activités.
AFN [1]	La logistique est une fonction "dont la finalité est la satisfaction des besoins exprimés ou latents, aux meilleures conditions économiques pour l'entreprise et pour un niveau de service déterminé. Les besoins sont de nature interne (approvisionnement de biens et de services pour assurer le fonctionnement de l'entreprise) ou externe (satisfaction des clients). La logistique fait appel à plusieurs métiers et savoir-faire qui concourent à la gestion et à la maîtrise des flux physiques et d'informations ainsi que des moyens".
Simchi-Levi <i>et al.</i> [91]	Un ensemble d'approches utilisées pour intégrer les fournisseurs, les producteurs, les dépôts et les centres de distribution, pour que le produit fini soit fabriqué et distribué avec la qualité requise, dans les délais exigés, de façon à minimiser le coût total tout en étant à la hauteur du niveau de service demandé.

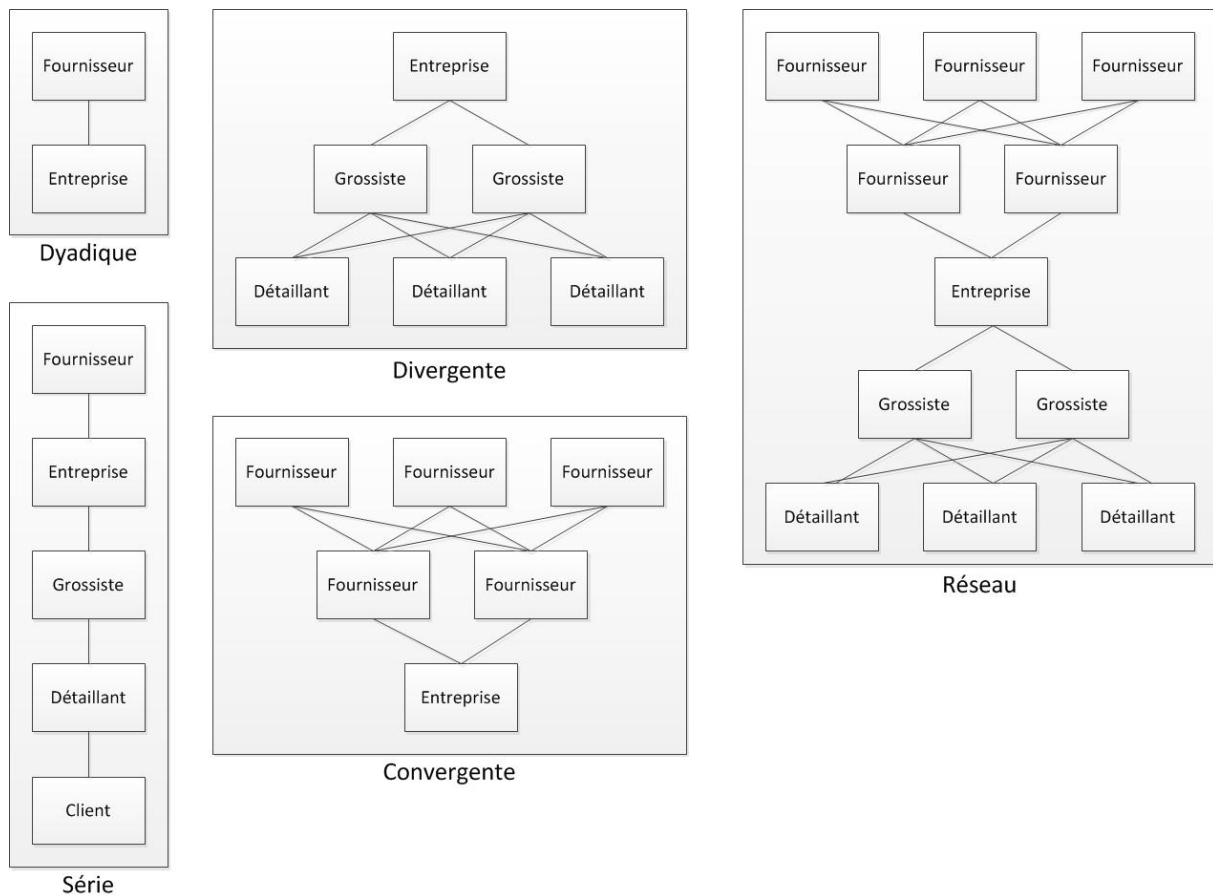


FIGURE 1.1: Différentes structures de la chaîne logistique (Huang et al. [49])

1.1.2 Structures de la chaîne logistique

Une chaîne logistique peut prendre plusieurs formes. Étant donnée une multitude de formes et de types de structures d'approvisionnements, de fabrication et de distribution, il s'avère difficile d'avoir une définition et une cartographie complètes de toutes les structures des chaînes. Une décomposition des structures typiques des chaînes logistiques est proposée par Huang *et al.* [49]. Elle se décline en 5 types : Série ; Divergente ; Dyadique ; Convergente et Réseau (voir l'illustration de la figure 1.1).

1.1.3 Niveaux de décision

Les problématiques rencontrées au sein d'une chaîne logistique ne relèvent pas toutes d'un seul et même niveau de décision. Il est, en effet, évident que des problématiques liées à la gestion de stocks et celles de localisation de sites de production ne peuvent être gérées dans

Niveau de décision ↑	Stratégique	Politique tarifaire	Différenciation	Allocation de capacités	Localisation de sites
	Tactique	Stocks de sécurité	Stocks de sécurité Planning de sécurité	MRP (Planification des besoins en matières)	Planification de la Production et du Transport
	Opérationnel	Commandes	Allocation de stocks	Ordonnancement de la production	Problème de Tournées de Véhicules
		Dyadique/Série	Divergent	Convergent	Réseau
		Structure de la Supply Chain →			

FIGURE 1.2: Description des niveaux de décision d'après Huang et al. [49]

un même horizon et n'ont pas les mêmes retombées sur le fonctionnement de la chaîne.

Dans la littérature, les auteurs s'accordent généralement sur 3 principaux niveaux de décision : stratégique, tactique et opérationnel. En particulier, Huang *et al.* [49] ont proposé une représentation (figure 1.2) des niveaux de décision en rapport avec les principales structures de chaînes logistiques.

- **Niveau stratégique** : Les décisions d'ordre stratégique s'effectuent sur le long terme (plus d'un an, en général) et affectent la chaîne dans toute sa globalité. Elles se rapportent à des décisions ayant trait à la localisation de sites, au lancement de gammes de produits, le choix de mode de transport, la mise en place d'un système d'information etc. ;
- **Niveau tactique** : Il s'agit ici de décisions sur le moyen terme (de l'ordre d'un an à quelques mois). Les décisions au niveau tactique prennent généralement le relais de celles du niveau stratégique. L'objectif étant de travailler plus en local et de se concentrer sur une portion de la chaîne. Parmi les décisions à ce niveau, on peut relever celles concernant l'affectation des clients aux dépôts, l'allocation des fournisseurs aux usines, l'allocation des produits dans les différents centres de production ;
- **Niveau opérationnel** : À ce niveau, des décisions sont prises pour gérer les activités sur un court horizon (1 jour à moins d'un mois en pratique). Ces décisions s'appliquent à des cas comme l'organisation de la production d'un atelier, l'affectation des machines, la planification de tournées de livraisons etc. Il est à noter qu'à un niveau plus fin, certains auteurs parlent de décisions en temps réel (de l'ordre de quelques minutes à

quelques heures). Ces dernières interviennent souvent en réponse à des aléas de dernière minute, ce qui est le cas lors de problèmes dynamiques de tournées de véhicules et d'ordonnements en temps réel.

1.1.4 Gestion de la chaîne logistique

D'après Simchi-Levi *et al.* [91], « la gestion d'une chaîne logistique ou *Supply Chain Management* est un ensemble d'approches utilisées pour intégrer efficacement les fournisseurs, les producteurs, les distributeurs, de manière à ce que la marchandise soit produite et distribuée à la bonne quantité, au bon endroit et au bon moment dans le but de minimiser les coûts et d'assurer le niveau de service requis par le client ». Cela dit, la gestion de la chaîne repose fondamentalement sur le pilotage d'ensembles de flux qui traversent ses différents maillons. On les catégorise généralement en 3 types : flux physique, flux d'information et flux financier.

- **Flux physique** : Le flux physique se rapporte aux mouvements de marchandises et de biens au niveau de la chaîne. Ces mouvements trouvent leur origine depuis l'approvisionnement en matières premières en passant par les activités de production et de transformation jusqu'à la livraison au consommateur final. Donc, le flux physique s'appuie fondamentalement sur 3 familles d'activités, qui sont la manutention, la transformation/production et la circulation ;
- **Flux d'information** : Moins palpable que le flux physique, il n'en demeure pas moins qu'il joue un rôle crucial dans l'écoulement des autres flux dans la chaîne. En effet, ce flux représente l'ensemble des transferts et des échanges de données entre les différents acteurs de la chaîne. A priori, on peut considérer les opérations commerciales d'achats et de ventes qui reposent essentiellement sur des échanges de données en termes de spécifications de coûts, délais, qualités et quantités ;
- **Flux financier** : Sur ce flux repose l'ensemble des activités de gestion et de comptabilité de la chaîne. Il peut être matérialisé par des factures clients/fournisseurs, des fiches de paies par exemple. Egalement, il concerne l'ensemble des investissements et des budgets dans une entreprise.

1.2 La logistique du transport

L'un des aspects de premier plan dans la logistique est le transport de marchandises. La logistique reposant notamment sur des mouvements de flux de marchandises, le transport se révèle donc incontournable. Selon Devernay [28], « le transport constitue le véritable "moteur" de cette logistique : sans une organisation de transport fiable, réactive, flexible et économique, la logistique devrait faire appel à des moyens coûteux en surfaces, manutention, magasinage et systèmes d'information, pour satisfaire les exigences de ses consommateurs. Si certaines de ces chaînes restent internes aux établissements bénéficiant de leurs propres unités de production, beaucoup remontent vers des fournisseurs multiples et variés, à travers des prestataires, dont les plus connus sont les transporteurs ».

Durant ces dernières décennies, la mondialisation de l'industrie combinée à l'expansion du commerce international a profondément remodelé les systèmes logistiques. Par voie de conséquence, la gestion des flux de transports est devenue très complexe. De plus, les coûts de transports représentant une part significative dans les dépenses d'exploitation des entreprises, la nécessité de solutions d'optimisation de plus en plus performantes se fait grandissante.

Plus précisément, dans la suite du document, l'intérêt est porté sur le problème de routage des véhicules (Vehicle Routing Problem). Il s'agit d'un problème d'optimisation combinatoire dont la résolution fait appel à des méthodes relevant de la recherche opérationnelle. Dans ces problèmes, un ou plusieurs véhicules doivent couvrir un réseau de transport pour livrer des marchandises à des clients ou couvrir les routes de ce réseau. L'impact des activités de transport sur les coûts en entreprises s'est intensifié à un point tel que des efforts d'optimisation s'imposent. En effet, Toth et Vigo [98] estiment les frais de transport entre 10% et 20% des prix finaux des marchandises sur le marché. Selon eux, une marge de l'ordre de 5% à 20% peut être économisée sur ces coûts de transport, grâce à des méthodes d'optimisation implémentées dans des procédures informatiques.

1.3 Contexte industriel

1.3.1 Le groupe Nestal

1.3.1.1 L'activité de Nestal

La thèse a eu lieu chez Nestal dans le cadre d'une convention CIFRE. Nestal est un groupe agroalimentaire dédié à la nutrition animale et regroupant 3 sociétés. Celles-ci sont Copam (2 usines à Saint-Martin et Velaines), Cadsar (1 usine à Pauvres) et Etienne (1 usine à Dompain) (voir figure 1.3). Une large gamme de produits correspondant à diverses familles d'animaux dans les catégories des bovins, ovins, chevaux, volailles, lapins (...) sont fabriqués dans les usines. Les matières premières sont essentiellement des produits céréaliers (soja, colza, blé, maïs...). Les produits finis, en général, consistent en des mélanges des matières qui sont granulés.

Les outils industriels n'étant pas homogènes, on ne peut produire tout partout. Les processus de production ne sont pas identiques dans tous les sites et surtout les capacités de production des sites diffèrent. En termes de production, les capacités journalières peuvent varier de 350 tonnes (Etienne) à 450 tonnes (Copam Saint-Martin, Cadsar) d'une usine à l'autre. Le rythme des activités de production connaît deux grands régimes. Un régime élevé qui peut aller jusqu'à la saturation des outils de production de certains sites en hiver (Octobre – Mars) et un régime moyen voire faible en été (Avril – Septembre). La production au niveau des usines de Nestal se fait sur commandes, les produits fabriqués correspondent à un carnet de commandes. Le système de production fonctionne à flux tirés et l'horizon de planification de la production est de moins de 24 heures (en général).

1.3.1.2 Le transport chez Nestal

Au niveau de NESTAL, le transport s'effectue à divers niveaux de la chaîne logistique. En amont, il concerne les réceptions de matières premières, en aval, les livraisons aux clients (particuliers, éleveurs, coopératives...). Egalement, s'ajoutent à la liste les livraisons inter-sociétés.

Il existe trois familles de ressources de transport qui sont utilisées : les camions-citernes, les camions-bennes et les camions-savoyardes. Les camions-citernes sont généralement utilisés

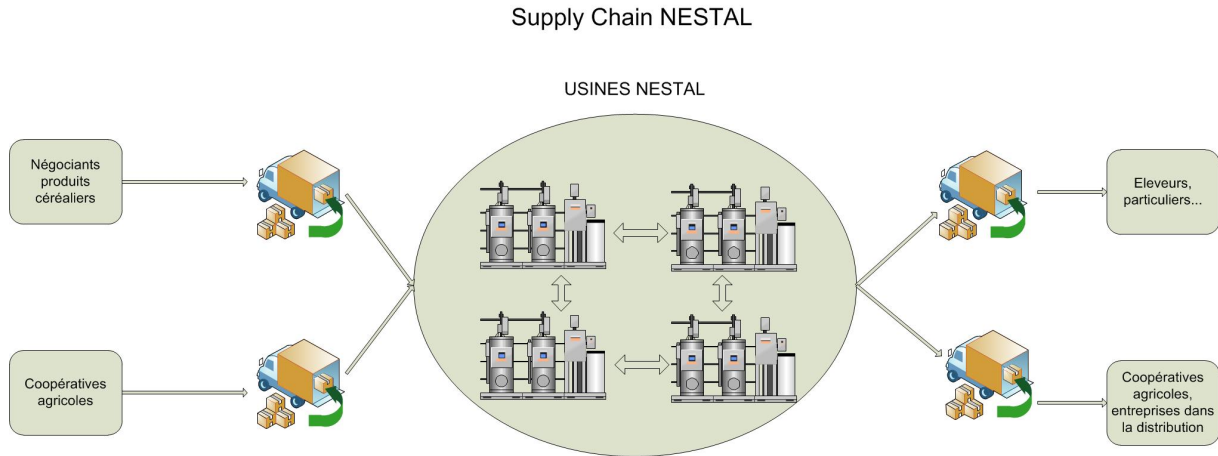


FIGURE 1.3: Schéma de la chaîne logistique de Nestal

dans les livraisons de produits en vrac en moyennes et grandes quantités (de l'ordre de 2 jusqu'à 26 tonnes). Les camions-bennes sont, en général, utilisées pour la réception de matières premières et la distribution de produits en vrac et en grandes quantités (20 à 30 tonnes) aux clients. Les camions-savoyardes ou semi-remorques de type "tautliner" servent à la réception et à la livraison de produits de négoce et de produits fabriqués et conditionnés aux clients pour des lots allant de quelques centaines de kilos à 22 tonnes en général. Les remorques des camions-savoyardes possèdent des parois souples et coulissantes sous forme de bâches, ce qui facilite les opérations de chargements et de déchargements.

1.3.1.3 Positionnement de la problématique industrielle

La volonté d'optimisation des flux allant des commandes des clients aux livraisons des produits est au cœur de la problématique industrielle. Cette problématique est centrée sur la logistique du transport. En effet, étant donnée la part importante des frais de transport dans les dépenses d'exploitations, l'optimisation des flux de distribution chez NESTAL constitue un enjeu stratégique majeur.

Les livraisons chez NESTAL sont assurées, en majeure partie, par des tournées de véhicules. En général, les problèmes de tournées de véhicules ou *Vehicle Routing Problems* (VRP) sont très présents dans la distribution. À partir de dépôt(s) où sont basés des véhicules, il s'agit de livrer des quantités connues de produits à un ensemble de clients. L'objectif consiste à déterminer, pour chaque véhicule, les clients desservis et l'ordre dans lequel ils sont visités en vue de minimiser le kilométrage parcouru, le temps de roulage ou d'autres critères.

Les livraisons effectuées lors des tournées de véhicules sont assurées, en général, par des camions-citernes compartimentés. Chaque compartiment est amené à recevoir une quantité d'un unique produit devant être livré en totalité à un seul client. Dans la littérature, il y a eu peu de travaux sur les problèmes de tournées de véhicules compartimentés ou *multi-compartment vehicle routing problems (MC-VRP)*. Un état de l'art récent a été réalisé par Derigs *et al.* [25]; ils ont souligné le fait que la plupart des travaux de recherche dans ce domaine se sont largement concentrés sur l'approvisionnement et la distribution de carburants. Concernant l'agroalimentaire, il existe une très maigre littérature traitant de ce type de problèmes. Néanmoins, certaines contraintes liées aux particularités de l'industrie agroalimentaire ajoutent une couche de complexité aux problèmes de tournées, qui nécessite d'être explorée par des approches scientifiques. Par exemple, les contraintes de contaminations croisées, qui obligent les transporteurs à séparer des produits non compatibles, peuvent poser des problèmes ardues d'affectations en plus des problèmes de routage.

Du point de vue scientifique, l'originalité de la problématique repose surtout sur la spécificité du chargement des lots aux compartiments des véhicules chez Nestlé. Dans le contexte en question, un lot correspond à une quantité d'un produit demandé par un client. D'un côté, il n'existe pas d'affectation prédéfinie d'une demande d_{ip} d'un client i pour un produit p à un ou des compartiments donnés dans un camion. D'autre part, le lot peut être scindé pour être réparti dans plus d'un compartiment d'un seul véhicule. En effet, une problématique de MC-VRP combinant ces deux aspects a été très peu étudiée, à notre connaissance, dans la littérature.

De nos jours, bien qu'il existe des logiciels commerciaux conçus pour la planification des tournées de véhicules, il n'en demeure pas moins qu'ils se révèlent souvent limités quand il s'agit de prendre en compte les contraintes spécifiques à certains métiers. L'objectif recherché par les éditeurs étant de fournir aux utilisateurs des résultats d'assez bonne qualité dans un délai relativement court, ils utilisent, généralement, des techniques d'optimisation de qualité moyenne. Sauf que de nos jours, les entreprises confrontées, aux défis de la concurrence et des enjeux d'ordre économique, sont de plus en plus demandeuses d'outils d'aide à la décision fournissant des résultats de très bonne qualité. Dès lors, les travaux de recherche en optimisation suscitent un intérêt sans cesse grandissant dans le milieu industriel.

1.4 Objectifs de la thèse

L'environnement dans lequel nos travaux se situent est celui de la logistique du transport dans l'industrie agroalimentaire. Le problème concerne l'optimisation des tournées de véhicules compartimentés. Nous intégrons certaines contraintes des métiers spécifiques à l'agroalimentaire dans la planification et l'optimisation des tournées de véhicules. En particulier, nous avons pris en compte des contraintes imposées par des mesures de prévention de la contamination croisée dans certains de nos modèles.

Le MC-VRP constitue le noyau dur de nos travaux de recherche. Parmi les contraintes, nous incluons des contraintes spécifiques au métier de notre partenaire industriel. Jusqu'à présent, la littérature en MC-VRP concerne principalement la distribution de carburants dans le secteur pétrolier. Néanmoins, dans le cadre de cette thèse, nous étudions une application du MC-VRP dans l'industrie agroalimentaire et, plus précisément, dans la nutrition animale.

La suite de la thèse est organisée de la manière suivante. Au deuxième chapitre, un état de l'art sur la logistique du transport routier ainsi que les approches de résolutions sont présentés. L'objectif étant de rappeler les concepts de base et le formalisme de modélisation nécessaires à la compréhension du problème classique de VRP. Les approches de résolution les plus connues seront décrites également.

Le troisième chapitre étudie un problème de MC-VRP avec des spécificités dans les modalités de chargement. Il s'agit d'un MC-VRP avec un compartiment de taille fixe dédié à chaque produit. Suite à une définition mathématique du problème, deux méthodes de résolution approchées sont proposées : le *Greedy Randomized Adaptive Search Procedure* (GRASP) et l'*Iterated Local Search procedure* (ILS)

Le quatrième chapitre est consacré à l'étude d'une généralisation du MC-VRP abordé au troisième chapitre, le problème de tournées de véhicules à compartiments flexibles. Les travaux relatifs à ce problème dans la littérature seront analysés. En plus d'une résolution exacte, les métaheuristiques GRASP et ILS seront utilisées pour sa résolution.

Le cinquième chapitre se focalise sur l'étude du MC-VRP avec décisions d'affectation de chaque compartiment à un client et un produit. A chaque compartiment, une demande d_{ip} d'un client i pour un produit p est affectée. Cette problématique se rencontre, en particu-

lier, dans la distribution dans le secteur de la nutrition animale. La problématique traitée est largement inspirée du contexte industriel chez Nestlé. La métaheuristique *Multi Start* ILS ou MS-ILS à laquelle est intégrée une heuristique d'optimisation du chargement a été développée pour résoudre le problème.

Le sixième chapitre est dédié à des travaux d'application industrielle. Nous avons appliqué des techniques d'optimisation des tournées de véhicules en vue de résoudre des problématiques de distribution chez Nestlé. Nous avons également évalué les performances d'un outil de planification de tournées chez Nestlé en comparant les résultats de tests effectués sur des instances industrielles.

La conclusion présente une synthèse des travaux réalisés et propose des perspectives de recherches futures.

La figure 1.4 présente la structure générale du mémoire. Elle représente les différentes problématiques de MC-VRP dans le cadre de cette thèse qui sont résolues par des méthodes exactes et approchées.

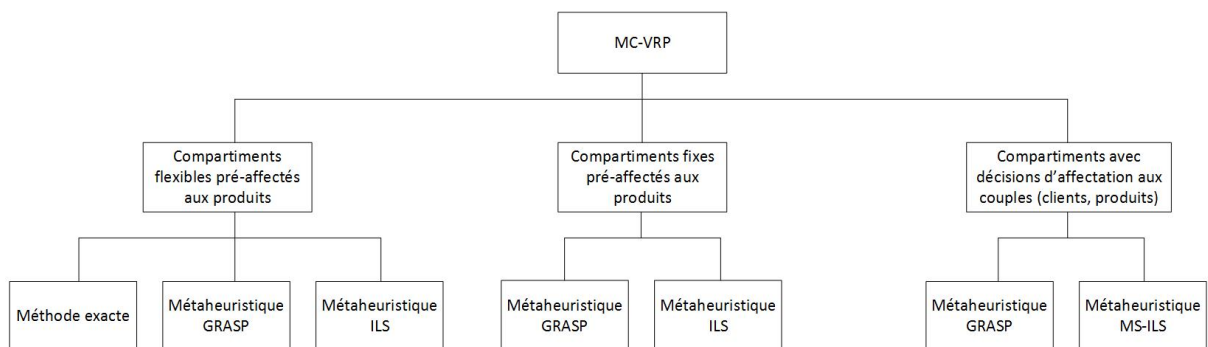


FIGURE 1.4: Plan des problématiques traitées dans le document

Chapitre 2

État de l'art

2.1 Introduction

Les activités du transport routier se sont considérablement accrues avec le développement des réseaux commerciaux au cours des dernières décennies. Cette forte croissance a eu pour conséquence, entre autres, de rendre beaucoup plus complexe la gestion des flux de transport. Depuis le début de la seconde moitié du 20^{ème} siècle, des spécialistes de la recherche opérationnelle se sont penchés sur la résolution des problèmes de transport routier. Dans leurs travaux, ils ont développé un grand nombre de méthodes de résolution. Dans ce chapitre, nous décrivons le problème de tournées de véhicules avec ses principales variantes. Nous soulignons l'évolution du problème de tournées de véhicules depuis ses origines dans les premiers travaux (Dantzig et Ramser [24], par exemple) jusqu'à ses variantes les plus récentes qui sont étudiées. Nous présentons également les méthodes de résolution qui sont proposées.

2.2 Problèmes de tournées de véhicules

Les problèmes de transport routier se trouvent au cœur des problématiques de la logistique. Avant de parvenir au consommateur final, un produit parcourt, en général, des centaines voire des milliers de kilomètres par voie terrestre. Les contextes et les moyens de transport n'étant pas toujours identiques, les problèmes de transport présentent des caractéristiques diverses. On distingue, en général, les problèmes où un véhicule est dédié à chaque client avec des trajets directs aller et retour (*truckload transportation* ou TL) et les problèmes avec des quantités moyennes livrées plus petites, ce qui permet à chaque véhicule de visiter plusieurs

clients (*less than truckload transportation* ou LTL). La première famille contient surtout des problèmes de flot de coût minimum tandis que la seconde est formée des problèmes de tournées. Dans la suite, nous nous concentrons sur les problèmes de tournées.

2.2.1 Les différents paramètres

2.2.1.1 Les véhicules

Les véhicules constituent les principales ressources indispensables à l'organisation du transport. Ils sont disponibles en quantité limitée et disposent également de capacité limitée. Ils transportent des poids qui sont limités soit à cause de contraintes techniques ou des réglementations en vigueur. Ils partent d'un dépôt et reviennent à leur point de départ ou à un autre dépôt.

2.2.1.2 Les demandes

Chaque client exprime une demande de livraison concernant un ou plusieurs produits. Ces derniers sont demandés en quantités bien définies. Le client précise également le lieu de livraison et, dans certains cas, les horaires de livraison.

2.2.1.3 Les contraintes

Les tournées de véhicules sont assujetties à un ensemble de contraintes. Elles peuvent être de plusieurs ordres.

Les contraintes les plus courantes sont celles concernant la capacité. La capacité d'un véhicule étant limitée, il est nécessaire d'augmenter le nombre de véhicules lorsqu'il faut transporter de grandes quantités. Les contraintes de capacité d'un véhicule peuvent, parfois, être spécifiques au compartimentage. Par exemple, des produits à congeler seront forcément transportés dans le compartiment froid d'un camion frigorifique à deux compartiments (compartiments froid et frais). La capacité du camion pour les produits à congeler est forcément celle du compartiment correspondant.

Une autre famille de contraintes concerne le facteur temporel. Le paramètre temporel est un facteur déterminant dans beaucoup de problèmes de tournées de véhicules. En effet, les opérations de livraisons et/ou de chargements requièrent un certain temps qui peut être proportionnel ou non aux quantités de produits. A cela peuvent s'ajouter des temps de préparatifs, une fois arrivé aux points de distribution et/ou de collecte. Il peut s'agir en l'occurrence du temps passé avec le client pour des formalités administratives, du temps de préparation du matériel... D'autre part, on peut rencontrer des contraintes horaires liées à la disponibilité des clients (horaires d'ouverture, de présence...). Plus précisément, on les appelle contraintes de fenêtres horaires. Ces contraintes peuvent être dures mais parfois molles (on peut les violer au prix de pénalités). Elles ont souvent un impact significatif sur l'ordonnancement des tournées dans la mesure où elles imposent une date de début et/ou de fin pour les visites chez les clients.

2.2.1.4 La fonction-objectif

La fonction-objectif consiste en une fonction de coût à minimiser ou une fonction de profit à maximiser. En ce qui a trait aux problèmes de VRP, on recherche généralement les objectifs suivants :

- Minimisation des distances parcourues : la somme des distances parcourues au cours des tournées est minimisée ;
- Minimisation des durées des tournées : l'objectif recherché est la minimisation des temps de route ;
- Minimisation du nombre de ressources de transport : le nombre de véhicules utilisés pour effectuer les tournées est minimisé.

Bien entendu, ces différents critères sont parfois convertis en coûts et additionnés.

2.2.2 Les principaux problèmes de tournées de véhicules

Dans le contexte logistique, le transport routier permet, avant tout, de relier le(s) producteur(s) et/ou vendeur(s) au(x) demandeurs(s). Cela dit, le transport routier assure des mouvements de flux entre offreurs et demandeurs. Dans un schéma classique, d'une part, les demandeurs ou clients sont caractérisés au moins par leurs demandes et leurs localisations

géographiques et d'autre part, le dépôt doit livrer les produits demandés au moyen de ressources de transport (personnel et/ou véhicules). Etant donné qu'il existe de nombreux schémas de tournées de livraisons possibles pour faire parvenir les produits aux clients, l'objectif recherché, lors de la résolution des problèmes de tournées, est de construire des tournées permettant de réduire au mieux les coûts de transport compte tenu des contraintes. Ces contraintes, telles que les capacités des ressources de transport, effectif des ressources disponibles, la législation, etc., sont, bien sûr, à prendre en compte. Celles-ci confèrent, d'ailleurs, une réelle complexité à la résolution des problèmes de tournées. Complexité qui sera explicitée au travers de la présentation des problèmes de tournées qui ont été étudiés dans la littérature.

De manière formelle, pour représenter les problèmes de transport, on fait appel aux graphes, ceux-ci étant, par définition, adaptés pour représenter les réseaux. En ce qui a trait à un réseau de transport de type VRP, on considère un graphe non orienté comprenant, d'une part, un ensemble de nœuds ou sommets et, d'autre part, un ensemble d'arêtes. Concrètement, le graphe G est défini par l'expression $G = (V, E, C)$ avec :

- $V = \{0, 1, \dots, n\}$ qui désigne l'ensemble des sommets ou nœuds, 0 représentant le nœud-dépôt ;
- $E = \{[i, j] : i, j \in V, i \neq j\}$ correspondant à l'ensemble des arêtes qui relient les différents nœuds ;
- $C = \{c_{ij} : i, j \in V\}$ une matrice qui définit pour chaque arête $[i, j]$ un coût de traversée qui peut être interprété comme la distance ou le temps de parcours, par exemple.

Concrètement, un nœud dans un graphe peut désigner un dépôt, un client ou un carrefour c'est-à-dire, en quelque sorte, tout endroit qu'un véhicule dessert sur son parcours. Une arête $[i, j]$ représente une route, un tronçon de route ou une rue reliant dans les deux sens les nœuds i et j (voir figure 2.1 illustrant un graphe sur un réseau routier reliant des clients à un dépôt). La connexion entre deux nœuds peut avoir une orientation prédéfinie comme dans le cas d'une rue à sens unique par exemple. Dans ce cas, nous utilisons un graphe orienté $G = (V, A, C)$ avec A désignant la famille des arcs. Chaque arc constitue un couple ordonné de nœuds. Entre les nœuds i et j , on peut avoir les arcs (i, j) ou (j, i) . Il en découle également un coût de déplacement spécifique à chaque arc tenant compte de l'orientation entre les sommets. Au niveau des réseaux routiers, en milieu rural, les rues à sens uniques sont pratiquement inexistantes, ce qui n'est pas forcément le cas dans les milieux urbains.

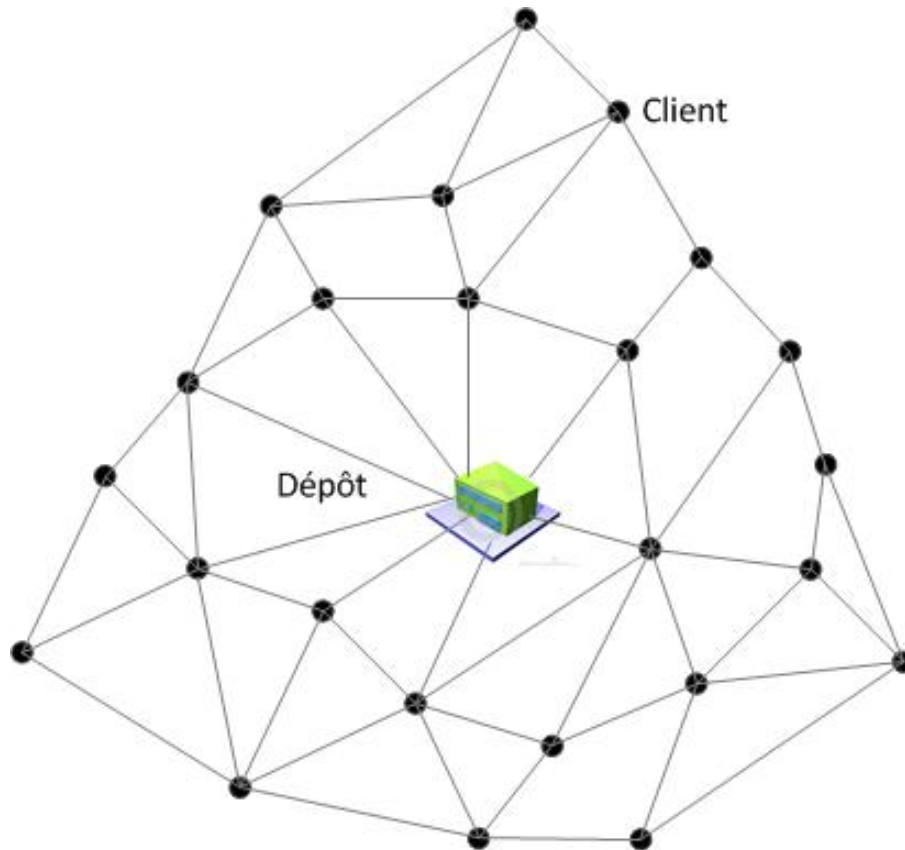


FIGURE 2.1: Graphe d'un réseau routier reliant des clients à un dépôt

A noter que pour des raisons de simplicité et pour réduire la taille des données traitées lors de calculs d'optimisation, le graphe d'origine est réduit pour ne garder que les nœuds désignant les clients et le dépôt. On obtient ainsi un graphe complet et valué qui est utilisé pour les méthodes de résolution. Les plus courts chemins sont calculés entre les nœuds et un graphe complet est obtenu où les connexions de chaque sommet à tous les autres sont représentés. Les calculs du plus court chemin peuvent s'appuyer sur deux méthodes : l'algorithme de Dijkstra [29] utilisant la technique de *fixation d'étiquettes*, et l'algorithme de Bellman [9] se basant sur la *correction d'étiquettes*.

2.2.2.1 Le problème du voyageur de commerce

Le problème du voyageur de commerce ou *Traveling Salesman Problem* (TSP) nait d'une problématique vécue par des vendeurs ou commerciaux qui se déplacent pour livrer ou rencontrer leurs clients. En effet, un vendeur a pour mission de visiter n villes et retourne à son point de départ après les avoir visitées une et une seule fois chacune. L'objectif du vendeur,

qui visite toutes les villes, est de définir un parcours qui minimise la distance totale. Plus formellement, un TSP est modélisé sous la forme d'un graphe où les villes sont représentées par des nœuds, et les liaisons entre elles par des arêtes. Dans le cadre de ce problème, un vendeur part d'un dépôt pour livrer les demandes des clients et rentre après avoir livré le dernier client. Le TSP peut être décomposé en ces différents éléments :

- Un ensemble de clients avec des demandes et des adresses connues.
- Une localité ou un dépôt d'où part et revient la tournée, c'est-à-dire le parcours d'un véhicule qui visite un ensemble de clients avant de revenir au dépôt.
- Une ressource de transport (humain, véhicule, ...)

La résolution de ce problème dans un graphe revient à trouver un cycle de coût minimal qui passe par chaque nœud précisément une seule fois. Il s'agit donc de trouver un cycle hamiltonien de coût minimal. A noter ici que la notion de coût peut refléter la distance, le temps ou le coût économique. Dans le problème en question, le voyageur peut tout aussi bien être considéré comme étant un véhicule devant livrer des produits à des clients situés dans des endroits géographiques distincts. Dans ce cas, la capacité du véhicule est nécessairement supérieure ou égale à la somme des demandes des clients.

En optimisation, ce problème est connu comme étant *NP-difficile*. Lorsque les distances sont symétriques sur les arêtes c'est-à-dire que les distances sont identiques quel que soit le sens de traversée des arêtes, le nombre total de solutions possibles, dans un graphe non orienté, est de l'ordre de $\frac{(n-1)!}{2}$ où n est le nombre de villes. Dans la littérature, une formulation très utilisée, a été proposée par Dantzig *et al.* [23]. Le TSP est défini sur un graphe $G = (V, A, C)$ où V désigne l'ensemble des sommets, comprenant un ensemble de clients $V' = \{1, \dots, n\}$ et un nœud-dépôt 0. A correspond à l'ensemble des arcs (i, j) avec $i, j \in V$. $C = \{c_{ij}\}$ est une application définissant les coûts de traversée des arcs. La formulation mathématique du TSP requiert la définition d'une variable de décision $x_{i,j}$.

$$x_{ij} = \begin{cases} 1 & \text{si le voyageur traverse l'arc } (i, j) \text{ où } (i, j) \in A \\ 0 & \text{sinon} \end{cases} \quad (2.1)$$

Une formulation classique du TSP en programme linéaire en nombres entiers peut être définie comme suit :

Fonction-objectif :

$$\min \sum_{(i,j) \in A} x_{ij} c_{ij} \quad (2.2)$$

Sous contraintes :

$$\sum_{i \in V} x_{ij} = 1 \quad \forall j \in V \quad (2.3)$$

$$\sum_{j \in V} x_{ij} = 1 \quad \forall i \in V \quad (2.4)$$

$$\sum_{i \in S} \sum_{j \in S, j \neq i} x_{ij} \leq |S| - 1 \quad \forall S \subset V', |S| \geq 2 \quad (2.5)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (2.6)$$

La fonction-objectif (2.2) cherche à minimiser le coût total du circuit parcouru. Les contraintes (2.3) et (2.4) imposent que le voyageur entre et sorte une seule fois de chaque nœud. L'élimination des sous-tours, c'est-à-dire des trajets passant par un sous-ensemble de nœuds sans retour au dépôt, est assurée par les contraintes (2.5).

La formulation du problème telle qu'exprimée à travers ce modèle ne permet pas sa résolution dans un délai raisonnable par un logiciel de programmation linéaire. En effet, le nombre des contraintes de sous-tours (2.5) est exponentiel ($2^n - 2$). Cela dit, dans la pratique, d'autres formulations permettant une résolution informatique dans des délais raisonnables sont utilisées. En particulier, une formulation proposée par Miller *et al.* [69] ne nécessite que $n^2 - 3n + 2$ contraintes.

Il existe de nombreuses variantes de ce problème dans la littérature (Applegate *et al.* [3]). En particulier, le problème de voyageurs de commerce multiples, également connu sous le nom de *multiple Traveling Salesman Problem* (mTSP), a été étudié, par exemple par Bektas [8]. Le mTSP consiste en une généralisation du TSP. Dans cette version, un nombre m de voyageurs sont à l'origine localisés au nœud-dépôt. L'objectif est de calculer le nombre m de tournées au départ du dépôt qui visitent une seule fois les nœuds clients et qui se terminent au dépôt. La somme des coûts des tournées obtenues doit être minimale.

Le mTSP s'est révélé assez proche d'autres variantes de problèmes de routage. En particulier, l'addition de quelques contraintes supplémentaires permet de le transformer en un pur problème de tournées de véhicules ou *Vehicle Routing Problem* (VRP). Pour ce, on peut, par exemple, considérer que les voyageurs sont des véhicules de capacité limitée, ce qui donne un problème de VRP avec contraintes de capacité ou *Capacitated VRP* (CVRP). Etant donnée sa proximité au VRP, la résolution du mTSP a été utilisée par certains auteurs (Mole *et al.* [72], Laporte *et al.* [60]) dans le cadre d'une approche hiérarchique pour la résolution de problèmes de VRP beaucoup plus complexes.

2.2.2.2 Le problème de tournées de véhicules (VRP)

Le VRP consiste en un problème d'optimisation des parcours d'un nombre k de véhicules devant effectuer des tournées de livraisons à n clients. Il a été introduit par Dantzig et Ramser [24] en 1954 sous le nom de "*Truck Dispatching Problem*".

Ce nom, peu précis, est souvent remplacé par, CVRP (*Capacitated VRP ou VRP avec contraintes de capacité*). Concrètement, une flotte de k véhicules identiques de capacité Q , basée au dépôt, doit effectuer des tournées de livraisons à des nœuds-clients. Chaque client est livré d'une quantité finie et est visité une seule fois par un véhicule de la flotte qui retourne au dépôt à la fin de sa tournée.

Le problème de VRP peut être modélisé mathématiquement sous la forme d'un programme linéaire en nombres entiers. Le VRP est défini sur un graphe orienté complet mais sans boucles $G = (V, A)$. Les clients sont représentés par des nœuds $i \in V' = V - \{0\}$ et expriment chacun une demande d_i . c_{ij} désigne le coût de parcours de l'arc (i, j) . La variable de décision utilisée est ainsi définie :

$$x_{ijk} = \begin{cases} 1 & \text{si le véhicule } k \text{ traverse l'arc } (i, j), \text{ avec } (i, j) \in A \text{ et } k \in K \\ 0 & \text{sinon} \end{cases} \quad (2.7)$$

Voici une des formulations existantes du CVRP en PLNE :

Fonction-objectif :

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk} \quad (2.8)$$

Sous contraintes :

$$\sum_{i \in V} x_{ijk} = 1 \quad \forall j \in V', \forall k \in K \quad (2.9)$$

$$\sum_{j \in V} x_{ijk} = 1 \quad \forall i \in V', \forall k \in K \quad (2.10)$$

$$\sum_{j \in V} x_{jik} = \sum_{j \in V} x_{ijk} = y_{ik} \quad \forall i \in V', \forall k \in K \quad (2.11)$$

$$\sum_{i \in V'} d_i \left(\sum_{j \in V} x_{ijk} \right) \leq Q \quad \forall k \in K \quad (2.12)$$

$$\sum_{j \in V} x_{0jk} \leq 1 \quad \forall k \in K \quad (2.13)$$

$$\sum_{i \in V} x_{i0k} \leq 1 \quad \forall k \in K \quad (2.14)$$

$$\sum_{i \in S} \sum_{j \in S, j \neq i} x_{ijk} \leq |S| - 1 \quad \forall S \subset V', |S| \geq 2, \forall k \in K \quad (2.15)$$

$$x_{ijk} \in \{0, 1\} \quad \forall (i, j) \in A, \forall k \in K \quad (2.16)$$

La fonction-objectif (2.8) cherche à minimiser la somme des coûts de toutes les tournées. Chaque client est desservi une seule fois conformément aux contraintes (2.9) et (2.10). La conservation de flot est assurée par (2.11); tout véhicule entrant dans un nœud doit le quitter. La somme des demandes livrées par n'importe quel véhicule ne doit, en aucun cas, dépasser sa capacité (2.12). Les contraintes (2.13) et (2.14) garantissent que chaque tournée commence et se termine au dépôt. Les contraintes (2.15) empêchent les sous-tours. Il existe aussi des versions avec un nombre polynomial de contraintes similaires aux contraintes de Miller *et al.* [69] pour le TSP. Les contraintes (2.16) sont des contraintes de binarité des x_{ijk} .

2.2.2.3 Variantes du VRP

Le VRP de base est, avant tout, une version simplifiée au regard de l'ensemble des problèmes de tournées de véhicules que l'on peut rencontrer dans la réalité. En effet, en fonction des contraintes à prendre en compte, plusieurs variantes du VRP ont été définies à travers les travaux de la littérature. Un certain nombre de paramètres permettant de classifier les problèmes de tournées ont été recensés dans les travaux de Bodin et Golden [13], Desrochers *et al.* [26] et [32]. Ces paramètres sont principalement :

1. Distances entre les nœuds :
 - (a) Distances symétriques
 - (b) Distances non symétriques
2. Nombre de dépôts :
 - (a) Un seul dépôt
 - (b) Deux ou plusieurs dépôts
3. Composition de la flotte :
 - (a) Véhicules homogènes
 - (b) Véhicules hétérogènes
4. Taille de la flotte :
 - (a) Un seul véhicule
 - (b) Plus d'un véhicule
5. Nature des demandes :
 - (a) Déterministe
 - (b) Stochastique
6. Longueur des tournées
 - (a) Restreinte
 - (b) Non restreinte
7. Durée des tournées
 - (a) Restreinte
 - (b) Non restreinte
8. Nombre de livraisons/visites à un client
 - (a) Unique

- (b) Multiple
- 9. Horaires de livraisons
 - (a) Non définis
 - (b) Définis de manière précise (rendez-vous)
 - (c) Définis dans des intervalles horaires souples (soft time windows)
 - (d) Définis dans des intervalles horaires rigides (hard time windows)
- 10. Nombre de tournées admises par véhicule
 - (a) Unique
 - (b) Multiple
- 11. Calcul des coûts de transport
 - (a) Fonction de la distance
 - (b) Fonction du temps
 - (c) Fonction du nombre de véhicules (avec possibilité d'achats)
- 12. Répartition des charges dans les véhicules
 - (a) Compartiment unique
 - (b) Multi-compartiments
- 13. Opérations de collectes
 - (a) Collectes et livraisons simultanées chez les clients
 - (b) Soit collecte soit livraison chez un client
- 14. Horizon temporel
 - (a) Mono-période
 - (b) Multi-périodes

Parmi les variantes les plus connues, en voici quelques-unes :

- ***Distance-Constrained Vehicle Routing Problem (DCVRP)*** : Le DCVRP se définit comme un CVRP auquel on ajoute une limitation de distance pour les tournées. L'objectif étant de faire des tournées de livraisons aux clients, il faut faire en sorte que chaque tournée soit de longueur inférieure ou égale à une distance définie. Entre autres, les auteurs Li *et al.* [63] ont proposé dans leurs travaux une méthode de résolution pour ce problème.

- ***Vehicle Routing Problem with Time Windows (VRPTW)*** : Le VRPTW ou problème de tournées de véhicules avec fenêtres horaires a pour but de déterminer des tournées de livraisons dans des intervalles horaires prédéfinis. Outre les contraintes du CVRP, les fenêtres horaires liées aux livraisons constituent la particularité du problème (Nagata et Bräysy [73], Labadi *et al.* [57]). Ces intervalles de temps peuvent être définis de manière rigide ou flexible. La flexibilité vient du fait d'admettre que les bornes peuvent varier dans une certaine mesure contrairement au cas où elles sont rigides.
- ***Vehicle Routing Problem with Backhauls (VRPB)*** : Le VRPB (ou *problème de tournées de véhicules avec collectes*) inclut, outre des livraisons, des collectes dans les tournées de livraisons. Il est question ici de répartir l'ensemble des clients en deux sous-ensembles : un groupe de clients à livrer et un autre groupe chez lesquels il faut faire des collectes. Pour résoudre ce problème, les livraisons sont effectuées en premier et les collectes en second pour éviter de réarranger les chargements des véhicules (Gajpal et Abad [38], Belmecheri *et al.* [11]).
- ***Vehicle Routing Problem with simultaneous Pick-up and Delivery (VRPPD)*** : Le VRPPD (problème de tournées de véhicules avec collectes et livraisons simultanées) a été introduit par Min [70]. A la différence du VRPB, lors du service chez chaque client, la livraison et la collecte sont effectuées simultanément, ce qui constitue la principale caractéristique de ce problème (Tang Montané et Galvão [95]).
- ***Multi-Depot Vehicle Routing Problem (MDVRP)*** : Il s'agit d'un problème de tournées où il existe plusieurs dépôts de départ qui sont localisés en différents endroits. Les véhicules partent de n'importe quel dépôt mais reviennent à leurs dépôts de départ (Ho *et al.* [46]).
- ***Heterogeneous fleet Vehicle Routing Problem (HVRP ou HFVRP)*** : Le HVRP a pour particularité une flotte de véhicules non uniformes (Gendreau *et al.* [40], Prins [84]). Cette différence peut être liée à plusieurs aspects tels que la capacité, les coûts d'utilisation etc.
- ***Periodic Vehicle Routing Problem (PVRP)*** : Le PVRP consiste en un VRP qui implique la construction de tournées sur une période de planification donnée tout en desservant les clients plus d'une fois (Vidal *et al.* [101]). Ce cas de figure arrive, généralement, lorsque des clients ont des demandes de gros volumes pour des capacités de stockage faibles.
- ***Stochastic Vehicle Routing Problem (SVRP)*** : Un problème de tournées est considéré stochastique dès lors que les valeurs d'un ou de plusieurs paramètres sont aléatoires. Généralement, des cas avec des demandes stochastiques sont étudiés (Novoa et

Storer [75]). Il est à noter que l'aspect aléatoire peut être attribué à d'autres paramètres tels que le nombre de clients, leur présence ou absence, le temps et le coût de transport.

- ***Dynamic Vehicle Routing Problem (DVRP)*** : Le DVRP est un problème où la planification des tournées est effectuée de manière dynamique. Elle est mise à jour au fur et à mesure de l'obtention de nouvelles informations sur les clients à livrer (nouveaux clients, par exemple) (Pillac *et al.* [80]). En cours de route, les tournées sont modifiées pour prendre en compte les nouveaux éléments.
- ***Multi-Compartment Vehicle Routing Problem (MC-VRP)*** : Lorsque dans un problème de tournées, les charges doivent être séparées dans un même véhicule, on utilise, en général, des véhicules compartimentés. Ces véhicules ont l'avantage de permettre le transport de produits incompatibles. Le but, comme dans le cas, d'un CVRP, consiste à construire les tournées au moindre coût en répartissant les charges dans les compartiments des véhicules (Derigs *et al.* [25], El Fallahi *et al.* [33]).

2.3 Le MC-VRP

Le MC-VRP ou *Multi-Compartment Vehicle Routing Problem* consiste en une généralisation du VRP qui concerne plus particulièrement cette thèse. Dans ce problème, les véhicules utilisés sont dotés de compartiments dans lesquels les produits sont répartis. En effet, des contraintes d'incompatibilité entre certains produits peuvent rendre le recours à des camions compartimentés nécessaire ou, du moins, plus bénéfique. Le MC-VRP est très présent dans les industries chimiques, alimentaires et pétro-chimiques, par exemple. En dépit de son application industrielle très large, ce problème a été peu étudié dans la littérature du VRP. En général, les travaux sur le MC-VRP se sont focalisés sur la distribution de carburants à des stations-services ou à des particuliers. Ces travaux, ne se limitant pas toujours au seul problème de transport, traitent parfois de la planification des approvisionnements.

Brown et Graves [15] ont développé une méthode en vue de la résolution d'un problème de minimisation des coûts de transport de produits pétroliers des terminaux de stockage vers les clients. Chaque commande correspondant à la charge d'un véhicule rempli, les routes construites ne sont constituées que d'un seul client. Les auteurs ont proposé, d'une part, une approche exacte, la programmation linéaire en nombres entiers, d'autre part, une approche reposant sur une heuristique qui résout des sous-problèmes d'affectation des commandes

aux véhicules en tenant compte des contraintes (incompatibilité produits, flotte hétérogène, ...). Brown *et al.* [14] ont étudié le même problème tout en considérant que, sur une route, plusieurs clients peuvent être visités. La distribution est planifiée en quatre phases : déterminer depuis quel terminal chaque commande doit être livrée, affecter chaque commande à un véhicule, ajuster les quantités à livrer aux capacités des véhicules et construire les routes.

Van der Bruggen *et al.* [100] ont résolu un problème de distribution de produits pétroliers. Ils ont proposé une approche hiérarchique en vue d'optimiser l'infrastructure de distribution entre des dépôts et des stations-services. Le problème global est décomposé en trois sous-problèmes de localisation (ouverture et fermeture de dépôts), affectation (associer les stations aux dépôts) et de tournées de véhicules (planification des livraisons et construction des routes). Les problèmes sont résolus de manière séquentielle. Après avoir appliqué une relaxation lagrangienne, une heuristique basée sur la méthode d'insertion de Gendreau *et al.* [39] a été utilisée pour résoudre le problème de construction de routes.

Avella *et al.* [4] ont considéré un problème de MC-VRP multi-périodique. Lors du chargement des camions, une contrainte exige que chaque compartiment soit entièrement rempli ou vide. Après avoir modélisé le problème comme un problème de partitionnement, une approche de résolution basée sur une heuristique et un algorithme de *branch and price* est appliquée.

Cornillier *et al.* [22] ont étudié un problème où il s'agit de livrer des produits pétroliers transportés par une flotte hétérogène de camions-citernes compartimentés. Sur une période donnée, un problème d'affectation des produits aux compartiments et un problème de tournées de véhicules ont été résolus. Une approche exacte a été développée par les auteurs. La partie affectation a été résolue au moyen d'un algorithme d'affectation dans un graphe biparti. Le problème de routage a été résolu à l'aide d'une approche de couplage de coût minimal dans un graphe non biparti.

Par ailleurs, outre l'industrie pétrolière, d'autres secteurs d'activités tels que le commerce, la distribution et la collecte d'ordures ont été pris en compte dans les principaux travaux sur le MC-VRP.

Chajakis et Guignard [17] se sont penchés sur un problème d'approvisionnements de boutiques. L'objectif consiste à minimiser les coûts de distribution et de réfrigération des produits. La flotte est composée de véhicules compartimentés où les produits transportés sont

conditionnés à différents niveaux de température (ambiant, frais, froid). Avec une approche de résolution exacte basée sur une relaxation lagrangienne, les auteurs ont cherché à déterminer la configuration des compartiments permettant d'optimiser les coûts.

Dans le secteur de la nutrition animale, Ruiz *et al.* [88] ont développé une approche exacte en deux phases. Durant la première phase, une énumération implicite est utilisée pour générer un ensemble des routes faisables et de bonne qualité. Ensuite, une méthode exacte est proposée en vue de sélectionner un sous-ensemble optimal de tournées.

El Fallahi *et al.* [33] ont étudié un problème de livraisons dans le contexte de l'agro-alimentaire. Il s'agit de livrer de l'aliment destiné au bétail au moyen de camions-citernes. Des règles d'hygiène et de sécurité alimentaires imposent de séparer les familles de produits entre elles (contraintes d'incompatibilité). Considérant que chaque famille de produits ne peut être chargée que dans un seul compartiment, les auteurs ont développé un *tabu search* et un algorithme mémétique pour résoudre le problème. La particularité du problème étudié est que les différentes familles de produits demandées par un client peuvent être apportées par plusieurs véhicules. Par contre, comme d'habitude, les livraisons fractionnées d'une même famille sont interdites.

Dans un contexte stochastique, Mendoza *et al.* [68] ont proposé deux approches basées sur l'algorithme mémétique. La première approche n'intègre pas directement la nature stochastique du problème mais recourt à une stratégie de capacité résiduelle des compartiments pour parer aux fluctuations des demandes. D'autre part, la seconde approche intègre de manière explicite la nature stochastique de la demande dans le processus de construction de route. Leurs expériences ont montré que la seconde approche garantit des solutions de meilleure qualité tandis que la première approche permet de réduire les temps de calculs.

2.4 Méthodes de résolution

Dans le cadre d'une démarche de résolution des problèmes de VRP, plusieurs méthodes ont été développées dans la littérature. En optimisation combinatoire, ces méthodes ont été classées en deux grandes familles (Laporte [59]) : les méthodes exactes et les méthodes heuristiques. Les méthodes exactes garantissent l'obtention d'une solution de la meilleure qualité possible également appelée *optimum*. Par contre, les méthodes heuristiques produisent géné-

ralement des solutions de bonne qualité sans garantir l'optimalité. La famille des méthodes heuristiques renferme plusieurs méthodes qui se classifient en trois catégories : heuristiques constructives, recherches locales et métaheuristiques.

2.4.1 Méthodes exactes

Les méthodes exactes, également appelées méthodes *complètes* ou optimales, procèdent par exploration intelligente des solutions afin d'en trouver une de coût minimal. Plusieurs méthodes exactes ont été proposées dans la littérature. Elles sont basées sur les approches de programmation linéaire, de programmation dynamique et de recherche arborescente.

- **Programmation linéaire** : Cette approche décrit le problème au moyen d'une formulation linéaire appelée Programme Linéaire en Nombres Entiers (PLNE). Cette formulation comprend 3 composantes : (1) l'ensemble des variables avec leurs bornes, des paramètres et hypothèses, (2) une fonction des variables de décision qu'il faut optimiser (fonction-objectif), (3) les contraintes exprimées sous forme d'équations. En ce qui a trait au VRP, plusieurs modèles linéaires ont été définis dont les principaux sont : le modèle de flot à deux indices ou *two-index vehicle flow formulation* (Laporte *et al.* [60]), le modèle de flot de commodités ou *commodity flow formulation* (Baldacci *et al.* [6]) et la formulation en tant que problème de partition d'ensemble ou *set partitioning problem*. Pour résoudre les PLNE, plusieurs logiciels sont proposés dans le commerce dont les plus connus sont CPLEX et XPRESS.
- **Programmation dynamique** : Initialement développée par Bellman [10], la programmation dynamique repose sur une théorie selon laquelle une décision optimale est composée d'un ensemble de sous-décisions optimales. Donc, la résolution d'un problème est effectuée en procédant par décomposition du problème en de petits sous-problèmes qui sont d'abord résolus avant de remonter progressivement à des sous-problèmes plus difficiles.
- **Recherche arborescente** : Cette méthode repose sur l'énumération implicite de l'espace des solutions c'est-à-dire que, contrairement à une énumération complète, les solutions ne sont pas évaluées une à une. La représentation de l'espace de solutions se fait sous la forme d'un arbre dont les nœuds désignent des solutions. Pour ce, la procédure de séparation évaluation (*branch and bound*) est très utilisée. Introduite par Land et Doig [58], la méthode *branch and bound* a été utilisée pour le VRP pour la première

fois par Christofides et Eilon [18]. Deux stratégies générales sont utilisées en vue de l'exploration de l'espace de solutions : l'exploration en profondeur d'abord (*depth-first search*) et l'exploration en largeur d'abord (*breadth-first search*). Pour les problèmes de tournées, les méthodes arborescentes sont souvent basées sur des modèles de PLNE.

En dépit de sa garantie d'optimalité, l'application des méthodes exactes à des problèmes de VRP s'est confrontée à des limitations d'ordre technique. En effet, les temps de calculs requis pour la résolution exacte de la plupart des problèmes de VRP de taille réelle sont dans le pire des cas exponentiels en fonction de la taille du problème. Le VRP étant NP-difficile, les puissances de calculs actuelles des ordinateurs ne permettent pas, en général, la résolution exacte de grands problèmes dans des délais raisonnables. Jusqu'à présent, les méthodes exactes les plus performantes appliquées au CVRP – une des versions les plus basiques du VRP – ne dépassent pas les 100 clients (Ralphs [85]).

2.4.2 Les heuristiques simples

Les heuristiques appartiennent à la classe des méthodes approchées c'est-à-dire celles qui fournissent des solutions de bonne qualité sans garantie d'optimalité. D'une manière générale, les heuristiques peuvent être définies, suivant Pearl [79], comme "des stratégies ayant recours à des informations facilement accessibles et même d'application générale en vue du contrôle de processus de résolution de problèmes d'ordre humain et technique". Dans le domaine du VRP, trois familles d'heuristiques sont recensées (Laporte et Semet [61]) :

- Heuristiques constructives ;
- Méthodes à deux phases ;
- Méthodes d'amélioration.

Heuristiques constructives :

Les méthodes constructives appelées *route construction heuristics* font partie des premières méthodes heuristiques utilisées pour la résolution du CVRP. Elles construisent les routes en procédant par ajout itératif de clients à celles-ci. La première heuristique proposée pour le VRP est la méthode des *savings* de Clarke et Wright [21] (CW). La stratégie de construction

est basée sur le calcul des économies (*savings*) obtenues en servant deux clients successivement sur une même route au lieu de deux routes distinctes. Tout en partant d'une solution où, pour chaque client, une route individuelle a été créée, une liste de *savings* est obtenue en calculant l'économie potentielle relative à la fusion de chaque paire de routes. Après avoir été classées par ordre décroissant en termes d'économie, les paires de nœuds sont évaluées en tentant de fusionner les deux routes qui les contiennent. Lorsque toutes les contraintes sont vérifiées (capacités, sous-tours, ...), la fusion est effectuée. Il existe deux modes d'implémentation de l'algorithme des *savings* : la version parallèle et la version séquentielle. La version parallèle produit, en général, de meilleurs résultats que la version séquentielle. Dans la littérature, des auteurs tels que Golden *et al.* [44], Paessens [78], Nelson *et al.* [74], Desrochers et Verhoog [27], Altinkemer et Gavish [2] et Wark et Holt [103] ont proposé des adaptations en vue d'améliorer la performance de la méthode.

Une autre approche classique de construction initialement développée par Mole et Jameson [71] est la méthode d'insertion. Son principe consiste à insérer des clients dans une tournée tout en s'assurant d'une augmentation minimale du coût (distance ou temps). A chaque insertion, entre deux clients consécutifs, dans une tournée, la variation de coût est évaluée. Celle provoquant la plus petite augmentation de coût est appliquée. D'autres facteurs peuvent être pris en compte pour évaluer les coûts, entre autres, l'évaluation de la distance du client à insérer par rapport au dépôt. En général, dans la littérature, on utilise deux versions de la méthode d'insertion : une version séquentielle Solomon [93] et une version parallèle Potvin et Rousseau [81]. La version séquentielle démarre une nouvelle tournée quand les insertions ne sont plus possibles dans la tournée courante. Dans la version parallèle, on effectue des insertions dans plusieurs tournées en cours de construction.

Méthodes à deux phases :

Cette approche consiste à décomposer le problème en deux phases. La décomposition procède suivant deux approches :

- Le *clustering* (regroupement) : Les clients sont répartis en sous-groupes, généralement, suivant un critère de proximité relative ;
- Le *routing* (planification de tournées) : L'ordre de visite des clients pour chaque tournée est calculé.

A partir de cette logique de décomposition, deux types d'heuristiques ont été développées : l'algorithme *route-first cluster-second* et l'algorithme *cluster-first route-second*. Le *cluster-first route-second* fait du regroupement d'abord et ensuite construit des tournées. Les méthodes *sweep algorithm* développée par Gillett et Miller [41] et *petal algorithm* proposée par Foster et Ryan [37], Ryan *et al.* [89] et par Renaud *et al.* [87] sont parmi les applications les plus connues de cette approche. Par ailleurs, la technique *route-first cluster-second*, initialement proposée par Beasley [7], procède par construction de tournées et ensuite regroupement. Pour ce, lors de la résolution des problèmes de VRP, une tournée géante (TSP) visitant tous les clients est construite. Quelques applications de cet algorithme ont été effectuées, entre autres, à travers les travaux de Bertsimas et Simchi-Levi [12] et de Haimovich et Rinnooy Kan [45].

Méthodes d'amélioration :

Les méthodes d'amélioration cherchent à améliorer une solution initiale en y appliquant des modifications successives au niveau des tournées. Ces heuristiques sont désignées sous le nom de *recherche locale*. Le principe de la recherche locale consiste à appliquer des modifications élémentaires à une solution donnée. Il s'agit principalement d'échanges d'arcs ou de mouvements de clients dans le but d'explorer le voisinage de la solution en cours dans l'espoir d'obtenir au moins une meilleure solution. Lorsqu'une meilleure solution est obtenue, elle remplace l'ancienne, dans le cas contraire, un optimum local est supposé atteint.

Deux types de modes d'exploration de voisinage existent : les voisinages du type *intra-route* et ceux du type *inter-route*. Les voisinages *intra-route* sont explorés à l'aide d'opérations impliquant une seule tournée tandis que, l'exploration *inter-route* concernent des opérations sur plus d'une tournée simultanément. Parmi les méthodes les plus connues, il existe le k -opt initialement développé par Lin [65] pour le TSP. $k = 2$ ou 3 (pour limiter le temps de calcul qui est proportionnel à n^k) désigne le nombre d'arêtes qui sont enlevées dans la solution et remplacées par d'autres. Une autre méthode, le Or-opt, consistant à remplacer une chaîne composée d'un nombre donné de clients par une autre, a été proposée par Or [76]. A part ces méthodes communément utilisées, d'autres approches d'exploration plus sophistiquées ont été développées et dont une revue a été réalisée à travers les travaux de Thompson et Psaraftis [97], Van Breedam [99] et Kindervater et Savelsbergh [54].

2.4.3 Les métaheuristiques

Une métaheuristique se définit comme un « processus-maître itératif qui guide et modifie les opérations d'heuristiques intégrées dans le but de produire des résultats de très bonne qualité » (Voss *et al.* [102]). A la différence des heuristiques simples, les métaheuristiques sont conçues pour s'échapper des optima locaux. Pour ce, elles combinent deux techniques lors de l'exploration de l'espace des solutions : *intensification* et *diversification*. L'*intensification* permet d'approfondir la recherche dans une région prometteuse jusqu'à en obtenir l'optimum local tandis que la *diversification* permet de s'extraire de cette région où l'optimum local a été trouvé dans le but d'obtenir probablement un optimum global qui se trouve ailleurs. Notons cependant que la plupart des métaheuristiques utilisent des heuristiques simples pour obtenir leurs solutions initiales. Dans la littérature, plusieurs métaheuristiques ont été proposées pour les problèmes de VRP. Elles peuvent, en général, être regroupées sous trois catégories :

- (1) Les métaheuristiques basées sur la recherche locale ;
- (2) Les méthodes travaillant sur un ensemble de solutions ;
- (3) Les métaheuristiques hybrides.

2.4.3.1 Les métaheuristiques basées sur la recherche locale

La recherche locale, appelée aussi méthode de descente, désigne une classe de méthodes appliquant des modifications élémentaires à une solution en vue d'obtenir des solutions localisées dans son *voisinage* à l'intérieur de l'espace de solutions. Le *voisinage* est exploré jusqu'à ce qu'une condition d'arrêt soit satisfaite. Ce mécanisme de recherche a été intégré dans plusieurs métaheuristiques dont, parmi les plus connues, figurent :

- **Le recuit simulé (ou *simulated annealing*)** est inspiré d'un processus de refroidissement utilisé en métallurgie. Ce processus consiste à diminuer progressivement et de manière contrôlée la température d'un métal à l'état liquide afin d'aboutir à une solidification de bonne qualité. Appliqué aux problèmes d'optimisation combinatoire, le recuit simulé, proposé par Kirkpatrick *et al.* [55], procède par tirage au sort d'une solution dans le voisinage de la solution courante tout en admettant avec un niveau de probabilité donné des solutions de moins bonne qualité. Cette probabilité est dépendante d'un paramètre indiquant la température. Initialement fixée à un niveau élevé,

la valeur de ce paramètre décroît au fur et à mesure du déroulement de la procédure jusqu'à s'annuler. L'efficacité de l'algorithme est avant tout lié à un très bon contrôle de la variation de la température dans le temps.

- Le ***Greedy Randomized Adaptive Search Procedure (GRASP)*** a été introduit par Feo et Resende [36], il génère de manière itérative une nouvelle solution à l'aide d'un algorithme glouton randomisé à laquelle une recherche locale est appliquée en vue de l'améliorer. Un ensemble de solutions sont créées de cette manière et la meilleure est retenue en tant que solution finale. Le principe de cette méthode repose donc sur un échantillonnage aléatoire (*random sampling*) de l'espace de solutions. En ce qui a trait aux récentes applications du GRASP à des problèmes de VRP, on peut se référer aux travaux de Li *et al.* [64] pour le VRPTW, par exemple.

- **La recherche locale itérative (ou *Iterated Local Search procedure (ILS)*)** est assez proche du GRASP en termes de stratégie d'exploration, sa différence réside dans le fait que la recherche locale est appliquée à une perturbation de la solution courante (Lourenço *et al.* [66], Prins [83]).

- **La recherche taboue** a été proposée par Glover [42], elle repose sur une stratégie d'exploration des solutions dans le voisinage tout en introduisant la notion de mémoire. Un ensemble de transformations sont appliquées à la solution en cours en admettant une certaine détérioration du coût. Les déplacements les plus récents dans l'espace de solutions sont mémorisés dans une liste dite « taboue ». Ceci permet d'éviter de revisiter des zones déjà explorées. La mémoire étant limitée, les mouvements amenant à chaque nouvelle solution admise sont enregistrés à la place de ceux de la plus ancienne solution sur la liste. Cette méthode a été largement utilisée dans les travaux sur les problèmes de VRP dont ceux de Gendreau *et al.* [39], Osman [77] et Taillard [94].

2.4.3.2 Les méthodes travaillant sur un ensemble de solutions

Ces méthodes, à la différence des méthodes de descente, initialisent la recherche en partant d'un ensemble diversifié de solutions. Un certain nombre de stratégies sont appliquées afin de faire évoluer cette population en fonction d'une technique définie par une procédure donnée. En effet, plusieurs algorithmes exploitant le principe d'évolution de population ont été développés dont les plus connus sont :

- **Les algorithmes génétiques**, initialement développés à travers les travaux de Holland [47], Goldberg [43] et Holland [48]. Ils se basent sur le principe de l'évolution naturelle des êtres vivants. Cette classe d'algorithmes applique des opérateurs dits génétiques pour faire évoluer la population de solutions. Il s'agit notamment d'opérateurs de *sélection*, de *croisement* et de *mutation* aux individus de la population. À partir du *croisement* de deux individus *parents*, deux individus *enfants* sont créés. Les *enfants*, quoique différents de leurs *parents*, héritent de parties de structures de ces derniers. L'opérateur de mutation a pour rôle de modifier aléatoirement la structure des *enfants* afin de maintenir un bon niveau de diversité dans la population. Au fur et à mesure de l'évolution de la population, une élite est obtenue grâce au fait que les meilleurs *gènes* sont transmis au fil des générations et que les individus les plus faibles sont éliminés par l'opérateur de *sélection*. Après un nombre fini de générations, la procédure s'arrête et le meilleur individu est retenu comme solution finale. Les algorithmes génétiques de base recourent à un mode de codage des solutions en nombre binaire. Ce codage ne convenant pas aux variantes du VRP, plusieurs adaptations ont été effectuées dont, parmi les meilleures, on retrouve celles de Baker et Ayechev [5] et Prins [82].

- **Les algorithmes de colonies de fourmis** sont une classe d'algorithmes relevant de l'approche d'optimisation inspirée des colonies de fourmis ou *Ant Colony Optimization* (ACO). Cette approche, proposée par Dorigo *et al.* [30] reproduit les schémas de comportements des fourmis qui partent à la recherche de nourriture pour leur colonie. Celles-ci mettent en place une forme de coopération leur permettant de trouver le chemin le plus court menant à une source de nourriture. Laissant sur leurs traces des dépôts de phéromones qui s'accumulent avec les allers-retours, l'ensemble des fourmis finissent par converger vers le parcours présentant la plus forte concentration de phéromones. Cette concentration dépend de la qualité du parcours en termes de longueur et de la source de nourriture à laquelle il mène. Pour résoudre les problèmes d'optimisation, l'ACO combine deux techniques : une méthode de construction et les mécanismes d'apprentissage. Des exemples d'applications peuvent être trouvés dans les travaux de Reimann *et al.* [86] et Yu et Yang [104].

- **L'optimisation par essais particuliers** a été initialement développée par Eberhart et Kennedy [31] et Kennedy et Eberhart [53] et est inspirée des comportements coopératifs dans les essaims d'abeilles ou les bancs de poissons. Le principe repose sur

un ensemble d'individus, appelés *particules*, répartis de manière aléatoire dans l'espace de solution où ils se déplacent à la recherche de solutions. Chaque *particule* est dotée d'une mémoire de sa meilleure solution et est apte à communiquer avec les autres dans son entourage. Régulièrement, chaque particule est appelée à choisir entre suivre sa voie sur la base de sa propre expérience ou suivre son meilleur voisin. Au fur et à mesure de l'avancement de la recherche dans l'espace des solutions, les particules tendent à converger vers la meilleure solution optimale. Dans le domaine du VRP, cette méthode a été appliquée à plusieurs variantes, en particulier le TSP (Shi *et al.* [90]) et le VRPTW (Zhu *et al.* [106]).

2.4.3.3 Métaheuristiques hybrides

Dans le but d'exploiter les points forts de différentes méthodes, on peut combiner des composantes provenant de métaheuristiques des deux classes précédentes. Ce qui donne lieu à des métaheuristiques dites "hybrides". L'hybridation peut s'opérer à deux niveaux : soit plusieurs méthodes de recherche coopèrent au sein d'une méthode hybride, soit une heuristique est insérée à l'intérieur d'une autre méthode. A titre d'exemple, on inclut souvent des recherches locales dans les algorithmes génétiques, ce qui donne des algorithmes dits "mémétiques" (Prins [82]).

2.5 Conclusion

Au cours de ce chapitre, nous avons décrit les problèmes de tournées de véhicules en partant de leurs origines jusqu'à leurs variantes les plus complexes. En effet, un problème de tournées peut désigner tout aussi bien un simple problème de voyageur de commerce (avec une seule tournée) qu'un problème avec de multiples contraintes (fenêtres de temps, flotte hétérogène, collectes...). Le VRP, pris au sens large, reste donc une classe de problèmes très riches et variés. Nous avons, également, présenté les principales méthodes de résolution du VRP et ses variantes. Ces méthodes sont divisées en deux grandes familles : les méthodes exactes et les méthodes approchées. Les méthodes approchées, quoique fournissant généralement des solutions non optimales, ne cessent d'être améliorées tant en qualité de solution qu'en temps de calculs. Dans les chapitres suivants, nous nous consacrerons à la variante du VRP qui se trouve au cœur de la problématique de cette thèse : le MC-VRP. Nous en ferons une description approfondie et proposerons nos approches de résolution.

Chapitre 3

MC-VRP avec un compartiment de taille fixe dédié à chaque produit

3.1 Introduction

Comme mentionné dans le précédent chapitre, le MC-VRP connaît une application très répandue dans le contexte industriel ainsi que dans la vie pratique. A travers ses nombreux cas d'application, plusieurs variantes du MC-VRP ont vu le jour. Dans ce chapitre, une classe de problèmes de MC-VRP sera étudiée. Il s'agit d'un MC-VRP avec un compartiment de taille fixe dédié à chaque produit. Par définition, nous considérons qu'un produit peut également désigner une famille de produits compatibles pouvant être transportés dans un même compartiment. En effet, dans l'industrie agroalimentaire, on trouve l'application de ce problème où, pour des raisons de sécurité et/ou d'hygiène lors du transport, les produits sont affectés aux compartiments une fois pour toutes.

3.2 Travaux antérieurs

Dans la littérature, à notre connaissance, très peu de travaux portent sur le MC-VRP avec un compartiment de taille fixe dédié à chaque produit. Rappelons les travaux d'El Fallahi *et al.* [33] qui ont résolu un problème de MC-VRP dans un contexte agroalimentaire. Ils ont étudié un problème de distribution dans une industrie agroalimentaire spécialisée en nutrition animale. Dans le but de prévenir tout risque de contamination croisée, chaque famille de produits est affectée de manière exclusive à un compartiment. De ce fait, chaque

compartiment ne peut recevoir qu'un seul type de produits. Les auteurs ont proposé deux approches pour résoudre le problème : un algorithme génétique combiné avec une méthode de *path relinking* pour la post-optimisation et une recherche taboue.

Derigs *et al.* [25] ont réalisé des travaux sur le MC-VRP tout en prenant en compte plusieurs de ses variantes. En particulier, ils ont étudié des contraintes liées à la compatibilité entre produits et compartiments dans un contexte agroalimentaire. Pour la résolution des problématiques de MC-VRP, ils ont développé un solveur polyvalent dont les composants heuristiques sont constitués de différentes approches alternatives telles que méthode de construction, méthode d'insertion, recherche locale et le *Large Neighborhood Search*.

3.3 Modèle mathématique

Dans cette section, le MC-VRP est modélisé sous la forme d'un PLNE. Un ensemble P de produits est livré aux clients tout en sachant que chaque client i commande une quantité d_{ip} pour un produit p . Pour chaque client i , il existe au moins une demande $d_{ip} \neq 0$. On dispose d'un ensemble K de véhicules de capacités homogènes Q . Chaque véhicule possède autant de compartiments que de produits p . Donc, la capacité d'un compartiment dédié à un produit p est désignée par w_p . Nous supposons que toute demande d_{ip} est inférieure ou égale à la capacité w_p d'un compartiment p . Les coûts c_{ij} des arcs sont interprétés comme des temps de parcours. A chaque tournée de livraisons, une durée maximale L est imposée. La durée de service identique pour chaque client est désignée par m . La durée d'une tournée correspond à son coût additionné à la somme des durées de service des clients visités. Le modèle mathématique est décrit ci-dessous :

Variables de décision :

$$x_{ijk} = \begin{cases} 1 & \text{si le véhicule } k \text{ traverse l'arc } (i, j) \text{ avec } (i, j) \in A \text{ et } k \in K \\ 0 & \text{sinon} \end{cases} \quad (3.1)$$

$$y_{ik} = \begin{cases} 1 & \text{si le véhicule } k \text{ visite le nœud } i \\ 0 & \text{sinon} \end{cases} \quad (3.2)$$

Fonction-objectif :

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk} \quad (3.3)$$

Sous contraintes :

$$\sum_{k \in K} y_{ik} = 1 \quad \forall i \in V' \quad (3.4)$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik} = y_{ik} \quad \forall i \in V', \forall k \in K \quad (3.5)$$

$$\sum_{i \in V'} d_{ip} y_{ik} \leq w_p \quad \forall k \in K, \forall p \in P \quad (3.6)$$

$$\sum_{i \in S} \sum_{j \in S, j \neq i} x_{ijk} \leq |S| - 1 \quad \forall S \subset V', |S| \geq 2, \forall k \in K \quad (3.7)$$

$$\sum_{(i,j) \in A} c_{ij} x_{ijk} + \sum_{i \in V'} m y_{ik} \leq L \quad \forall k \in K \quad (3.8)$$

$$x_{ijk} \in \{0, 1\} \quad \forall (i, j) \in A, \forall k \in K \quad (3.9)$$

La fonction-objectif (3.3) a pour but de minimiser la somme des coûts de toutes les tournées. Les contraintes (3.4) assurent que chaque client est desservi une seule fois. La conservation des flots est imposée par (3.5). Les demandes associées à un produit ne peuvent en aucun cas dépasser la capacité allouée à ce produit dans le véhicule. En d'autres termes, les contraintes (3.6) exigent que la capacité du compartiment affecté au produit p soit au moins égale à la somme des demandes pour le produit p . Les contraintes de sous-tours sont imposées en (3.7). L'expression en (3.8) définit la borne supérieure de la longueur de chaque tournée. Les contraintes (3.9) précisent le domaine de définition binaire des variables x_{ijk} .

3.4 Résolution heuristique

Les solveurs de PLNE étant encore limités en termes de performances de calculs, nous avons recours à des méthodes approchées pour résoudre le MC-VRP étudié dans ce chapitre. Ces méthodes sont désignées sous l'appellation de méthodes approchées ou heuristiques car elles ne peuvent garantir des solutions optimales mais plutôt des solutions de bonne qualité plus ou moins proches de l'optimum. Parmi un grand nombre de méthodes approchées existantes, nous avons opté pour deux métaheuristiques : le *Greedy Randomized Adaptive Search Procedure* (GRASP) et la méthode *Iterated Local Search procedure* (ILS). Le GRASP a connu une application très vaste dans le domaine du VRP (Prins [83]). Egalement, l'ILS a été largement appliqué aux problèmes de VRP (Lourenço *et al.* [66]). Il est reconnu pour sa simplicité d'implémentation, sa robustesse et sa grande efficacité.

3.4.1 Résolution à l'aide d'un GRASP

3.4.1.1 Description générale du GRASP

Le GRASP, développé par Feo et Bard [35], procède à un échantillonnage aléatoire de l'espace des solutions. Il s'agit d'un processus itératif où chaque itération comprend 2 étapes :

- La construction d'une solution avec une heuristique gloutonne ;
- L'amélioration de celle-ci par une recherche locale.

Le GRASP procède par itération à l'aide d'un algorithme glouton auquel on introduit un paramètre aléatoire (*randomisation*) pour diversifier les solutions. La randomisation repose sur une liste restreinte (*Restricted Candidate List - RCL*) de choix possibles à chaque étape de la construction de solution. Chaque solution obtenue est ensuite améliorée par une recherche locale. La meilleure solution trouvée est considérée comme la solution globale.

A chaque itération, pendant la phase de construction, une solution est construite en partant d'une solution vide qui est complétée au fur et à mesure par insertion d'un élément sélectionné dans un ensemble de candidats. La sélection du prochain élément, à insérer dans une solution partielle, est déterminée par l'évaluation de tous les éléments candidats selon un critère. Celui-ci sert à exprimer l'augmentation du coût de la fonction-objectif qui serait due à l'insertion d'un nouvel élément dans une solution en construction. Ce qui permet la

création d'une liste restreinte d'éléments candidats (*Restricted Candidate List* ou *RCL*) formés par les meilleurs éléments. Ces derniers sont considérés comme les meilleurs parce que leur insertion induit les plus petites augmentations de coûts possibles. Ce qui correspond à l'aspect glouton de l'algorithme.

Dans le cas du VRP, une heuristique randomisée possible est la suivante. A chaque itération, on peut procéder en construisant une tournée en insérant de proche en proche les sommets non encore visités, la distance étant le coût à minimiser. L'introduction d'aléa dans le choix d'un sommet de la *RCL* en vue de son insertion à la solution partielle définit l'aspect probabiliste de l'algorithme (*randomisation*). Cette sélection aléatoire dans une liste restreinte de bons candidats permet la construction de solutions diverses avec un assez bon niveau de qualité. Après chaque nouvelle insertion d'élément à la solution en construction, la *RCL* est mise à jour et les coûts d'insertion sont réévalués (*adaptive*). Quand on ne peut plus ajouter de sommets à la tournée actuelle, on la termine en revenant au dépôt. S'il reste des clients à visiter, on construit de la même manière une nouvelle tournée.

A la fin de la construction, la solution obtenue n'est pas forcément optimale, elle peut être souvent améliorée. Pour ce, une recherche locale est utilisée en vue de trouver la meilleure solution dans le voisinage de la solution initiale. La recherche locale consiste à explorer le voisinage d'une solution de départ en quête d'une meilleure solution. On définit généralement le voisinage d'une solution à l'aide d'une transformation élémentaire ou locale de sa structure. L'efficacité de la recherche locale dépend fortement de la qualité et de la diversité des solutions issues de la phase de construction. Car, outre les aspects tels que la structure du voisinage, la technique de recherche et la rapidité d'évaluation des fonctions de coûts de voisinage, la solution de départ joue un rôle prépondérant dans l'efficacité de la recherche locale.

La recherche dans le voisinage est effectuée suivant 2 stratégies : la meilleure amélioration (*best improvement*) et la première amélioration (*first improvement*). Dans le cas de la stratégie du *best improvement*, après une évaluation exhaustive de tous les voisinages de la solution de départ, la meilleure solution est retenue. En ce qui a trait au *first improvement*, la solution courante est remplacée par la première solution améliorante. L'exploration se poursuit jusqu'à ce qu'elle ne parvienne plus à améliorer la solution en cours. Les trajectoires générées étant différentes, les deux stratégies peuvent donner des solutions différentes. Toutefois, celle du (*first improvement*) a le mérite de nécessiter moins de temps de calcul tandis que le (*best improvement*) montre une plus grande probabilité de convergence précoce. En VRP, deux

types de voisinages sont utilisés : les voisinages du type *intra-route* et ceux du type *inter-route*. Les voisinages *intra-route* sont explorés à l'aide d'opérations impliquant une seule tournée tandis que l'exploration *inter-route* concerne des opérations sur plus d'une tournée simultanément.

3.4.1.2 Les composants du GRASP appliqué au MC-VRP

3.4.1.2.1 Phase de construction de solution

Durant la première phase, au lieu d'utiliser une méthode de construction de tournées comme expliquée à la page précédente, chaque solution a été construite par une version parallèle de l'heuristique de *Clarke and Wright* adaptée aux caractéristiques du problème de MC-VRP. Cette méthode appartient à la famille des heuristiques basées sur les économies. Ces heuristiques ont l'avantage d'être relativement simples, rapides et faciles à implémenter. En plus, des expériences numériques réalisées par Laporte et Semet [61] sur des problèmes test de Christofides *et al.* [19] montrent que les solutions de l'algorithme de *Clarke and Wright* suivi d'une recherche locale 3-opt sur chaque tournée se situent en moyenne à 6.71% des meilleures solutions connues sinon, sans recherche locale, on est à plus de 10%, ce qui n'est déjà pas mal.

Au départ, cette heuristique crée une tournée directe pour chaque client. Ensuite, elle fusionne de manière itérative deux routes en une seule, ce qui résulte en une économie de distance car on suppose que l'inégalité triangulaire est vérifiée. Ensuite, toutes les économies s_{ij} possibles sont calculées et sont triées en ordre décroissant. Après quoi, à chaque itération, l'algorithme considère alors la plus grande économie disponible et étudie la possibilité de fusionner les routes correspondantes tout en respectant les contraintes (respect des capacités avec la procédure *ChargerComp()* (voir algorithme 1), durée, distance, compatibilité produits/compartiments...). Par exemple, la fusion de la tournée $R_i = (0, \dots, i, 0)$ avec la tournée $R_j = (0, j, \dots, 0)$ crée la tournée $R_i \oplus R_j = (0, \dots, i, j, \dots, 0)$ tout en générant l'économie suivante : $s_{ij} = c_{0i} + c_{0j} - c_{ij}$ (voir illustration de la figure 3.1). La solution est finalisée lorsqu'il n'est plus possible de fusionner des tournées ou lorsque la liste des économies disponibles a été complètement exploitée.

Afin de générer des solutions diversifiées au fur et à mesure des itérations du *GRASP*, un niveau d'aléa est introduit dans le processus de construction des solutions. A chaque étape de la construction, l'ensemble des décisions possibles sont consignées dans une liste restreinte

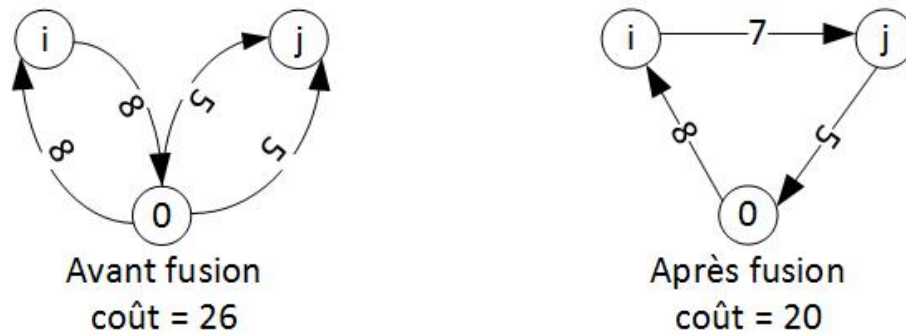


FIGURE 3.1: Fusion de tournées par la méthode de Clarke and Wright

RCL. Concrètement, la *randomisation* a été traduite, dans notre cas, par une légère perturbation aléatoire du processus de constitution de la liste des économies s_{ij} . En effet, de manière aléatoire, un faible pourcentage parmi toutes les économies possibles ne seront pas intégrées dans la liste, ce qui permet, par conséquent, de *randomiser* le processus de fusion (voir algorithme 2 pour plus de détails). D'où, le caractère probabiliste de la construction de tournées.

La définition de la stratégie de *randomisation* s'avère cruciale dans la performance de l'algorithme. C'est ce qui détermine la manière dont la *RCL* sera construite. En effet, deux stratégies sont utilisées. La première consiste à remplir de manière aléatoire la liste *RCL* à partir de l'ensemble des décisions faisables et de sélectionner la meilleure sur la base d'un critère glouton. La deuxième place dans la liste les meilleures décisions trouvées par l'algorithme et, ensuite, sélectionne aléatoirement l'une d'entre elles. Nous avons utilisé la première stratégie dans notre approche.

Algorithme 1 Algorithme de la procédure *ChargerComp()*

ENTRÉE(S): Capacité Q d'un véhicule

ENTRÉE(S): Charges q_{kp} affectées aux compartiments p dans chaque tournée R_k à modifier

si $(\forall p, \forall k, q_{kp} \leq w_p)$ **alors**

Renvoyer VRAI

sinon

Renvoyer FAUX

finsi

Algorithme 2 Génération d'une liste des $SavingsList(i, j, s_{ij})$ triés en ordre décroissant de manière randomisée : $RandSavingsListGen()$

ENTRÉE(s): liste des nœuds $(i, j) \in A$, nombre total de nœuds n , matrice des coûts c_{ij} ,

paramètre de randomisation : $0 < \alpha < 1$

$SavingsList \leftarrow \emptyset$

pour $i = 1, \dots, n$ **faire**

pour $j = i + 1, \dots, n$ **faire**

 Générer aléatoirement une valeur $Rand$ comprise entre 0 et 1

si $Rand < \alpha$ **alors**

$$s_{ij} = c_{0i} + c_{0j} - c_{ij}$$

$SavingsList \leftarrow s_{ij}$

finsi

fin pour

fin pour

Trier $SavingsList$ par valeurs décroissantes des s_{ij}

$Renvoyer(SavingsList)$

3.4.1.2.2 Phase d'amélioration

Les solutions générées lors de la phase de construction sont généralement de bonne qualité sans nécessairement être optimales. C'est la raison pour laquelle elles sont soumises à une phase d'optimisation assurée par une recherche locale. Celle-ci procède par exploration du voisinage sous forme de mouvements qui effectuent des déplacements et des échanges de nœuds en impliquant une route ou plus. Nous avons utilisé les mouvements ci-dessous :

- **L'opérateur de relocalisation (*relocation*)** déplace un nombre donné de sommets d'une position vers une autre dans la même route ou vers une autre (voir illustration 3.2) ;
- **L'opérateur d'échange (*exchange*)** échange un nombre donné de sommets dans une route ou entre deux routes (voir illustration 3.3) ;
- **L'opérateur 2-Opt** enlève deux arcs sans noeud en commun et reconnecte les différentes portions de route tout en évitant de reconstituer les arcs préalablement défaits (voir illustration 3.4). La recombinaison de tronçons de route(s) crée de nouveaux arcs qui remplacent les arcs enlevés. Toutefois, les opérations peuvent donner lieu à trois cas de figures. D'une part, dans le cas d'une seule route, la portion se trouvant entre les deux arcs sera inversée par le mouvement. D'autre part, lorsqu'il s'agit de deux routes,

les nouvelles routes peuvent être construites de deux manières différentes. Admettons que chaque route soit sectionnée en tronçon n° 1 et tronçon n° 2 de telle sorte que le tronçon n° 1 désigne la première portion de la route sectionnée et le n° 2 correspond à la deuxième partie. Dans un cas, les nouvelles routes sont formées en connectant respectivement les tronçons n° 1 et n° 2 de deux routes sectionnées. Dans un autre cas de figure, les routes sont formées en reliant les tronçons de mêmes numéros. Cela implique nécessairement un changement d'orientation d'un des tronçons dans chaque route nouvellement constituée.

Bien entendu, ces mouvements sont évalués seulement s'ils respectent la capacité des véhicules et la durée maximale des tournées.

3.4.1.2.3 Vue algorithmique

Le schéma algorithmique du GRASP est présenté ci-dessous (Algorithme 4). A chaque nouvelle itération, une nouvelle solution *Soln* est générée. Chacune des solutions est construite au départ par un *CW* randomisé avant d'être optimisé par une recherche locale (*LS*). Le nombre total d'itérations est déterminé par le paramètre *MaxGRASPIter*. C'est l'un des paramètres-clés de l'algorithme dans la mesure où il détermine l'envergure de l'échantillonnage de l'espace des solutions.

3.4.2 Résolution à l'aide d'un ILS

3.4.2.1 Description de l'ILS

L'idée de base de l'ILS repose sur le fait de déployer la recherche non pas dans tout l'espace de solutions mais plutôt de la concentrer dans des sous-espaces définis par le bassin d'attraction constitué par des solutions localement optimales. La procédure ILS est constituée de trois principaux composants : génération d'une solution initiale, recherche locale et perturbation. Comme illustré dans l'algorithme, l'ILS calcule un optimum local initial *Soln**. Puis, de manière itérative, les étapes suivantes sont exécutées : (1) une copie *Soln* de *Soln** est créée (2) *Soln* est perturbée (3) on applique la recherche locale sur la solution perturbée (4) si un nouvel optimum local est obtenu, on remplace *Soln** par *Soln*. L'exploration s'arrête au bout d'un nombre défini (*MaxLSIter*) d'itérations de la recherche locale.

Algorithme 3 Heuristique de Clarke and Wright randomisée : *RandomCW*

ENTRÉE(S): ensemble $A = \{a_{ij}\}$ des arcs, matrice des coûts c_{ij}

Créer une solution initiale $Soln$ en attribuant une tournée directe à chaque client

Générer une liste de *savings* randomisés : *RandSavingsListGen()*

pour $i = 1, \dots, n$ **faire**

pour $j = i + 1, \dots, n$ **faire**

si $s_{ij} > 0$ **alors**

 Vérifier si i et j sont aux extrémités de 2 tournées distinctes R_i et R_j

 Vérifier le respect des capacités des compartiments en exécutant : *ChargerComp()*

 Vérifier le respect les contraintes de longueur de $R_i \oplus R_j$

 Si OK, exécuter la fusion de R_i et R_j

finsi

fin pour

fin pour

Renvoyer ($Soln$)

Algorithme 4 Structure générale du GRASP

ENTRÉE(S): ensemble $A = a_{ij}$ des arcs, matrice des coûts c_{ij}

Initialiser $coût(Soln^*) \leftarrow +\infty$

pour $Iter = 1, \dots, MaxGRASPIter$ **faire**

 Appliquer heuristique de Clarke and Wright randomisée : *RandomCW*

LS($Soln$)

si $coût(Soln^*) > coût(Soln)$ **alors**

$Soln^* \leftarrow Soln$

finsi

fin pour

Renvoyer($Soln^*$)

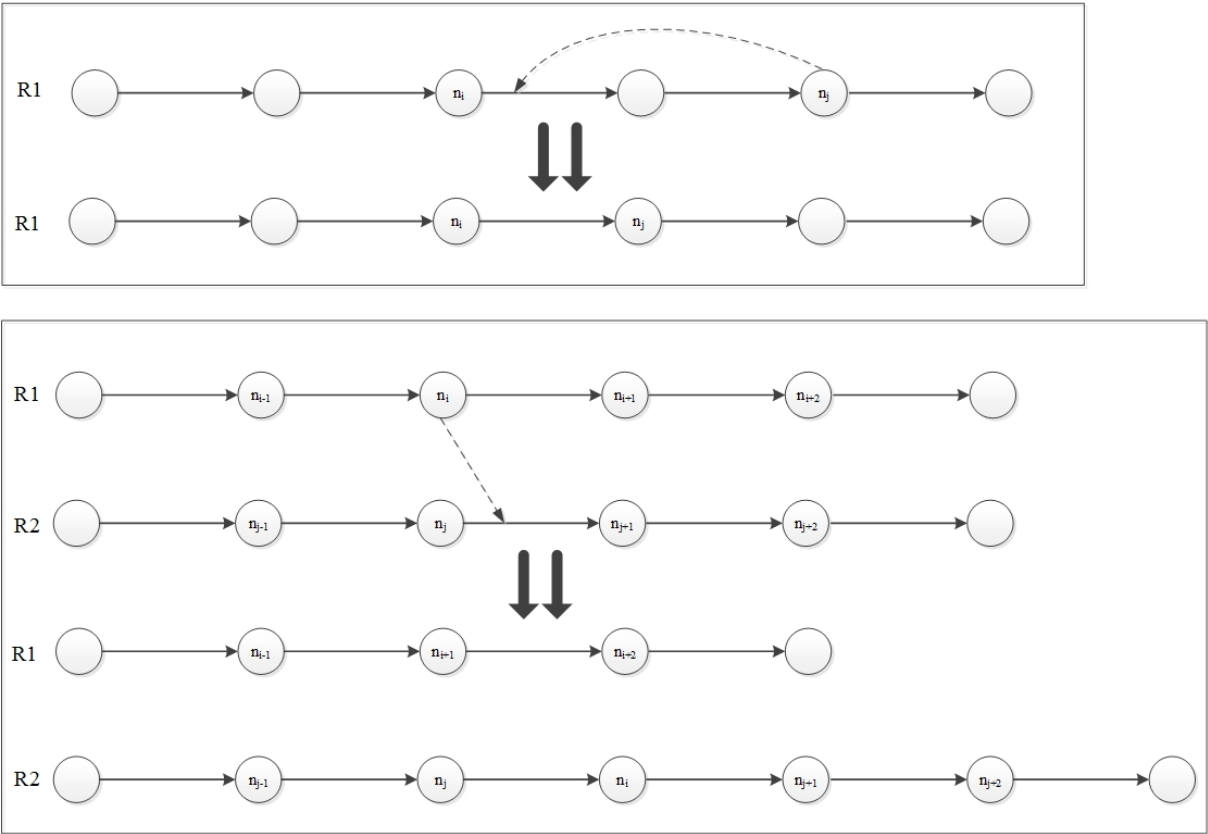


FIGURE 3.2: *Mouvements de relocalisation illustrés sur une seule et deux routes*

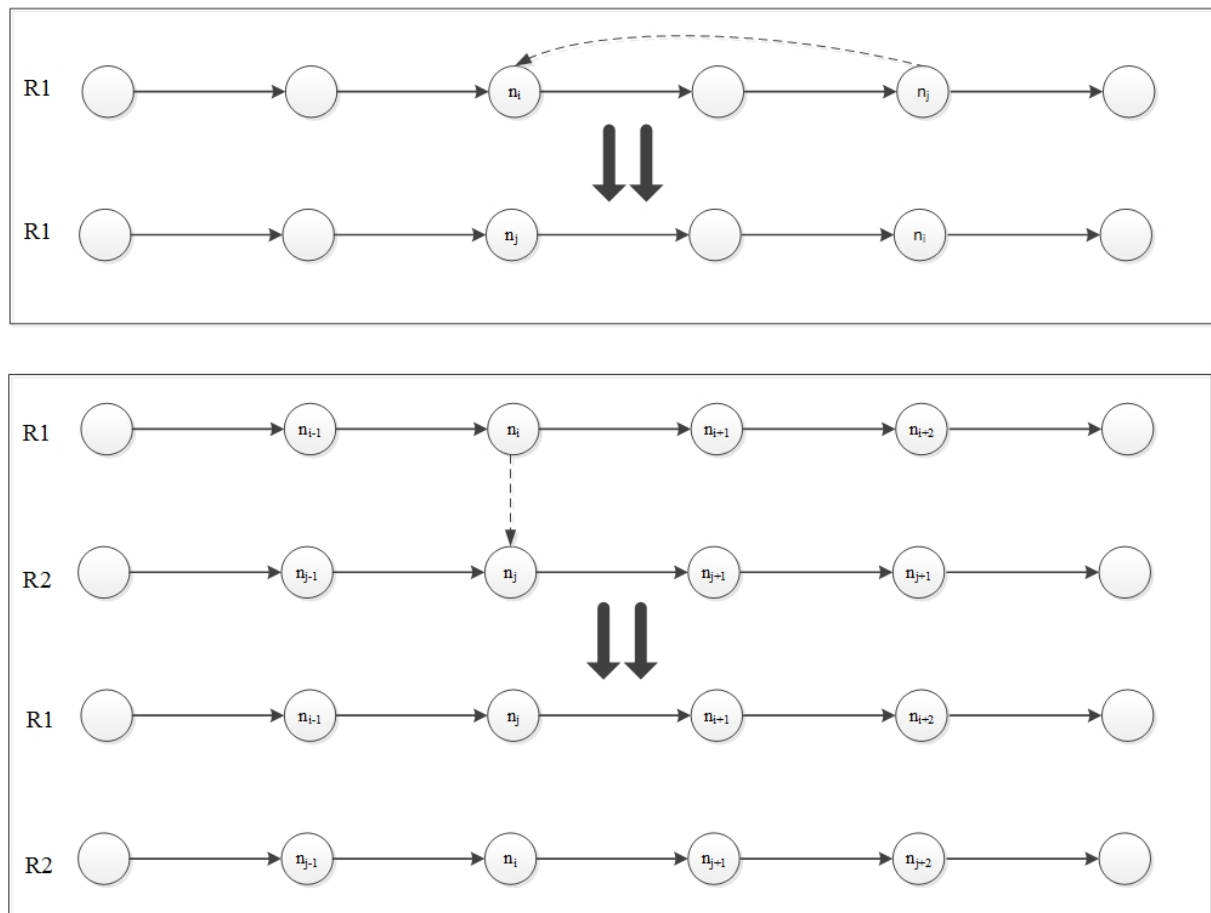


FIGURE 3.3: *Mouvements d'échange illustrés sur une seule et deux routes*

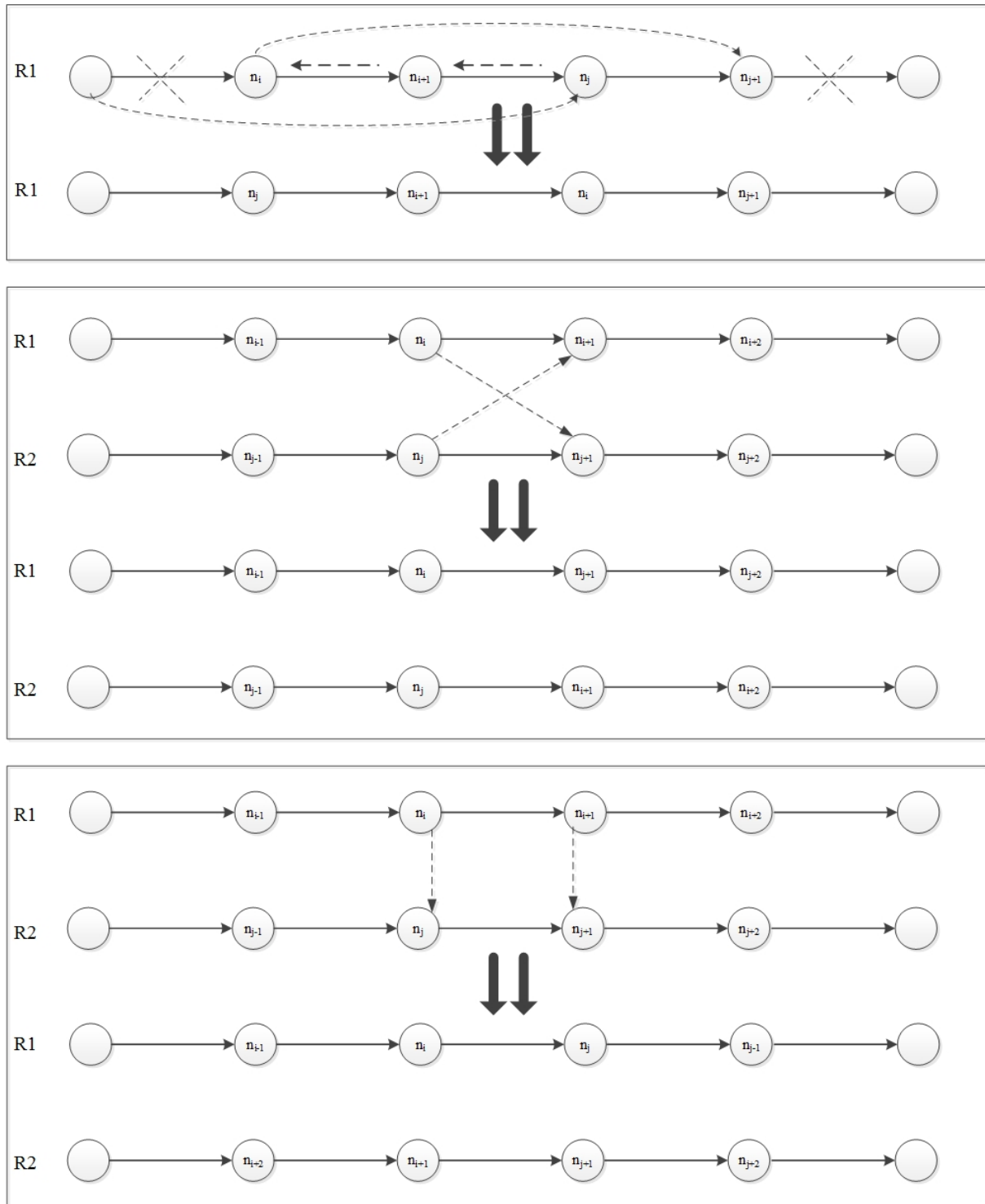


FIGURE 3.4: Mouvements 2-Opt illustrés sur une seule et deux routes

Algorithme 5 Algorithme de la procédure ILS

ENTRÉE(S): ensemble $A = \{a_{ij}\}$ des arcs, matrice des coûts c_{ij}

ENTRÉE(S): $MaxLSIter$, $LimitMaxPerturbIter$

Générer $Soln^*$ à l'aide de l'heuristique de *Clarke and Wright*

Appliquer une recherche locale : $LS(Soln^*)$

$Iter \leftarrow 1$

$MaxPerturbIter \leftarrow 1$

répéter

$Iter \leftarrow Iter + 1$

On prend une copie de la solution : $Soln \leftarrow Soln^*$

$Perturbation(Soln)$

$LS(Soln)$

si $coût(Soln) < coût(Soln^*)$ **alors**

$Soln^* \leftarrow Soln$

$MaxPerturbIter \leftarrow 1$

sinon

$MaxPerturbIter \leftarrow Min(MaxPerturbIter + 1, LimitMaxPerturbIter)$

finsi

jusqu'à $Iter = MaxLSIter$

$Renvoyer(Soln^*)$

3.4.2.2 Les composants de l'ILS

3.4.2.2.1 Génération d'une solution initiale

L'ILS procède, avant tout, à l'amélioration d'une solution initiale. Deux méthodes peuvent être employées pour en générer une. On peut recourir à une méthode purement aléatoire, une méthode constructive ou une méthode de fusion. Nous avons choisi d'utiliser une méthode de fusion dans notre cas. Car, une telle méthode ainsi qu'une méthode constructive présentent deux avantages en comparaison à une méthode aléatoire. D'une part, une recherche locale appliquée à une solution issue d'une de ces deux méthodes produit, généralement, une solution de meilleure qualité que lorsqu'il s'agit d'une méthode aléatoire. D'autre part, lorsque la solution initiale est générée par une méthode gloutonne, en moyenne, une recherche locale, passe par moins de phases intermédiaires d'amélioration (vallées) avant de parvenir à l'optimum local.

Précisément, comme méthode de génération de solutions de départ, nous avons utilisé la version parallèle de l'heuristique de *Clarke and Wright*. Etant originalement conçue pour des problèmes de CVRP, elle a été adaptée aux spécificités du problème de MC-VRP que nous étudions ici. En particulier, les tests de capacité ont été appliqués à chaque compartiment pour voir si une fusion de deux tournées est faisable.

3.4.2.2.2 Recherche locale

La recherche locale est utilisée de manière itérative pour tenter d'améliorer la solution en cours. La structure de voisinage repose sur des mouvements élémentaires comme pour le GRASP décrit en chapitre 3.

3.4.2.2.3 Perturbation

L'une des limites d'une recherche locale est la possibilité d'être piégée par un optimum local qui serait de qualité bien inférieure à celle de l'optimum global. Pour permettre au LS d'échapper à ce piège, une méthode de perturbation est intégrée à l'ILS. Son principe rappelle le fonctionnement du *simulated annealing* qui fait varier la température en vue de permettre une plus large exploration de l'espace de solutions.

Précisément, la perturbation en ILS consiste à appliquer quelques mouvements aléatoirement à la solution en cours en vue de changer sa structure de manière significative. Pour empêcher à la recherche locale de défaire facilement les modifications apportées par la perturbation, ses mouvements sont sélectionnés pour être différents de ceux de cette dernière. Ce qui évite à l'ILS de revenir sur des optima locaux récemment visités. Le niveau de perturbation appliquée influe sur la performance de l'ILS. Un faible niveau de perturbation risque de ne pas pouvoir extraire la solution d'un optimum local tandis qu'un niveau trop élevé de perturbation peut produire le même effet qu'une pure randomisation dans la mesure où la solution perturbée est fortement dénaturée, ce qui réduit la probabilité de s'approcher de l'optimum global.

Dans le contexte du MC-VRP que nous étudions ici, nous avons utilisé des mouvements d'échange pour la procédure de perturbation. Pour produire une perturbation, deux tournées sont sélectionnées de manière aléatoire et ensuite deux clients choisis aléatoirement sont échangés entre ces dernières. Pour le paramétrage du niveau de perturbation, nous avons opté pour une approche adaptative. En effet, lors du processus de perturbation, le nombre de mouvements d'échanges augmente jusqu'à un nombre maximal $MaxPerturbIter$ tant que l'ILS n'obtient pas de nouvelle solution améliorante. La valeur de $MaxPerturbIter$ est réinitialisée dès qu'il y a une nouvelle amélioration. La procédure de perturbation est illustrée par l'algorithme 6 pour plus de détails.

3.5 Résultats numériques

3.5.1 Génération des instances

En vue de réaliser des tests numériques, nous avons adapté des instances de Christofides *et al.* [19] aux caractéristiques de la problématique étudiée. Les instances comprennent entre 50 et 199 clients et sont au nombre de 14 parmi lesquelles 7 instances avec une limite de longueur sur les tournées. Les modifications portent sur : a) l'insertion de compartiments aux véhicules b) la définition des produits demandés c) la répartition de la demande de chaque client entre les différents produits.

Pour pouvoir vérifier plus facilement les solutions, quoique notre algorithme puisse gérer jusqu'à dix compartiments, nous en avons défini deux par véhicule. La somme de leurs capa-

Algorithme 6 Algorithme de la procédure de perturbation : *Perturbation()*

ENTRÉE(S): *Soln, MaxPerturbIter*

PerturbIter \leftarrow 0

répéter

PerturbIter \leftarrow *PerturbIter* + 1

Initialiser : *Amélioration* \leftarrow *FAUX*

TentativesPerturb \leftarrow 0

répéter

Sélectionner aléatoirement 2 tournées distinctes R_1, R_2 dans *Soln*

Sélectionner aléatoirement les noeuds $n_1 \in R_1$ et $n_2 \in R_2$

Un test de faisabilité du mouvement est effectué (test de violation des capacités en exécutant *ChargerComp()* et des longueurs max des tournées)

si *TestFaisabilité(Soln)* = VRAI **alors**

Récupérer le coût actuel de *Soln* : *Valeur* \leftarrow *coût(Soln)*

Appliquer le mouvement d'échange sur n_1 et n_2 :

Soln \leftarrow *Echange(Soln, n₁, n₂)*

finsi

jusqu'à *TestFaisabilité(Soln)* = VRAI

si *coût(Soln)* < *Valeur* **alors**

Amélioration \leftarrow VRAI

finsi

Si *Soln* est améliorée, la procédure de perturbation peut être arrêtée prématurément

jusqu'à *Amélioration* = VRAI OU *PerturbIter* = *MaxPerturbIter*

Renvoyer(Soln)

cités est égale à la capacité initialement définie dans l'instance correspondante de Christofides *et al.* [19]. Ces derniers ne possèdent pas forcément la même taille. En effet, sont définies des instances représentant des scénarios avec des véhicules de compartiments identiques et différents.

Pour les demandes, nous avons gardé la même quantité initiale d_i pour chaque client i dans les instances d'origine de Christofides *et al.* [19]. Cependant, comme nous avons fixé le nombre de produits à 2, chaque demande d_i est répartie en 2 demandes, une par produit : $d_i = d_{i1} + d_{i2}$ avec d_{i1} ou d_{i2} pouvant être nulle.

Les modifications effectuées ont permis d'obtenir huit types d'instances (MC1, MC2, MC3, MC4, MC5, MC6, MC7, MC8) adaptées à la problématique traitée dans ce chapitre. Les modifications prennent en compte deux contextes : véhicules avec compartiments homogènes (MC1, MC2, MC3, MC4) et véhicules avec compartiments hétérogènes (MC5, MC6, MC7, MC8).

1. En MC1, la demande de chaque client est également répartie entre deux produits.
2. En MC2, le nombre de produits (lots) commandés par les clients diffère suivant une règle précise. Dans chaque instance, à un client dont le numéro est impair est attribué un nombre impair de produits commandés tandis qu'un client de numéro pair commande un nombre pair de produits.
3. En MC3, chaque client commande deux lots de quantités différentes. Les quantités correspondant aux lots demandés par un client sont déterminées en fonction d'une suite géométrique de raison $u = 1,25$. La quantité d_{ip} demandée par un client i pour un produit p est donnée par la formule suivante $d_{ip} = d_{ip-1}u^{p-1}$.
4. En MC4, le nombre de lots demandés par chaque client varie suivant la même règle qu'en MC2.
5. A part les caractéristiques des véhicules dont les compartiments sont hétérogènes, la définition des types d'instances MC5, MC6, MC7 et MC8 est respectivement identique à celle de MC1, MC2, MC3 et MC4. Les tailles des compartiments ont été définies suivant une logique basée sur une suite géométrique de raison $u = 1,25$. La taille du compartiment p (alloué au type de produits p) d'un véhicule k est donnée par l'expression : $w_p = w_{p-1}u^{p-1}$.

3.5.2 Paramétrage

La machine utilisée pour réaliser les expérimentations numériques est un PC DELL Studio XPS disposant d'un processeur Intel Core i7 avec une fréquence de 1.6 GHz et une mémoire vive de 8 Go tournant sous Microsoft Windows 7. Nous avons utilisé le langage de programmation C++ du compilateur Microsoft Visual Studio 2012.

La méthode GRASP possède deux principaux paramètres : un paramètre de randomisation α et le nombre d'itérations $MaxGRASPIter$. Il ne suffit pas de leur donner une valeur au hasard pour assurer l'efficacité de la méthode. Il faut, bien au contraire, déterminer leurs meilleures valeurs. Cela nécessite, bien sûr, de trouver un compromis entre les valeurs des solutions obtenues et les temps de calcul requis.

Le paramètre α a été fixé à 0.8 après avoir réalisé des essais sur les cinq premières instances dans la catégorie MC1. A priori, nous avons fixé le paramètre $MaxGRASPIter$ à 1500. Les 1500 itérations sont largement suffisantes d'un point de vue statistique pour pouvoir observer l'influence des différentes valeurs du paramètre α sur la qualité des solutions obtenues. Lors des essais, α a pris les différentes valeurs $\{0.75, 0.775, 0.8, 0.825, 0.85, 0.875, 0.9, 0.925\}$. La meilleure solution, en moyenne, a été obtenue pour $\alpha = 0.8$ comme indiquée dans le tableau 3.1. Il est à noter également que la moyenne des solutions pour les cinq instances varie très peu pour les différentes valeurs de α allant de 0.75 à 0.925. Ce qui indique que notre méthode GRASP est robuste.

En partant de la meilleure valeur trouvée pour le paramètre α , des essais sont effectués pour déterminer la valeur de $MaxGRASPIter$. Etant donnée que la valeur de $MaxGRASPIter$ définit l'ampleur de l'échantillonnage de l'espace de solutions, il est évident que plus elle est grande, plus il est probable de trouver la solution optimale. Donc, le but recherché ici est plutôt de trouver un bon compromis entre la qualité des solutions et le temps de calcul nécessaire. En incrémentant, de 500 à chaque fois, la valeur de $MaxGRASPIter$ dans l'intervalle $[500, 4000]$, nous avons exécuté la méthode GRASP sur les trois premières instances de MC1. Dans le tableau 3.2, sont affichées les valeurs moyennes des solutions et des temps de calcul du processeur. De 500 à 4000 itérations, l'écart entre la plus petite valeur moyenne des solutions (736.53) et la plus grande (742.75) est d'environ 0.8% tandis que le temps de calcul a été multiplié par environ 7.6 en passant d'un temps moyen de 10.59 sec. à 80.4 sec. Les solutions obtenues par le GRASP en fonction du nombre d'itérations tendent à converger à partir de 2000 itérations comme on peut le voir dans le tableau 3.2. En effet, avec 2000

α	Moyenne des solutions GRASP
0.75	922.4
0.775	926
0.8	919.7
0.825	927.1
0.85	926.9
0.875	925.9
0.9	925
0.925	924.9

TABLE 3.1: *Moyenne des solutions GRASP pour différentes valeurs du paramètre de randomisation*

$MaxGRASPIter$	Moyenne des solutions	Moyenne des temps de calcul CPU en sec
500	742.75	10.59
1000	739.26	20.74
1500	737.54	30.8
2000	736.69	41.9
2500	737.14	52.49
3000	737.69	60.92
3500	737.54	69.6
4000	736.53	80.4

TABLE 3.2: *Moyenne des solutions et des temps de calcul pour différentes valeurs du nombre d'itérations GRASP*

itérations, au seizième près, on obtient le même résultat qu'avec 4000. Nous avons finalement fixé à $MaxGRASPIter$ à 2000.

La méthode ILS possède deux principaux paramètres : un paramètre permettant de fixer le niveau de perturbation ($LimitMaxPerturbIter$) et un paramètre définissant le nombre d'itérations de la recherche locale ($MaxLSIter$). Des tests sont réalisés en vue de régler les valeurs des différents paramètres de la méthode.

En premier lieu, suite à des essais, le paramètre $LimitMaxPerturbIter$ a été fixé à 3. Pour ce, nous avons établi un budget 1500 itérations de recherche locale. Ce qui permet de réaliser un

$LimitMaxPerturbIter$	Moyenne des solutions ILS
2	916.59
3	915.56
4	920.65
5	920.76

TABLE 3.3: Moyenne des solutions ILS pour différentes valeurs du paramètre concernant le niveau de perturbation

échantillonnage suffisant de l'espace de solutions en vue de faire une analyse de sensibilité concernant le paramètre $LimitMaxPerturbIter$. Sur les cinq premières instances de MC1, la procédure ILS a été appliquée pour différentes valeurs de $LimitMaxPerturbIter$ allant de 2 à 5. En moyenne, comme affiché dans le tableau 3.3, les meilleurs résultats ont, en moyenne, été obtenus pour $LimitMaxPerturbIter = 3$.

Une fois la valeur de $LimitMaxPerturbIter$ fixée, le paramètre $MaxLSIter$ a été déterminé suite à des tests sur les trois premières instances de MC1 (tableau 3.4). Nous avons procédé en faisant varier $MaxLSIter$ de 500 à 4000 par incrément de 500. L'analyse de sensibilité montre que l'évolution des coûts moyens atteint un seuil de 912.85. Ce dernier est atteint à partir d'un budget de 2500 itérations de la recherche locale. Au delà de ce nombre d'itérations, les résultats ont plutôt tendance à converger. Les nouvelles valeurs obtenues fluctuent légèrement au-dessus du seuil déjà atteint. Par conséquent, la valeur de $MaxLSIter$ a été fixée à 2500.

3.5.3 Application au MC-VRP

Le paramétrage ayant été défini, les méthodes GRASP et ILS sont appliquées aux instances de MC-VRP de types : MC1, MC2, MC3, MC4, MC5, MC6, MC7 et MC8 (voir les tableaux 3.5, 3.6, 3.7 et 3.8). Les résultats obtenus pour les deux méthodes sont comparés. Les colonnes nommées ILS et GRASP contiennent les coûts des différentes solutions. La colonne $SolRef$ affiche les meilleurs résultats obtenus, à notre connaissance, pour le problème de MC-VRP le plus proche de notre problématique qui a été étudié par El Fallahi *et al.* [33] dans la littérature. El Fallahi *et al.* [33] ont également considéré le problème de MC-VRP avec livraisons multiples, plus précisément un client peut être livré lors de 2 tournées distinctes pour une seule demande. Les instances qui ont été résolues par ces derniers sont

<i>MaxLSIter</i>	Moyenne des solutions	Moyenne des temps de calcul CPU en sec
500	918.27	5.28
1000	917.68	11.1
1500	914.92	16.69
2000	913.86	21.93
2500	912.85	26.64
3000	915.99	32.69
3500	915.78	37.92
4000	914.21	43.33

TABLE 3.4: *Moyenne des solutions et des temps de calcul pour différents nombres d'itérations de recherche locale*

identiques à celles de MC1. Il en résulte que les comparaisons des résultats calculés avec ceux du BKS MC-VRP dans les tableaux sont plus significatives dans le cas des résultats pour MC1 que pour les autres types d'instances. Par ailleurs, dans les colonnes GAP_1 et GAP_2 des comparaisons sont effectuées respectivement entre les résultats de l'ILS et du GRASP et entre ceux de l'ILS et les solutions de référence (SolRef). Ces écarts sont calculés comme suit :

$$GAP_1 = \frac{(\text{Coût GRASP}) - (\text{Coût ILS})}{(\text{Coût ILS})} \times 100\%$$

$$GAP_2 = \frac{(\text{Coût SolRef}) - (\text{Coût ILS})}{(\text{Coût ILS})} \times 100\%$$

Globalement, l'ILS améliore les résultats du GRASP. En moyenne, pour les différents types d'instances MC1, MC2, MC3, MC4, MC5, MC6, MC7, MC8, les écarts moyens constatés au niveau des résultats sont respectivement de l'ordre de 0.4%, 0.05%, 0.03%, 0.16%, 0.56%, 0.28%, 0.66%, 0.41%. Les améliorations apportées par l'ILS sont assez faibles, ce qui s'explique, en grande partie, par la nature des problèmes traités. Ces derniers, contrairement au

Instance	n	ILS		GRASP		SolRef	%	%
		Coût	Temps CPU	Coût	Temps CPU	Coût	GAP_1	GAP_2
MC1-1	50	524.93	7.06	524.61	42.72	524.6	-0.06	-0.06
MC1-2	75	850.68	15.41	855.17	48.26	855.8	0.53	0.6
MC1-3	100	829.63	26.99	837.1	73.6	876.8	0.9	5.69
MC1-4	150	1044.18	63.97	1064.86	147.72	1089.6	1.98	4.35
MC1-5	199	1326.15	112.92	1338.77	269.43	1389.6	0.95	4.78
MC1-6	50	556.68	2.79	555.43	30.99	571.4	-0.22	2.64
MC1-7	75	929.33	8.65	920.11	58.4	933	-0.99	0.39
MC1-8	100	866.87	18.15	870.88	96.05	969.2	0.46	11.8
MC1-9	150	1176.35	32.14	1185.02	231.72	1230.9	0.74	4.64
MC1-10	199	1429.54	52.21	1465.85	388.96	1520.1	2.54	6.33
MC1-11	120	1049.19	52.29	1044.65	106.69	1046.1	-0.43	-0.29
MC1-12	100	819.56	28.18	819.56	88.06	820.6	0	0.13
MC1-13	120	1564.39	42.24	1554.01	163.28	1547.4	-0.66	-1.09
MC1-14	100	867.17	26.6	866.37	95.27	866.9	-0.09	-0.03
<i>Moyenne</i>							<i>0.4</i>	<i>2.85</i>
MC2-1	50	641.05	7.43	644.96	42.26	524.6	0.61	-18.17
MC2-2	75	1047.11	16.17	1055.1	58.73	855.8	0.76	-18.27
MC2-3	100	1003.61	38.11	1012.99	88.65	876.8	0.93	-12.64
MC2-4	150	1323.67	96.86	1339.17	199.32	1089.6	1.17	-17.68
MC2-5	199	1732.86	188.54	1733.59	399.44	1389.6	0.04	-19.81
MC2-6	50	642.76	5.46	641.04	36.37	571.4	-0.27	-11.1
MC2-7	75	1068.17	12.71	1068.44	47.36	933	0.03	-12.65
MC2-8	100	1008.72	26.81	1019.53	83.89	969.2	1.07	-3.92
MC2-9	150	1341.51	58.63	1364.24	213.83	1230.9	1.69	-8.25
MC2-10	199	1726.68	127.98	1735.19	398.45	1520.1	0.49	-11.96
MC2-11	120	1556.38	63.26	1519.54	148.44	1046.1	-2.37	-32.79
MC2-12	100	1141.16	28.26	1140.26	100.48	820.6	-0.08	-28.09
MC2-13	120	1653.92	43.39	1603.49	147.4	1547.4	-3.05	-6.44
MC2-14	100	1141.84	22.05	1138.66	69.44	866.9	-0.28	-24.08
<i>Moyenne</i>							<i>0.05</i>	<i>-16.13</i>

TABLE 3.5: Comparaison de l'ILS et du GRASP sur les types d'instances MC1 et MC2

Instance	n	ILS		GRASP		SolRef	%	%
		Coût	Temps CPU	Coût	Temps CPU	Coût	GAP_1	GAP_2
MC3-1	50	552.96	6.2	552.96	31.94	524.6	0	-5.13
MC3-2	75	900.83	14.63	896.67	56.82	855.8	-0.46	-5
MC3-3	100	869.64	28.37	872.47	89.85	876.8	0.33	0.82
MC3-4	150	1097.2	88.08	1116.21	177.11	1089.6	1.73	-0.69
MC3-5	199	1425.61	182.85	1431.79	338.62	1389.6	0.43	-2.53
MC3-6	50	558.99	3.28	556.68	41.71	571.4	-0.41	2.22
MC3-7	75	944.07	9.54	941.99	54.27	933	-0.22	-1.17
MC3-8	100	869.6	18.7	874.97	97.02	969.2	0.62	11.45
MC3-9	150	1223.81	21.46	1212.64	224.89	1230.9	-0.91	0.58
MC3-10	199	1449.03	80.18	1499.73	409.97	1520.1	3.5	4.9
MC3-11	120	1260.4	76.79	1211.05	133.91	1046.1	-3.92	-17
MC3-12	100	945.45	26.79	946.43	96	820.6	0.1	-13.21
MC3-13	120	1561.03	36.39	1554.12	161.07	1547.4	-0.44	-0.87
MC3-14	100	955.89	20.33	956.11	92.67	866.9	0.02	-9.31
<i>Moyenne</i>							<i>0.03</i>	<i>-2.5</i>
MC4-1	50	628.18	6.03	626.56	27	524.6	-0.26	-16.49
MC4-2	75	1020.48	16.61	1021.79	40.05	855.8	0.13	-16.14
MC4-3	100	984.42	38.74	992.37	100.5	876.8	0.81	-10.93
MC4-4	150	1302.7	108.78	1310.12	207.39	1089.6	0.57	-16.36
MC4-5	199	1686.23	194.61	1673.67	377.23	1389.6	-0.74	-17.59
MC4-6	50	632.52	4.79	625.2	41.69	571.4	-1.16	-9.66
MC4-7	75	1050.74	10.59	1052.81	52.19	933	0.2	-11.21
MC4-8	100	990.73	28.31	997.04	91.18	969.2	0.64	-2.17
MC4-9	150	1312.73	61.97	1337.37	214.98	1230.9	1.88	-6.23
MC4-10	199	1687.53	129.06	1706.44	389.99	1520.1	1.12	-9.92
MC4-11	120	1495.27	60.09	1475.83	144.79	1046.1	-1.3	-30.04
MC4-12	100	1132.54	26.28	1132.62	97.41	820.6	0.01	-27.54
MC4-13	120	1580.9	26.94	1580.07	150.29	1547.4	-0.05	-2.12
MC4-14	100	1131.83	24.2	1136.44	97.96	866.9	0.41	-23.41
<i>Moyenne</i>							<i>0.16</i>	<i>-14.27</i>

TABLE 3.6: Comparaison de l'ILS et du GRASP sur les types d'instances MC3 et MC4

Instance	n	ILS		GRASP		SolRef	%	%
		Coût	Temps CPU	Coût	Temps CPU	Coût	GAP_1	GAP_2
MC5-1	50	556.03	5.11	557.46	25.74	524.6	0.26	-5.65
MC5-2	75	899.63	11.87	910.1	41.33	855.8	1.16	-4.87
MC5-3	100	869.95	26.57	879.75	76.32	876.8	1.13	0.79
MC5-4	150	1109.94	95.44	1125.87	183.03	1089.6	1.44	-1.83
MC5-5	199	1424.93	162.08	1446.03	353.98	1389.6	1.48	-2.48
MC5-6	50	558.99	2.92	557.49	31.33	571.4	-0.27	2.22
MC5-7	75	944.58	10.44	957.57	50.28	933	1.38	-1.23
MC5-8	100	875.38	17.63	875.93	94.42	969.2	0.06	10.72
MC5-9	150	1184.12	33.22	1200.39	229.5	1230.9	1.37	3.95
MC5-10	199	1468.15	82.35	1495.98	408.48	1520.1	1.9	3.54
MC5-11	120	1262.53	72.23	1234.61	131.23	1046.1	-2.21	-17.14
MC5-12	100	961.98	33.1	960.88	94.62	820.6	-0.11	-14.7
MC5-13	120	1552.29	42.63	1552.74	157.58	1547.4	0.03	-0.32
MC5-14	100	963.85	26.94	966.34	101.8	866.9	0.26	-10.06
<i>Moyenne</i>							<i>0.56</i>	<i>-2.65</i>
MC6-1	50	677.72	7.84	677.68	32.76	524.6	-0.01	-22.59
MC6-2	75	1133.45	20.42	1143.07	44.13	855.8	0.85	-24.5
MC6-3	100	1074.47	40.76	1077.08	83.2	876.8	0.24	-18.4
MC6-4	150	1430.3	97.36	1441.57	212.02	1089.6	0.79	-23.82
MC6-5	199	1884.62	175.11	1878.21	404.57	1389.6	-0.34	-26.27
MC6-6	50	677.72	4.25	677.68	34.66	571.4	-0.01	-15.69
MC6-7	75	1141.49	10.09	1142.61	47.87	933	0.1	-18.26
MC6-8	100	1070.7	25.12	1077.95	84.38	969.2	0.68	-9.48
MC6-9	150	1441.82	60.52	1453.83	210.42	1230.9	0.83	-14.63
MC6-10	199	1873.93	117.06	1877.06	399.57	1520.1	0.17	-18.88
MC6-11	120	1687.13	64.97	1682.31	142.13	1046.1	-0.29	-38
MC6-12	100	1259.94	35.19	1256.84	109.28	820.6	-0.25	-34.87
MC6-13	120	1714.16	48.13	1720.65	146.01	1547.4	0.38	-9.73
MC6-14	100	1249.83	30	1259.52	91.76	866.9	0.78	-30.64
<i>Moyenne</i>							<i>0.28</i>	<i>-21.84</i>

TABLE 3.7: Comparaison de l'ILS et du GRASP sur les types d'instances MC5 et MC6

Instance	n	ILS		GRASP		SolRef	%	%
		Coût	Temps CPU	Coût	Temps CPU	Coût	GAP_1	GAP_2
MC7-1	50	524.61	5.82	530.26	31.77	524.6	1.08	0
MC7-2	75	847.94	12.1	854.72	42.83	855.8	0.8	0.93
MC7-3	100	830.99	26.78	832.4	77.59	876.8	0.17	5.51
MC7-4	150	1046.06	60.21	1064.72	151.68	1089.6	1.78	4.16
MC7-5	199	1333.84	111.2	1354.28	272.99	1389.6	1.53	4.18
MC7-6	50	556.68	3.13	555.43	34.75	571.4	-0.22	2.64
MC7-7	75	919.64	6.04	933.79	49.82	933	1.54	1.45
MC7-8	100	869.6	16.07	870.31	94.92	969.2	0.08	11.45
MC7-9	150	1179.19	33.77	1194.63	227.97	1230.9	1.31	4.39
MC7-10	199	1429.69	46.88	1456.62	392.41	1520.1	1.88	6.32
MC7-11	120	1049.04	39.27	1045.55	105.51	1046.1	-0.33	-0.28
MC7-12	100	819.56	19.97	819.56	89.45	820.6	0	0.13
MC7-13	120	1554.05	34.7	1552.42	156.71	1547.4	-0.1	-0.43
MC7-14	100	869.2	23.95	866.53	95.67	866.9	-0.31	-0.26
<i>Moyenne</i>							<i>0.66</i>	<i>2.87</i>
MC8-1	50	668.76	6.87	670.15	34.56	524.6	0.21	-21.56
MC8-2	75	1092.39	16.28	1095.5	45.62	855.8	0.28	-21.66
MC8-3	100	1044.05	30.61	1060.68	83	876.8	1.59	-16.02
MC8-4	150	1398.43	111	1403.17	202.45	1089.6	0.34	-22.08
MC8-5	199	1824.64	191.53	1825.57	392.48	1389.6	0.05	-23.84
MC8-6	50	668.02	4.11	672.37	31.07	571.4	0.65	-14.46
MC8-7	75	1116.02	10.88	1124.72	45.07	933	0.78	-16.4
MC8-8	100	1052.6	18.79	1062.02	84.29	969.2	0.89	-7.92
MC8-9	150	1399.52	54.48	1417.76	209.84	1230.9	1.3	-12.05
MC8-10	199	1837.67	120.18	1830.76	395.73	1520.1	-0.38	-17.28
MC8-11	120	1647.16	66.71	1615.28	143.64	1046.1	-1.94	-36.49
MC8-12	100	1189.93	33.19	1198.51	95.8	820.6	0.72	-31.04
MC8-13	120	1709.03	43.61	1706.05	107.43	1547.4	-0.17	-9.46
MC8-14	100	1186.14	31.43	1202.17	71.41	866.9	1.35	-26.91
<i>Moyenne</i>							<i>0.41</i>	<i>-19.8</i>

TABLE 3.8: Comparaison de l'ILS et du GRASP sur les types d'instances MC7 et MC8

VRP classique, présentent un espace de solutions faisables beaucoup plus restreint à cause de l'ajout de contraintes fortes telles que celles de compartimentage et celles de compatibilité entre les produits et les compartiments. Donc, la probabilité pour l'ILS de trouver des solutions améliorantes lors de son exploration de l'espace des solutions a été fortement atténuée.

Néanmoins, la comparaison des solutions trouvées pour les instances de MC1 avec les meilleures solutions que nous connaissons pour des instances similaires montre une amélioration significative des résultats. En moyenne, l'ILS améliore de 2.85% les résultats de El Fallahi *et al.* [33]. Ces derniers ont proposé un algorithme mémétique et une recherche taboue pour résoudre des instances identiques à celles de MC1. Ces deux méthodes sont reconnues comme étant très compétitives dans la littérature sur le VRP. Ce qui confirme la grande efficacité de la méthode ILS que nous avons proposée.

Par ailleurs, les résultats trouvés pour les autres types d'instances (MC2, MC3, MC4, MC5, MC6, MC7, MC8) sont nettement moins bons que les valeurs des solutions de référence. Cependant, il est à noter que les instances considérées sont différentes de celles d'El Fallahi *et al.* [33] (MC1 plus précisément). En effet, dans ces instances, les ratios entre les volumes des lots affectés aux compartiments et les ratios entre les tailles des compartiments sont différents.

3.6 Conclusion

A travers ce chapitre, le problème de MC-VRP avec un compartiment de taille fixe dédié à chaque produit a été étudié. Après une modélisation mathématique du problème sous la forme d'un PLNE, les méthodes de GRASP et d'ILS ont été proposées pour la résolution. Des comparaisons ont été effectuées entre les méthodes sur la base des solutions trouvées. Egalement, une comparaison des résultats de la meilleure des méthodes a été effectuée avec les meilleures solutions connues pour un problème de MC-VRP similaire à l'un des cas étudiés.

La comparaison des résultats montre un léger avantage en faveur de l'ILS. Les écarts moyens sont de l'ordre de 0.4%, 0.05%, 0.03%, 0.16%, 0.56%, 0.28%, 0.66%, 0.41% sur les huit types d'instances MC1, MC2, MC3, MC4, MC5, MC6, MC7, MC8 qui ont été résolues. Ces faibles écarts sont dus à la nature des contraintes. Par exemple, dans le cas des instances à deux

compartiments, deux contraintes de type sac à dos doivent être respectées au lieu d'une dans le VRP, ce qui rend le problème plus difficile.

Toutefois, la méthode ILS que nous avons proposée, s'est révélée très efficace en comparaison avec d'autres méthodes éprouvées dans la résolution des problèmes de VRP telles que la recherche taboue et l'algorithme mémétique. En particulier, les résultats obtenus par l'ILS sont inférieurs de 2.85% en moyenne par rapport aux meilleures solutions obtenues par El Fallahi *et al.* [33] qui avaient utilisé la recherche taboue et l'algorithme mémétique. Leurs résultats sont, à notre connaissance, les meilleurs connus dans la littérature pour les instances définies en MC1. Les résultats que nous avons obtenus ont été présentés à la conférence ISC 2012 (Joseph *et al.* [51]).

Chapitre 4

Extension à des compartiments de tailles flexibles

4.1 Introduction

Dans le chapitre précédent, nous nous sommes penchés sur le problème de MC-VRP avec un compartiment de taille fixe dédié à chaque produit demandé. Dans le présent chapitre, nous étudions une généralisation de ce problème. Nous la désignons sous le nom de VRP à compartiments flexibles ou *Flexible Compartments VRP* (FC-VRP). Son application se rencontre souvent dans des cas de transport de produits dans des camions frigorifiques avec compartiments.

4.2 Travaux antérieurs

Le FC-VRP a été très peu étudié dans la littérature. Certains auteurs ont travaillé sur des problématiques connexes de transport impliquant le compartimentage en dehors du champ des problèmes de tournées de véhicules. Par exemple, dans le transport maritime, Fagerholt et Christiansen [34] ont résolu un problème d'ordonnancement de cargos dans un contexte de collectes et de livraisons de marchandises. Le chargement est réparti dans des compartiments flexibles. L'espace dédié au chargement est partitionné à l'aide de cloisons amovibles, ce qui rend le compartimentage flexible. Une approche de partitionnement d'ensemble a été développée en vue de résoudre le problème en deux étapes : l'affectation des produits aux cargos et leur ordonnancement.

Dans un contexte de VRP, Chajakis et Guignard [17] ont travaillé sur un problème d'approvisionnements de boutiques en produits frais. Le compartimentage des véhicules étant configurable, ils ont étudié des modèles d'optimisation du dimensionnement des compartiments en vue de réduire les coûts.

Derigs *et al.* [25] ont intégré, outre des contraintes complexes de compatibilité entre produits et compartiments, la flexibilité des compartiments dans leurs travaux sur le MC-VRP. Ils ont présenté une approche globale de résolution basée sur la méthode d'optimisation *Large Neighborhood Search (LNS)*.

4.3 Modèle mathématique

Dans cette section, un FC-VRP est modélisé sous la forme d'un PLNE. Nous considérons une version du FC-VRP où la cloison entre deux compartiments successifs d'un véhicule peut être déplacée avec un pas discret. Dans le modèle ci-dessous, un FC-VRP avec une flotte homogène est représenté. Les véhicules disposant de compartiments flexibles possèdent les mêmes caractéristiques. Ils ont la même capacité Q . La capacité de chaque compartiment peut varier entre w_{min} et w_{max} avec un pas Δ , (voir schéma 4.1). Un ensemble P de produits est livré aux clients. Les demandes des clients sont telles que chaque client i commande une quantité non nulle d_{ip} pour au moins un produit p pouvant toujours tenir dans un seul compartiment. Chaque compartiment p ne peut contenir des demandes-clients que pour le produit p . Il y a donc autant de compartiments que de produits. L est la durée maximale autorisée pour chaque tournée et m le temps de service chez chaque client. Outre la variable de décision généralement utilisée en VRP x_{ijk} , d'autres variables sont introduites dans ce modèle :

$$x_{ijk} = \begin{cases} 1 & \text{si le véhicule } k \text{ traverse l'arc } (i, j) \text{ où } (i, j) \in A \text{ et } k \in K \\ 0 & \text{sinon} \end{cases} \quad (4.1)$$

$$y_{ik} = \begin{cases} 1 & \text{si le véhicule } k \text{ visite le nœud } i \\ 0 & \text{sinon} \end{cases} \quad (4.2)$$

w_{kp} : taille choisie pour le compartiment p du véhicule k

n_{kp} : nombre d'incrémentaires élémentaires de capacité Δ qu'il faut ajouter à la capacité minimale w_{min} pour obtenir la capacité requise pour le compartiment p du véhicule k

D'autres notations utilisées sont déjà introduites pour le CVRP en chapitre 2 section 2.2.2.2.

Voici la formulation du FC-VRP en PLNE :

Fonction-objectif :

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk} \quad (4.3)$$

Sous contraintes :

$$\sum_{k \in K} y_{ik} = 1 \quad \forall i \in V' \quad (4.4)$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik} = y_{ik} \quad \forall i \in V', \forall k \in K \quad (4.5)$$

$$\sum_{(i,j) \in A} c_{ij} x_{ijk} + \sum_{i \in V'} m y_{ik} \leq L \quad \forall k \in K \quad (4.6)$$

$$\sum_{i \in V'} d_{ip} y_{ik} \leq w_{kp} \quad \forall k \in K, \forall p \in P \quad (4.7)$$

$$\sum_{p \in P} w_{kp} = Q \quad \forall k \in K \quad (4.8)$$

$$w_{min} \leq w_{kp} \leq w_{max} \quad \forall k \in K, \forall p \in P \quad (4.9)$$

$$\sum_{i \in S} \sum_{j \in S, j \neq i} x_{ijk} \leq |S| - 1 \quad \forall S \subset V', |S| \geq 2, \forall k \in K \quad (4.10)$$

$$n_{kp} \geq \left(\sum_{i \in V'} d_{ip} y_{ik} - w_{min} \right) / \Delta \quad \forall k \in K, \forall p \in P \quad (4.11)$$

$$w_{kp} = w_{min} + n_{kp} \Delta \quad \forall k \in K, \forall p \in P \quad (4.12)$$

$$n_{kp} \in \mathbb{N} \quad \forall k \in K, \forall p \in P \quad (4.13)$$

$$x_{ijk} \in \{0, 1\} \quad \forall (i, j) \in A, \forall k \in K \quad (4.14)$$

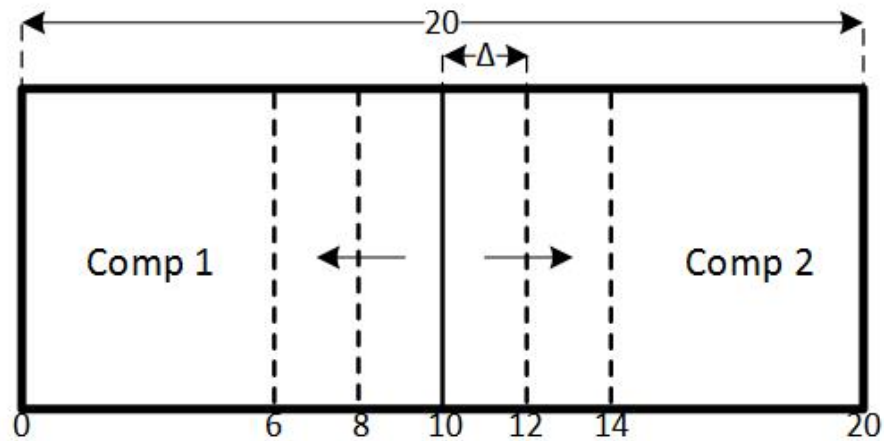


FIGURE 4.1: Illustration du dimensionnement des compartiments d'un véhicule

La fonction-objectif (4.3) a pour but de minimiser la somme des coûts de toutes les tournées. Les contraintes (4.4) assurent que chaque client est desservi une seule fois. La conservation des flots est imposée par (4.5). Une longueur maximale est imposée à chaque tournée en (4.6). Les demandes affectées à un compartiment ne peuvent en aucun cas violer sa capacité obtenue après configuration (4.7). La capacité de chaque véhicule est égale à la somme des capacités des compartiments après dimensionnement (4.8). Les inéquations (4.9) définissent les bornes pour les capacités des compartiments. A noter que les tailles des compartiments sont supposées varier dans le même intervalle. Les contraintes de sous-tours sont définies en (4.10). Les contraintes (4.11) et (4.12) déterminent le dimensionnement des compartiments. Elles permettent de dimensionner les tailles des compartiments au plus près des volumes de leurs chargements. Dans ce modèle, nous avons considéré une version où les capacités des compartiments varient de manière discrète mais notre modèle peut être facilement adapté à une version où les variations de capacités sont faites de manière continue. Pour ce, il suffirait de relaxer les variables w_{kp} et d'enlever les contraintes (4.11), (4.12) et (4.13).

4.4 Résolution exacte

La résolution du modèle mathématique, appliquée à des instances spécifiques, a permis d'obtenir des résultats numériques. Pour ce, nous avons utilisé le modelleur OPL Studio du solveur

Client	Qté Produit 1	Qté Produit 2
1	3	4
2	11	19
3	4	12
4	4	5
5	8	13
6	4	11
7	5	14

TABLE 4.1: *Demande des clients par produit*

CPLEX (version 12.2) qui permet de trouver des solutions optimales. Comme matériel, nous avons utilisé un PC Dell OPTIPLEX GX260 doté d'un processeur Pentium(R) D CPU 3.4GHz et de 1 Go de RAM tournant sur Windows XP. Le FC-VRP, étant une généralisation du VRP, est complexe à résoudre. Donc, des petits problèmes ont été définis en vue d'appliquer la méthode exacte.

4.4.1 Protocole

Les problèmes utilisés lors des expérimentations numériques ont été créés à partir des instances proposées par Christofides *et al.* [19] pour le CVRP. Nous avons fait des extractions de petits problèmes à partir de la première instance de la liste.

La taille des problèmes créés est limitée à 10 clients. Le plus petit débute avec 3 clients (en l'occurrence, les 3 premiers de la première instance de Christofides *et al.* [19]) et la taille est augmentée graduellement jusqu'à atteindre 10 clients. Quelques modifications ont été nécessaires pour transformer les problèmes extraits en purs problèmes de FC-VRP. Le contexte défini correspond à un problème de tournées avec une flotte de véhicules de capacité = 40, des clients qui sont livrés en deux types de produits. Les véhicules possèdent deux compartiments ayant chacun une capacité qui peut varier entre 10 et 30.

En ce qui a trait aux demandes des clients pour chaque produit, les quantités à livrer figurant dans l'instance originale ont été divisées en deux parties. La répartition qui est faite de manière arbitraire est présentée dans le tableau 4.1.

	F1	F2	F3	F4	F5	F6	F7	F8
Nbre de clients	3	4	5	6	7	8	9	10
Fonction-objectif	107.707	140.901	159.45	173.649	213.764	229.018	261.282	275.167
Temps CPU	0.38	38.56	26.43	29.36	133.94	48.48	52.35	353.095

TABLE 4.2: Résultats de la résolution exacte des problèmes de FC-VRP

4.4.2 Application

A travers le tableau 4.2, les résultats de la méthode exacte appliquée sur 8 problèmes de FC-VRP (F1, F2, F3, F4, F5, F6, F7, F8) sont présentés. Les résultats numériques qui sont pris en compte sont les valeurs de la fonction-objectif et les temps de calculs. La fonction-objectif correspond ici à la somme des coûts des tournées.

Nous avons fait un essai avec 11 clients, nous avons rencontré des problèmes de mémoire. De 3 à 10 clients, on passe de moins d'une seconde à près de 6 minutes. Cette rapide augmentation du temps de calcul montre clairement que l'approche de résolution exacte ne permet pas de résoudre des instances de grande taille dans des délais raisonnables. Pour ces problèmes de grande taille proches de ceux qu'on trouve dans la réalité, nous avons recours à d'autres approches appelées méthodes approchées ou heuristiques.

4.5 Adaptation du GRASP et de l'ILS

Pour pouvoir gérer les compartiments flexibles, le GRASP et l'ILS proposés dans le précédent chapitre ont été modifiés. Outre l'affectation des charges aux compartiments, un problème de dimensionnement de ces compartiments est résolu. Précisément, nous avons modifié la procédure *ChargerComp()* (voir algorithme 7) qui détermine les dimensions les mieux adaptées des compartiments tout en tenant compte de leurs charges respectives. Cette fonction, appelée lors des transformations opérées sur les tournées (mouvements de recherche locale, fusions ou perturbations), permet également de vérifier la faisabilité du chargement d'une tournée.

Algorithme 7 Algorithme de la procédure *ChargerComp()*

ENTRÉE(S): Capacité Q d'un véhicule, pas Δ utilisé pour la variation de la taille d'un compartiment

ENTRÉE(S): Charges q_{kp} affectées aux compartiments p dans chaque tournée R_k à modifier

si $(\forall p, \forall k, q_{kp} \leq w_{max})$ ET $(\forall k, \sum_{p \in P} q_{kp} \leq Q)$ **alors**

$\forall p, \forall k, w_{kp} = w_{min} + \lceil (q_{kp} - w_{min}) / \Delta \rceil \times \Delta$

si $\forall k, \sum_{p \in P} w_{kp} \leq Q$ **alors**

Renvoyer VRAI

sinon

Renvoyer FAUX

finsi

sinon

Renvoyer FAUX

finsi

4.6 Résultats numériques

4.6.1 Génération des instances

En vue de réaliser des tests numériques, nous avons adapté des instances de Christofides *et al.* [19] aux caractéristiques de la problématique étudiée. Les modifications portent sur :

- l'introduction de compartiments de taille variable aux véhicules
- la définition de produits, de familles ou types de produits
- la division de la demande de chaque client en deux parts comme au chapitre 3.

Pour les compartiments, nous avons choisi de limiter le nombre à deux pour faciliter les expérimentations mais notre méthode peut gérer bien plus de compartiments. En divisant par deux la taille du véhicule fournie dans les instances de Christofides *et al.* [19], la taille par défaut de chaque compartiment est ainsi définie. Ensuite, nous avons défini une marge de +/- 25% pour l'intervalle de variation de la taille de chaque compartiment autour de la valeur de sa taille par défaut. Chaque compartiment est affecté à un produit. Cela dit, le compartiment p correspond au produit p .

Pour les demandes des clients, nous avons gardé la même quantité par client que dans les instances originales. Toutefois, nous avons défini deux produits pouvant être livrés aux clients. Donc, la demande de chaque client est répartie en 2 parts : d_{i1} pour le produit 1 et d_{i2} pour

le produit 2.

En effectuant des variations au niveau des demandes des clients dans les instances originales de Christofides *et al.* [19], nous avons généré quatre catégories d'instances (FC1, FC2, FC3, FC4) qui correspondent chacune à un scénario :

1. Dans la catégorie FC1, les clients expriment chacun une demande scindée en deux lots de même quantité pour deux produits.
2. Dans le cas de FC2, le nombre de produits commandés varie d'un client à l'autre. Pour ce, une règle a été adoptée. Dans chaque instance, un client dont le numéro est impair commande un nombre impair de produits (1, en l'occurrence) et un client dont le numéro est pair demande un nombre pair de produits (2, en l'occurrence). Dans le cas où deux produits sont commandés par un client, la demande est répartie en deux parts égales.
3. En ce qui concerne FC3, chaque demande concerne deux produits mais en quantités différentes. Pour ce, les quantités associées aux produits dans une demande d'un client sont calculées en fonction d'une suite géométrique de raison $u = 1.25$. La quantité d_{ip} demandée par un client i pour un produit p est donnée par la formule suivante $d_{ip} = d_{ip-1}u^{p-1}$. La demande cumulée par client demeure, bien sûr, égale à la demande dans l'instance originale.
4. En FC4, le nombre de produits commandés varie d'un client à l'autre suivant la même règle qu'en FC2. De plus, lorsqu'un client commande deux produits, la quantité par produit varie comme en FC3.

4.6.2 Paramétrage

La machine utilisée pour pour les expérimentations est la même que celle utilisée dans le chapitre 3 section 3.5.2. Nous avons utilisé le langage de programmation C++ du compilateur Microsoft Visual Studio 2012.

Comme au chapitre 3, nous avons utilisé deux méthodes pour les expérimentations : le GRASP et l'ILS. La méthode GRASP possède deux principaux paramètres : un paramètre de randomisation α et le nombre d'itérations $MaxGRASPIter$. Le paramètre α a été fixé à 0.8 et le paramètre $MaxGRASPIter$ à 2000. La méthode ILS, d'autre part, possède le paramètre $LimitMaxPerturbIter$ que nous avons fixé à 3 et le paramètre $MaxLSIter$ défini

à 2500.

4.6.3 Application au FC-VRP

L'ILS et le GRASP ont été appliqués aux instances modifiées de Christofides *et al.* [19] (FC1, FC2, FC3, FC4) avec les paramètres définis. Les solutions obtenues pour les deux méthodes sont comparées. Les résultats sont affichés dans les tableaux 4.3, 4.4, 4.5 et 4.6. Les colonnes ILS et GRASP contiennent les coûts des différentes solutions. La colonne BKS VRP affiche les meilleurs résultats (*Best Known Solution*) pour les problèmes VRP connus dans la littérature. Car, à notre connaissance, il n'y pas eu de références dans la littérature sur des instances similaires aux nôtres permettant de comparer les résultats. Au final, dans les colonnes GAP_1 , GAP_2 et GAP_3 des comparaisons sont établies entre ILS et GRASP, entre ILS et BKS VRP et entre GRASP et BKS. Ces écarts sont calculés comme suit :

$$GAP_1 = \frac{(\text{Coût GRASP}) - (\text{Coût ILS})}{(\text{Coût ILS})} \times 100\%$$

$$GAP_2 = \frac{(\text{Coût ILS}) - (\text{Coût BKS})}{(\text{Coût BKS})} \times 100\%$$

$$GAP_3 = \frac{(\text{Coût GRASP}) - (\text{Coût BKS})}{(\text{Coût BKS})} \times 100\%$$

Les résultats obtenus par l'ILS sont meilleurs que ceux du GRASP. Pour les instances de type FC1, l'écart (0.3%) s'est révélé très faible. Nous n'avons pas observé une réelle dominance d'une méthode par rapport à l'autre. Par contre, pour les instances de types FC2, FC3 et FC4, les écarts sont significatifs. Ils sont respectivement de l'ordre de 17.44%, 2.82% et de 15.12% en faveur de l'ILS. Il en ressort que, sur ces instances-là, l'ILS démontre une meilleure performance que le GRASP.

Les deux méthodes ont par ailleurs montré une grande efficacité lorsqu'elles sont confrontées aux BKS des instances de VRP de Christofides *et al.* [19]. Celles-ci sont assez proches de la structure des instances FC1. Car, pour la création des instances FC1, nous avons procédé en scindant en deux parties égales les demandes dans les instances d'origine et nous avons considéré deux compartiments identiques dans les véhicules sans modifier leurs capacités initiales. Donc, le VRP sur les instances de Christofides *et al.* [19] est une relaxation du FC-VRP sur les instances FC1. Il suffit de fusionner les deux compartiments et les deux demandes. Par conséquent, la solution optimale du VRP sur l'instance d'origine fournit donc une borne inférieure pour notre problème. Or, nos deux méthodes, ILS et GRASP, se sont rapprochées respectivement de 1.07% et de 1.37% des BKS du VRP. Notons qu'il est remarquable de se rapprocher de si près des BKS d'une version relaxée de notre problème. Cela confirme que nos méthodes sont très compétitives.

Instance	n	ILS		GRASP		BKS VRP	%	%	%
		Coût	Temps CPU	Coût	Temps CPU	Coût	GAP_1	GAP_2	GAP_3
FC1-1	50	524.93	7.06	524.61	27.19	524.61	-0.06	0.06	0
FC1-2	75	850.68	15.41	851.92	51.23	835.26	0.15	1.85	1.99
FC1-3	100	829.63	26.99	832.92	104.99	826.14	0.4	0.42	0.82
FC1-4	150	1044.18	63.97	1061.62	193.09	1028.42	1.67	1.53	3.23
FC1-5	199	1326.15	112.92	1347.02	352.15	1291.45	1.57	2.69	4.3
FC1-6	50	556.68	2.79	555.43	38.04	555.43	-0.22	0.23	0
FC1-7	75	929.33	8.65	926.22	62.73	909.68	-0.33	2.16	1.82
FC1-8	100	866.87	18.15	870.7	124.21	865.94	0.44	0.11	0.55
FC1-9	150	1176.35	32.14	1181.99	338.34	1162.55	0.48	1.19	1.67
FC1-10	199	1429.54	52.21	1452.07	667.9	1395.85	1.58	2.41	4.03
FC1-11	120	1049.19	52.29	1043.65	144.08	1042.11	-0.53	0.68	0.15
FC1-12	100	819.56	28.18	819.56	118.12	819.56	0	0	0
FC1-13	120	1564.39	42.24	1550.12	252.25	1541.14	-0.91	1.51	0.58
FC1-14	100	867.17	26.6	866.37	129.17	866.37	-0.09	0.09	0
<i>Moyenne</i>							<i>0.3</i>	<i>1.07</i>	<i>1.37</i>

TABLE 4.3: Comparaison de l'ILS et du GRASP sur les types d'instances FC1

Instance	n	ILS		GRASP		BKS VRP	%	%	%
		Coût	Temps CPU	Coût	Temps CPU	Coût	GAP_1	GAP_2	GAP_3
FC2-1	50	564.1	19.22	647	35.05	524.61	14.7	7.53	23.33
FC2-2	75	896.3	19.96	1053.42	72.04	835.26	17.53	7.31	26.12
FC2-3	100	879.58	32.43	1011.98	112.31	826.14	15.05	6.47	22.49
FC2-4	150	1134.24	35.87	1334.67	260.32	1028.42	17.67	10.29	29.78
FC2-5	199	1386.39	56.1	1722.46	554.7	1291.45	24.24	7.35	33.37
FC2-6	50	616.66	22.06	646	36.74	555.43	4.76	11.02	16.31
FC2-7	75	971.88	18.86	1068.44	61.04	909.68	9.94	6.84	17.45
FC2-8	100	968.73	29.55	1015.58	109.98	865.94	4.84	11.87	17.28
FC2-9	150	1284.14	33.64	1357.76	322.08	1162.55	5.73	10.46	16.79
FC2-10	199	1519.18	41.22	1737.82	607.51	1395.85	14.39	8.84	24.5
FC2-11	120	1049.19	41.73	1508.93	206.7	1042.11	43.82	0.68	44.8
FC2-12	100	820.92	28.14	1139.11	137.2	819.56	38.76	0.17	38.99
FC2-13	120	1574.49	30.52	1594.32	198.8	1541.14	1.26	2.16	3.45
FC2-14	100	868.66	23.66	1142.39	109.2	866.37	31.51	0.26	31.86
<i>Moyenne</i>							<i>17.44</i>	<i>6.52</i>	<i>24.75</i>

TABLE 4.4: Comparaison de l'ILS et du GRASP sur les types d'instances FC2

Instance	n	ILS		GRASP		BKS VRP	%	%	%
		Coût	Temps CPU	Coût	Temps CPU	Coût	GAP_1	GAP_2	GAP_3
FC3-1	50	549.93	18.02	552.96	44.82	524.61	0.55	4.83	5.4
FC3-2	75	896.3	20.81	893	66.47	835.26	-0.37	7.31	6.91
FC3-3	100	876.83	33.18	874.4	100.13	826.14	-0.28	6.14	5.84
FC3-4	150	1129.16	74.58	1115.58	226.32	1028.42	-1.2	9.8	8.48
FC3-5	199	1386.39	109.81	1427.35	486.33	1291.45	2.95	7.35	10.52
FC3-6	50	556.68	2.74	556.68	38.01	555.43	0	0.23	0.23
FC3-7	75	944.54	7.76	946.03	60.68	909.68	0.16	3.83	4
FC3-8	100	909.51	26.73	869.6	117.98	865.94	-4.39	5.03	0.42
FC3-9	150	1189.61	22.2	1211.51	363.01	1162.55	1.84	2.33	4.21
FC3-10	199	1476.56	70.85	1473.74	669.03	1395.85	-0.19	5.78	5.58
FC3-11	120	1049.19	41.46	1210.54	184.3	1042.11	15.38	0.68	16.16
FC3-12	100	821.29	21.54	945.45	124.22	819.56	15.12	0.21	15.36
FC3-13	120	1552.12	32.68	1548.16	218.11	1541.14	-0.26	0.71	0.46
FC3-14	100	869.2	17.66	957.54	104.88	866.37	10.16	0.33	10.52
<i>Moyenne</i>							<i>2.82</i>	<i>3.9</i>	<i>6.72</i>

TABLE 4.5: Comparaison de l'ILS et du GRASP sur les types d'instances FC3

Instance	n	ILS		GRASP		BKS VRP	%	%	%
		Coût	Temps CPU	Coût	Temps CPU	Coût	GAP_1	GAP_2	GAP_3
FC4-1	50	564.1	22.6	626.56	48.57	524.61	11.07	7.53	19.43
FC4-2	75	896.3	19.71	1017.04	62.68	835.26	13.47	7.31	21.76
FC4-3	100	879.58	31.88	989.79	104.8	826.14	12.53	6.47	19.81
FC4-4	150	1134.24	36.43	1305.71	257.42	1028.42	15.12	10.29	26.96
FC4-5	199	1386.33	58.15	1682.51	545.67	1291.45	21.36	7.35	30.28
FC4-6	50	616.66	21.63	627.15	39.77	555.43	1.7	11.02	12.91
FC4-7	75	971.88	10.51	1047.29	58.66	909.68	7.76	6.84	15.13
FC4-8	100	968.73	24.53	989.66	109.51	865.94	2.16	11.87	14.29
FC4-9	150	1267.81	31.9	1334.26	332.89	1162.55	5.24	9.05	14.77
FC4-10	199	1512.83	43.22	1695.51	612.69	1395.85	12.08	8.38	21.47
FC4-11	120	1049.19	41.32	1468.99	211.53	1042.11	40.01	0.68	40.96
FC4-12	100	820.92	28.17	1133.64	133.39	819.56	38.09	0.17	38.32
FC4-13	120	1574.49	32.25	1580.37	197.22	1541.14	0.37	2.16	2.55
FC4-14	100	868.66	17.36	1135.44	94.7	866.37	30.71	0.26	31.06
<i>Moyenne</i>							<i>15.12</i>	<i>6.38</i>	<i>22.12</i>

TABLE 4.6: Comparaison de l'ILS et du GRASP sur les types d'instances FC4

4.7 Conclusion

Dans ce chapitre, une généralisation du VRP a été étudiée : le *Flexible Compartment Vehicle Routing Problem* ou FC-VRP. Après avoir appliqué une méthode de résolution exacte sur des problèmes de petites tailles, nous avons proposé les méthodes GRASP et ILS en vue de résoudre des problèmes de tailles plus importantes. Nous avons établi des comparaisons entre les solutions de ces deux méthodes. Etant donné qu'à notre connaissance, il n'existe pas de références dans la littérature sur des problèmes similaires, nous avons confronté nos résultats avec des BKS des instances de VRP de Christofides *et al.* [19]. La comparaison des résultats de l'ILS et du GRASP sur les instances FC1 montre un très faible écart de l'ordre de 0.3% en moyenne. Par contre, nous observons une nette dominance de l'ILS lors des comparaisons sur les instances FC2, FC3 et FC4. Les écarts moyens culminent à 17.44% sur les instances FC2 en faveur de l'ILS. Sur les autres instances FC3 et FC4, nous observons des écarts en moyenne de l'ordre de 2.82% et de 15.12%. Globalement, il en ressort que l'ILS s'est démarqué positivement en termes de performances par rapport au GRASP.

En outre, les résultats trouvés pour les instances de type FC1 dont la structure est la plus proche de celle des instances originales de Christofides *et al.* [19] sont comparés avec les meilleures solutions connues ou BKS en VRP. Les écarts constatés sont très faibles, 1.07% et de 1.37% respectivement pour l'ILS et le GRASP. Il est très remarquable de se rapprocher autant des BKS sur les instances de Christofides *et al.* [19]. En effet, ces instances étant une relaxation de nos instances FC1 qui sont obtenues en coupant en deux parties égales le compartiment et les demandes des clients dans le VRP d'origine. Les solutions optimales des instances de Christofides *et al.* [19] constituent des bornes inférieures aux solutions de FC1. Même lorsque les BKS du VRP n'ont pas été prouvées être des solutions optimales, il serait toutefois exceptionnel de les battre dans le cas des instances FC1. Signalons que plus de 50 métaheuristiques ont été publiées pour le VRP et que les BKS n'ont pas été améliorées depuis 2005. Ce qui confirme que les méthodes ILS et GRASP que nous avons proposées se révèlent très compétitives. De plus, le travail que nous avons réalisé sur le FC-VRP a donné lieu à une conférence ROADEF'11 (Joseph *et al.* [52]).

Les méthodes ILS et GRASP proposées ici ont permis d'étudier le FC-VRP qui, à notre connaissance, a fait l'objet de peu de travaux dans la littérature. Par ailleurs, le FC-VRP étant une généralisation du MC-VRP. Dans le chapitre suivant, nous aborderons une spécificité du MC-VRP. Il s'agit d'un problème de MC-VRP avec décisions d'affectation de chaque compartiment à un client et un produit.

Chapitre 5

Le MC-VRP avec décisions d'affectation de chaque compartiment à un client et un produit

5.1 Introduction

A travers ce chapitre, le MC-VRP avec décisions d'affectation de chaque compartiment de taille fixe à un client et un produit sera étudié. Lors du chargement d'un camion, une décision d'affectation de chaque couple (client, produit) est prise. Contrairement aux deux chapitres précédents, l'affectation d'un compartiment à un produit n'est pas définie par défaut et il n'y a pas d'incompatibilité entre compartiments et produits. Egalement, une demande d'un client pour un produit peut être répartie dans plusieurs compartiments (voir illustration 5.1). Cette problématique connaît une application industrielle dans le contexte agroalimentaire, en particulier, dans le secteur de la nutrition animale. Par exemple, chez Nestal, cette problématique est présente dans les tournées de livraisons en camions-citernes compartimentés. Les produits transportés étant des granulés, on ne peut pas utiliser un débitmètre lors d'une livraison, contrairement au contexte de livraisons de carburants liquides. Donc, la quantité à livrer à un client est pesée au dépôt et mise dans un compartiment.

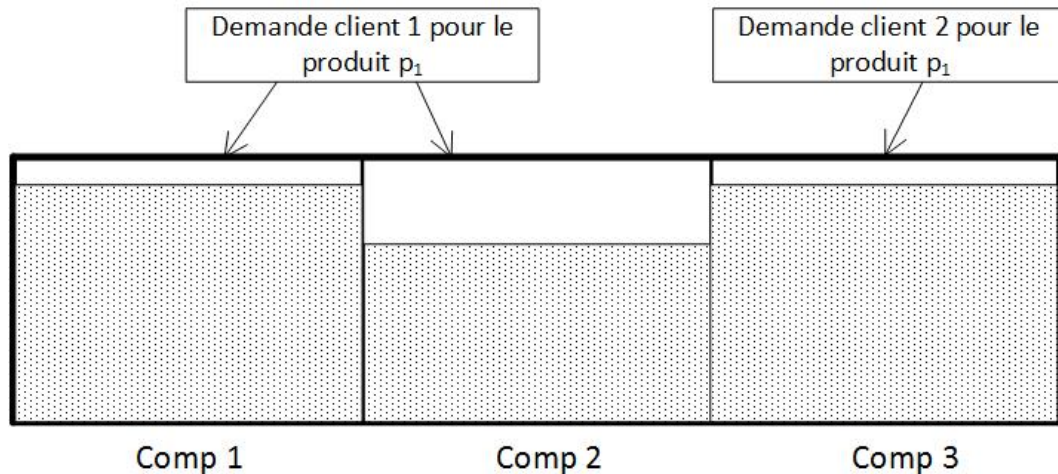


FIGURE 5.1: Exemple de chargement de compartiments

5.2 Travaux antérieurs

Dans la littérature, à part une étude réalisée par Ruiz *et al.* [88], nous n'avons pas relevé de travaux dédiés à cette problématique. En effet, les travaux de ces derniers portent sur l'optimisation des tournées de livraisons d'une industrie agroalimentaire localisée dans la péninsule ibérique. Au départ de chaque usine, des aliments destinés au bétail sont livrés au moyen de camions compartimentés. Pour résoudre ce problème, Ruiz *et al.* [88] ont proposé une approche qui procède en deux étapes. Dans un premier temps, une méthode d'énumération implicite est utilisée pour générer un sous-ensemble de routes de bonne qualité. Dans un second temps, un programme linéaire est utilisé pour sélectionner des routes formant une solution optimale.

5.3 Modèle mathématique

Dans cette section, une modélisation du MC-VRP avec décisions d'affectation de chaque compartiment de taille fixe à un client et un produit sous la forme d'un PLNE est présentée. Un ensemble P de produits est livré aux clients. Un client i commande une quantité $d_{ip} \geq 0$ du produit p . Pour l'ensemble de ses demandes, chaque client est livré en une seule visite. On dispose d'une flotte K de véhicules homogènes. Chaque véhicule, de capacité Q , possède un ensemble T de compartiments. Le nombre de compartiments n'est pas forcément égal au nombre de produits. Cependant, lors d'une tournée, le nombre de demandes

non nulles d_{ip} à livrer doit être, au plus, égal au nombre de compartiments. Une demande d_{ip} peut être scindée pour être chargée dans plus d'un compartiment. Chaque arc (i, j) a un coût de traversée c_{ij} . La durée de service m est identique pour chaque client. La longueur d'une tournée correspond à son coût additionné à la somme des durées de service des clients visités comme exprimée dans (5.8). Le modèle est présenté ci-dessous. D'autres notations non définies ici qui y sont utilisées sont déjà introduites dans le chapitre précédent.

Variables de décision :

$$x_{ijk} = \begin{cases} 1 & \text{si le véhicule } k \text{ traverse l'arc } (i, j) \text{ où } (i, j) \in A \text{ et } k \in K \\ 0 & \text{sinon} \end{cases} \quad (5.1)$$

$$y_{ik} = \begin{cases} 1 & \text{si le véhicule } k \text{ visite le nœud } i \\ 0 & \text{sinon} \end{cases} \quad (5.2)$$

$$z_{ip}^{kt} = \begin{cases} 1 & \text{si le compartiment } t \text{ du véhicule } k \text{ est affecté à } d_{ip} \\ 0 & \text{sinon} \end{cases} \quad (5.3)$$

$$q_{ip}^{kt} : \text{quantité de la demande } d_{ip} \text{ transportée dans le compartiment } t \text{ du véhicule } k \quad (5.4)$$

Fonction-objectif :

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk} \quad (5.5)$$

Sous contraintes :

$$\sum_{k \in K} y_{ik} = 1 \quad \forall i \in V' \quad (5.6)$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik} = y_{ik} \quad \forall i \in V', \forall k \in K \quad (5.7)$$

$$\sum_{(i,j) \in A} c_{ij} x_{ijk} + \sum_{i \in V'} m y_{ik} \leq L \quad \forall k \in K \quad (5.8)$$

$$\sum_{i \in S} \sum_{j \in S, j \neq i} x_{ijk} \leq |S| - 1 \quad \forall S \subset V', |S| \geq 2, \forall k \in K \quad (5.9)$$

$$q_{ip}^{kt} \leq w_t z_{ip}^{kt} \quad \forall i \in V', \forall p \in P, \forall k \in K, \forall t \in T \quad (5.10)$$

$$\sum_{i \in V'} \sum_{p \in P} z_{ip}^{kt} \leq 1 \quad \forall k \in K, \forall t \in T \quad (5.11)$$

$$\sum_{t \in T} q_{ip}^{kt} = d_{ip} y_{ik} \quad \forall i \in V', \forall p \in P, \forall k \in K \quad (5.12)$$

$$x_{ijk} \in \{0, 1\} \quad \forall (i, j) \in A, \forall k \in K \quad (5.13)$$

$$y_{ik} \in \{0, 1\} \quad \forall i \in V', \forall k \in K \quad (5.14)$$

$$q_{ip}^{kt} \in \mathbb{R}^+ \quad \forall k \in K, \forall t \in T, \forall i \in V', \forall p \in P \quad (5.15)$$

La fonction-objectif (5.5) minimise la somme des coûts de toutes les tournées. Les contraintes (5.6) assurent que chaque client est visité une seule fois. La conservation des flots est imposée par (5.7). L'expression en (5.8) définit la borne supérieure de la longueur de chaque tournée. L'interdiction des sous-tours est exprimée en (5.9). Les quantités des demandes d_{ip} affectées aux compartiments doivent être inférieures ou égales à leurs capacités respectives (5.10). L'expression en (5.11) assurent qu'une unique demande d_{ip} est affectée partiellement ou entièrement au compartiment t d'un véhicule k . Les contraintes (5.12) imposent qu'une demande d_{ip} soit chargée dans un ou des compartiments d'un véhicule. En (5.13), (5.14) et (5.15), les domaines de définition des variables de décision sont précisés.

5.4 Résolution heuristique

Nous avons opté pour une résolution par des méthodes heuristiques, étant données les limites d'une résolution exacte du modèle mathématique par des solveurs commerciaux comme CPLEX. Nous avons appliqué deux méthodes : le *Multi Start* ILS ou MS-ILS et le GRASP. En plus de leur simplicité d'implémentation, ces méthodes sont connues pour leur performance.

5.4.1 Description du MS-ILS

Le MS-ILS, proposé dans les travaux de Prins [83], est une métaheuristique dont le principe repose sur l'hybridation de deux autres métaheuristicques : le GRASP et l'ILS. Il emprunte au GRASP sa technique de randomisation et à l'ILS, l'ensemble de sa procédure. Précisément, le MS-ILS se base sur le principe de redémarrages successifs d'une procédure ILS appliquée à des solutions initiales distinctes. Cette approche a pour but de limiter les itérations improductives de la recherche locale dans des voisinages non prometteurs. Ce qui, par conséquent, donne lieu à des redémarrages de l'ILS. La structure globale du MS-ILS est illustrée dans l'algorithme 12.

5.4.2 Optimisation du chargement des véhicules compartimentés

Outre le problème de tournées de véhicules, il se pose un problème de *packing* ou de chargement des compartiments. En plus de l'optimisation des tournées, des décisions d'affectation des demandes aux compartiments doivent être effectuées. Ces affectations doivent être effectuées en vue de réduire le coût des tournées. D'où, la nécessité de recourir à une méthode d'optimisation en vue de maximiser le taux de chargement des véhicules.

5.4.2.1 Travaux dans la littérature

Dans la littérature, plusieurs approches ont été proposées pour résoudre le problème de chargement des compartiments de véhicules. Certaines de ces approches ne sont pas forcément dédiées au contexte du VRP. En particulier, le problème de remplissage de cases ou *Bin Packing* (Johnson [50]) qui consiste à minimiser le nombre de cases identiques utilisées pour ranger des objets de différentes tailles a connu une large application dans le MC-VRP. Le *Bin Packing* étant un problème NP-difficile, plusieurs méthodes heuristiques ont été développées pour sa résolution, dont voici quelques-unes :

1. *First Fit* (FF) : on place au fur et à mesure les objets dans la première case pouvant les recevoir ;
2. *Worst Fit* (WF) : on place au fur et à mesure les objets dans la case la moins remplie ;
3. *Best Fit* (BF) : on place, au fur et à mesure, les objets dans la case la plus remplie ;

4. *First Fit Decreasing* (FFD), *Best Fit Decreasing* (BFD) et *Worst Fit Decreasing* (WFD) : on trie les objets en fonction de leurs tailles par ordre décroissant puis on applique FF, BF ou WF.

Toutefois, dans la littérature spécifique au MC-VRP, outre des approches heuristiques basées sur le *Bin Packing*, les travaux sur l'optimisation du chargement utilisent des méthodes diverses. Par exemple, Yuceer [105] a résolu un problème d'affectation de différents types de carburants aux compartiments de véhicules approvisionnant des stations-services. Les compartiments d'un véhicule sont de tailles différentes et un seul type de carburant peut être chargé dans un compartiment. Partant de l'hypothèse que les stations ont des capacités de stockage illimitées, l'objectif recherché est de maximiser le délai avant la prochaine demande de livraison.

Bukchin et Sarin [16] ont poursuivi ce même objectif dans un contexte d'approvisionnement d'une station-service par un camion-citerne. Ils ont proposé deux approches : une approche statique et une approche dynamique. L'approche dynamique réoptimise l'affectation des produits aux compartiments de manière cyclique tandis que l'approche statique calcule, de manière définitive, une affectation des produits aux compartiments permettant de réduire la fréquence des livraisons nécessaires pour satisfaire les demandes.

Avella *et al.* [4] ont étudié un problème d'optimisation de tournées de livraisons de carburants à des stations-services à partir d'entrepôts. En plus du problème de tournées, il ont résolu un problème d'allocation des compartiments aux demandes des clients pour différents carburants. Pour ce, ils ont proposé une heuristique et une approche exacte. Leur heuristique combine une technique d'évaluation des tournées basée sur la méthode des *savings* et une technique de remplissage des compartiments proche du *First Fit Decreasing*. Pour la résolution exacte, les auteurs ont utilisé une formulation basée sur le partitionnement d'ensemble (*Set Partitioning*). Ensuite, se basant sur la solution de leur heuristique comme borne supérieure, ils ont appliqué la méthode de *Branch-and-Price* pour la résolution.

El Fallahi *et al.* [33] ont résolu un problème de tournées de livraisons de véhicules compartimentés dans un contexte agroalimentaire avec une flotte homogène. Les livraisons multiples sont admises à condition que chaque demande pour un produit soit livrée de manière intégrale en une seule fois. Le problème d'optimisation du chargement des véhicules ne s'est pas posé dans la mesure où chaque compartiment d'un véhicule est préaffecté à un produit. Le

nombre de produits à livrer ne dépasse donc pas le nombre de compartiments par véhicule. Par ailleurs, pour l'optimisation des tournées, les auteurs ont proposé deux méthodes : une recherche taboue et un algorithme mémétique.

5.4.2.2 Technique utilisée pour l'optimisation du chargement

Pour le problème que nous étudions ici, nous utilisons une approche heuristique combinant une approche de BFD et l'exploitation de données précalculées (*preprocessing*) en vue d'optimiser les calculs des tests de chargement des compartiments. En effet, le sous-problème d'affectation des demandes de produits p aux compartiments se révèle d'une grande complexité. En particulier, le fait que chaque demande d_{ip} puisse occuper plusieurs compartiments renforce la complexité combinatoire du calcul d'affectation. Car, dans les cas où la demande d'un client pour un produit dépasse la capacité du plus grand des compartiments disponibles, il faut tester différents scénarios de division de cette demande pour la charger dans plusieurs compartiments.

Compte tenu de cette complexité, nous avons opté pour une approche heuristique rapide pour ne pas alourdir davantage les calculs d'optimisation de tournées. La procédure de chargement étant appelée dans la phase de tests de faisabilité des tournées calculées, par conséquent, le sous-problème de chargement est résolu un très grand nombre (plusieurs dizaines de milliers) de fois lors de l'exécution de l'algorithme d'optimisation des tournées. Une résolution exacte pénaliserait fortement les temps de calculs.

La technique du *preprocessing* et l'approche du *Best Fit Decreasing* (BFD), coopèrent donc de façon à alléger le temps de calcul. La phase du *preprocessing*, basée sur un mécanisme de mémorisation, permet d'éviter toute redondance des calculs lors des simulations de chargements des véhicules. Pour ce, le problème de chargement des compartiments est assimilé à un problème d'affectation dans un graphe biparti. D'une part, il y a un ensemble de demandes d_{ip} caractérisées par leurs poids et, d'autre part, il y a les compartiments d'un véhicule définis par leurs capacités. Une demande d_{ip} peut être chargée dans plus d'un compartiment tandis qu'un compartiment ne peut être affecté qu'à une seule demande à la fois.

Lors d'un chargement, le mode opératoire est ainsi défini : de manière itérative, une demande donnée dans une liste est chargée dans un nombre optimisé de compartiments vides dans un véhicule. Comme il n'existe pas d'incompatibilité entre produits et compartiments, nous pouvons considérer qu'il s'agit de charger des compartiments avec une liste de quantités données. Nous avons donc déterminé l'ensemble $B = \{b_1, \dots, b_i, \dots\}$ des quantités distinctes et non nulles des demandes des clients pour chaque produit. Nous avons également défini la liste des combinaisons de compartiments pouvant être rencontrée lors des chargements, ce qui correspond à la liste des sous-ensembles de compartiments $\mathbb{P}(T) = \{P_1(T), \dots, P_i(T), \dots\}$. Donc, un scénario élémentaire de chargement consiste à charger une quantité b_i dans un ensemble $P_j(T)$ de compartiments. Afin d'accélérer les calculs lors des tests de chargement, nous avons créé une structure, mémorisant les résultats des scénarios élémentaires de chargement (algorithme 8). Chaque scénario possible est identifié et, dès qu'un scénario est rencontré pour la première fois lors d'une fusion de tournées, d'un mouvement de recherche locale ou d'une perturbation sur des tournées, nous gardons en mémoire les informations concernant la validité de la tournée, les compartiments occupés et les compartiments restés vides dans une structure de données. Celle-ci est exploitée par une approche basée sur le BFD à laquelle nous recourons pour les opérations de chargement.

Nous avons adapté le BFD aux caractéristiques du problème étudié dans ce chapitre pour effectuer les calculs d'affectation lors du chargement. Le BFD est connu dans la littérature pour son efficacité et sa rapidité. La solution fournie par un BFD est de bonne qualité, au pire des cas, Simchi-Levi [92] a prouvé que sa valeur est 1.5 fois plus élevée que celle de la solution optimale. A priori, le BFD s'applique au problème de *Bin Packing* où il faut remplir des cases identiques avec un certain nombre d'objets. Par contre, dans le cas précis du problème que nous étudions, les cases ne sont pas forcément identiques, elles peuvent être de tailles différentes et chacune ne peut recevoir qu'une seule demande d_{ip} (entièrement ou partiellement). Pour adapter le BFD au problème, il suffit de trier les compartiments à remplir par ordre de capacités croissantes et les remplir dans cet ordre. Par analogie au *Bin Packing*, cela revient donc à remplir en priorité les cases dont les capacités disponibles sont plus petites. En outre, étant donné que chaque compartiment ne peut recevoir qu'une demande d_{ip} au maximum, il n'est pas nécessaire de trier les lots par ordre de quantités croissantes car le résultat sera toujours le même.

A travers l'algorithme 8, nous illustrons précisément le fonctionnement de la méthode d'optimisation du chargement que nous avons proposée. Cet algorithme ne s'exécute pas de manière autonome mais est appelé au cours de l'exécution des métaheuristiques MS-ILS (algorithme 12) et GRASP (algorithme 13). Il est utilisé lors des tests de faisabilité à chaque mouvement dans la recherche locale ou perturbation et, également à chaque création ou fusion de tournées (algorithme de Clarke et Wright [21]). Toutes les transformations appliquées changent au maximum 2 tournées. Au cours de son exécution, il procède, d'abord, par la création des structures de données pour mémoriser l'ensemble des combinaisons de compartiments possibles et les résultats de calculs de chargement des demandes dans des compartiments.

Algorithme 8 Procédure de chargement des compartiments : *ChargerComp()*

ENTRÉE(S): Ensemble $P_j(T)$ de compartiments à charger initialement, les capacités w_t des compartiments, liste des l quantités b_i devant être chargées

ENTRÉE(S): Tableau de valeurs booléennes indiquant si chaque scénario de chargement est déjà simulé ou non : $ScénarioSimulé[P_j(T)][b_i]$

ENTRÉE(S): Tableau de valeurs booléennes indiquant si chaque scénario déjà simulé est réalisable ou non : $ScénarioRéalisable[P_j(T)][b_i]$

pour $i = 1, \dots, l$ **faire**

si $ScénarioSimulé[P_j(T)][b_i] = FAUX$ **alors**

$ScénarioSimulé[P_j(T)][b_i] \leftarrow VRAI$

Initialiser par défaut : $ScénarioRéalisable[P_j(T)][b_i] \leftarrow FAUX$

si $b_i \leq Capacité(P_j(T))$ **alors**

Ranger les compartiments de $P_j(T)$ par ordre de capacités croissantes

Initialiser : $QtéàCharger \leftarrow b_i$; $NbCompRemplis \leftarrow 0$

si $\exists t \in P_j(T)$ tel que $w_t \geq b_i$ **alors**

Prendre le plus petit compartiment t tel que $w_t \geq b_i$

Mise à jour quantité restante : $QtéàCharger \leftarrow 0$

$NbCompRemplis \leftarrow NbCompRemplis + 1$

sinon

répéter

Prendre le plus grand compartiment vide t

$QtéàCharger \leftarrow QtéàCharger - w_t$

$NbCompRemplis \leftarrow NbCompRemplis + 1$

jusqu'à ($QtéàCharger \leq 0$ OU $NbCompRemplis = Card(P_j(T))$)

finsi

si $QtéàCharger \leq 0$ **alors**

$ScénarioRéalisable[P_j(T)][b_i] = VRAI$

finsi

finsi

finsi

si $ScénarioRéalisable[P_j(T)][b_i] = FAUX$ **alors**

Sortir de la boucle

sinon

Mettre à jour $P_j(T)$ pour correspondre aux compartiments vides

finsi

fin pour

Renvoyer $ScénarioRéalisable[P_j(T)][b_i]$

Algorithme 9 Rappel de la procédure de génération d'une liste randomisée de *savings* : *RandSavingsListGen()*

ENTRÉE(S): liste des nœuds $(i, j) \in A$, nombre total de nœuds n , matrice des coûts c_{ij} , paramètre de randomisation : $0 < \alpha < 1$

SavingsList $\leftarrow \emptyset$

pour $i = 1, \dots, n$ **faire**

pour $j = i + 1, \dots, n$ **faire**

 Générer aléatoirement une valeur *Rand* comprise entre 0 et 1

si $Rand < \alpha$ **alors**

$s_{ij} = c_{0i} + c_{0j} - c_{ij}$

SavingsList $\leftarrow s_{ij}$

finsi

fin pour

fin pour

Trier *SavingsList* par valeurs décroissantes des s_{ij}

Renvoyer(*SavingsList*)

Algorithme 10 Clarke and Wright randomisée adaptée au problème de MC-VRP avec décisions d'affectation : *RandomCW()*

ENTRÉE(S): ensemble $A = \{a_{ij}\}$ des arcs, matrice des coûts c_{ij}

Créer une solution initiale *Soln* en attribuant une tournée directe à chaque client

Générer une liste de *savings* randomisés : *RandSavingsListGen()*

pour $i = 1, \dots, n$ **faire**

pour $j = i + 1, \dots, n$ **faire**

si $s_{ij} > 0$ **alors**

 Vérifier si i et j sont aux extrémités de 2 tournées distinctes R_i et R_j

 Vérifier si le chargement pour $R_i \oplus R_j$ est possible en exécutant *ChargerComp()*

 Vérifier le respect les contraintes de longueur de $R_i \oplus R_j$

 Exécuter la fusion de R_i et R_j

 Mettre à jour *Soln*

finsi

fin pour

fin pour

Renvoyer (*Soln*)

Algorithme 11 Procédure de perturbation adaptée au problème de MC-VRP avec décisions d'affectation : *Perturbation()*

ENTRÉE(S): *Soln, MaxPerturbIter*

PerturbIter \leftarrow 0

répéter

PerturbIter \leftarrow *PerturbIter* + 1

Initialiser : *Amélioration* \leftarrow FAUX

TentativesPerturb \leftarrow 0

répéter

Sélectionner aléatoirement 2 tournées distinctes R_1, R_2 dans *Soln*

Sélectionner aléatoirement les noeuds $n_1 \in R_1$ et $n_2 \in R_2$

Un test de faisabilité du mouvement est effectué (test de violation de capacité des compartiments avec *ChargerComp()* et des longueurs max des tournées)

si *TestFaisabilité(Soln)* = VRAI **alors**

Récupérer le coût actuel de *Soln* : *Valeur* \leftarrow *coût(Soln)*

Appliquer le mouvement d'échange sur n_1 et n_2 :

Soln \leftarrow *Echange(Soln, n₁, n₂)*

finsi

jusqu'à *TestFaisabilité(Soln)* = VRAI

si *coût(Soln)* < *Valeur* **alors**

Amélioration \leftarrow VRAI

finsi

Si *Soln* est améliorée, la procédure de perturbation peut être arrêtée prématurément

jusqu'à *Amélioration* = VRAI OU *PerturbIter* = *MaxPerturbIter*

Renvoyer(Soln)

Algorithme 12 Algorithme de la procédure MS-ILS

ENTRÉE(s): Ensemble $B = \{b_i\}$ des quantités distinctes et non nulles des demandes des clients pour chaque produit

ENTRÉE(s): $MaxLSIter$, $MaxIterNonAméliorant$

ENTRÉE(s): $MaxGlobalIter$, $LimitMaxPerturbIter$

Générer les partitions de l'ensemble des compartiments d'un véhicule :

$\mathbb{P}(T) = \{P_i(T)\}$ avec $P_i(T) \neq \emptyset$

Initialiser : $Soln^{**} \leftarrow +\infty$

$GlobalIter \leftarrow 0$

répéter

Appliquer heuristique de Clarke and Wright randomisée : $RandomCW()$

Appliquer une recherche locale tout en vérifiant le respect des contraintes de capacité avec $ChargerComp()$ et de longueur : $LS(Soln)$

$Soln^* \leftarrow Soln$

$GlobalIter \leftarrow GlobalIter + 1$

$Iter \leftarrow 1$

$MaxPerturbIter \leftarrow 1$

$IterNonAméliorant \leftarrow 0$

répéter

$Iter ++$

$GlobalIter \leftarrow GlobalIter + 1$

Appliquer une perturbation : $Perturbation(Soln)$

Appliquer une recherche locale tout en vérifiant le respect des contraintes de capacité avec $ChargerComp()$ et de longueur : $LS(Soln)$

si $cout(Soln) < cout(Soln^*)$ **alors**

$Soln^* \leftarrow Soln$

$MaxPerturbIter \leftarrow 1$

$IterNonAméliorant \leftarrow 0$

sinon

$MaxPerturbIter \leftarrow \text{Min}(MaxPerturbIter + 1, LimitMaxPerturbIter)$

$IterNonAméliorant \leftarrow IterNonAméliorant + 1$

finsi

jusqu'à $Iter = MaxLSIter$ OU $IterNonAméliorant = MaxIterNonAméliorant$ OU
 $GlobalIter = MaxGlobalIter$

si $cout(Soln^*) < cout(Soln^{**})$ **alors**

$Soln^{**} \leftarrow Soln^*$

finsi

jusqu'à $GlobalIter = MaxGlobalIter$

$Renvoyer(Soln^{**})$

Algorithme 13 GRASP adapté au problème de MC-VRP avec décisions d'affectation

ENTRÉE(S): ensemble $A = a_{ij}$ des arcs, matrice des coûts c_{ij}

Initialiser $\text{coût}(\text{Soln}^*) \leftarrow +\infty$

pour $Iter = 1, \dots, \text{MaxGRASPIter}$ **faire**

 Appliquer heuristique de Clarke and Wright randomisée : $\text{RandomCW}()$

 Appliquer une recherche locale tout en vérifiant le respect des contraintes de capacité avec $\text{ChargerComp}()$ et de longueur : $\text{LS}(\text{Soln})$

si $\text{coût}(\text{Soln}^*) > \text{coût}(\text{Soln})$ **alors**

$\text{Soln}^* \leftarrow \text{Soln}$

fin

fin pour

$\text{Renvoyer}(\text{Soln}^*)$

5.5 Résultats numériques

5.5.1 Génération des instances

En vue de réaliser des tests numériques, des instances sont créées grâce à une adaptation de celles de Christofides *et al.* [19]. Les modifications concernent principalement l'insertion de compartiments aux véhicules. La capacité attribuée à chaque véhicule par Christofides *et al.* [19] est répartie entre 10 compartiments. Les compartiments ne sont pas forcément de tailles homogènes.

A partir des modifications, deux instances sont générées : MC'1 et MC'2. En MC'1, chaque véhicule comporte 10 compartiments de tailles homogènes et chaque client ne demande qu'un seul lot ou produit. Par ailleurs, en MC'2, les tailles des compartiments sont hétérogènes. Chacun possède sa propre taille. Les tailles des compartiments sont dimensionnées en fonction d'une suite géométrique de raison $r = 1.15$. D'où, cette expression qui a été utilisée pour calculer la taille d'un compartiment t : $w_{kt} = w_{kt-1}r^{t-1} \quad t \in T, t > 1$.

5.5.2 Paramétrage

La machine utilisée pour les expérimentations est la même que celle utilisée dans le chapitre 3 section 3.5.2. Le langage de programmation C++ a été utilisé avec le compilateur Visual Studio 2012.

La métaheuristique MS-ILS que nous avons utilisée pour la résolution possède plusieurs paramètres. Il a donc été nécessaire de procéder à des essais en vue de trouver de bonnes valeurs aux paramètres. Ce qui a permis d'assurer l'efficacité de l'algorithme dans son contexte d'application.

Suite à des essais, nous avons fixé le paramètre de randomisation α à 0.8. Le nombre d'itérations de la procédure de recherche locale (*MaxGlobalIter*) a été limité à 3000. A chaque nouveau démarrage de l'ILS, le nombre d'applications de la recherche locale (*MaxLSIter*) est fixé à 1500. Le nombre maximal d'applications consécutives de la recherche locale sans amélioration (*MaxIterNonAméliorant*) est limité à 375 essais. La limite du niveau de perturbation (*LimitMaxPerturbIter*) appliquée à une solution est fixée à 3.

Le paramétrage du GRASP est similaire à celui utilisé au chapitre 3 section 3.5.2 sauf pour le paramètre du nombre d'itérations *MaxGRASPIter* dont la valeur a été fixée à 3000. Ce choix est fait dans le but donner le même budget de recherche locale que le MS-ILS.

5.5.3 Application au MC-VRP

Les métaheuristic MS-ILS et GRASP ont été appliquées sur les instances de types MC'1 et MC'2 que nous avons définies. Les résultats des deux méthodes ont été comparés entre eux. En l'absence de résultats sur des problèmes similaires dans la littérature, nous avons concentré notre analyse sur la comparaison des deux méthodes. Les résultats des comparaisons sont présentés dans les tableaux 5.1 et 5.2, en particulier, dans la colonne GAP_1 . Les écarts obtenus sont calculés comme suit :

$$GAP_1 = \frac{(\text{Coût GRASP}) - (\text{Coût MS-ILS})}{(\text{Coût MS-ILS})} \times 100\%$$

Les résultats obtenus permettent de constater une sensible amélioration des solutions par le MS-ILS comparativement au GRASP. Les solutions sont, en moyenne, améliorées de 0.2% pour les instances MC'1 et de 0.34% pour les instances MC'2. Les plus grands écarts, qui sont de l'ordre de 3.42% et de 3.52%, ont été respectivement observés sur les instances MC'1-7 et MC'2-7.

Les contraintes liées au mode de chargement des véhicules réduisent fortement l'espace des solutions réalisables. Le fait que chaque compartiment ne peut être utilisé que pour un seul lot, il devient très difficile de rapprocher le taux de chargement des véhicules de 100%. Il résulte donc qu'en dépit de la capacité du MS-ILS à réaliser un meilleur échantillonnage de l'espace de solutions comparativement au GRASP, les possibilités d'amélioration se sont révélées limitées.

Instance	GRASP		MS-ILS		%
	Coût	Temps CPU	Coût	Temps CPU	
MC'1-1	547.14	48.05	541.57	8.26	1.03
MC'1-2	861.04	60.83	844.1	21.92	2.01
MC'1-3	888.69	82.47	889.74	100.29	-0.12
MC'1-4	1125.7	143.94	1138.7	96.01	-1.14
MC'1-5	1385.64	247.81	1417.05	165.35	-2.22
MC'1-6	557.49	48.1	556.68	3.58	0.15
MC'1-7	943.55	60.31	912.34	8.09	3.42
MC'1-8	944.05	84.53	920.75	49.68	2.53
MC'1-9	1261.12	160.07	1229.76	87.64	2.55
MC'1-10	1531.12	268.16	1502.58	230.42	1.9
MC'1-11	1088.4	125.42	1176.7	269.38	-7.5
MC'1-12	827.5	92.45	826.3	74.8	0.15
MC'1-13	1575.56	123.67	1575.66	136.2	-0.01
MC'1-14	866.37	93.44	866.53	53.11	-0.02
<i>Moyenne</i>					<i>0.2</i>

TABLE 5.1: Comparaison du MS-ILS et du GRASP sur les instances de type MC'1

Instance	GRASP		MS-ILS		%
	Coût	Temps CPU	Coût	Temps CPU	GAP_1
MC'2-1	547.14	50.5	540.126	10.249	1.3
MC'2-2	865.76	58.87	847.671	22.198	2.13
MC'2-3	888.51	80.71	889.735	74.25	-0.14
MC'2-4	1135.67	143.85	1138.7	108.046	-0.27
MC'2-5	1395.35	247.99	1426.2	213.948	-2.16
MC'2-6	557.49	49.66	556.679	3.537	0.15
MC'2-7	945.33	59.23	913.229	7.524	3.52
MC'2-8	939	83.49	918.727	50.342	2.21
MC'2-9	1266.72	158.64	1226.57	90.994	3.27
MC'2-10	1523.64	267.77	1502.78	248.561	1.39
MC'2-11	1085.38	124.24	1153.08	283.175	-5.87
MC'2-12	822.93	92.17	828.968	64.92	-0.73
MC'2-13	1575.75	122.21	1574.6	108.621	0.07
MC'2-14	868.1	92.45	868.518	38.658	-0.05
<i>Moyenne</i>					<i>0.34</i>

TABLE 5.2: Comparaison du MS-ILS et du GRASP sur les instances de type MC'2

5.6 Conclusion

A travers ce chapitre, le MC-VRP avec décisions d'affectation de chaque compartiment de taille fixe à un client et un produit a été étudié. D'abord, le problème a été formalisé avec un modèle linéaire. Ensuite, les métaheuristiques MS-ILS et GRASP ont été utilisées pour résoudre des instances de Christofides *et al.* [19] modifiées pour être adaptées au problème.

Pour l'analyse des résultats, une comparaison des solutions trouvées par les deux méthodes a été effectuée. En l'absence de résultats sur un problème similaire dans la littérature à notre connaissance, les solutions respectivement trouvées par chaque méthode ont été comparées entre elles en calculant la distance qui les sépare. Les écarts moyens montrent une sensible amélioration apportée par le MS-ILS en comparaison au GRASP. En effet, les écarts moyens en faveur du MS-ILS sont de l'ordre de 0.2% pour les instances MC'1 et de 0.34% pour les instances MC'2.

Toutefois, les améliorations interviennent plus d'une fois sur deux sur les instances qui ont été résolues. Ce qui confirme que l'efficacité du MS-ILS est meilleure dans le contexte étudié, sans qu'il y ait pour autant une nette domination. Cela s'explique, en partie, par des contraintes de chargement très restrictives qui réduisent fortement l'espace de solutions réalisables et, par conséquent, les possibilités d'amélioration.

Les techniques d'optimisation proposées dans ce chapitre et qui ont été validées sur des instances théoriques seront utilisées sur des instances réelles provenant de l'activité de distribution chez Nestal dans le prochain chapitre.

Chapitre 6

Application industrielle

6.1 Introduction

Dans les précédents chapitres, plusieurs variantes du MC-VRP ont été étudiées. Nous avons développé des méthodes d'optimisation en vue les résoudre. Nous avons réalisé des tests sur des cas issus de la littérature ou sur des problèmes créés à partir de ceux-ci. Les résultats obtenus se sont révélés très compétitifs. Nous avons donc appliqué nos méthodes sur des cas industriels. Pour cela, nous avons utilisé des problèmes issus du contexte de distribution de Nestal.

6.2 L'activité de Nestal

Rappelons que Nestal est une industrie agroalimentaire spécialisée dans la nutrition animale. Elle est composée de quatre usines de fabrication d'aliments pour le bétail. Son activité repose, d'une part, sur la transformation de matières et, d'autre part, sur la revente de produits conditionnés. La production génère deux types de flux : un flux de produits conditionnés en sacs de 25 kilos sur des palettes et un flux de produits en vrac. La distribution est organisée en fonction des types de flux (conditionnés ou non) et des volumes commandés. Les livraisons s'effectuent en camions-citernes et camions-bennes pour les demandes de produits en vrac. Les camions-citernes sont dotés de compartiments hermétiques et servent à livrer différents lots d'au minimum 2 tonnes chacun. Les camions-bennes sont utilisés pour livrer des demandes de lots d'au moins 20 tonnes pour un seul produit à la fois. D'autre part, les livraisons des produits conditionnés s'effectuent en camions-savoyardes équipés en matériels

de manutention appropriés (chariots).

La demande globale est éclatée sur des lots de petits volumes (en sacs, des lots en moyenne d'une tonne et, en vrac, des lots en moyenne d'environ 4 tonnes). La majorité des commandes (plus de 55%) sont passées moins de 2 jours avant leurs dates de livraison.

Par ailleurs, sur l'année, on enregistre des variations saisonnières au niveau de la demande. Plus précisément, pour la demande des produits fabriqués et livrés en vrac, on enregistre un effet de saisonnalité au courant de l'année qui est caractérisée par deux saisons : « hiver » qui s'étend de novembre à mars saturant quelques usines même au-delà de leurs capacités nominales et « été » d'avril à octobre avec une charge dépassant rarement 80% de la capacité des usines.

La saison « hiver » est caractérisée par une demande accrue d'aliments pour le bétail étant donné que pendant cette période de l'année les éleveurs sont obligés de nourrir leur bétail à l'aide de farines et de granulés faute de pouvoir les nourrir en pleine nature (herbes, choux, maïs etc.). Tandis que pendant la période « été », les éleveurs nourrissent leur bétail dans les prairies, ce à quoi ils ajoutent des compléments alimentaires qui leur sont procurés par Nestal.

La répartition géographique de la demande autour des sites de Nestal ne se fait pas de manière symétrique. Les livraisons des commandes à partir des usines ne se font pas nécessairement à une clientèle de proximité.

La distribution représente une part importante dans l'activité de Nestal. La quasi-totalité de la distribution des produits commandés se fait en dehors des sites de production. Donc, les frais de transport de marchandises pèsent énormément au niveau des dépenses d'exploitation du groupe. Le service de transport chez Nestal est totalement externalisé. Il repose sur un partenariat avec un noyau de transporteurs privilégiés dont les tournées sont entièrement planifiées par Nestal. La flotte utilisée est hétérogène (camions-bennes, camions-bennes à fond mouvant, camions-citernes équipées de vis et ou soufflerie, savoyarde etc.) permettant ainsi de répondre à des besoins diversifiés de transport en fonction de la nature des produits.

L'optimisation des tournées de livraisons et d'approvisionnements est gérée indépendamment par plusieurs services (services des approvisionnements, d'ordonnements des tournées de

produits en sacs et en vrac et planification des livraisons en bennes). L'optimisation des tournées des produits en vrac livrés dans les camions-citernes et des produits conditionnés livrés en camions-savoyardes est assurée par un logiciel commercial qui prend en charge les problèmes de tournées de véhicules hétérogènes avec des contraintes de dépôts multiples et des fenêtres de temps.

Par conséquent, le contexte de livraisons chez Nestal implique trois types de problématiques de livraisons. Les livraisons en camions-citernes et savoyardes correspondent à des problématiques de tournées de véhicules. Pour les camions-savoyardes, il s'agit de problématiques de tournées de véhicules avec contraintes de capacité (CVRP) tandis qu'en ce qui concerne les camions-citernes, il s'agit de problèmes de tournées de véhicules multi-compartimentés (MCVRP). D'autre part, les livraisons en camions-bennes ne génèrent pas de tournées. Chaque livraison concerne une source et une destination unique, il s'agit, donc, de problèmes de livraisons à clients uniques (*Truck Load*). Dans le cadre de nos études d'optimisation menées chez Nestal, nous nous sommes particulièrement concentrés sur des problématiques de tournées en camions-citernes et camions-savoyardes.

6.3 Etude de lissage de la distribution journalière des produits conditionnés

6.3.1 Contexte

La distribution des produits conditionnés (fabriqués ou revendus) chez Nestal s'effectue au départ des magasins des sites de production (Saint Martin, Pauvres, Etienne). A Saint Martin, une forte variation des volumes journaliers de livraisons était observée (voir tableau 6.1), ce qui engendrait parfois des hausses d'activité qui perturbaient l'organisation déjà en place. Etant données les diverses activités de préparation nécessaires pour la mise en place des livraisons, le personnel du magasin de Saint Martin se trouvait parfois débordé dans les situations de pics de tonnage. Compte tenu de la problématique, nous avons réalisé une étude d'optimisation portant sur deux aspects : 1) la reconfiguration cartographique des zones de livraisons des produits en sacs 2) le lissage de la charge de distribution journalière du magasin du site de Saint Martin.

Jour de livraison	Demande moyenne	Ecart-Type	Min (90% des cas)	Max (90% des cas)
Lundi	63	13	47	79
Mardi	49	14	31	67
Mercredi	53	13	37	70
Jeudi	88	8	77	98
Vendredi	75	10	62	88

TABLE 6.1: *Données statistiques sur les fluctuations de la demande journalière exprimée en tonnes pendant les périodes chargées de 2009*

6.3.2 Mode opératoire

6.3.2.1 Constitution d'un échantillon de données

En vue d'alléger les calculs, un échantillon de données sur les périodes de hausse de l'activité de distribution a été constitué. En effet, les livraisons ont été regroupées par commune ; les points de livraisons dans chaque commune sont rassemblés en un seul point identifié à cette commune et les demandes sont cumulées. Le risque de pertes d'informations est minime dans la mesure où il y a, en moyenne, moins de 2 points de livraisons par commune livrée. La période échantillon choisie s'étale sur les deux semaines 40 et 41 qui sont les plus chargées au cours de l'année 2009.

6.3.2.2 Sectorisation de la zone de distribution

Une nouvelle cartographie de distribution a été définie. La zone de livraison est divisée en différents secteurs de livraison permettant de lisser la distribution journalière sur une semaine et d'optimiser les parcours kilométriques. La méthode de balayage ou *sweep algorithm* proposée par Gillett et Miller [41] a été utilisée pour définir les secteurs. Pour ce, le problème a été traité comme un TSP géant périodique. Tout en veillant au respect des contraintes journalières de cumul de temps de conduite et de capacité, un véhicule visite l'ensemble des clients d'un secteur associé à un jour donné. Chaque secteur est dans un premier temps dimensionné pour avoir une demande moyenne correspondant à la demande moyenne par jour dans une semaine de travail.

Jour de livraison	Demande moyenne	Ecart-Type	Min (90% des cas)	Max (90% des cas)
Lundi	64	9	53	76
Mardi	70	16	50	91
Mercredi	63	12	47	79
Jeudi	73	10	60	86
Vendredi	57	9	46	68

TABLE 6.2: *Données statistiques après lissage sur les fluctuations journalières des charges de livraisons en tonnes pendant les périodes de pics en 2009*

6.3.2.3 Simulation des tournées

Une fois les secteurs prédéfinis par la méthode *sweep algorithm*, la faisabilité des tournées nécessaires par secteur est évaluée en mode réel en vue de finaliser leur dimensionnement. Une simulation des tournées de livraisons a été effectuée en utilisant la méthode de Clarke et Wright [21]. Au cas où un secteur n'est pas validé, le processus de sectorisation est relancé en faisant en sorte que le secteur en question soit délesté d'au moins 5% de sa capacité initiale. Pour les besoins de cette simulation, la période de référence considérée concerne les deux semaines 40 et 41 de l'année 2009.

6.3.3 Application

L'étude réalisée a permis de lisser la charge de distribution journalière et, également, de réduire les coûts de livraisons. En ce qui a trait à la charge de distribution journalière, les fluctuations journalières ont été fortement réduites comme l'indiquent les données du tableau 6.2 et la courbe de comparaison entre la situation avant et après (figure 6.1). D'autre part, les coûts de livraisons sont réduits en raison d'un meilleur regroupement des zones géographiques visitées, ce que montre la comparaison des cartographies avant et après (figures 6.2 et 6.3). Pour faciliter la lecture des cartographies, les différentes zones de livraisons sont représentées par des points coloriés en fonction de leur affectation à un jour donné de livraison. Les économies en termes de distances totales parcourues sont de 6.21% en moyenne. Le ratio distance parcourue par tonne livrée (km/tonne) a chuté de 5.94% en moyenne.

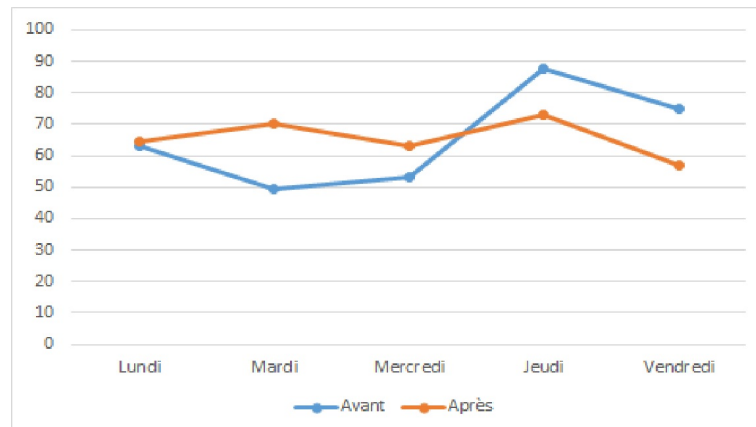


FIGURE 6.1: Comparaison des fluctuations journalières des charges de livraisons (en tonnes) avant et après lissage en périodes de pics en 2009

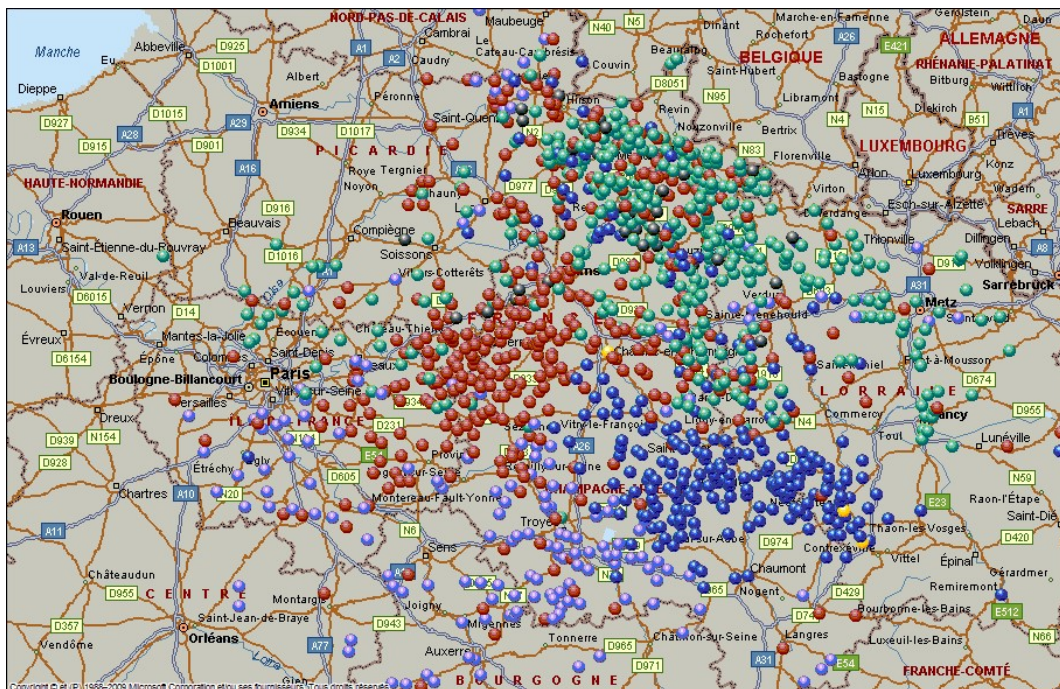


FIGURE 6.2: Cartographie de l'affectation journalière des zones de livraisons avant

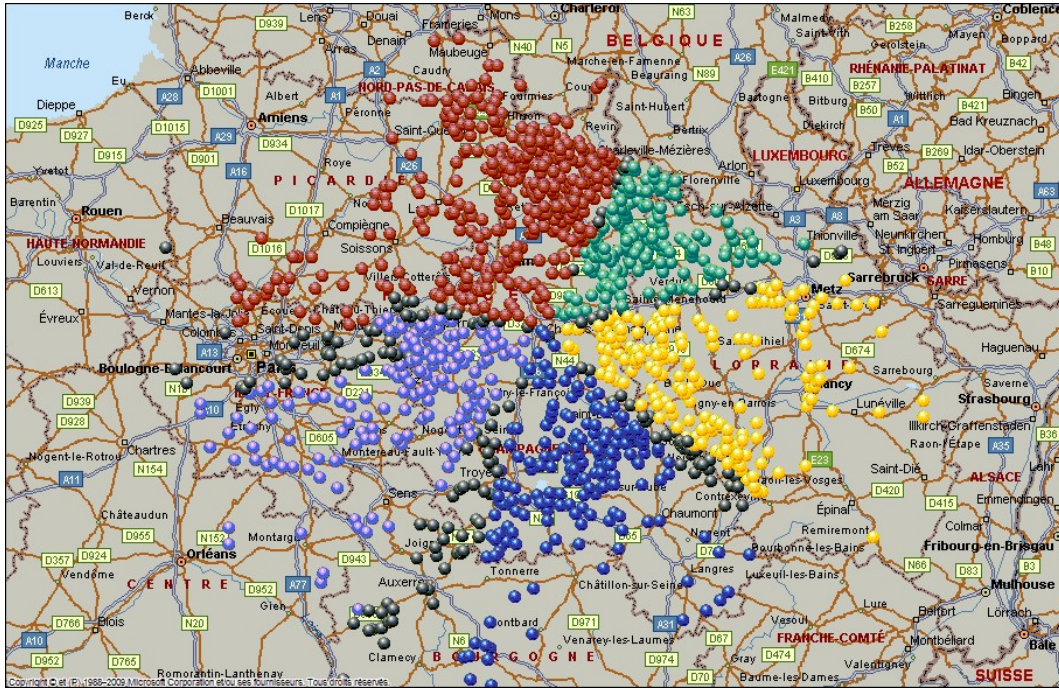


FIGURE 6.3: Cartographie de l'affectation journalière des zones de livraisons après

6.4 Evaluation des performances de l'outil d'optimisation des tournées chez Nestal

6.4.1 Contexte

Pour le calcul de ses tournées de livraisons, Nestal utilise un logiciel commercial, AXIODIS de la société OPTILOGISTIC. Outre les calculs des tournées, cet outil est utilisé pour la gestion de la flotte, des postes de conduite etc... En ce qui concerne les livraisons en véhicules compartimentés, il aide à la planification des tournées multi-dépôts avec des contraintes d'incompatibilité de lots et une flotte de véhicules hétérogènes. Compte tenu des résultats encourageants que nous avons obtenus avec les méthodes que nous avons proposées, nous avons décidé de comparer nos modèles par rapport au module d'optimisation des tournées de l'outil. Nos méthodes n'étant pas adaptées à certaines contraintes, nous avons évidemment restreint le champ d'analyse à des problèmes de MC-VRP avec des véhicules homogènes. Plus précisément, le contexte dans lequel nous avons réalisé les simulations de tournées prend en compte les différents aspects suivants :

1. Chaque véhicule est rattaché à un seul dépôt. Il part et rentre au même dépôt.
2. Le paramétrage des temps de chargement dans les différentes usines et les temps de

déchargement chez les clients est proportionnel au tonnage. Le débit de chargement et de déchargement est de 20 tonnes/heure. A cela s'ajoute un temps fixe de 5 minutes utilisé lors de chaque arrêt en vue d'un chargement ou déchargement. Ce temps est nécessaire pour les différentes manœuvres d'approche.

3. Chaque camion dispose de 8 compartiments : 5 compartiments de 4 tonnes et 3 de 2 tonnes.
4. Le total des heures de conduite lors des tournées quotidiennes pour un camion ne doit pas dépasser 9 heures, en général, et exceptionnellement 10 heures (2 fois/semaine). L'amplitude est égale aux heures de conduite auxquelles on ajoute 45 minutes de pause.

6.4.2 Mode opératoire

Les instances utilisées pour les tests sont nommées en accolant la première lettre de la ville du site au numéro du problème. Par exemple, l'instance n°1 du site situé à Saint Martin sur le pré est nommé S-Pb1. Pour constituer ces instances, un échantillon de 10 jours de livraisons a été retenu. D'une part, une planification des tournées a été réalisée avec l'outil utilisé chez Nestal. Le paramétrage de l'outil a été adapté pour résoudre les problèmes dans le contexte que nous avons défini : 1) flotte homogène 2) chaque véhicule rentre à son dépôt de départ. Ensuite, une optimisation des tournées a été effectuée en utilisant la méthode MS-ILS que nous avons développée pour le MC-VRP.

6.4.3 Résultats industriels

Les résultats obtenus suite aux tests sont affichés dans le tableau 6.3. Le tableau représente les distances cumulées parcourues par les véhicules chez Nestal lors des tournées. Nous considérons que les distances kilométriques correspondent aux coûts des tournées et que les coûts fixes sont nuls. Les deux colonnes de droite affichent les écarts entre les solutions obtenues (nombre de véhicules et km parcourus) par les différentes méthodes utilisées. Pour des raisons de confidentialité, les informations concernant le nombre de kilomètres et de véhicules utilisés ne sont pas affichées.

$$GAP_1 = \frac{(\text{NbrV Nestal}) - (\text{NbrV MS-ILS})}{(\text{NbrV MS-ILS})} \times 100\%$$

Instance	GAP_1 en %	GAP_2 en %
D-Pb1	16.67	14.32
D-Pb2	0	3.8
D-Pb3	0	1.42
D-Pb4	0	-2.65
D-Pb5	0	5.59
D-Pb6	11.11	4.38
D-Pb7	0	3.05
D-Pb8	0	-0.85
D-Pb9	9.09	9.62
D-Pb10	0	15.06
S-Pb1	12.5	10.01
S-Pb2	0	2.24
S-Pb3	0	1.1
S-Pb4	10	12.38
S-Pb5	10	11.61
S-Pb6	0	-4.07
S-Pb7	0	1.6
S-Pb8	9.09	7.81
S-Pb9	0	-2.63
S-Pb10	0	3.36
V-Pb1	12.5	5.68
V-Pb2	0	0.41
V-Pb3	12.5	7.94
V-Pb4	0	-0.47
V-Pb5	16.67	10.82
V-Pb6	0	1.94
V-Pb7	0	-0.5
V-Pb8	0	1.79
V-Pb9	0	8.59
V-Pb10	0	3.43
<i>Moyenne</i>	<i>3.31</i>	<i>4.55</i>

TABLE 6.3: Résultats industriels

$$GAP_2 = \frac{(\text{Coût Nestal}) - (\text{Coût MS-ILS})}{(\text{Coût MS-ILS})} \times 100\%$$

Les résultats affichés dans le tableau 6.3 montrent que la méthode MS-ILS améliore les tournées calculées par l'outil chez Nestal de 4.55% en moyenne. Il est à noter que la distribution en camions-citernes représente près de 70% de l'activité chez Nestal. Donc, cette amélioration de 4.55% peut potentiellement s'appliquer à 70% des activités de distribution. Nous observons également une baisse considérable de la flotte nécessaire pour les tournées de 3.31%. Les gains peuvent aller jusqu'à 15.06% (instance D-Pb10) en ce qui a trait aux distances et jusqu'à 16.67% en termes de réduction de flottes (instances D-Pb1, V-Pb5). Toutefois, on remarque, dans de rares cas, de meilleurs résultats fournis par l'outil utilisé chez Nestal. Cela reste assez exceptionnel dans la mesure où nos tests basés sur un échantillon représentatif de 40 instances confirment l'écart entre les performances. Par conséquent, le MS-ILS s'est donc avéré nettement plus performant que l'outil utilisé chez Nestal.

6.5 Etudes et travaux divers

En dehors des travaux de recherche et de ceux décrits précédemment, j'ai réalisé chez Nestal d'autres travaux et études dans le cadre de missions qui m'ont été confiées. Je les cite brièvement ci-dessous :

– Analyse et extraction de données

Etant donné qu'au début de la thèse, la société Nestal était récemment créée à partir d'une fusion-acquisition d'autres sociétés, les bases de données logistiques n'étaient pas homogénéisées. J'ai réalisé un travail de « data mining » c'est-à-dire d'analyse, de nettoyage et de formattage de données sur les bases de l'ensemble des usines pendant plusieurs mois. J'ai dû, pour cela, faire le tour des différents services (service clients, transport, production) où les saisies de données sont effectuées pour apprendre à interpréter les données. Car, celles-ci

étaient mélangées dans les fichiers dont je disposais pour faire mon travail.

– **Etude d'aménagement d'un magasin**

J'ai conduit un projet d'aménagement du magasin central chez Nestlé. J'ai conçu le plan d'aménagement de l'un des bâtiments de stockage. J'ai déterminé les types de palettiers et leur configuration dans le but d'optimiser leur taux d'occupation. Précisément, j'ai dimensionné la profondeur des alvéoles des palettiers pour permettre un stockage optimisé des différentes familles de produits. Par exemple, à des familles de produits dont les stocks moyens étaient faibles, j'ai affecté des palettiers pouvant recevoir une palette en profondeur, car on est sûr de réduire au minimum les espaces vides. Cela dit, une colonne de 3 niveaux avec une profondeur d'une palette par niveau affectée à des produits de famille A peut être remplie à 100% si on dispose d'un petit stock de 3 palettes de ces produits. Elle serait, par contre, remplie à 50% si elle avait une profondeur de 2 palettes par niveau.

– **Etude d'optimisation des flux d'approvisionnements**

J'ai réalisé une étude d'optimisation des flux d'approvisionnements des produits conditionnés. Une cartographie des flux a été effectuée. Une refonte des processus du système d'approvisionnements a été faite. J'ai défini un nouveau mode opératoire des demandes d'approvisionnements des usines basé sur une stratégie de mutualisation. J'ai mis en place une politique de réapprovisionnements basée à la fois sur une approche calendaire et de quantités minimales de commandes pour favoriser la massification des flux transportés. Egalement, j'ai proposé un nouvel arbitrage des modes de transport où ils ne sont plus définis, par défaut, par les fournisseurs (en « franco » ou en « départ ») mais sélectionnés après calculs des coûts comparatifs.

– **Conception d'outils d'aide à la décision**

J'ai conçu plusieurs outils logiciels d'aide à la décision en vue d'apporter des solutions à certains problèmes ou besoins. Par exemple, sur Excel, j'ai programmé avec VBA un outil de préfacturation des transports en camions-bennes chez Nestlé. Sur l'outil Access, j'ai programmé des requêtes en SQL pour automatiser des extractions de données dans le logiciel ERP utilisé dans l'entreprise. J'ai également conçu un outil d'analyse statistique des livraisons en termes de tonnages, fréquences et de distances parcourues.

6.6 Conclusion

A travers ce chapitre, nous avons présenté les applications industrielles de nos travaux de recherche chez Nestal. Dans un premier temps, nous avons résolu un problème de livraisons périodiques de produits conditionnés. Une nouvelle cartographie de distribution basée sur une approche de sectorisation en utilisant la méthode de Gillett et Miller [41]. Des gains substantiels de 6.21% ont été obtenus sur les coûts kilométriques de transport. Egalement, au niveau cartographique, une meilleure cohérence des zones de livraisons périodiques a été obtenue.

Dans un second temps, concernant l'activité de distribution en véhicules compartimentés, nous avons réalisé des tests pour évaluer les performances de l'outil de planification de tournées utilisé chez Nestal. Tous les types de contraintes rencontrés pendant les livraisons chez Nestal n'ont pas été inclus. Ceci pour pouvoir comparer dans les mêmes conditions que celles définies par les hypothèses retenues pour appliquer notre méthode MS-ILS. Nous avons donc utilisés des instances compatibles à la fois avec l'outil de Nestal et notre méthode. Les résultats obtenus sur un échantillon de 40 instances montrent que notre méthode MS-ILS est plus compétitive que l'outil utilisé chez Nestal. Le MS-ILS améliore de 4.55% les résultats obtenus chez Nestal.

Chapitre 7

Conclusion et perspectives

Au cours des dernières décennies, la mondialisation de l'industrie a provoqué une forte expansion du commerce international. La croissance exponentielle des échanges a profondément remodelé les systèmes logistiques. L'intérêt pour l'organisation des activités logistiques est devenu prépondérant d'autant plus que leurs coûts représentent une part importante des prix de revient des produits.

Les activités logistiques couvrent, entre autres, des domaines tels que les approvisionnements, le stockage, la production, le transport. En particulier, la logistique du transport concerne principalement le transport routier. Pour pouvoir livrer des marchandises à des clients, il s'avère nécessaire de planifier le routage de véhicules dans un réseau routier. Ce type de problème est complexe à résoudre efficacement. C'est pourquoi des chercheurs en recherche opérationnelle ont défini des méthodes d'optimisation pour résoudre ce problème.

Le but de cette thèse a été de traiter le problème de tournées de véhicules compartimentés (MC-VRP) dans le contexte de l'industrie de la nutrition animale. Ce sujet de thèse a été défini par Nestlé qui est une industrie agroalimentaire spécialisée dans la nutrition animale. Les travaux de cette thèse se sont donc concentrés en grande partie sur la distribution dans le domaine agroalimentaire.

Le problème étudié dans cette thèse est très présent dans la distribution dans le contexte de la nutrition animale. Il s'agit d'un problème de MC-VRP avec décisions d'affectation de chaque compartiment à un couple (client, produit). Nous considérons que ce couple forme un lot et, lors du chargement, chaque lot est divisible. Ce qui permet sa répartition dans plus d'un compartiment lorsque c'est nécessaire. Toutefois, ce problème a fait l'objet de

peu d'études dans la littérature des problèmes de MC-VRP. Notre contribution porte sur le développement d'une méthode MS-ILS permettant de résoudre le problème de tournées de véhicules compartimentés avec décisions d'affectation de chaque compartiment à un produit et à un compartiment. Notre méthode s'étend également aux problèmes incluant la notion de flexibilité de la taille des compartiments et d'incompatibilité entre produits et compartiments.

Le premier chapitre a présenté les différentes structures de la chaîne logistique. Une classification des niveaux de décisions correspondant aux différentes problématiques logistiques a été effectuée. Une description de la logistique du transport a été faite. Le chapitre a également présenté le contexte industriel de Nestlé qui a motivé notre étude des problèmes de tournées de véhicules compartimentés ou MC-VRP (*Multi-Compartment Vehicle Routing Problem*).

Le deuxième chapitre porte sur une synthèse de la littérature concernant les problèmes de transport routier. Les différentes méthodes de résolution des problèmes de tournées de véhicules ont été passées en revue. Il en est ressorti que le problème de MC-VRP abordé dans cette thèse est peu étudié, en particulier, dans le contexte de la distribution dans l'industrie de la nutrition animale.

Dans le troisième chapitre, le problème de MC-VRP avec un compartiment de taille fixe dédié à chaque produit a été étudié. Outre le GRASP (*Greedy Randomized Adaptive Search Procedure*), la méthode ILS (*Iterated Local Search procedure*) a été proposée. Elles ont été appliquées sur des instances modifiées de Christofides *et al.* [19]. Les résultats de l'ILS ont été comparés avec ceux obtenus dans les travaux d'El Fallahi *et al.* [33] pour le MC-VRP avec livraisons uniques. A notre connaissance, ce sont les meilleurs résultats pour ce problème dans la littérature. Ils ont été améliorés de 2.85% par notre méthode ILS.

Le quatrième chapitre a présenté une extension du problème abordé au troisième chapitre. Le problème de tournées de véhicules à compartiments flexibles (FC-VRP). Les méthodes d'optimisation ILS et GRASP ont été proposées. Leurs performances respectives ont été comparées. Il en résulte une nette dominance de l'ILS par rapport au GRASP. Nous avons confronté leurs solutions aux meilleures solutions connues (BKS) des instances de Christofides *et al.* [19]. Pour les instances dont la structure reste la plus proche de celle des instances originales de Christofides *et al.* [19], nos résultats se sont approchés à 1.07% et de 1.37% des BKS respectivement pour l'ILS et le GRASP. Ce qui confirme que nos méthodes GRASP et ILS sont très compétitives.

Le cinquième chapitre a présenté un MC-VRP avec décisions d'affectation de chaque compartiment à un client et un produit. Une méthode de *Multi Start* ILS (MS-ILS), à laquelle est intégrée une heuristique d'optimisation du chargement, a été proposée pour la résolution. Cette heuristique d'optimisation du chargement repose sur l'exploitation de données précalculées (*preprocessing*) et sur l'approche du *Best Fit Decreasing* (BFD). En l'absence de résultats sur un problème similaire dans la littérature à notre connaissance, les solutions du MS-ILS ont été comparées avec celles d'un ILS. Les écarts moyens montrent une sensible amélioration apportée par le MS-ILS en comparaison à l'ILS. En effet, les écarts moyens en faveur du MS-ILS sont de l'ordre de 0.2% et de 0.34% sur deux groupes d'instances.

Le sixième chapitre est dédié à des travaux d'application industrielle. Dans un premier temps, nous avons résolu un problème de livraisons périodiques de produits conditionnés. Une nouvelle cartographie de distribution basée sur une approche de sectorisation en utilisant la méthode de Gillett et Miller [41] a été proposée. Un lissage des charges de livraisons journalières a été obtenu. Des gains de 6.21% ont été obtenus sur les coûts kilométriques de transport. Dans un second temps, des tests d'évaluation des performances de l'outil de planification de tournées de véhicules compartimentés utilisé chez Nestal ont été réalisés. Les résultats obtenus sur un échantillon de 40 instances montrent que notre méthode MS-ILS améliore les résultats obtenus par l'outil de Nestal de 4.55%.

Nous aurions aimé développer plus de contributions théoriques mais, comme toutes les thèses CIFRE, de nombreuses tâches annexes ont dû être effectuées chez Nestal en plus du sujet de recherche proprement dit. Par exemple, par rapport à un sujet académique disposant de problèmes-tests et de données publiquement disponibles, la recherche de données nécessaires a pris un temps considérable dans l'entreprise. Les données se trouvaient souvent dans plusieurs logiciels ou bases de données, et dans des formats à uniformiser. Certaines données étaient même sur des bordereaux-papiers et elles ont dû être saisies.

Les résultats obtenus dans cette thèse sont très encourageants, il serait donc intéressant de poursuivre les travaux recherche sous d'autres angles. Il est souhaitable de développer une méthode exacte pouvant résoudre des instances de grande taille pour le MC-VRP. La résolution par des méthodes à populations basées, par exemple, sur les approches d'optimisation par colonies de fourmis ou par essaim particulière serait une piste intéressante à explorer. Nous envisageons, à l'avenir, d'intégrer d'autres familles de contraintes en plus de

celles traitées dans cette thèse. En particulier, les contraintes de dépôts multiples, de flottes hétérogènes, de fenêtres horaires et de livraisons multiples seront étudiées. En dernier lieu, compte tenu des résultats obtenus chez Nestlé, il serait souhaitable que d'autres partenariats de recherche soient mis en place dans le cadre d'études d'optimisation conjointe de la production, du stockage et de la distribution.

Bibliographie

- [1] *NF X 50-600, Logistique – Fonction et démarche logistiques.* AFNOR, 1999.
- [2] K ALTINKEMER et B GAVISH : Parallel Savings Based Heuristics for the Delivery Problem. *Operations Research*, 39:456–469, 1991.
- [3] D L APPLGATE, R E BIXBY, V CHVÁTAL et W J COOK : *The Traveling Salesman Problem : a computational study.* Princeton University Press, Princeton, NJ, 2007.
- [4] Pasquale AVELLA, Maurizio BOCCIA et Antonio SFORZA : Solving a fuel delivery problem by heuristic and exact approaches. *European Journal of Operational Research*, 152(1):170–179, janvier 2004. ISSN 03772217. URL <http://linkinghub.elsevier.com/retrieve/pii/S0377221702006768>.
- [5] Barrie M. BAKER et M.a. AYECHER : A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, 30(5):787–800, avril 2003. ISSN 03050548. URL <http://linkinghub.elsevier.com/retrieve/pii/S0305054802000515>.
- [6] R. BALDACCI, E. HADJICONSTANTINOÛ et a. MINGOZZI : An Exact Algorithm for the Capacitated Vehicle Routing Problem Based on a Two-Commodity Network Flow Formulation. *Operations Research*, 52(5):723–738, octobre 2004. ISSN 0030-364X. URL <http://or.journal.informs.org/cgi/doi/10.1287/opre.1040.0111>.
- [7] J E BEASLEY : Route first—Cluster second methods for vehicle routing. *Omega*, 11(4):403–408, 1983. ISSN 0305-0483. URL <http://www.sciencedirect.com/science/article/pii/0305048383900336>.
- [8] Tolga BEKTAS : The multiple traveling salesman problem : an overview of formulations and solution procedures. *Omega*, 34(3):209–219, juin 2006. ISSN 03050483. URL <http://linkinghub.elsevier.com/retrieve/pii/S0305048304001550>.
- [9] R E BELLMAN : On a routing problem. *Quart. Appl. Math.*, 18:87–90, 1958.
- [10] Richard BELLMAN : Dynamic programming and stochastic control processes. *Information and Control*, 1(3):228–239, septembre 1958. ISSN 00199958. URL <http://linkinghub.elsevier.com/retrieve/pii/S0019995858800030>.

- [11] Farah BELMECHERI, Christian PRINS, Farouk YALAOUI et Lionel AMODEO : Particle swarm optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows. *Journal of Intelligent Manufacturing*, 24(4): 775–789, mars 2012. ISSN 0956-5515. URL <http://link.springer.com/10.1007/s10845-012-0627-8>.
- [12] Dimitris J BERTSIMAS et David SIMCHI-LEVI : A New Generation of Vehicle Routing Research : Robust Algorithms, Addressing Uncertainty. *Operations Research*, 44(2):286–304, 1996. URL <http://or.journal.informs.org/content/44/2/286.abstract>.
- [13] L BODIN et B GOLDEN : Classification in vehicle routing and scheduling. *Networks*, 11(2):97–108, 1981.
- [14] Gerald G BROWN, Carol J ELLIS, Glenn W GRAVES et David RONEN : Real-Time, Wide Area Dispatch of Mobil Tank Trucks. *Interfaces*, 17(1):107–120, 1987. URL <http://interfaces.journal.informs.org/content/17/1/107.abstract>.
- [15] Gerald G BROWN et Glenn W GRAVES : Real-Time Dispatch of Petroleum Tank Trucks. *Management Science*, 27(1):19–32, 1981.
- [16] Yossi BUKCHIN et Subhash C. SARIN : Discrete and dynamic versus continuous and static loading policy for a multi-compartment vehicle. *European Journal of Operational Research*, 174(2):1329–1337, octobre 2006. ISSN 03772217. URL <http://linkinghub.elsevier.com/retrieve/pii/S0377221705003796>.
- [17] Emmanuel D. CHAJAKIS et Monique GUIGNARD : Scheduling Deliveries in Vehicles with Multiple Compartments. *Journal of Global Optimization*, 26:43–78, 2003.
- [18] N. CHRISTOFIDES et S. EILON : An Algorithm for the Vehicle- dispatching Problem. *Operations Research*, 20(3):309–318, 1969.
- [19] N CHRISTOFIDES, A MINGOZZI et P TOTH : The vehicle routing problem. In N CHRISTOFIDES, A MINGOZZI, P TOTH et C SANDY, éditeurs : *Combinatorial Optimization*, pages 315–338. Wiley, Chichester, 1979.
- [20] M L CHRISTOPHER : *Logistics and Supply Chain Management*. Pitman Publishing, London, 1992.
- [21] G CLARKE et J W WRIGHT : Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12:568–581, 1964.
- [22] Fabien CORNILLIER, Faye F. BOCTOR, Gilbert LAPORTE et Jacques RENAUD : A heuristic for the multi-period petrol station replenishment problem. *European Journal of Operational Research*, 191(2):295–305, décembre 2008. ISSN 03772217. URL <http://linkinghub.elsevier.com/retrieve/pii/S0377221707008910>.

- [23] G. DANTZIG, R FULKERSON et S. JOHNSON : Solution of a Large-Scale Traveling-Salesman Problem. *Journal of the Operations Research Society of America*, 2(4):393–410, 1954. URL <http://www.jstor.org/stable/166695>.
- [24] G B DANTZIG et J H RAMSER : The truck dispatching problem. *Operations Research*, 12:81–91, 1959.
- [25] U DERIGS, J GOTTLIEB, J KALKOFF, M PIESCHE, F ROTHLAUF et U VOGEL : Vehicle routing with compartments : applications, modelling and heuristics. *OR Spectrum*, 33(4):885–914, 2011.
- [26] M DESROCHERS, J K LENSTRAA et M W P SAVELSBERGH : A classification scheme for vehicle routing and scheduling problems. *European Journal of Operational Research*, 46(3):322–332, 1990.
- [27] M DESROCHERS et T W VERHOOG : *A matching based savings algorithm for the vehicle routing problem*. Numéro GERAD - G-89-04. Cahiers du GERAD, 1989.
- [28] P DEVERNAY : Le transport, "moteur" de la chaîne logistique. *Logistique et Management*, 5(1):75–79, 1997.
- [29] E. W. DIJKSTRA : A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, décembre 1959. ISSN 0029-599X. URL <http://www.springerlink.com/index/10.1007/BF01386390>.
- [30] M DORIGO, V MANIEZZO et A COLORNI : Ant system : optimization by a colony of cooperating agents. In *IEEE International Conference on systems, Man and Cybernetics-Part B*, volume 26, pages 29–41, 1996.
- [31] R C EBERHART et J KENNEDY : A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pages 39–43, Nagoya-Japan, 1995.
- [32] Burak EKSIÖGLU, Arif Volkan VURAL et Arnold REISMAN : The vehicle routing problem : A taxonomic review. *Computers & Industrial Engineering*, 57(4):1472–1483, novembre 2009. ISSN 03608352. URL <http://linkinghub.elsevier.com/retrieve/pii/S0360835209001405>.
- [33] Abdellah EL FALLAHI, Christian PRINS et Roberto WOLFLER CALVO : A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research*, 35(5):1725–1741, mai 2008. ISSN 03050548. URL <http://linkinghub.elsevier.com/retrieve/pii/S0305054806002516>.
- [34] K. FAGERHOLT et M. CHRISTIANSEN : A combined ship scheduling and allocation problem. *The Journal of the Operational Research Society*, 51(7):834–842, 2000.

- [35] Thomas A FEO et Jonathan F BARD : Flight Scheduling and Maintenance Base Planning. *Management Science*, 35(12):1415–1432, 1989. URL <http://mansci.journal.informs.org/content/35/12/1415.abstract>.
- [36] Thomas A. FEO et Mauricio G.C. RESENDE : Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, 6(2):109–133, 1995. URL <http://dx.doi.org/10.1007/BF01096763>.
- [37] B. A. FOSTER et D. M. RYAN : An Integer Programming Approach to the Vehicle Scheduling Problem. *Operational Research Quarterly (1970-1977)*, 27(2):367–384, 1976. URL <http://www.jstor.org/stable/3009018>.
- [38] Yuvraj GAJPAL et P.L. ABAD : Multi-ant colony system (MACS) for a vehicle routing problem with backhauls. *European Journal of Operational Research*, 196(1):102–117, juillet 2009. ISSN 03772217. URL <http://linkinghub.elsevier.com/retrieve/pii/S0377221708002300>.
- [39] Michel GENDREAU, Alain HERTZ et Gilbert LAPORTE : New Insertion and Postoptimization Procedures for the Traveling Salesman Problem. *Operations Research*, 40(6):1086–1094, 1992. URL <http://or.journal.informs.org/content/40/6/1086.abstract>.
- [40] Michel GENDREAU, Gilbert LAPORTE, Christophe MUSARAGANYI et Eric D. TAILLARD : A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, 26:1153–1173, 1999.
- [41] Billy E GILLETT et Leland R MILLER : A Heuristic Algorithm for the Vehicle-Dispatch Problem. *Operations Research*, 22(2):340–349, 1974. URL <http://or.journal.informs.org/content/22/2/340.abstract>.
- [42] Fred GLOVER : Tabu Search : A Tutorial. *Interfaces*, 20(4):74–94, 1990. URL <http://interfaces.journal.informs.org/content/20/4/74.abstract>.
- [43] David E GOLDBERG : *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, 1 édition, 1989. ISBN 0201157675.
- [44] B L GOLDEN, T L MAGNANTI et H Q NGUYEN : Implementing vehicle routing algorithms. *Networks*, 7(2):113–148, 1977. ISSN 1097-0037. URL <http://dx.doi.org/10.1002/net.3230070203>.
- [45] M HAIMOVICH et A H G RINNOOY KAN : Bounds and Heuristics for Capacitated Routing Problems. *Mathematics of Operations Research*, 10(4):527–542, 1985. URL <http://mor.journal.informs.org/content/10/4/527.abstract>.
- [46] William HO, George T.S. HO, Ping Ji et Henry C.W. LAU : A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial*

- Intelligence*, 21(4):548–557, juin 2008. ISSN 09521976. URL <http://linkinghub.elsevier.com/retrieve/pii/S0952197607000887>.
- [47] J H HOLLAND : *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor, 1975.
- [48] J H HOLLAND : *Adaptation In Natural And Artificial Systems : An Introductory Analysis With Applications To Biology, Control, And Artificial Intelligence*. Bradford Books. MIT Press, 1992. ISBN 9780262581110.
- [49] George Q. HUANG, Jason S. K. LAU et K. L. MAK : The impacts of sharing production information on supply chain dynamics : A review of the literature. *International Journal of Production Research*, 41(7):1483–1517, janvier 2003. ISSN 0020-7543. URL <http://www.tandfonline.com/doi/abs/10.1080/0020754031000069625>.
- [50] David S. JOHNSON : Fast algorithms for bin packing. *Journal of Computer and System Sciences*, 8(3):272–314, juin 1974. ISSN 00220000. URL <http://linkinghub.elsevier.com/retrieve/pii/S0022000074800267>.
- [51] C D JOSEPH, C PRINS, F YALAOUI et L AMODEO : A Multi Start Iterated Local Search algorithm for the multi compartment vehicle routing problem. In *ISC 2012, Industrial Simulation Conference. EUROSIS-ETI Publication, ISBN :978-90-77381-71-7*, pages 125–129, Brno, Czech Republic, 2012.
- [52] C D JOSEPH, F YALAOUI, C PRINS et L AMODEO : Problème de tournées de véhicules avec compartiments à taille variable. In *ROADEF'11 : Société Française de la Recherche Opérationnelle et d'Aide à la Décision*, Saint-Etienne, France, 2011.
- [53] J KENNEDY et R C EBERHART : A discrete binary version of the particle swarm algorithm. In *IEEE International Conference on systems, Man and Cybernetics*, volume 5, pages 4104–4108, Orlando-FL-USA, 1997.
- [54] G A P KINDERVATER et M W P SAVELSBERGH : Vehicle routing : Handling edge exchanges. In E H L AARTS et J K LENSTRA, éditeurs : *Local Search in Combinatorial Optimization*, pages 337–360. Wiley, Chichester, 1997.
- [55] S. KIRKPATRICK, C. D. GELATT et P. M. VECCHI : Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [56] L R KOPCZAK : Logistics partnerships and supply chain restructuring : survey results from the U.S. computer industry. *Production and Operations Management*, 6(3):226–247, 1997.
- [57] Nacima LABADI, Christian PRINS et Mohamed REGHIOUI : A MEMETIC ALGORITHM FOR THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS. *RAIRO - Operations Research*, 42(3):415–431, 2008.

- [58] A. H. LAND et A. G. DOIG : An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.
- [59] G. LAPORTE : Fifty Years of Vehicle Routing. *Transportation Science*, 43(4):408–416, octobre 2009. ISSN 0041-1655. URL <http://transci.journal.informs.org/cgi/doi/10.1287/trsc.1090.0301>.
- [60] G LAPORTE, Y NOBERT et M DESROCHERS : Optimal routing under capacity and distance restrictions. *Operations Research*, 33(5):1050–1073, 1985.
- [61] G LAPORTE et F SEMET : Classical heuristics for the capacitated VRP. In Paolo TOTH et Daniele VIGO, éditeurs : *The Vehicle Routing Problem*, pages 109–128. Society for Industrial and Applied Mathematics, 2002. ISBN 978-0-89871-851-5. URL <http://epubs.siam.org/doi/abs/10.1137/1.9780898718515>.
- [62] H L LEE et C BILLINGTON : Material management in decentralized supply chains. *Operations Research*, 41(5):835–847, 1993.
- [63] C.-L. LI, D SIMCHI-LEVI et M DESROCHERS : On the distance constrained vehicle-routing problem. *Operations Research*, 40(4):790–799, 1992.
- [64] Zhiye LI, Songshan GUO, Fan WANG et Andrew LIM : Improved GRASP with Tabu Search for Vehicle Routing with Both Time Window and Limited Number of Vehicles. In Bob ORCHARD, Chunsheng YANG et Moonis ALI, éditeurs : *Innovations in Applied Artificial Intelligence*, pages 552–561. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-22007-7. URL http://dx.doi.org/10.1007/978-3-540-24677-0_57.
- [65] Shen LIN : Computer solutions of the travelling salesman problem. *Bell System Technical Journal*, 44:2245–2269, 1965.
- [66] Helena R LOURENÇO, Olivier C MARTIN et Thomas STÜTZLE : Iterated Local Search. In Fred GLOVER et Gary A. KOCHENBERGER, éditeurs : *Handbook of Metaheuristics*, volume 57 de *International Series in Operations Research & Management Science*, chapitre 11, pages 320–353. Springer US, 2003. ISBN 978-1-4020-7263-5. URL http://dx.doi.org/10.1007/0-306-48056-5_11.
- [67] R R LUMMUS et R J VOKURKA : Defining supply chain management : a historical perspective and practical guidelines. *Industrial Management & Data Systems*, pages 11–17, 1999.
- [68] Jorge E. MENDOZA, Bruno CASTANIER, Christelle GUÉRET, Andrés L. MEDAGLIA et Nubia VELASCO : A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. *Computers & Operations Research*, 37(11):1886–1898, novembre 2010. ISSN 03050548. URL <http://linkinghub.elsevier.com/retrieve/pii/S0305054809001713>.

- [69] C. E. MILLER, A. W. TUCKER et R. A. ZEMLIN : Integer Programming Formulation of Traveling Salesman Problems. *Journal of the ACM (JACM)*, 7(4):326–329, 1960.
- [70] Hokey MIN : The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A : General*, 23(5):377–386, 1989. ISSN 0191-2607. URL <http://www.sciencedirect.com/science/article/pii/019126078990085X>.
- [71] R. H. MOLE et S. R. JAMESON : A Sequential Route-building Algorithm Employing a Generalised Savings Criterion. *Operational Research Quarterly (1970-1977)*, 27(2):503–511, 1976.
- [72] R H MOLE, D G JOHNSON et K WELLS : Combinatorial analysis for route first-cluster second vehicle routing. *Omega*, 11(5):507–512, 1983.
- [73] Yuichi NAGATA et Olli BRÄYSY : A powerful route minimization heuristic for the vehicle routing problem with time windows. *Operations Research Letters*, 37(5):333–338, septembre 2009. ISSN 01676377. URL <http://linkinghub.elsevier.com/retrieve/pii/S0167637709000662>.
- [74] Marvin D NELSON, Kendall E NYGARD, John H GRIFFIN et Warren E SHREVE : Implementation techniques for the vehicle routing problem. *Computers & Operations Research*, 12(3):273–283, 1985. ISSN 0305-0548. URL <http://www.sciencedirect.com/science/article/pii/0305054885900267>.
- [75] Clara NOVOA et Robert STORER : An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 196(2):509–515, juillet 2009. ISSN 03772217. URL <http://linkinghub.elsevier.com/retrieve/pii/S0377221708003172>.
- [76] Ilhan OR : *Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking*. Thèse de doctorat, Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 1976.
- [77] Ibrahim Hassan OSMAN : Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41(4):421–451, décembre 1993. ISSN 0254-5330. URL <http://dx.doi.org/10.1007/BF02023004><http://www.springerlink.com/index/10.1007/BF02023004>.
- [78] H PAESSENS : The savings algorithm for the vehicle routing problem. *European Journal of Operational Research*, 34(3):336–344, 1988. ISSN 0377-2217. URL <http://www.sciencedirect.com/science/article/pii/0377221788901543>.
- [79] J. PEARL : *Heuristics : Intelligent search strategies for computer problem solving*. Addison-Wesley Publishing Co., Inc., Reading, MA, New-York, 1984.

- [80] Victor PILLAC, Michel GENDREAU, Christelle GUÉRET et Andrés L. MEDAGLIA : A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11, février 2013. ISSN 03772217. URL <http://linkinghub.elsevier.com/retrieve/pii/S0377221712006388>.
- [81] Jean-Yves POTVIN et Jean-Marc ROUSSEAU : A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research*, 66(3):331–340, 1993. ISSN 0377-2217. URL <http://www.sciencedirect.com/science/article/pii/0377221793902218>.
- [82] Christian PRINS : A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985–2002, octobre 2004. ISSN 03050548. URL <http://linkinghub.elsevier.com/retrieve/pii/S0305054803001588>.
- [83] Christian PRINS : A GRASP x Evolutionary Local Search Hybrid for the Vehicle Routing Problem. In Francisco BAPTISTA PEREIRA et Jorge TAVARES, éditeurs : *Bio-inspired Algorithms for the Vehicle Routing Problem*, volume 161 de *Studies in Computational Intelligence*, pages 35–53. Springer Berlin Heidelberg, 2009. ISBN 978-3-540-85151-6. URL http://dx.doi.org/10.1007/978-3-540-85152-3_2.
- [84] Christian PRINS : Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence*, 22(6):916–928, septembre 2009. ISSN 09521976. URL <http://linkinghub.elsevier.com/retrieve/pii/S0952197608001693>.
- [85] T. K. RALPHS : Parallel branch and cut for capacitated vehicle routing. *Parallel Computing*, 29:607–629, 2003.
- [86] Marc REIMANN, Karl DOERNER et Richard F HARTL : D-Ants : Savings Based Ants divide and conquer the vehicle routing problem. *Computers & Operations Research*, 31(4):563–591, avril 2004. ISSN 03050548. URL <http://linkinghub.elsevier.com/retrieve/pii/S0305054803000145>.
- [87] Jacques RENAUD, Fayez F. BOCTOR et Gilbert LAPORTE : An Improved Petal Heuristic for the Vehicle Routing Problem. *Journal of the Operational Research Society*, 47(2):329–336, 1996. URL <http://www.jstor.org/stable/2584352>.
- [88] Rubén RUIZ, Concepción MAROTO et Javier ALCARAZ : A decision support system for a real vehicle routing problem. *European Journal of Operational Research*, 153(3):593–606, mars 2004. ISSN 03772217. URL <http://linkinghub.elsevier.com/retrieve/pii/S0377221703002650>.

- [89] David M. RYAN, Curt HJORRING et Fred GLOVER : Extensions of the Petal Method for Vehicle Routing. *The Journal of the Operational Research Society*, 44(3):289–296, 1993. URL <http://www.jstor.org/stable/2584199>.
- [90] X H SHI, Y C LIANG, H P LEE, C LU et Q X WANG : Particle swarm optimization-based algorithms for TSP and generalized TSP. *Information Processing Letters*, 103(5):169–176, 2007. ISSN 0020-0190. URL <http://www.sciencedirect.com/science/article/pii/S0020019007000804>.
- [91] D SIMCHI-LEVI, P KAMINSKY et E SIMCHI-LEVI : *Designing and managing the supply chain. Concepts, strategies and case studies*. McGraw-Hill\Irwin, Boston, second édition, 2003.
- [92] David SIMCHI-LEVI : New worst-case results for the bin-packing problem. *Naval Research Logistics*, 41(4):579–585, juin 1994. ISSN 0894069X. URL <http://doi.wiley.com/10.1002/1520-6750%28199406%2941%3A4%3C579%3A%3AAID-NAV3220410409%3E3.0.CO%3B2-G>.
- [93] Marius M SOLOMON : Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*, 35(2):254–265, 1987. URL <http://or.journal.informs.org/content/35/2/254.abstract>.
- [94] É TAILLARD : Parallel iterative search methods for vehicle routing problems. *Networks*, 23(8):661–673, 1993. ISSN 1097-0037. URL <http://dx.doi.org/10.1002/net.3230230804>.
- [95] Fermín Alfredo TANG MONTANÉ et Roberto Diéguez GALVÃO : A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research*, 33(3):595–619, mars 2006. ISSN 03050548. URL <http://linkinghub.elsevier.com/retrieve/pii/S0305054804001741>.
- [96] S TAYUR, R GANESHAN et M MAGAZINE : *Quantitative models for supply chain management*, volume 17 de *International Series in Operations Research & Management Science*. Springer, 1999.
- [97] Paul M THOMPSON et Harilaos N PSARAFTIS : Cyclic Transfer Algorithm for Multi-vehicle Routing and Scheduling Problems. *Operations Research*, 41(5):935–946, 1993. URL <http://or.journal.informs.org/content/41/5/935.abstract>.
- [98] Paolo TOTH et Daniele VIGO, éditeurs. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, 2002. ISBN 978-0-89871-851-5. URL <http://epubs.siam.org/doi/abs/10.1137/1.9780898718515>.
- [99] A VAN BREEDAM : *An analysis of the behavior of heuristics for the vehicle routing problem for a selection of problems with vehicle-related, customer-related, and*

- time-related constraints*. Thèse de doctorat, PhD dissertation, University of Antwerp, Belgium, 1994.
- [100] Lambert VAN DER BRUGGEN, Ruud GRUSON et Marc SALOMON : Reconsidering the distribution structure of gasoline products for a large oil company. *European Journal of Operational Research*, 81(3):460–473, mars 1995. ISSN 03772217. URL <http://linkinghub.elsevier.com/retrieve/pii/037722179400189J>.
- [101] T. VIDAL, T. G. CRAINIC, M. GENDREAU, N. LAHRICHI et W. REI : A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems. *Operations Research*, 60(3):611–624, juillet 2012. ISSN 0030-364X. URL <http://or.journal.informs.org/cgi/doi/10.1287/opre.1120.1048>.
- [102] Stefan VOSS, Ibrahim H OSMAN et Catherine ROUCAIROL, éditeurs. *Meta-Heuristics : Advances and Trends in Local Search Paradigms for Optimization*. Kluwer Academic Publishers, Norwell, MA, USA, 1999. ISBN 0792383699.
- [103] P WARK et J HOLT : A Repeated Matching Heuristic for the Vehicle Routing Problem. *The Journal of the Operational Research Society*, 45(10):1156–1167, 1994. ISSN 01605682. URL <http://www.jstor.org/stable/2584478>.
- [104] Bin YU et Zhong Zhen YANG : An ant colony optimization model : The period vehicle routing problem with time windows. *Transportation Research Part E : Logistics and Transportation Review*, 47(2):166–181, mars 2011. ISSN 13665545. URL <http://linkinghub.elsevier.com/retrieve/pii/S136655451000089X>.
- [105] Umit YUCEER : A multi-product loading problem : a model and solution method. *European Journal of Operational Research*, 101(3):519–531, 1997.
- [106] Qing ZHU, Limin QIAN, Yingchun LI et Shanjun ZHU : An Improved Particle Swarm Optimization Algorithm for Vehicle Routing Problem with Time Windows. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1386–1390, Vancouver-BC, 2006.

Cadet David JOSEPH

Doctorat : Optimisation et Sûreté des Systèmes

Année 2013

Optimisation des flux : application aux problèmes de distribution en nutrition animale

Cette thèse porte sur l'étude du problème de tournées de véhicules compartimentés (Multi-Compartment Vehicle Routing Problem ou MC-VRP) dans le contexte de l'industrie de la nutrition animale. Les travaux de recherche et d'application sont concentrés sur la distribution dans le domaine agroalimentaire.

En dépit d'une large application industrielle, les problèmes de MC-VRP ont été peu étudiés dans la littérature scientifique. Trois variantes du MC-VRP sont traitées dans cette thèse. D'une part, nous proposons les algorithmes "Greedy Randomized Adaptive Search Procedure" (GRASP) et "Iterated Local Search procedure" (ILS) pour résoudre le MC-VRP avec un compartiment de taille fixe dédié à chaque produit. Une extension de ce problème à des compartiments de tailles variables (Flexible Compartments Vehicle Routing Problem ou FC-VRP) est également résolue. D'autre part, nous proposons un GRASP et un Multi Start ILS pour la résolution du MC-VRP avec décisions d'affectation de chaque compartiment à un client et un produit.

En dernier lieu, des travaux d'application industrielle sont présentés. Des tests d'évaluation de performances ont été réalisés dans un contexte de distribution d'aliments au bétail chez la société Nestlé. Des outils d'aide à la décision ont été développés et mis en place dans cette société.

Mots clés : recherche opérationnelle - optimisation combinatoire - logistique (organisation) - transport - métaheuristiques - industries agroalimentaires - animaux, nutrition.

Flow Optimization: Application to Distribution Problems in the Context of Animal Nutrition Industry

This research concerns solving the Multi-Compartment Vehicle Routing Problem (MC-VRP) in the context of animal nutrition industry. Research and application work focuses on distribution in food industry.

Despite its vast application in industry, little attention has been paid to the MC-VRP. We address three classes of MC-VRP in this research. Firstly, we propose two metaheuristics, "Greedy Randomized Adaptive Search Procedure" (GRASP) and "Iterated Local Search procedure" (ILS), in order to solve a MC-VRP with a fixed-sized compartment dedicated to each product. Also, an extension of this problem to variable-sized compartments which we call Flexible Compartments Vehicle Routing Problem (FC-VRP) is studied. Further, we propose a GRASP and a Multi Start ILS to solve a MC-VRP problem with assignment decisions of each compartment to one client and one product.

Finally, some application work is presented. Experiments intended to measure performance in the context of food distribution to cattle were conducted for Nestlé company. Decision support tools had been developed and implemented for this company.

Keywords: operations research - combinatorial optimization - business logistics - transportation - food industry and trade - animal nutrition.

Thèse réalisée en partenariat entre :

