



HAL
open science

Optimisation de la distribution de marchandises avec sous-traitance du transport : une problématique chargeur

Gauthier Soleilhac

► **To cite this version:**

Gauthier Soleilhac. Optimisation de la distribution de marchandises avec sous-traitance du transport : une problématique chargeur. Recherche opérationnelle [math.OA]. Ecole nationale supérieure Mines-Télécom Atlantique, 2022. Français. NNT : 2022IMTA0303 . tel-03807647

HAL Id: tel-03807647

<https://theses.hal.science/tel-03807647>

Submitted on 10 Oct 2022

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'ÉCOLE NATIONALE SUPÉRIEURE
MINES-TÉLÉCOM ATLANTIQUE BRETAGNE
PAYS-DE-LA-LOIRE - IMT ATLANTIQUE

ÉCOLE DOCTORALE N° 601
*Mathématiques et Sciences et Technologies
de l'Information et de la Communication*
Spécialité : *Informatique*

Par

Gauthier SOLEILHAC

**Optimisation de la distribution de marchandises avec sous-
traitance du transport : une problématique chargeur**

Thèse présentée et soutenue à Nantes, le 12 juillet 2022

Unité de recherche : Laboratoire des Sciences du Numérique de Nantes (LS2N)

Thèse N° : 2022IMTA0303

Rapporteurs avant soutenance :

Michel GENDREAU Professeur, École Polytechnique de Montréal
Nicolas JOZEFOWIEZ Professeur, Université de Lorraine

Composition du Jury :

Président :	Éric PINSON	Professeur, Institut de Mathématiques Appliquées, Angers
Examineurs :	Diego CATTARUZZA	Maître de Conférences, Centrale Lille
	Michel GENDREAU	Professeur, École Polytechnique de Montréal
	Nicolas JOZEFOWIEZ	Professeur, Université de Lorraine
	Yannick KERGOSIEN	Maître de Conférences, École Polytechnique de l'Université de Tours
Dir. de thèse :	Olivier PÉTON	Professeur, IMT Atlantique
Co-dir. de thèse :	Fabien LEHUÉDÉ	Professeur, IMT Atlantique
Co-encadrante :	Juliette MEDINA	Responsable R&D, CRC Services

Invité :

Xavier PERRAUDIN Président Directeur Général, CRC Services



REMERCIEMENTS

Je tiens à remercier mes encadrants Olivier et Fabien, pour leurs précieux conseils et leurs encouragements.

Je remercie également Juliette et Xavier pour m'avoir accordé leur confiance et pour leur soutien au cours de ces trois années de thèse.

Je souhaite aussi remercier mes rapporteurs de thèse Michel GENDREAU et Nicolas JOSEFOWIEZ pour avoir accepté de juger mon travail et pour leurs retours sur mon manuscrit.

Merci à tous mes collègues à IMT Atlantique et à CRC Services pour tous les moments de détente ou d'entraide que nous avons partagé.

Je remercie enfin ma famille et Charline pour leur soutien inconditionnel et leurs encouragements.

TABLE DES MATIÈRES

Introduction	9
1 Problématique industrielle	13
1.1 CRC Services	13
1.2 Problématiques des clients de CRC Services	15
1.2.1 Le marché	15
1.2.2 Différents niveaux de décision	16
1.2.3 La sous-traitance du transport	16
1.3 Les outils proposés par CRC Services	20
1.3.1 L'organisation et la mutualisation du transport	20
1.3.2 CoLivRi	21
1.4 La Recherche et le Développement chez CRC Services	24
2 Contributions Scientifiques et Techniques	27
2.1 Le problème de tournées de véhicules avec transporteurs FTL et LTL	27
2.2 Le problème de collectes et livraisons avec plateformes intermédiaires	28
2.3 Applications chez CRC Services	29
3 Littérature	31
3.1 Sous-traitance du transport	32
3.1.1 Tarification Less Than Truckload	32
3.1.2 Tarification FTL	37
3.1.3 Flotte hétérogène	38
3.2 Sous-traitance des derniers kilomètres à un transporteur régional	39
3.2.1 Problèmes de tournées avec plateformes intermédiaires	40
3.2.2 Problèmes de tournées généralisés ou avec options	41
3.2.3 Synthèse	43
3.3 Caractéristiques liées aux commandes	44
3.3.1 Commodity Constrained Split Delivery Vehicle Routing Problem	44
3.3.2 Fenêtres de temps multiples	45

4	Problème de tournées de véhicules avec transporteurs FTL et LTL	49
4.1	Introduction	50
4.2	Définition du problème	52
4.3	Méthode de résolution	54
4.3.1	Description de l'algorithme LNS-SPP	56
4.3.2	Opérateurs du LNS	58
4.3.3	Gestion de la flotte limitée	60
4.4	Expérimentations numériques	65
4.4.1	Paramètres de l'algorithme	65
4.4.2	Résultats sur les instances de HFF-VRP-TW	66
4.4.3	Résultats sur les instances de HFF-VRP-TW-PC	69
4.4.4	Instances et résultats sur le VRP-FLC	70
4.5	Cas d'étude : combinaison d'expéditions FTL et LTL	73
4.5.1	Description	73
4.5.2	Expérimentations	74
4.6	Conclusions	76
5	Problème de collectes et livraisons avec plateformes intermédiaires	79
5.1	Modélisation du problème de collectes et livraisons avec plateformes intermédiaires	80
5.1.1	Coûts et réalisabilité des tournées	82
5.1.2	Modélisation mathématique du PDPTF avec options	83
5.2	Small and Large Neighborhood Search	85
5.2.1	Opérateurs du SLNS	87
5.2.2	Évaluation des options	90
5.2.3	Problème de partitionnement d'ensemble	91
5.3	Expérimentations numériques	92
5.3.1	Vehicle Routing Problem with Time Windows and Transshipment Facilities	93
5.3.2	Generalized Vehicle Routing Problem with Time Windows	95
5.3.3	Vehicle Routing Problem with Multiple Time Windows	97
5.3.4	Commodity-Constrained Split-Delivery Vehicle Routing Problem	98
5.3.5	Discussion	100
5.4	Conclusion	100

6 Applications industrielles	103
6.1 Introduction	103
6.2 Mise en œuvre des algorithmes développés au sein de CRC Services	105
6.2.1 Données d'entrée	105
6.2.2 Données de sortie	106
6.2.3 Intégration dans CoLivRi	106
6.2.4 Perspectives	108
6.3 Cas d'étude	109
6.3.1 Projet Auptilait	109
6.3.2 Optimisation de la distribution de flux mutualisés depuis un CRC vers des grandes surfaces spécialisées	111
6.3.3 Optimisation du transport de marchandises pour un acteur de l'ultra- frais	112
6.4 Conclusion	113
Conclusion	115
Bibliographie	119
A Résultats de la méthode Large Neighborhood Search	128
A.1 Résultats détaillés de la méthode LNS sur les instances VRPPC et VRPFLC	128
B Résultats de la méthode Small and Large Neighborhood Search	137
B.1 Résultats détaillés des instances VRPTWTF	137
B.2 Résultats détaillés des instances FSMTWTF	139
B.3 Résultats détaillés des instances CSDVRP	141
B.4 Résultats détaillés des instances VRPMTW	148

INTRODUCTION

Dans un contexte de crise sanitaire, écologique et énergétique, les chaînes logistiques sont soumises à de fortes perturbations. Elles doivent malgré ce contexte atteindre des objectifs de plus en plus élevés : réduire leurs émissions de gaz à effet de serre (GES) tout en maintenant des coûts acceptables pour les industriels et un niveau de service satisfaisant pour leurs clients.

Aujourd’hui, le fret représente 9% des émissions de gaz à effet de serre de la France (The Shift Project 2022). Dans le cadre du Plan de Transformation de l’Économie Française (PTEF), et pour tendre vers une neutralité carbone d’ici 2050, le Shift Project présente plusieurs axes de travail, impactant directement l’organisation du transport. Pour rendre le transport routier plus efficace, le PTEF propose différentes mesures afin d’améliorer le taux de remplissage dans les camions : l’organisation de la mutualisation, la réduction des cadences d’envoi et la généralisation de pratiques de maximisation des chargements.

Les travaux présentés dans ce manuscrit sont issus d’une thèse CIFRE, réalisée dans le cadre d’une collaboration entre la société CRC Services et IMT Atlantique au sein de l’équipe modelis du LS2N. Nous nous intéressons à des problématiques d’optimisation des transports de *chargeurs*. Ces acteurs sont des distributeurs ou industriels achetant du transport à différents prestataires logistiques afin d’acheminer des marchandises depuis différents sites de production ou de stockage vers différents points de vente ou plateformes logistiques.

Dans le monde du logiciel, l’optimisation des transports est notamment réalisée par les logiciels de tournées de véhicules. Ces logiciels cherchent à déterminer les tournées d’un ensemble de véhicules, de manière à servir un ensemble de clients, tout en respectant des contraintes telles que la capacité des véhicules, des fenêtres de temps pour la livraison et en minimisant des critères d’optimisation tels que le coût ou le nombre de kilomètres parcourus. Dans le domaine du transport de marchandises, le logiciel de tournées de véhicules est un outil classiquement dédié aux transporteurs. Le transporteur organise son réseau de transport et ses tournées de collecte et livraison de manière à réaliser les meilleures consolidations possibles de ses commandes et à minimiser le nombre total de kilomètres à parcourir pour satisfaire ses clients.

Un constat surprenant est que de nombreux transporteurs se passent encore de ce type d'outil en dépit du potentiel d'économie et de réduction des émissions de gaz à effet de serre (GES). Un argument souvent relevé est que les outils s'adaptent mal à certaines contraintes métiers. Sans entrer dans les raisons du choix des transporteurs, un potentiel d'optimisation encore peu exploité existe en amont du transporteur : en effet, de nombreux chargeurs s'intéressent de manière croissante au pilotage de leurs transports. Lorsqu'elles livrent les mêmes clients (typiquement les points de livraison de la grande distribution), ces entreprises cherchent parfois à d'organiser leur transport de manière conjointe dans le cadre de projets de *mutualisation* de manière à optimiser la performance de leurs opérations tant au niveau économique qu'au niveau environnemental.

Aujourd'hui, ces entreprises sont particulièrement sensibles à l'efficacité de leur transport. Pour autant, l'optimisation des transports du point de vue d'un producteur, fournisseur de la grande distribution par exemple, ne pose pas les mêmes problématiques, contraintes, fonctions de coût, que pour les transporteurs. Pour cette catégorie d'entreprises, il n'existe pas de logiciel adapté.

Dans cette thèse, nous appliquons des techniques issues de la Recherche Opérationnelle afin de résoudre plusieurs problèmes de tournées de véhicules et de répondre aux problématiques de ces chargeurs.

La résolution de ces problèmes doit permettre à CRC Services de mener des études pour ses clients, par exemple quantifier des gains liés à l'utilisation d'une nouvelle stratégie de transport. À terme, les algorithmes développés seront intégrés à la plateforme numérique « CoLivRi », afin d'être utilisés directement par les clients de CRC Services, pour un usage stratégique ou pour résoudre des problèmes opérationnels.

Plan de la thèse

Cette thèse est composée de trois parties : la première partie regroupe les Chapitres 1, 2 et 3 qui présentent le contexte des travaux ; la seconde partie est composée des Chapitres 4 et 5 qui présentent les principales contributions méthodologiques de la thèse ; enfin la dernière partie est constituée du Chapitre 6 qui présente un certain nombre d'applications dans lesquels les contributions méthodologiques ont été employées. Plus précisément :

Le Chapitre 1 présente la société CRC Services et les problématiques rencontrées par ses clients, des chargeurs, achetant des services de transport pour acheminer des marchandises dans des réseaux de la grande distribution. Nous décrivons l'organisation du transport et les différents types d'itinéraires possibles pour les commandes dans ce réseau,

et nous présentons les différents outils proposés par CRC Services pour accompagner ses clients et améliorer leur organisation de transport.

En s'appuyant sur les problématiques décrites au Chapitre 1, le Chapitre 2 présente les deux principaux problèmes d'optimisation étudiés au cours de cette thèse, ainsi que les contributions scientifiques de ces travaux.

Le Chapitre 3 propose une revue de la littérature sur des problèmes d'optimisation présentant les principales caractéristiques rencontrées dans les réseaux logistiques de la grande distribution.

Le Chapitre 4 aborde un problème de distribution de marchandises depuis un dépôt vers un ensemble de clients, dans lequel les chargements sont répartis entre plusieurs transporteurs. Ces transporteurs proposent chacun une flotte de véhicule limitée, et des tarifs variés. De multiples contraintes imposées par le réseau, les transporteurs ou les clients sont prises en compte. Pour résoudre ce problème nous proposons une heuristique combinant une Recherche à Voisinage Large (Large Neighborhood Search, LNS) et la résolution d'un problème de partitionnement d'ensemble.

Le Chapitre 5 étudie un problème de distribution dans un réseau comprenant plusieurs sites de production et de multiples plateformes intermédiaires. Les marchandises peuvent être livrées directement chez les clients ou sur l'une de ces plateformes. Dans ce cas, un transporteur se charge d'assurer la livraison depuis la plateforme jusqu'au client en échange d'une rémunération. Nous proposons une modélisation de ce problème comme un problème de collectes et livraisons avec *options*, où différentes options associées à chaque commande permettent de déterminer le site d'où la commande partira et son passage ou non par une plateforme intermédiaire. Nous proposons une heuristique de type Small and Large Neighborhood Search (SLNS) pour résoudre ce problème.

Le Chapitre 6 présente les applications des algorithmes développés au cours de cette thèse dans l'entreprise CRC Services. Nous y décrivons comment les outils de recherche opérationnelle conçus pendant cette thèse ont servi à supporter la prise de décision dans le cadre d'études en transport.

PROBLÉMATIQUE INDUSTRIELLE

1.1	CRC Services	13
1.2	Problématiques des clients de CRC Services	15
1.2.1	Le marché	15
1.2.2	Différents niveaux de décision	16
1.2.3	La sous-traitance du transport	16
1.3	Les outils proposés par CRC Services	20
1.3.1	L'organisation et la mutualisation du transport	20
1.3.2	CoLivRi	21
1.4	La Recherche et le Développement chez CRC Services	24

1.1 CRC Services

CRC Services a été créée en décembre 2015 sous la forme d'une filiale de la société 4S Network pour lui permettre de concentrer ses activités numériques de R&D notamment en optimisation du transport de marchandises. Celles-ci ont été initiées dès 2013 autour du concept de routeur de flux, le CRC « Centre de Routage Collaboratif », vecteur d'une mutualisation des flux suivant le concept de l'Internet Physique (Montreuil et al. 2013; Ballot et al. 2014).

Ce concept de mutualisation a pour objectif de fédérer des acteurs de toutes tailles autour d'un réseau de transport composés de hubs ou plateformes CRC. Ces plateformes constituent les noeuds de massification et de consolidation du réseau de transport. Contrairement aux hubs logistiques classiques des réseaux de transporteur, les CRC sont des plateformes utilisées par différents transporteurs spécialistes de leur région. Cette approche non propriétaire permet ainsi une collaboration horizontale entre transporteurs et apporte de la transparence et une garantie d'indépendance pour les utilisateurs du réseau.

Plus globalement, l'approche CRC se veut une offre écologique et « gagnant-gagnant » pour tous les acteurs du réseau : les *chargeurs*, qui achètent le transport afin que leurs

marchandises soient livrées, les *transporteurs* et autres prestataires logistiques, qui vendent et réalisent le transport et les *clients* qui reçoivent les commandes. L'approche CRC permet une coordination en amont des chargeurs qui partagent un même ensemble de clients.

Suite à la création du concept de CRC, identifiant un manque de maturité numérique sur le marché du transport de marchandises, CRC Services décide de développer en parallèle une offre numérique autour d'une plateforme web appelée « CoLivRi ». Le développement de cette plateforme a été réalisé dans le cadre du projet Open Network, financé par l'ADEME au sein du programme d'investissement d'avenir. Ce projet a été réalisé en partenariat avec l'équipe SLP (Systèmes Logistiques et de Production) du Laboratoire des Sciences du Numérique de Nantes (LS2N, UMR CNRS 6004), le laboratoire CSG des Mines ParisTech, GS1 France et TAB Rail Road, un prestataire logistique spécialisé dans le transport multi-modal. L'objectif du projet Open Network était de développer de nouvelles fonctionnalités pour faciliter la généralisation de la mutualisation du transport de marchandises et ainsi, proposer de nouveaux services à valeur ajoutée à des entreprises de toutes tailles en « digitalisant » les opérations de transport tout au long de la chaîne d'approvisionnement.

La solution proposée via la plateforme CoLivRi a pour but d'améliorer la performance d'une organisation transport sur 3 axes : économique, environnemental (émissions de gaz à effet de serre, GES), et qualité de service :

- **La performance économique** de la chaîne logistique est souvent le premier aspect considéré par un fournisseur qui cherchera à réduire la part de son budget de transport. Les économies réalisées sur le budget de transport d'un fournisseur peuvent se refléter sur le prix de vente de ses produits.
- **La performance environnementale** se caractérise par une réduction de l'émission de Gaz à Effet de Serre (GES). Un transport plus efficace (camions mieux remplis, moins de kilomètres parcourus) aura un effet bénéfique sur cet aspect mais également sur l'aspect économique. L'utilisation de modes de transport alternatifs (ferry, rail-route ou véhicules électriques) peut également avoir un effet positif pour réduire les émissions de GES.
- **Le taux de service** d'une chaîne logistique mesure la part de marchandises livrées dans les temps et sans anomalie chez le client. Un retard de livraison sur un magasin ou autre plateforme logistique peut entraîner un refus et la nécessité pour le transporteur de revenir livrer les marchandises à une autre date. Ces reports de

livraisons peuvent affecter la disponibilité des produits en magasins.

Ainsi, depuis sa création, CRC Services a développé deux types d'outils pour accompagner ses clients :

- Des outils organisationnels, notamment en les orientant vers des solutions de transport mutualisées et en travaillant autour du concept de Centre de Routage Collaboratif (CRC) (Medina 2016 ; Medina et al. 2019). Ce sont des moyens mis à disposition pour traiter une partie des flux mais qui visent à optimiser l'ensemble des flux. Ces outils sont mis en œuvre dans des activités de conseil fortement basées sur des simulations. Ces dernières s'appuient sur des modèles et algorithmes d'optimisation développés par la R&D de CRC Services.
- Des outils numériques autour de la traçabilité, de l'optimisation et de la mutualisation du transport de marchandises. Ces outils sont proposés par le biais d'une plateforme CoLivRi et peuvent être proposés indépendamment ou pour accompagner, piloter ou évaluer la pertinence d'une nouvelle organisation du transport. Cet accompagnement est souvent considéré comme la suite logique des missions de conseils réalisées au niveau organisationnel. Les algorithmes développés ou adaptés lors des missions de conseils sont alors intégrés et packagés dans CoLivRi. Les clients de CRC Services sont formés et peuvent ainsi simuler en toute indépendance de nouveaux scénarii en fonction de l'évolution de leur organisation transport.

1.2 Problématiques des clients de CRC Services

1.2.1 Le marché

Les clients de CRC Services sont pour la plupart des chargeurs, industriels, distributeurs ou grossistes, opérant dans l'univers des Produits de Grande Consommation (PGC), de la Restauration Hors Domicile ou Hors Foyer (RHD/RHF) ou de la Distribution de produits spécialisés (bricolage, sport ...). Ils acheminent leurs marchandises depuis un ou plusieurs sites de production ou de stockage, vers des sites de livraisons (magasins, entrepôts ou plateformes de passage à quai (Cross-Dock)). Pour cela, les chargeurs disposent rarement de leurs propres moyens de transport, et font donc appel aux services de *transporteurs* ou de commissionnaires de transport.

1.2.2 Différents niveaux de décision

Afin de définir une stratégie de transport performante, le chargeur doit prendre un ensemble de décisions que l'on peut définir selon plusieurs niveaux.

- Au niveau *stratégique*, le chargeur peut prendre des décisions sur la structure de son réseau logistique. Ces décisions incluent notamment le nombre et la localisation de ses plateformes de production, de stockage et d'expédition en fonction de la localisation de ses clients.
- A un niveau *tactique*, le chargeur doit sélectionner les schémas de transport en fonction de la typologie et du volume de ses flux. Il devra notamment sélectionner ses transporteurs et construire son plan de transport tactique. Différents schémas de transport possibles sont détaillés en sous-section 1.2.3.
- Le niveau *opérationnel* concerne la préparation d'un plan de transport réellement mis en place lors de l'expédition de ses marchandises au jour J. A ce niveau, les décisions sont souvent restreintes à l'application du process d'expédition.

1.2.3 La sous-traitance du transport

Pour expédier leurs marchandises, les industriels sous-traitent le transport et font appel à un ensemble de transporteurs ou prestataires logistiques spécialisés. Chaque transporteur dispose de sa propre flotte de véhicules, de contraintes spécifiques sur les tournées qu'il peut réaliser et de ses règles pour tarifier sa prestation de transport. Il peut également proposer l'utilisation de ses propres hubs de consolidation.

Cette sous-section définit dans un premier temps les différents tarifs proposés par les transporteurs puis les différents types d'itinéraires possibles et les contraintes que peuvent imposer les transporteurs.

1.2.3.1 Tarification du transport

Les tournées réalisées par les transporteurs peuvent être définies et tarifées de deux manières.

- En *Full-Truckload* (FTL), le chargeur réserve la capacité entière d'un véhicule pour lui faire effectuer un trajet donné. C'est le chargeur qui définit les sites visités et les horaires de passage à chaque point de la tournée. Chaque tournée a un coût qui a été négocié avec le transporteur.

Le coût d'une tournée FTL peut être basé sur un coût selon la distance parcourue, mais d'autres méthodes de calcul sont couramment utilisées. Par exemple, son coût peut être déterminé par le coût du point le plus éloigné du point de départ, auquel on rajoute un coût (souvent appelé coût d'ouverture de porte) pour chaque arrêt supplémentaire du véhicule dans la tournée.

La Figure 1.1 montre un exemple d'un tel tarif déterminé par des zones. Le coût est celui de la zone la plus éloignée traversée (425 €) auquel on ajoute 3 coûts d'ouverture de porte (30 €). Le coût total de la tournée est donc de 515€

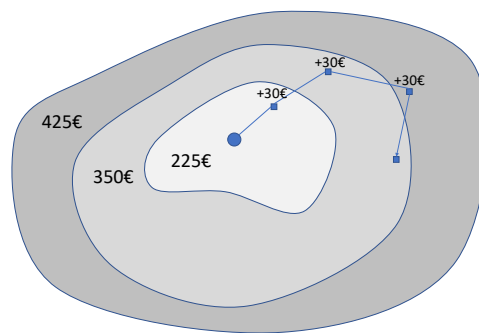


FIGURE 1.1 – Tarif FTL déterminé par des zones concentriques et des coûts d'ouverture de porte

- En *Less-Than-Truckload* (LTL), le transporteur propose un tarif pour chaque commande en fonction de son origine, de sa destination et de sa taille (en palettes ou en poids). Le coût est proportionnel à la distance et au volume transporté. Le plus souvent, le coût à la palette (ou au kg) décroît avec la quantité expédiée. L'organisation du transport est alors à la charge du transporteur, qui est libre de faire passer les marchandises dans son propre réseau afin de trouver une consolidation efficace pour livrer l'ensemble de ses flux.

La Figure 1.2 illustre un exemple de tarif unitaire qui décroît avec le poids de la commande.

1.2.3.2 Schémas de transport

Différents schémas de transport peuvent être utilisés ou combinés pour acheminer efficacement les marchandises du chargeur depuis sa ou ses plateformes d'expédition vers ses clients. Ces schémas sont illustrés en Figure 1.3.

- Le groupage (1.3a) est un mode d'expédition dans lequel les marchandises de plu-

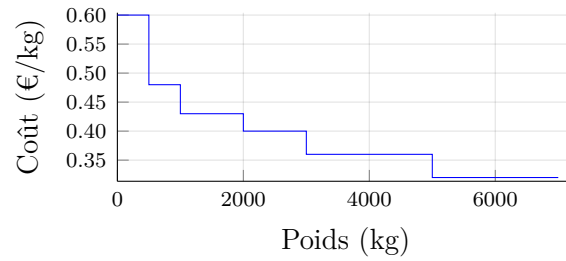


FIGURE 1.2 – Tarif LTL dont le coût unitaire décroît en fonction du poids expédié

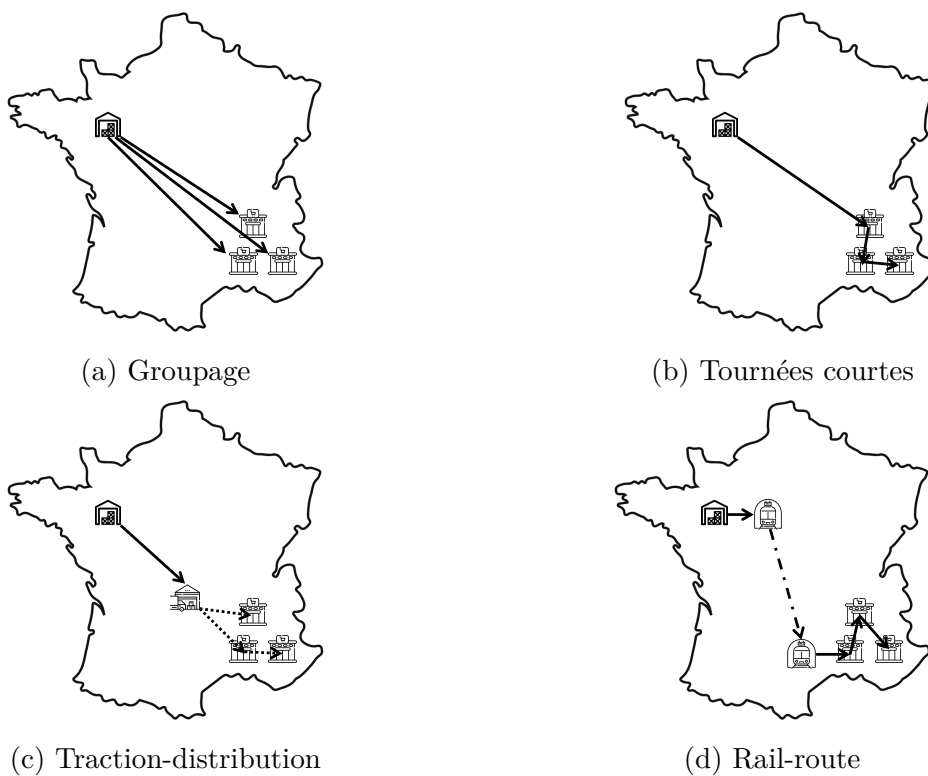


FIGURE 1.3 – Différents schémas de transport

sieurs chargeurs sont regroupées dans le même camion, à l'initiative du transporteur, sans que les différents chargeurs ne connaissent ni la composition, ni le trajet des véhicules, ni les autres chargeurs participant à la composition du véhicule. C'est le transporteur qui se charge de trouver d'autres flux pour remplir le véhicule ou de faire passer les marchandises par ses plateformes pour consolidation. C'est donc le transporteur qui dans ce cas bénéficie économiquement de la mutualisation, chaque chargeur payant son transport indépendamment de la quantité globale suivant un tarif Less-Than-Truckload négocié avec le transporteur.

- Les tournées (1.3b), associées au tarif Full-Truckload, sont souvent courtes dans les réseaux de la grande distribution : elles sont limitées à 3 voire 4 points de livraison afin d'éviter les risques de retard sur les plateformes clients. Ce mode de transport est privilégié lorsque le chargeur a des flux suffisants vers un groupe de clients et des heures de livraison chez ces clients qui permettent de les livrer au sein d'une même tournée.
- La traction-distribution (1.3c) consiste à acheminer des marchandises destinées à différents clients vers un hub ou CRC. Les clients sont ensuite livrés depuis ce hub. Des tarifs LTL ou FTL peuvent être considérés aussi bien pour la partie traction en amont du hub que pour la distribution aux clients en aval. Ce schéma de transport est également propice à la mutualisation des flux. Chaque industriel peut mieux remplir ses véhicules en amont des hubs et bénéficier également de conditions tarifaires plus avantageuses grâce à la mutualisation multi-chargeurs opérée en aval de hubs.
- Différents schémas de transport multi-modaux peuvent être utilisés comme le rail-route (1.3d) ou la combinaison de transport routier avec du transport maritime ou fluvial. Les camions sont alors utilisés pour les premiers et derniers kilomètres. Ce mode de transport est privilégié pour un chargeur voulant réduire son impact écologique mais peut également être moins cher qu'un transport entièrement routier. Il est en revanche plus contraignant pour l'organisation du transport du chargeur du fait du plus grand nombre de segments à considérer : pré-acheminement route jusqu'au terminal ferroviaire, segment rail entre le terminal ferroviaire de départ et celui d'arrivée, segment de post-acheminement route jusqu'au point de livraison final ou jusqu'au CRC.

Dans cette thèse nous nous intéressons principalement aux schémas de transport Groupage, Tournées, Traction-Distribution et à leurs combinaisons.

1.3 Les outils proposés par CRC Services

Cette section décrit les différents outils que CRC Services propose à ses clients pour améliorer l'efficacité de leur transport. Les outils organisationnels sont décrits en sous-Section 1.3.1 et les outils numériques, hébergés sur la plateforme CoLivRi, sont décrits en sous-Section 1.3.2.

1.3.1 L'organisation et la mutualisation du transport

CRC Services propose des outils différents et complémentaires pour aider ses clients dans l'organisation de leur transport.

1.3.1.1 Mutualisation des moyens de transport

Un chargeur peut réaliser des économies substantielles et grandement diminuer l'impact environnemental de sa logistique en mutualisant ses flux avec ceux d'autres chargeurs. C'est particulièrement le cas lorsque les quantités expédiées par un chargeur seul ne sont pas suffisantes pour obtenir un bon remplissage des camions. En regroupant leurs flux, les chargeurs peuvent obtenir des tarifs FTL plus avantageux. Cette mutualisation peut également être bénéfique pour les clients si elle concerne plusieurs de leurs fournisseurs, car une livraison groupée permet de réduire le nombre de prises de rendez-vous et le temps nécessaire pour la réception de la marchandise.

Toutefois, la collaboration horizontale entre plusieurs chargeurs implique de nombreux changements dans l'organisation et la gestion de leur réseau logistique. La réussite d'une telle collaboration est fortement liée à l'implication des chargeurs dans le projet et à l'intervention d'un partenaire tiers, aussi bien pour la mise en place du projet que pour son exploitation. CRC Services assure ce rôle de tiers de confiance.

1.3.1.2 Centres de Routage Collaboratifs

Les CRC sont des plateformes de passage à quai (cross-docking) opérées par des prestataires logistiques qui servent de hub de consolidation. Un chargeur peut choisir de systématiquement livrer certains de ses clients par un CRC, mais ce choix peut également être dynamique, en fonction des volumes, des contraintes horaires et des autres opportunités de consolidation. Si l'on s'intéresse à la distribution des commandes en aval du CRC, on pourra également étudier un plan de transport depuis celui-ci en le considérant

comme une plateforme d'expédition. La consolidation des tournées en aval se fera alors en fonction des fenêtres horaires pour livrer les commandes mais aussi de l'heure à laquelle elles arrivent sur le CRC.

1.3.1.3 Cadencement des passages de commande

Le cadencement des commandes clients dans un réseau de grande distribution consiste à définir des jours fixes sur lesquels les magasins passent commande ou sur lesquels ils sont livrés. Imposer à des magasins dans une région donnée de passer commande le même jour permet d'assurer un meilleur remplissage des camions et donc de bénéficier de meilleurs tarifs FTL pour le chargeur. Selon les typologies de flux, de produits et de destinataires, les cadences de livraison peuvent être très différentes : dans le secteur de l'ultra frais, certains magasins ou plateformes seront livrés 5 à 6 fois par semaine. Dans d'autres secteurs comme celui du bricolage, les cadences sont moins fortes et les chargeurs sont soumis à une livraison une fois ou deux par semaine, voire une fois tous les 15 jours. De plus, dans le cas d'une seule livraison par semaine, il ne serait pas possible de cadencer l'ensemble des magasins d'un réseau sur la même journée, car cela poserait des problèmes de capacité au niveau de la production ou de la main d'œuvre nécessaire à la préparation et au chargement pour le fournisseur, au transport et passage à quai pour les transporteurs et à la réception pour les magasins. Chaque type d'acteur peut avoir différentes préférences : un chargeur préférera livrer moins fréquemment afin de massifier ses volumes dans le moins de camions possible et ainsi diminuer ses coûts de transport. Un client, au contraire, préférera être livré plus fréquemment, afin de minimiser ses stocks et diminuer le risque de rupture de produits en rayon. L'enjeu est donc d'identifier le meilleur équilibre en respectant les contraintes de chacun. Par exemple, des clusters de magasins pouvant être livrés ensemble peuvent être déterminés et suivis d'une répartition du cadencement de ces clusters sur la semaine afin de lisser les expéditions pour le chargeur.

1.3.2 CoLivRi

Dans la plateforme CoLivRi, CRC Services axe son travail autour de trois problématiques :

- La traçabilité, la communication et l'interconnexion : la grande majorité des outils de communication entre les acteurs de la chaîne logistique transport sont encore aujourd'hui le mail, le téléphone et des tableurs (principalement Microsoft Excel).

Il en est de même avec la traçabilité des marchandises qui est souvent réalisée de façon semi-manuelle, les standards utilisés dans ce domaine datent d'une trentaine d'années, et les interconnexions entre systèmes d'informations sont difficiles au vu des technologies peu récentes utilisées dans ces outils. L'objectif de cet axe de recherche consiste donc à faciliter la communication entre tous les acteurs d'une chaîne transport, via la contextualisation et la sauvegarde des échanges. L'interconnexion doit être facile et facilitée, en proposant un outil capable de s'interconnecter à tous les standards, de sorte à intégrer et interpréter tout type de données de traçabilité des marchandises.

- L'interprétation de la donnée : l'ensemble des données collectées doit être interprété de sorte à pouvoir apprendre des données afin de prendre de meilleures décisions, et publier des tableaux de bord (GES / Service / Coût) pour faciliter le pilotage de la chaîne transport.
- L'optimisation des flux de transport : L'utilisation d'algorithmes d'optimisation ancrés dans les process transports doivent permettre une meilleure prise de décision à tous les niveaux : stratégique, lorsqu'un réseau transport doit être dimensionné, tactique, par exemple pour les prévisions de moyens de transport, et opérationnel, pour la prise de décision au jour le jour.

Ces problématiques sont déclinées dans CoLivRi sous la forme de modules accessibles ou non à l'utilisateur selon son abonnement à la plateforme. Chaque module est utilisable indépendamment, mais un usage multi-modules permet de travailler à tous les niveaux de décisions (modules d'optimisation stratégique et opérationnel) et d'assurer la bon fonctionnement de mise en place de la stratégie choisie grâce aux modules de traçabilité et de revue de performance.

1.3.2.1 Optimisation stratégique

Le module d'optimisation stratégique vise à évaluer différentes stratégies de transport. En fonction d'un jeu de données de flux (historique ou prévisionnel), et des contraintes transporteurs, contraintes clients ainsi que les tarifs réels, le chargeur peut rejouer ses données pour structurer sa stratégie transport et sélectionner les schémas de transports et transporteurs adaptés à ses flux.

1.3.2.2 Optimisation opérationnelle

En fonction des résultats de l'optimisation stratégique, le module d'optimisation opérationnelle construit les plans de transport opérationnels en fonction des commandes réellement passées ou des prévisions de ventes des jours à venir.

1.3.2.3 Traçabilité

Une fois les commandes expédiées, le module de traçabilité permet de suivre le transport de ses marchandises et est alerté en cas de problème. Ce module, ainsi que le module suivant d'analyse de la performance, est accessible à l'ensemble des acteurs de la chaîne transport au niveau de chaque commande : chargeur, transporteurs, et clients peuvent suivre et interagir ensemble sur CoLivRi en cas de problème. Pour chaque problème ou anomalie de transport, une responsabilité est associée automatiquement à l'acteur responsable.

1.3.2.4 Analyse de la performance (KPI)

Le module KPI (Key Performance Indicators) permet d'analyser les performances à partir de données issues des trois modules précédents sur les axes de performances économiques, GES et de services. Différents écrans permettent de projeter les données sur des graphes ou des cartes qui peuvent être filtrées suivant les besoins de l'utilisateur.

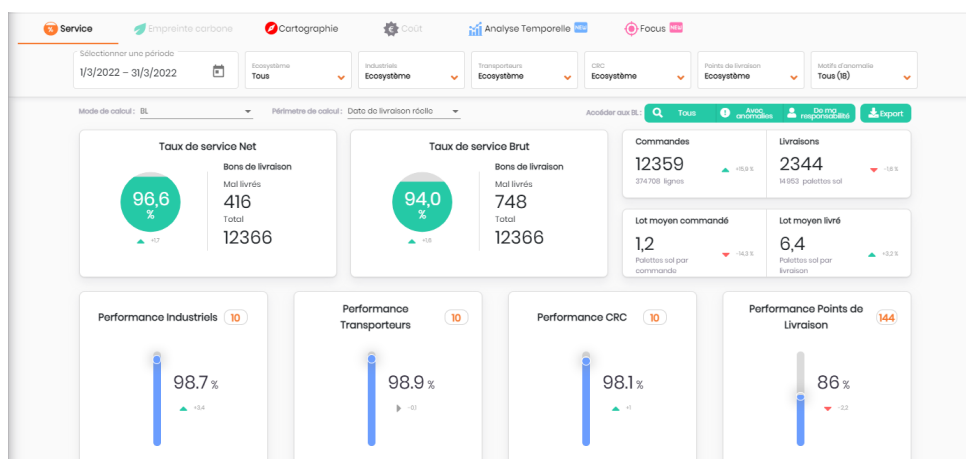


FIGURE 1.4 – Tableau de bord de suivi des KPIs de la plateforme CoLivRi

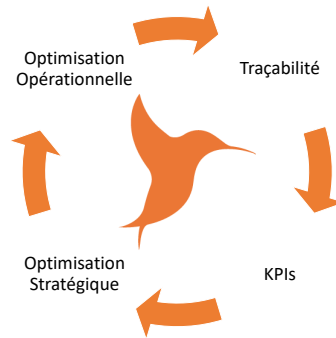


FIGURE 1.5 – Offre de CoLivRi axée sur 4 modules

1.4 La Recherche et le Développement chez CRC Services

La R&D est au coeur de la stratégie de CRC Services. En effet, CRC Services souhaite se positionner sur son marché en se démarquant de la concurrence grâce à une forte expertise en optimisation du transport de marchandises, plus particulièrement en développant des algorithmes issus des dernières avancées de la recherche académique. Pour ce faire, CRC Services investit depuis plusieurs années dans la recherche et collabore avec des partenaires académiques, notamment IMT Atlantique. Entre 2013 et 2016, 4S Network, dont CRC Services est la filiale, finance la thèse de Juliette Medina, qui rejoint ensuite l'équipe CRC Services pour développer les services numériques et la recherche et développement autour de l'optimisation du transport. En parallèle, le projet de R&D « CRC Open Network » est retenu à l'appel à projets « Logistique et Intermodalité » de l'ADEME. Ce projet permet de continuer la collaboration avec les partenaires académiques historiques de CRC Services, IMT Atlantique et Mines ParisTech. Deux post-doctorats sont financés pour continuer à mener des travaux de recherche autour de l'optimisation du transport de marchandises dans un contexte de mutualisation. En 2020, afin de pérenniser les travaux de recherche en cours, IMT Atlantique et CRC Services postulent à l'ANR pour le financement d'un laboratoire commun « LABCOM », qui est accordé début 2021. Le CRC Lab est ainsi créé fin 2021 pour travailler sur des problématiques de transport dans des contextes de données incertaines (volumes, durées de transport, etc). En parallèle, CRC Services s'intéresse également aux techniques d'apprentissage des données afin de

proposer des prédictions de retard à ses clients en fonction de son historique de données de traçabilité.

Dans ce cadre, nous nous intéressons dans cette thèse CIFRE à la résolution de problèmes de transport tactiques et opérationnels auxquels font face les chargeurs. L'intérêt des algorithmes développés au cours de cette thèse pour CRC Services est double. Ils doivent pouvoir être utilisés par CRC Services pour mener à bien diverses études stratégiques ou tactiques pour conseiller ses clients, et ils devront, à terme, venir enrichir l'offre d'optimisation sur la plateforme CoLivRi, afin d'être proposés à ses clients pour une utilisation tactique ou opérationnelle. Les algorithmes développés doivent donc être robustes aux différentes contraintes imposées par les différents acteurs du réseau, et prendre en compte différents schémas de transport présentés en Sous-Section 1.2.3 ainsi que les différents modes de tarification proposés par les transporteurs.

CONTRIBUTIONS SCIENTIFIQUES ET TECHNIQUES

Afin de répondre aux problématiques rencontrées par les chargeurs sous-traitant leur transport, nous définissons et proposons des méthodes de résolution pour deux problèmes de distribution rencontrés par des clients de la société CRC Services : le *Problème de tournées de véhicules avec transporteurs FTL et LTL* et le *Problème de collectes et de livraison avec plateformes intermédiaires*.

2.1 Le problème de tournées de véhicules avec transporteurs FTL et LTL

Nous considérons un problème de distribution auquel est confrontée une entreprise qui expédie des marchandises d'un entrepôt central vers de multiples clients. Plusieurs transporteurs proposent des tarifs et des capacités différents pour deux modes d'expédition : les marchandises peuvent être expédiées en LTL, avec des tarifs dépendant de la taille de l'expédition, ou en FTL, avec des flottes hétérogènes. Ce problème est formulé comme une variante d'un problème de tournées de véhicules avec flotte privée et transporteur commun. Nous intégrons des contraintes spécifiques pour traiter le fait que la « flotte privée » est composée de véhicules proposés par différents transporteurs proposant des tarifs FTL. En pratique, ces transporteurs peuvent s'engager sur un nombre limité de véhicules.

Les principales contributions de ce travail sont les suivantes :

- Nous introduisons le VRP-FLC (Vehicle Routing Problem with FTL and LTL Carriers) dans lequel les commandes sont servies en FTL avec une flotte hétérogène proposée par différents transporteurs ou en LTL pour un coût pré-déterminé. Dans ce problème, nous introduisons une *limite transporteur* qui définit, pour chaque

transporteur, le nombre maximal de véhicules mobilisables.

- Nous proposons une matheuristique qui combine la recherche à voisinage large (LNS) et la résolution d'un problème de partitionnement d'ensemble (Set Partitioning Problem, SPP). Les heuristiques utilisées pour la réparation d'une solution sont reprises ou inspirées des heuristiques de liste proposées par Christiaens et Vanden Berghe (2020). Elles permettent d'effectuer un grand nombre d'itérations et compensent l'absence de recherche locale au sein du LNS.
- Afin d'intégrer efficacement les contraintes de la flotte lors de la réparation d'une solution, nous adaptons un algorithme de filtrage issu de la programmation par contraintes (Régis 1996).
- Nous montrons que notre méthode est compétitive sur des instances de problèmes proches de la littérature et proposons des instances pour le VRP-FLC.
- Nous appliquons le VRP-FLC à une étude de cas réelle et montrons que des économies substantielles peuvent être générées en combinant des stratégies d'expédition en FTL et en LTL.

2.2 Le problème de collectes et livraisons avec plateformes intermédiaires

Nous considérons un problème de distribution dans un réseau comprenant plusieurs points d'expédition pouvant appartenir à un ou plusieurs industriels, des CRC ou autres plateformes logistiques, que l'on appellera plateformes intermédiaires et un ensemble de clients à servir. Les commandes peuvent être livrées directement aux clients depuis le site fournisseur, ou transiter par une plateforme intermédiaire. Un prestataire externe se charge alors de la distribution de la commande depuis la plateforme jusqu'au client en échange d'une rémunération. Nous proposons une modélisation de ce problème où chaque commande est associée à un ensemble d'options. Une option est définie par un lieu de collecte, un lieu de livraison, des fenêtres horaires associées à la collecte et à la livraison, et par un coût. L'objectif est de sélectionner une option pour chaque commande et de décider d'un ensemble de tournées minimisant les coûts de transport et les coûts de distribution associés aux options. Nous proposons également de modéliser l'expédition en LTL par des tournées, similaires aux tournées FTL, mais dont le coût est déterminé par la quantité expédiée. Cette représentation permet d'obtenir des coûts plus proches de la réalité lorsque plusieurs commandes sont expédiées en LTL vers une même destination.

Nos principales contributions sur cette problématique sont les suivantes :

- Nous formalisons la problématique du Problème de Collectes et Livraisons avec Plateformes Intermédiaires (Pickup and Delivery Problem with Transshipment Facilities, PDPTF).
- Nous modélisons ce problème comme un PDPTW avec options, qui représente une extension du generalized PDPTW où un coût peut être associé à chaque option. Ce formalisme permet de prendre en compte les différents sites de collecte et de livraison possibles pour une commande ainsi que les fenêtres horaires multiples associées à la collecte et à la livraison. Un coût sur les options modélise les coûts de distribution proposés par les transporteurs régionaux depuis les plateformes intermédiaires.
- Nous proposons une matheuristique dans laquelle une recherche Small and Large Neighborhood Search est combinée à la résolution d'un problème de partitionnement d'ensemble. Cette matheuristique diffère de la Recherche à Voisinage Large dans le fait qu'elle utilise deux types de voisinages. A chaque itération, la solution courante est partiellement détruite, soit par de petites destructions, soit par des destructions larges. Les petites destructions, combinées avec des heuristiques de réparation rapides, permettent d'effectuer un grand nombre d'itérations. Les destructions larges diversifient la recherche et permettent d'échapper aux minima locaux.
- Nous implémentons une approche adaptative de réparation des solutions reposant sur l'attribution d'un score aux options. Ce mécanisme permet d'accentuer la recherche sur les options les plus prometteuses et d'effectuer un plus grand nombre d'itérations en ne testant pas l'insertion de mauvaises options.
- Nous évaluons notre méthode sur plusieurs variantes de problèmes de la littérature présentant des caractéristiques similaires : problèmes avec options de livraison, avec fenêtres de temps multiples ou avec de multiples commodités. Nous montrons que notre méthode est compétitive vis-à-vis des publications les plus récentes sur ces benchmarks.

2.3 Applications chez CRC Services

Les algorithmes développés tout au long de cette thèse ont été utilisés pour supporter les activités de CRC Services. Le Chapitre 6 décrit l'intégration progressive des algo-

rithmes d'optimisation dans la plateforme CoLivRi et présente leur utilisation dans le cadre de plusieurs cas d'études réalisés au sein de l'entreprise CRC Services. Ces cas d'études sont des applications directes des travaux de recherche présentés dans ce manuscrit et nous ont guidé dans la définition des problèmes étudiés.

ÉTAT DE L'ART : TRANSPORT SOUS-TRAITÉ DANS UN RÉSEAU DE DISTRIBUTION

3.1	Sous-traitance du transport	32
3.1.1	Tarification Less Than Truckload	32
3.1.2	Tarification FTL	37
3.1.3	Flotte hétérogène	38
3.2	Sous-traitance des derniers kilomètres à un transporteur régional	39
3.2.1	Problèmes de tournées avec plateformes intermédiaires	40
3.2.2	Problèmes de tournées généralisés ou avec options	41
3.2.3	Synthèse	43
3.3	Caractéristiques liées aux commandes	44
3.3.1	Commodity Constrained Split Delivery Vehicle Routing Problem	44
3.3.2	Fenêtres de temps multiples	45

Ce chapitre d'état de l'art a pour but d'identifier les principales publications traitant de problèmes d'optimisation présentant des caractéristiques proches de celles rencontrées dans les réseaux logistiques de la grande distribution.

Les problématiques traitées dans cette thèse sont d'ordre tactique ou opérationnel et consistent à résoudre des problèmes de tournées de véhicules (Toth et Vigo 2014) intégrant diverses contraintes du réseau ou de ses acteurs.

Nous proposons une revue de la littérature en fonction de la provenance de ces contraintes.

La Section 3.1 traite des problèmes dans lesquels le chargeur ne dispose pas de sa propre flotte de véhicules, mais sous-traite les expéditions depuis son site à différents types de transporteurs. Dans la Section 3.2 nous intégrons la possibilité de faire transiter

les marchandises par des hubs logistiques. Cette caractéristique nous intéresse notamment dans le cas où on souhaite livrer un transporteur intermédiaire sur sa plateforme dans une région éloignée des points d’expédition. Ce transporteur peut alors concentrer des flux et mieux optimiser ses tournées pour la distribution de sa région. La Section 3.3 aborde des caractéristiques particulières concernant les commandes dans les problèmes étudiés.

3.1 Sous-traitance du transport

Dans cette section, nous intéressons à deux aspects liés à la sous-traitance par le chargeur de son transport à des prestataires logistiques : les transporteurs proposent différents tarifs LTL et FTL, et disposent de flottes hétérogènes et limitées.

3.1.1 Tarification Less Than Truckload

Dans la littérature en transport, le terme LTL désigne en général les transporteurs spécialisés dans le transport de commandes dont les quantités sont inférieures à la capacité d’un camion (Hall 1985). Il est essentiellement issu d’une littérature nord-américaine pour des réseaux de très grandes taille opérés par de gros transporteurs. Pour Bakir et al. (2021), un transporteur LTL consolide des expéditions à l’intérieur d’un réseau de hubs logistiques de manière à proposer un service efficace en coût. Les problèmes d’optimisation associés à ce type de transporteur sont souvent des problèmes de conception de réseau de service (Crainic et Hewitt 2021). Dans ces réseaux, un grand nombre de travaux ont donné lieu à l’intégration de coûts linéaires par morceaux dans les modèles d’optimisation (Frangioni et Gendron 2021). Dans le cadre de l’activité de CRC Services, la thèse de Juliette Medina (Medina 2016) s’est intéressée au transport LTL et à des problématiques de conception de réseaux d’un réseau mutualisé à l’échelle de la France. Dans le cadre de cette thèse on s’intéressera plus particulièrement à l’intégration d’expéditions via des transporteurs LTL au sein de problèmes de tournées de véhicules.

Dans le contexte de transport nationaux en France, les prestataires logistiques proposant un service de transport LTL fournissent à leurs clients une matrice de coûts. Cette matrice, valable pour un point de départ donné, indique le coût d’expédition en fonction de la distance à parcourir et de la quantité chargée (nombre de palettes ou poids). Il est donc possible de pré-calculer ce coût d’expédition LTL pour l’ensemble des commandes et d’imposer à chaque commande d’être livrée soit par une tournée avec un tarif FTL, soit

en LTL pour un coût donné. Dans le contexte d'un problème de tournées de véhicules, cette modélisation correspond à celle du Vehicle Routing Problem with Private fleet and Common carrier (VRPPC). Le VRPPC, formulé pour la première fois par Chu (2005) appartient à la famille des problèmes de tournées de véhicules avec profits (Archetti et al. 2014b). Il consiste à livrer un ensemble de commandes soit à l'aide d'une flotte privée, en payant un coût fixe pour les véhicules utilisés et des coûts variables à la distance ou à la durée, soit en déléguant la livraison à un transporteur externe pour un coût pré-déterminé pour chaque commande.

Dans le VRPPC, l'expéditeur peut faire appel au transporteur externe pour deux raisons. Soit à cause du dimensionnement de sa flotte privée, qui n'est pas suffisante pour livrer l'ensemble des commandes, soit pour des raisons économiques : déléguer certaines commandes au transporteur externe peut réduire le nombre de véhicules utilisés et permettre une meilleure utilisation de ses propres véhicules, en améliorant le taux de remplissage.

Heuristiques pour le VRPPC

De nombreuses méthodes approchées sont proposées pour résoudre le VRPPC.

Chu (2005), Bolduc et al. (2007) et Bolduc et al. (2008) proposent des heuristiques de construction suivies d'une recherche locale. Bolduc et al. (2008) montrent que le VRPPC peut être reformulé comme un problème de tournées de véhicules avec flotte hétérogène fixe (HFFVRP : Heterogeneous Fixed Fleet Vehicle Routing Problem) en remplaçant le transporteur externe par un véhicule factice. Ce véhicule utilise une matrice de coûts spécifique dans laquelle le coût de livraison d'un client par le transporteur externe est reporté sur les arcs entrants du client. Côté et Potvin (2009) proposent une recherche tabou pour résoudre le VRPPC utilisant trois voisinages : la réinsertion d'un client dans une autre tournée (*shift*), l'échange de deux clients entre deux tournées (*swap*) et un voisinage défini comme un sous-ensemble du 4-opt.

Potvin et Naud (2011) résolvent également le VRPPC avec une recherche tabou. Les voisinages *shift* et *swap* sont également utilisés, mais une recherche basée sur des chaînes d'éjection est opérée si les deux premiers voisinages ne permettent pas d'améliorer la solution courante et que le coût de cette solution est suffisamment proche du coût de la meilleure solution trouvée. Liu et al. (2010) étudient une variante d'un problème de tournées sur les arcs. Un transporteur reçoit un ensemble de requêtes qu'il doit livrer soit avec sa propre flotte, soit en la sous-traitant à un transporteur externe en LTL. Un

second ensemble de requêtes provenant d'autres transporteurs peut être pris en charge par le transporteur en échange d'une compensation.

Stenger et al. (2013) étudient un problème de tournées multi-dépôts dans lequel les clients peuvent être servis depuis un ensemble de dépôts privés ou via un ensemble de dépôts externes pour un coût dépendant de leur distance à ce dépôt. Les auteurs proposent d'utiliser un type de véhicule factice pour modéliser les clients servis par un dépôt externe. Une méthode dite *Adaptive Variable Neighborhood Search* (AVNS) est proposée et permet de résoudre des instances comprenant jusqu'à 360 clients et 9 dépôts.

Vidal et al. (2015) proposent une modélisation unifiant plusieurs variantes de VRP avec profits. Une représentation dite *exhaustive* des solutions est utilisée : les clients sous-traités gardent une position théorique prometteuse dans les tournées de la flotte privée. Une procédure de sélection permet de réintégrer certains de ces clients efficacement après la modification d'une solution (par exemple après une itération de recherche locale). Lorsqu'une tournée est modifiée, un problème de plus court chemin sous contraintes de ressources est résolu afin de déterminer optimalement les clients servis par le véhicule de la flotte privée et ceux servis par le transporteur externe. Des essais numériques sont proposés en utilisant cette représentation dans trois métaheuristiques : une recherche locale avec multi-start (MS-LS), une recherche locale itérée avec multi-start (MS-ILS) et un algorithme génétique UHGS (Vidal et al. 2014).

Gahm et al. (2017) étudient une variante du VRPPC posée selon le point de vue d'un chargeur. Le transport des marchandises peut être délégué à plusieurs transporteurs LTL. Ceux-ci offrent des réductions selon la quantité totale de marchandises qui leur est confiée. Il est également possible d'affréter des véhicules d'un transporteur externe pour effectuer des tournées. Ces véhicules sont disponibles via deux tarifications. Une flotte de véhicules est proposée par un transporteur dont les coûts dépendent seulement de la distance parcourue. Leur utilisation est sujette à une distance minimum. Un autre transporteur propose également une flotte de véhicules. Ces véhicules sont disponibles pour un coût fixe mais leur utilisation est limitée par une distance maximum. Le chargeur dispose également d'une flotte hétérogène privée. Un PLNE ainsi que trois variantes d'un Variable Neighborhood Search (VNS) sont proposés et testés sur les instances adaptées de celles de Bolduc et al. (2008). Euchi (2017) propose de résoudre le VRPPC avec une recherche tabou. La méthode utilise trois voisinages : 1-exchange et 2-exchange et un voisinage utilisant des chaînes d'éjections. Bulhões et al. (2018) introduisent le VRP with Service Level Constraints (VRPSL) dans lequel des contraintes de taux de service viennent imposer de

livrer une partie des clients avec la flotte privée. Une méthode de type branch-and-price et un algorithme génétique sont proposés pour résoudre le problème sur des instances allant jusqu'à 200 clients. On peut noter que le VRPPC est équivalent à un VRPSL avec une contrainte de taux de service nulle. Goeke et al. (2019) proposent une métaheuristique LNS pour résoudre le VRPPC reposant sur une décomposition du problème en sous-problèmes. Un sous-problème P_r est défini pour chaque tournée r dans la solution courante. Le sous-problème P_r ne considère que les clients de la tournée r , les clients d'un nombre aléatoire de tournées les plus proches de r et les clients proches servis par le transport externe. Les opérateurs de destruction et de réparation ne sont alors appliqués qu'à ce sous-problème et la réparation est acceptée si elle améliore la qualité de la solution. La méthode est couplée avec une recherche locale de type Variable Neighborhood Descent (VND).

Méthodes exactes pour le VRPPC

Plusieurs méthodes exactes pour le VRRPC ont également été proposées. Elles reposent toutes sur de la génération de colonnes et abordent différentes variantes du problème.

En plus de la métaheuristique citée dans la section précédente, Goeke et al. (2019) proposent une méthode de type branch-and-cut-and-price (BCP). Le problème maître consiste en un problème de partitionnement d'ensembles (SPP : Set Partitioning Problem) dont les variables sont les tournées utilisées. Le problème de «pricing» est un problème de plus court chemin élémentaire avec contraintes de ressources; il est résolu avec un algorithme de programmation dynamique.

Dabia et al. (2019) étudient une variante d'un VRPPC avec une flotte hétérogène limitée, des fenêtres horaires et un transporteur externe offrant des réductions. Ces réductions dépendent de la quantité totale de marchandises confiée au transporteur externe et prennent la forme d'une fonction linéaire par morceaux. Les auteurs proposent deux modèles mathématiques sous la forme d'un SPP sur les tournées. Deux formulations sont proposées, avec ou sans variables additionnelles pour indiquer quels clients sont servis par le transporteur externe. La remise sur le tarif LTL est modélisée via un Multiple Choice Model (Croxtton et al. 2003). Les deux modèles sont résolus avec une méthode de Branch-and-Cut-and-Price et évalués sur des instances adaptées de celles proposées par Liu et Shen (1999).

Baller et al. (2020) étudient le VRP with Partial Outsourcing (VRPPO). Dans cette

variante du VRPPC, la demande d'un client peut être en partie sous-traitée à un transporteur externe. Le client est alors livré en deux temps, une fois par la flotte privée et une fois par le transporteur externe. Les auteurs proposent deux modèles mathématiques formulés sur les tournées et une méthode de Branch-and-Cut-and-Price pour les résoudre. Les bénéfices liés à la sous-traitance partielle d'une partie des commandes sont évalués sur les instances de VRPPC de Dabia et al. (2019) et les instances de SDVRPTW de Desaulniers (2010). Les plus grands bénéfices sont obtenus sur des instances comprenant des commandes dépassant la capacité totale des véhicules qui, dans la variante VRPPC doivent être livrées entièrement par le transporteur externe.

Synthèse et discussion

Référence	Problème	Méthodes exactes	Méthodes heuristiques	TW
<i>Flotte homogène</i>				
Côté et Potvin 2009	VRPPC		TS	
Liu et al. 2010	ARP-PC		MA	
Vidal et al. 2015	VRPPC		UHGS	
Bulhões et al. 2018	VRP-SL		HGS	
<i>Flotte hétérogène</i>				
Chu 2005	VRPPC		Savings + LS	
Bolduc et al. 2007	VRPPC		SRI	
Bolduc et al. 2008	VRPPC		RIP	
Potvin et Naud 2011	VRPPC		TS	
Ceschia et al. 2011	VRPPC [1]		TS	
Gahm et al. 2017	VRPPCdR+ [2], [3]		RVND	
Euchi 2017	VRPPC		TS	
Dabia et al. 2019	VRPPC-TW+[2]	BPC		✓
Goeke et al. 2019	VRPPC	BPC	LNS+VND	
Baller et al. 2020	VRP-PO	BPC		✓

LS = Local Search, TS = Tabu Search, MA = Memetic Algorithm, HGS = Hybrid Genetic Search, SRI = Selection, Routing and Improvement, RIP = Randomized construction - Improvement - Perturbation, (U)HGS = (Unified) Hybrid Genetic Search, RVND = Randomized Variable Neighborhood Descent, BPC = Branch and Price and Cut, LNS = Large Neighborhood Search [1] Multiples transporteurs FTL, [2] Réductions selon le volume expédié en LTL, [3] Multiples transporteurs LTL

TABLEAU 3.1 – Publications concernant le Vehicle Routing Problem with Common Fleet and Private Carrier (VRPPC).

Le tableau 3.1 résume les principales références listées dans cette section. La variante considérée du problème est indiquée en colonne 2, les problèmes sont séparés selon le type

de flotte considéré : homogène ou hétérogène. Les colonnes 3 et 4 indiquent les méthodes exactes ou heuristiques proposées pour résoudre le problème. La colonne 5 indique si la présence de fenêtres de temps est prise en compte.

Le VRPPC est le plus souvent utilisé pour optimiser le plan de transport d'un transporteur, ou d'un chargeur disposant de ses propres moyens de transport et dans lequel une partie du transport seulement est sous-traitée en Less-Than-Truckload via le transporteur externe. Le cas d'une flotte hétérogène est souvent considéré mais les véhicules sont seulement hétérogènes en terme de capacité et de coût fixe. Seuls Ceschia et al. (2011) considèrent un cas où la "flotte privée" est en fait constituée de différentes flottes proposées par plusieurs transporteurs proposant des tarifs variés.

On remarque aussi que peu d'auteurs ont étudié les problèmes de VRPPC avec fenêtre de temps et que celles-ci sont le plus souvent prises en compte dans des méthodes exactes où elles aident à la résolution du problème. Pourtant, le recours à la sous-traitance peut être bénéfique pour livrer un client difficile d'accès en raison de ses contraintes horaires.

3.1.2 Tarification FTL

La majorité des publications sur les problèmes de tournées de véhicules s'intéressent à la planification du transport pour un transporteur, ou pour un chargeur qui dispose de sa propre flotte privée. Les coûts de ces flottes sont le plus souvent modélisés par un coût fixe qui modélise l'usage du véhicule, un coût horaire représentant le coût du conducteur et un coût au km qui modélise le prix de l'énergie (<https://www.cnr.fr/formule-trinome>). En revanche, peu de travaux étudient des problèmes de transport du point de vue du chargeur dans lequel l'intégralité du transport est acheté à des prestataires externes.

Dans la variante du VRPPC considérée par Gahm et al. (2017), certains des véhicules sont proposés par un transporteur externe pour des tarifs uniquement fixes (resp. variables) sous contrainte d'une distance parcourue maximale (resp. minimale). À notre connaissance, ce modèle de coût n'est pas pratiqué par les transporteurs français.

Lindsey et al. (2013) étudient un problème de collecte et de livraisons dans lequel le transport est intégralement sous-traité. Toutes les commandes peuvent être expédiées en LTL pour un tarif prédéfini, ou acheminées via deux tournées tarifées en FTL. Une première tournée collecte les commandes et les achemine jusqu'à une plateforme de crossdock pour consolidation ; une seconde tournée se charge de la distribution des commandes depuis la plateforme de crossdock. Le tarif des tournées FTL est déterminé par la distance parcourue multiplié par un tarif dit «truckload lane rate» qui dépend du premier et du

dernier point visité. À ce coût vient s'ajouter un coût d'ouverture de porte à chaque arrêt intermédiaire dans la tournée. Une tournée est réalisable s'il existe des tarifs pour un ensemble de sous-chemins empruntés ou entre le premier et dernier point de la tournée. Un modèle sur les tournées est proposé et permet de résoudre des instances de petite taille. Pour les instances de plus grande taille, les auteurs proposent une décomposition en fonction des types de tournées considérées et résolvent séquentiellement les différents sous-problèmes.

Ceschia et al. (2011) étudient une variante riche d'un problème de tournées de véhicules proche du VRPPC dans lequel la flotte hétérogène est composée de véhicules proposés par différents transporteurs. Afin de traiter les différents tarifs proposés par les transporteurs, une recherche tabou dans laquelle le coût des tournées peut être calculé via un programme externe est proposée. Les auteurs ont identifié et implémenté quatre catégories de coûts utilisés en pratique. Ces coûts sont définis par un coût fixe d'utilisation du véhicule auquel vient s'ajouter : (i) un coût selon la distance parcourue ; (ii) un coût en €/kg en fonction du point le plus éloigné de la tournée ; (iii) un coût selon la distance parcourue si le remplissage du camion est inférieur à un seuil ou un coût en €/kg en fonction du point le plus éloigné de la tournée sinon ; (iv) un coût en €/kg en fonction du point le plus éloigné de la tournée, qui décroît avec la charge du camion.

3.1.3 Flotte hétérogène

Optimiser les transports d'un chargeur en ayant recours à plusieurs transporteurs implique de considérer plusieurs types de véhicules. Dans la littérature transport, cette problématique est inhérente aux problèmes de tournées de véhicules avec flottes hétérogènes. On peut distinguer deux grandes catégories de problèmes de tournées avec flotte hétérogène : le Fleet-Size-and-Mix VRP (FSMVRP) et le Heterogeneous Fixed Fleet VRP (HFFVRP).

Le FSMVRP, introduit par Golden et al. (1984), considère différents types de véhicules, tous théoriquement disponibles en quantité illimitée. L'objectif du problème est de définir un ensemble de tournées et de leur affecter un type de véhicule afin de minimiser en premier lieu le nombre de véhicules utilisés, puis le coût d'utilisation des véhicules afin de servir les demandes des clients.

Dans le HFFVRP (Taillard 1999), le nombre de véhicules de chaque type est limité. Puisque la composition de la flotte est déjà déterminée, le nombre de véhicules utilisés est rarement considéré comme une variable à minimiser.

La présence d'une flotte hétérogène dans les problèmes de tournées est une caractéristique souvent présente et la littérature à ce sujet est abondante. On pourra se référer à Koç et al. (2016) pour un état de l'art sur les FSMVRP et les HFFVRP.

Le FSMVRP est souvent étudié dans une version prenant en compte des fenêtres de temps, le FSMVRPTW. En revanche, le HFFVRP avec fenêtres de temps (HFFVRPTW) est beaucoup moins étudié.

Paraskevopoulos et al. (2008) proposent une heuristique en deux phases intégrant un VNS et une recherche tabou. La méthode est testée sur les instances de Liu et Shen (1999), modifiées pour limiter le nombre de véhicules de chaque type. L'objectif considéré est la minimisation de la somme de la durée des tournées. Koç et al. (2015) résolvent le HFFVRPTW avec un algorithme génétique. Les solutions sont encodées par un «tour géant» et la méthode de SPLIT de Prins (2009) est enrichie pour également affecter un type de véhicules aux tournées générées. Les auteurs reprennent les instances de Paraskevopoulos et al. (2008) et considèrent deux objectifs : la minimisation de la distance totale et la minimisation de la somme des durées des tournées. Fachini et Armentano (2020) proposent deux méthodes exactes basées sur une décomposition de Benders pour résoudre le HFFVRPTW. L'objectif considéré est la distance totale.

3.2 Sous-traitance des derniers kilomètres à un transporteur régional

Cette section traite des caractéristiques inhérentes aux réseaux de grande distribution considérés, à savoir : la possibilité de regrouper les commandes vers des hubs logistiques afin de consolider les tournées, et la présence de multiples sites fournisseurs.

Dans la littérature en tournées de véhicules, les hubs logistiques sont principalement intégrées dans les problèmes de VRP avec cross-docking (Lee et al. 2006 ; Wen et al. 2009) ou de PDP avec cross-docking (Santos et al. 2013). Ces problèmes sont posés du point de vue du transporteur qui opère une flotte de véhicules et une plateforme de cross-docking. Dans notre revue de la littérature, nous nous intéressons plus particulièrement à l'utilisation de transporteurs régionaux, capables de prendre en charge un flux et réalisant lui-même ses tournées. Ce type de prestataire peut-être apparenté à un transporteur LTL devant être livré sur sa plateforme.

3.2.1 Problèmes de tournées avec plateformes intermédiaires

Baldacci et al. (2017) définissent le Vehicle Routing Problem with Transshipment Facilities (VRPTF) dans lequel les commandes peuvent être livrées directement depuis le dépôt, ou par le biais de plateformes intermédiaires appartenant à des prestataires logistiques externes. Puisque la distribution des marchandises depuis ces plateformes est externalisée, les tournées de distribution *aval* ne sont pas intégrées au problème. Pour livrer un client depuis une plateforme intermédiaire, un véhicule doit acheminer les marchandises jusqu'à la plateforme et le chargeur paye le prestataire logistique pour assurer la distribution depuis la plateforme. Le coût de distribution depuis une plateforme dépend de la distance séparant la plateforme du client. Baldacci et al. (2017) proposent deux modèles mathématiques, un premier sur les arcs et un second formulé comme un Set Partitioning Problem, ainsi qu'une méthode de type branch-and-cut-and-price pour résoudre des instances contenant jusqu'à 142 clients et 18 plateformes intermédiaires.

Alcaraz et al. (2019) étudient une variante riche d'un problème de tournées de véhicules comprenant plusieurs dépôts de véhicules et points de collecte, des fenêtres de temps, une flotte hétérogène et des contraintes de régulation horaire pour les conducteurs. Les marchandises peuvent être livrées directement via la flotte ou par un prestataire logistique externe depuis une plateforme intermédiaire d'une manière similaire à celle du VRPTF. Les auteurs proposent une heuristique de construction et un VNS.

Dans un contexte de logistique urbaine, Sitek et Wikarek (2019) étudient le Capacitated Vehicle Routing Problem with Pickup and Alternative Delivery (CVRPPAD). Un ensemble de marchandises doit être acheminé depuis un ensemble de points de collecte vers leurs destinations. Les marchandises peuvent également être livrées parmi un ensemble de consignes. La livraison d'une commande dans une consigne induit une pénalité dans la fonction objectif. Les auteurs proposent une approche consistant à d'abord pré-traiter le problème avec une programmation par contrainte puis à le résoudre avec une programmation mathématique. Une heuristique de construction est également proposée pour résoudre des instances de plus grand taille.

Friedrich et Elbert (2022) définissent le Vehicle Routing Problem with Time Windows and Transshipment Facilities (VRPTWTF), qui étend le VRPTF en considérant une flotte hétérogène et des fenêtres de temps pour la livraison des clients et des plateformes intermédiaires. Les coûts de distribution ne dépendent plus seulement de la distance entre le client et la plateforme mais aussi de la quantité de marchandises. Les auteurs proposent un Adaptive Large Neighborhood Search (ALNS) combiné à une recherche locale

et proposent de nouvelles instances adaptées des instances de Solomon (1987) incluant 100 clients et 5 plateformes intermédiaires.

3.2.2 Problèmes de tournées généralisés ou avec options

Si on ne considère pas le coût associé à la livraison d'une commande via une plateforme intermédiaire, le VRPTWTF se réduit au Problème de Tournées de Véhicules Généralisé avec Fenêtres Horaires (Generalized Vehicle Routing Problem with Time Windows, GVRPTW), introduit par Moccia et al. (2012). Dans le GVRPTW, chaque commande est associée à un cluster composé de différents sites de livraison possibles, et chacun de ces sites est associé à une fenêtre horaire. L'objectif est de servir exactement un site parmi chaque cluster tout en minimisant le nombre de véhicules utilisés et les coûts de livraison. Moccia et al. (2012) proposent de résoudre le GVRPTW avec une Recherche Tabou. Reyes et al. (2017) étudient le Vehicle Routing Problem with (Home and) Roaming Delivery Locations (VRP(H)RDL), un cas particulier du GVRPTW où les fenêtres horaires des options d'un même cluster ne se chevauchent pas. Le problème est appliqué dans un contexte de livraison urbaine où les commandes peuvent entre autres être livrées dans le coffre de voitures particulières. Dans le VRPHRDL, les clients peuvent également être livrés à leur domicile sans contrainte de fenêtre de temps. Les auteurs proposent une heuristique de construction et une recherche locale. Une méthode exacte de type branch-and-price est proposée par Ozbaygin et al. (2017) pour résoudre le VRPRDL et le VRPHRDL. Yuan et al. (2021) proposent une heuristique basée sur la génération de colonnes pour résoudre le GVRPTW. Le problème maître consiste à résoudre un Set Covering Problem (SCP) sur un ensemble de tournées. De nouvelles tournées sont obtenues par recherche locale sur les tournées ayant un coût réduit nul dans la solution. La méthode permet de trouver 10 nouvelles meilleures solutions parmi les 20 instances de GVRPTW de Moccia et al. (2012) et trouve la totalité des solutions optimales pour les instances de VRPRDL avec 40 clients. Zhou et al. (2018) introduisent le Multi-Depot Two-Echelon Vehicle Routing Problem with Delivery Options (MD-TEVRP-DO), un problème de tournées à deux échelons dans lequel les clients peuvent être livrés à domicile ou parmi un ensemble de consignes. Un premier échelon contient un ensemble de tournées partant des dépôts et livrant un ensemble de satellites. Le second échelon contient des tournées partant des différents satellites et livrant les clients à leur domicile ou parmi un ensemble de consignes. Chaque commande n'est disponible qu'à un unique dépôt et un coût est associé à la livraison d'une commande dans une consigne. Les auteurs proposent une méthode basée sur un algorithme

génétique et présentent des résultats sur une instance réelle contenant 164 clients et deux dépôts. Des instances sont également proposées pour le problème et comportant jusqu'à 12 dépôts et 200 clients. Mancini et Gansterer (2021) étudient le Vehicle Routing Problem with Private and Shared Delivery Locations. Dans ce problème, chaque commande doit être livrée chez le client dans une fenêtre horaire donnée, ou à tout moment de la journée parmi un ensemble de consignes que le client a préalablement sélectionné. La livraison d'une commande dans une consigne entraîne une compensation financière au client, elle est donc pénalisée par un même montant dans la fonction objectif. Les auteurs proposent deux mathheuristiques : un ILS et un LNS dans lequel les solutions sont reconstruites en résolvant un PLNE. Dumez et al. (2021a) et Tilk et al. (2021) introduisent le Vehicle Routing Problem with Delivery Options, dans lequel les clients peuvent être livrés chez eux ou parmi un ensemble de consignes à capacité limitée. Les clients associent un niveau de préférence à chaque lieu de livraison possible. La livraison d'une commande dans une consigne n'entraîne pas de coût supplémentaire, mais une contrainte de niveau de service force la satisfaction des préférences exprimées par les clients. Le VRPDO est résolu de manière exacte par Tilk et al. (2021) avec une méthode branch-and-price-and-cut et de manière heuristique par Dumez et al. (2021a) avec un LNS couplé à un SPP. Dumez et al. (2021b) proposent un LNS combinant petites et grandes destructions, et deux composantes exacts : la résolution d'un SPP et une recherche locale reposant sur un voisinage de Balas-Simonetti (Balas et Simonetti 2001). La méthode permet de trouver au total 81 nouvelles meilleures solutions sur un ensemble d'instances pour le VRPDO et différents problèmes proches : le VRPHDL, le GVRPTW et le VRPMTW. Dragomir et al. (2022) étudient un problème de collecte et de livraison généralisé (Savelsbergh et Sol 1995) appliqué à un contexte de vente en ligne de particulier à particulier. Le vendeur est supposé mobile au cours de la journée et fournit au transporteur plusieurs options définies par un lieu de collecte et une fenêtre horaire. L'acheteur fournit également plusieurs options en fonction de ses déplacements, mais il peut aussi préciser une personne alternative pour recevoir les marchandises. Enfin, il peut aussi être livré dans une consigne qui reste ouverte toute la journée. Les auteurs proposent un modèle mathématique ainsi qu'une méthode appelée Multi-Start Adaptive Large Neighborhood Search (MS-ALNS) combinée avec une recherche locale.

Référence	Problème étudié	Dernier kilomètre sous-traité	Fenêtres de temps	Flotte hétérogène	Collecte	Méthode
Baldacci et al. 2017	VRPTF	✓				BCP
Alcaraz et al. 2019	Rich VRP	✓	✓	✓		VNS, TS, SA
Sitek et Wikarek 2019	CVRPPAD	✓	✓	✓	✓	CLP
Friedrich et Elbert 2022	VRPTWTF	✓	✓	✓		ALNS
Moccia et al. 2012	GVRPTW		✓			TS
Reyes et al. 2017	VRPRDL		✓			GRASP + LS
Ozbaygin et al. 2017	VRPRDL		✓			BP
Zhou et al. 2018	MD-TEVRP-DO	1				HEA x
Mancini et Gansterer 2021	VRPPSDL	1	✓			LNS, ILS
Dumez et al. 2021a	VRPDO		✓			ALNS
Dragomir et al. 2022	PDPAL		✓		Multiples	MS-ALNS

¹ Pénalités sur la livraison de commandes en consigne

TABLEAU 3.2 – Publications concernant des problèmes de tournées avec options

3.2.3 Synthèse

Le Tableau 3.2 synthétise les différentes publications concernant des problèmes de tournées avec de multiples options de livraison. Ce formalisme est utilisé dans deux situations :

- Dans des problèmes de distribution où un prestataire externe propose de prendre en charge la livraison finale d’un client depuis une plateforme. Le coût de cette distribution est alors répercuté sur le coût de l’option consistant à livrer la commande sur la plateforme de ce prestataire.
- Dans des problèmes de logistique urbaine dans lesquels les commandes peuvent être livrés dans des consignes où à différents endroits au cours de la journée en fonction de la mobilité des clients.

Dans l’état de l’art, la majorité des publications utilisant le formalisme des options de multiples points de livraison considèrent un unique dépôt ou point de collecte pour chaque commande. Les travaux connexes de Dragomir et al. (2022) utilisent des options pour modéliser un point de collecte mobile au cours de la journée en plus des différentes possibilités de livraison. Le formalisme des options permet de modéliser de multiples points de collecte possibles pour une même commande en fonction des produits demandés. Cette idée sera développée au Chapitre 5.

3.3 Caractéristiques liées aux commandes

Dans les problèmes de transport avec sous-traitance décrits au Chapitre 1, un certain nombre de caractéristiques sont directement liées aux commandes. Ces objectifs ou contraintes peuvent être liés à la nature des commandes, aux clients ou aux points de livraison. Nous avons en effet mentionné les caractéristiques suivantes :

- Un magasin peut attendre dans la même journée la réception de plusieurs commandes correspondant à différents produits ou différents types de produits, par exemple en fonction du rayon : produits laitiers, fruits et légumes etc... La livraison de l'ensemble des produits attendus par le magasin peut se faire en plusieurs fois, mais l'ensemble des produits d'un même rayon doivent être livrés en une fois. On peut retrouver cette caractéristique si on considère un problème d'expédition depuis un CRC, ou la présence de différents fournisseurs proposant plusieurs types de produits.
- Certains sites, notamment des magasins, ne peuvent décharger qu'un nombre limité de camions en parallèle, que ce soit pour des raisons de place ou de main d'œuvre. Ces sites ont donc tendance à proposer un ou plusieurs créneaux horaires possibles à leurs fournisseurs afin d'éviter de coïncider avec d'autres livraisons déjà planifiées.

Dans la littérature, ces caractéristiques sont étudiées respectivement dans le Commodity constrained Split Delivery Vehicle Routing Problem (C-SDVRP) et le problème de tournées de véhicules avec fenêtres de temps multiples (VRPMTW : Vehicle Routing Problem with Multiple Time Windows).

3.3.1 Commodity Constrained Split Delivery Vehicle Routing Problem

Le C-SDVRP est un problème de tournées de véhicules dans lequel différents produits ou types de produits, désignés par le terme *commodité*, doivent être livrés à un ensemble de clients. Un client peut être livré plusieurs fois, selon le nombre de commodités qu'il attend, mais la livraison de chaque commodité doit se faire en une seule fois.

Nakao et Nagamochi (2007) introduisent ce problème sous le nom de Discrete Split Delivery VRP et proposent une heuristique de construction basée sur une programmation dynamique.

Ceselli et al. (2009) étudient un problème riche de tournées de véhicules avec de mul-

tiples dépôts dans lequel les commandes sont constituée d'un ensemble de palettes, chaque palette comprenant plusieurs produits. Chaque palette peut provenir de différents dépôts selon les produits demandés et certains types de produits sont incompatibles : ils ne peuvent pas être livrés par le même véhicule. Un client peut être livré en plusieurs fois, mais une palette doit être livrée en une seule fois. Les auteurs proposent une méthode par génération de colonnes en trois phases. Une première phase consiste à générer des tournées livrant entièrement la demande de chaque client. Une seconde phase considère la livraison en plusieurs des commandes les plus volumineuses. Une troisième phase considère la livraison de chaque commodité.

Archetti et al. (2014a) étudient différentes variantes de VRP avec de multiples commodités et nomment le C-SDVRP. Le C-SDVRP est résolu avec une recherche tabou en dupliquant les sommets des clients pour chaque commodité demandée. Des instances adaptées des instances VRPTW de Solomon (1987) sont proposées dans lesquelles les clients demandent jusqu'à trois types de produits différents.

Archetti et al. (2015) proposent une méthode branch-and-price-and-cut pour le C-SDVRP et résolvent des instances contenant jusqu'à 40 clients et 3 commodités par client.

Gu et al. (2019) proposent de résoudre le C-SDVRP avec un ALNS combiné à plusieurs opérateurs de recherche locale agissant sur les clients ou sur les commodités. À chaque fois qu'une nouvelle meilleure solution est trouvée, une recherche locale basée sur la résolution de différents programmes linéaires est opérée afin de trouver une meilleure affectation des commodités aux tournées. Un programme linéaire est résolu pour chaque client et correspond à la résolution d'un problème de localisation (FLP : Facility Location Problem).

3.3.2 Fenêtres de temps multiples

Le problème de tournées de véhicules avec fenêtres de temps multiples, introduit par Favaretto et al. (2007), est une extension du VRPTW dans laquelle les clients proposent de multiples créneaux horaires dans lesquels ils peuvent être livrés.

Favaretto et al. (2007) étudient un un VRPMTW périodique. Les clients doivent être livrés plusieurs fois au cours d'un horizon de temps donné et des contraintes imposent d'espacer les différentes visites pour un même client. L'objectif est de minimiser le nombre de véhicules utilisés, puis la durée totale des tournées. Le problème est résolu avec une métaheuristique de type colonie de fourmis (ACO : Ant Colony Optimization). Des instances de VRPMTW sans contraintes de périodicité sont proposées pour évaluer la performance

de leur méthode. Tricoire et al. (2010) étudient le Multi-period Orienteering Problem with Multiple Time Windows (MuPOPTW) dans lequel un ensemble de clients peut être livré une fois au cours d'un horizon de temps découpé en plusieurs périodes. Sur chaque période, les clients peuvent proposer une à deux fenêtres de temps, ou zéro s'ils ne peuvent pas être livrés au cours de cette période. L'objectif du problème est de déterminer une tournée pour chaque période afin de maximiser un ensemble de profits associés à chaque client visité. Les auteurs proposent de résoudre le problème avec un VNS. Une procédure exacte pour déterminer les fenêtres horaires utilisées pour une tournée est proposée. Belhaiza et al. (2014) proposent une méthode dite Hybrid Variable Neighborhood Tabu Search (HVNTS) pour résoudre le VRPMTW. La durée des tournées est minimisée en adaptant leur heure de départ afin de réduire les temps d'attente aux différents points de livraison. La méthode est évaluée sur les instances de Favaretto et al. (2007) et sur de nouvelles instances adaptées de celles de Solomon (1987). Deux objectifs sont considérés : la minimisation de la distance totale parcourue et la minimisation de la somme des durées des tournées. Une fonction de coût incluant des pénalités est utilisée afin de considérer des solutions infaisables au cours de la recherche ; ces pénalités incluent le non respect des fenêtres horaires, de la capacité des véhicules et de la durée maximale des tournées. Belhaiza et al. (2017) améliorent de 0.54% (resp. 0.37%) les résultats obtenus sur ces nouvelles instances pour la minimisation de la distance (resp. durée) en proposant de nouveaux opérateurs de recherche locale. Beheshti et al. (2015) étudient une variante bi-objectif du VRPMTW dans laquelle les clients associent une préférence à chaque fenêtre horaire. Les objectifs considérés sont la réduction des coûts de distribution et la maximisation des préférences exprimées par les clients. Belhaiza et M'Hallah (2016) adaptent leur méthode HVNTS à une variante du VRPMTW à trois objectifs : la minimisation du coût des tournées, la maximisation de fonctions d'utilité pour les transporteurs et pour les clients. Belhaiza et al. (2019) proposent un algorithme génétique pour résoudre le VRPMTW utilisant des opérateurs de recherche locale visant à améliorer la faisabilité des tournées vis-à-vis des fenêtres horaires et une recherche suivant un schéma ruin-and-recreate (Schrimpf et al. 2000).

Larsen et Pacino (2019) proposent une heuristique de type (ALNS) pour résoudre le VRPMTW. Une extension des Forward Time Slacks (Savelsbergh 1992) permet de vérifier la réalisabilité d'une insertion en temps constant. Les expérimentations numériques sur les instances de Belhaiza et al. (2014) montrent que la méthode proposée est très efficace et permet de trouver 45 nouvelles meilleures solutions sur les 48 instances lorsque l'objectif

considéré est la minimisation de la distance et 47 nouvelles meilleures solutions lorsque la durée totale des tournées est minimisée. Hoogeboom et al. (2020) proposent également une généralisation des Forward Time Slacks aux fenêtres multiples. La méthode d'insertion proposée est mise en œuvre dans une recherche à voisinage variable et améliore les résultats de Belhaiza et al. (2017) sur 22 des 48 instances proposées. Dumez et al. (2021b) proposent de résoudre le VRPMTW comme un cas particulier du VRPDO ou du GVRPTW, dans laquelle chaque fenêtre horaire représente une option de livraison. La matheuristique LNS-SPP développée pour le VRPDO est appliquée sur les instances VRPMTW de Belhaiza et al. (2014) et améliore les résultats de Larsen et Pacino (2019) en trouvant 23 nouvelles meilleures solutions.

PROBLÈME DE TOURNÉES DE VÉHICULES AVEC TRANSPORTEURS FTL ET LTL

4.1	Introduction	50
4.2	Définition du problème	52
4.3	Méthode de résolution	54
4.3.1	Description de l'algorithme LNS-SPP	56
4.3.2	Opérateurs du LNS	58
4.3.3	Gestion de la flotte limitée	60
4.4	Expérimentations numériques	65
4.4.1	Paramètres de l'algorithme	65
4.4.2	Résultats sur les instances de HFF-VRP-TW	66
4.4.3	Résultats sur les instances de HFF-VRP-TW-PC	69
4.4.4	Instances et résultats sur le VRP-FLC	70
4.5	Cas d'étude : combinaison d'expéditions FTL et LTL	73
4.5.1	Description	73
4.5.2	Expérimentations	74
4.6	Conclusions	76

Dans ce chapitre, nous étudions un problème de tournées de véhicules visant à optimiser le transport de marchandises depuis un unique site de départ vers un ensemble de clients. Les marchandises peuvent être expédiées en LTL, ou en FTL par différents transporteurs proposant des flottes hétérogènes de véhicules.

4.1 Introduction

Nous nous intéressons à la politique de distribution d'un chargeur depuis un unique site vers un ensemble de sites clients. Le site d'où partent les marchandises peut être un site de production du chargeur ou un hub mutualisé sur lequel plusieurs fournisseurs viennent déposer les marchandises. Par la suite, nous appelons ce site le dépôt. Le chargeur livre fréquemment un ensemble de clients, répartis sur des centaines de kilomètres autour du dépôt. Les volumes commandés par chaque client sont variables d'un jour à l'autre. Le chargeur ne dispose pas de sa propre flotte de véhicules pour la distribution des produits. Ainsi, chaque jour d'expédition, les commandes sont réparties entre plusieurs transporteurs, chacun ayant ses propres tarifs de transport, sa propre flotte de véhicules et ses propres contraintes de gestion.

Dans ce contexte nous cherchons à combiner les modes de transport FTL et LTL présentés au chapitre 1 sur la base des modes de fonctionnement suivant :

En *Full-Truckload* (FTL), le chargeur définit une tournée et l'horaire de cette tournée tout en respectant un ensemble de contraintes opérationnelles qui peuvent différer d'un transporteur à l'autre. Il réserve entièrement un véhicule pour lui faire effectuer cette tournée.

Les tournées de type FTL partent du dépôt et livrent une ou plusieurs commandes à un ou plusieurs clients en respectant un certain nombre de contraintes. Certaines contraintes, comme les fenêtres horaires de livraison, sont liées aux clients. D'autres contraintes classiques concernent la capacité du véhicule ou la charge maximale, ainsi que la compatibilité entre les véhicules et leur chargement (par exemple, transport réfrigéré, conteneurs spécifiques, transport de matières dangereuses, etc.) Les transporteurs peuvent spécifier un nombre maximal d'arrêts dans une tournée, un détour maximal sous forme de limite absolue, ou en pourcentage de la longueur initiale de la tournée (Lindsey et al. 2013). Ensuite, la législation impose des contraintes liées à la durée maximale des tournées.

Les tarifs FTL sont préalablement établis dans un contrat de collaboration entre un transporteur et le chargeur. Le tarif FTL proposé par un transporteur peut être basé sur un coût au kilomètre, mais plusieurs autres méthodes de calcul sont fréquemment utilisées. Par exemple, le coût peut être déterminé par la zone la plus éloignée livrée par la tournée, plus un coût dit d'*ouverture de porte* pour chaque arrêt intermédiaire.

Chaque transporteur a ses propres règles de calcul qui dépendent de nombreux facteurs internes (coûts d'entretien du véhicule, coût des chauffeurs, carburant, péages,

etc.) ainsi que de facteurs de compétitivité (probabilité de trouver une cargaison pour le voyage de retour, présence de transporteurs concurrents, etc.). Le coût d'une tournée donnée dépend également de la liste des sites visités et du type de véhicule utilisé.

En *Less-Than-Truckload* (LTL) le transporteur propose un coût pour chaque commande en fonction de la distance entre le site de production et sa destination et de la quantité (poids, volume) à livrer. Dans ce cadre, le transporteur organise l'optimisation des transports dans son propre réseau. Par conséquent, toutes les contraintes internes au réseau logistique, telles que les fenêtres de temps, sont gérées par le transporteur. Le tarif LTL est particulièrement avantageux pour les envois de volumes faibles. En contrepartie, le coût à l'unité logistique est plus élevé que lors de la réservation d'un camion complet en FTL.

La relation entre les chargeurs et les transporteurs impose un ensemble de contraintes commerciales, liées à la gestion de la flotte. L'un des principaux défis pour les chargeurs est de trouver des véhicules disponibles sur des marchés de transport de plus en plus tendus. Pour les transporteurs, il s'agit d'assurer qu'un volume suffisant leur soit confié d'une manière régulière afin de prévoir un nombre suffisant de véhicules. Ainsi, les deux parties peuvent s'accorder sur un nombre de véhicules que le transporteur va pouvoir mettre à disposition chaque jour. Pour chaque transporteur, la *limite transporteur* définit le nombre maximal de véhicules pouvant être utilisé par le chargeur au cours d'une journée. Cette limite transporteur garantit au chargeur qu'un certain nombre de véhicules lui sont réservés. Elle aide également le transporteur à contrôler le nombre de véhicules alloués à ses différents clients. La littérature considère généralement l'utilisation de ressources sous-traitées sans contraintes (capacité infinie, aucun engagement de volume) et d'une seule flotte de véhicules homogènes ou hétérogènes (voir par exemple Chu (2005), Bolduc et al. (2008) et Dabia et al. (2019)). Dans ce chapitre, nous considérons plusieurs flottes hétérogènes proposées par différents transporteurs, chacun proposant un nombre limité de véhicules. Les véhicules de ces flottes sont regroupés en types de véhicule identiques, où chaque type de véhicule est disponible en une quantité limitée, et possède ses propres contraintes et son propre modèle de coûts.

Les principales contributions de ce chapitre sont les suivantes :

- Nous introduisons un nouveau problème d'optimisation : le problème de tournées de véhicules avec transporteurs FTL et LTL (*VRP-FLC* : Vehicle Routing Problem with FTL and LTL Carriers). Ce problème se caractérise par la possibilité de livrer les commandes soit en FTL soit en LTL, par le choix entre plusieurs transporteurs

FTL, avec une flotte de véhicules hétérogène avec des contraintes sur la disponibilité des véhicules.

- Nous proposons une matheuristique combinant une Recherche à Voisinage Large et la résolution d’un Problème de partitionnement (Set Partitioning Problem, SPP) sur l’ensemble des tournées générées par le LNS.
- Nous intégrons des modèles de coûts de tournées variés et des contraintes réalistes via une fonction «boîte noire» combinant des informations incrémentales et statiques pour évaluer efficacement la faisabilité et le coût des insertions.
- Nous introduisons des stratégies de filtrage pour intégrer efficacement des contraintes spécifiques sur la flotte de véhicules dans les opérateurs LNS.
- Nous évaluons notre approche sur des instances de problèmes similaires de la littérature scientifique et montrons que notre méthode est compétitive par rapport à l’état de l’art.
- Nous introduisons un nouvel ensemble d’instances pour le VRP-FLC intégrant de multiples transporteurs.
- Nous explorons comment la combinaison des expéditions FTL et LTL peut permettre de réaliser des économies substantielles par rapport aux approches 100% FTL ou LTL.

Ce chapitre est structuré comme suit : La section 4.2 présente le problème et le modèle mathématique correspondant. La section 4.3 décrit la méthode de résolution proposée. Dans la section 4.4, l’algorithme proposé est évalué sur des instances de référence pour des problèmes d’optimisation très similaires. Ensuite, nous présentons des résultats numériques sur de nouvelles instances. Enfin, la section 4.6 fournit des remarques finales et suggère des pistes de recherche futures.

4.2 Définition du problème

Le VRP-FLC considère un expéditeur unique et un ensemble de clients à livrer. Soit $\mathcal{N} = \{1, \dots, |\mathcal{N}|\}$ l’ensemble des commandes passées par ces clients. Une commande est un ensemble de produits qui doivent être livrés en une seule fois. Elle est caractérisée par l’adresse du lieu de livraison, une taille (nombre de palettes, et/ou poids), une date de mise à disponibilité et un ensemble de fenêtres horaires dans lesquelles le client peut être livré. Les commandes partent toutes d’un même lieu qu’on appelle le *dépôt*. Chaque commande peut être expédiée soit en mode FTL, soit en mode LTL. En mode LTL, le

coût d'expédition c_n^{LTL} est donné *a priori* pour chaque commande $n \in \mathcal{N}$. Lorsqu'elles sont expédiées en mode FTL, les commandes sont livrées par un ensemble de tournées. Une tournée $r \in \Omega$ part du dépôt, et livre un sous-ensemble donné de commandes à leurs clients respectifs. Toutes les tournées satisfont un certain nombre de contraintes imposées par les expéditeurs, les transporteurs et la législation (par exemple, le nombre maximal d'arrêts, la durée maximale ou les détours, l'accès aux sites clients). Désignons par Ω l'ensemble de toutes les tournées réalisables.

Soit $\mathcal{K} = \{1, \dots, |\mathcal{K}|\}$ l'ensemble des transporteurs FTL, chacun proposant une flotte hétérogène de véhicules. Pour chaque transporteur $k \in \mathcal{K}$, l'ensemble $\mathcal{V}^k = \{1, \dots, |\mathcal{V}^k|\}$ décrit les différents types de véhicules disponibles. De plus, nous définissons $\mathcal{V} = \bigcup_{k \in \mathcal{K}} \mathcal{V}^k$ l'ensemble des types de véhicules. Nous notons k^v le transporteur associé au type de véhicule $v \in \mathcal{V}$. On suppose que tous les types de véhicules ont la même vitesse de déplacement. En raison des contraintes de capacité et des diverses contraintes métier, les tournées ne sont pas compatibles avec tous les types de véhicules. Nous désignons par Ω_v le sous-ensemble de tournées pouvant être effectuées par un véhicule de type $v \in \mathcal{V}$ et \mathcal{V}_r la *liste de compatibilité* d'une tournée r , c'est-à-dire la liste des types de véhicules qui peuvent effectuer la tournée r . En outre, nous désignons par Ω^n le sous-ensemble de tournées qui peuvent livrer une commande $n \in \mathcal{N}$.

Le coût d'une tournée $r \in \Omega$ effectuée par un véhicule de type $v \in \mathcal{V}$ est noté $c_{r,v}$. Tout comme Ceschia et al. (2011), nous avons implémenté notre méthode de résolution de façon à ce que cette valeur puisse être calculée par une fonction « boîte noire » pouvant être adaptée aux types de tarifs variés proposés par les différents transporteurs. Chaque type de véhicule $v \in \mathcal{V}$ est disponible en une quantité limitée Q_v . Enfin, pour chaque transporteur, le nombre total de véhicules $v \in \mathcal{V}^k$ pouvant être utilisé est borné par la quantité ub_k , que l'on appelle « limite transporteur ».

Le VRP-FLC est modélisé par le programme linéaire en variables binaires (4.1)–(4.6). Pour toute tournée $r \in \Omega$ et tout type de véhicule $v \in \mathcal{V}^r$, la variable binaire $x_{r,v}$ prend pour valeur 1 un véhicule de type v effectue la tournée r , et 0 sinon. Pour toute commande $n \in \mathcal{N}$, la variable binaire y_n prend pour valeur 1 si la commande n est expédiée en LTL, et 0 sinon.

$$\min \quad z = \sum_{r \in \Omega} \sum_{v \in \mathcal{V}_r} c_{r,v} x_{r,v} + \sum_{n \in \mathcal{N}} c_n^{LTL} y_n \quad (4.1)$$

s.t.

$$\sum_{r \in \Omega^n} \sum_{v \in \mathcal{V}_r} x_{r,v} + y_n = 1 \quad \forall n \in \mathcal{N} \quad (4.2)$$

$$\sum_{r \in \Omega_v} x_{r,v} \leq Q_v \quad \forall v \in \mathcal{V} \quad (4.3)$$

$$\sum_{v \in \mathcal{V}^k} \sum_{r \in \Omega_v} x_{r,v} \leq ub_k \quad \forall k \in \mathcal{K} \quad (4.4)$$

$$x_{r,v} \in \{0, 1\} \quad \forall r \in \Omega, \forall v \in \mathcal{V}_r \quad (4.5)$$

$$y_n \in \{0, 1\} \quad \forall n \in \mathcal{N} \quad (4.6)$$

La fonction objectif minimise l'ensemble des coûts de transports. La première somme représente le coût des tournées FTL, la deuxième somme représente les coûts d'expédition des commandes en LTL. Les contraintes (4.2) assurent que chaque commande est livrée une fois et une seule, soit par une tournée, soit en LTL. Les contraintes (4.3) modélisent le nombre limité de véhicules de chaque type. Les contraintes (4.4) modélisent la limite transporteur pour chaque transporteur.

Une solution du *VRP-FLC* est un ensemble de tournées FTL partant du dépôt et livrant un sous-ensemble de commandes, tel que chaque tournée est affectée à un type de véhicule et que la limite transporteur et les contraintes sur le nombre de véhicules de chaque type soient respectés. On peut noter qu'il existe toujours une solution réalisable consistant à expédier l'ensemble des commandes en LTL.

4.3 Méthode de résolution

Cette section décrit la matheuristique LNS-SPP et son adaptation pour résoudre le VRP-FLC. Le LNS a été introduit pour la première fois par Shaw (1998). Il consiste à explorer de grands voisinages en appliquant de manière itérative des opérateurs de retrait (ou destruction) et d'insertion (ou réparation) (Ropke et Pisinger 2006a), également connus sous le nom d'heuristiques *ruin et recreate* (Schrimpf et al. 2000)). Les opérateurs de retrait sont des algorithmes heuristiques qui sélectionnent une certaine quantité de commandes et les enlèvent de la solution actuelle. La sortie de l'opérateur de destruction

est une *solution incomplète* et une *banque de requêtes*, où sont stockés les commandes supprimées. Les opérateurs de réparation sont des algorithmes heuristiques qui réinsèrent les commandes contenues dans la banque de requêtes dans la solution incomplète, avec pour objectif d’obtenir une nouvelle solution réalisable. Les opérateurs de destruction et de réparation sont appliqués de manière itérative jusqu’à ce qu’un critère d’arrêt soit atteint, par exemple, un nombre d’itérations ou une limite de temps. Le potentiel du LNS pour résoudre une grande variété de problèmes de tournées de véhicules a été souligné par Ropke et Pisinger (2006a), Ropke et Pisinger (2006b) et Pisinger et Ropke (2007) et dans l’état de l’art sur le LNS (Pisinger et Ropke 2019).

Nous soulignons trois caractéristiques majeures de notre implémentation du LNS. Premièrement, une source d’inspiration est la métaheuristique récente «ruin and recreate» de Christiaens et Vanden Berghe (2020), basée sur de petites destructions et des heuristiques d’insertion rapides. Grâce à ces deux facteurs, le LNS peut effectuer un très grand nombre d’itérations, ce qui compense en quelque sorte l’absence de recherche locale. L’algorithme de Christiaens et Vanden Berghe (2020) se montre compétitif sur de nombreux problèmes de tournées de véhicules, y compris le très étudié VRPTW. Ces opérateurs sont détaillés dans la section 4.3.2. Deuxièmement, une spécificité de notre approche est que, au cours de la réparation d’une solution, nous n’affectons pas immédiatement les tournées à un type de véhicule. Nous suivons l’approche proposée par Tellez et al. (2018) et maintenons une liste des types de véhicules compatibles avec chaque tournée. Afin de prendre en compte la limite transporteur et le nombre limité de véhicules disponibles pour chaque type, cette liste de véhicule compatibles est filtrée à l’aide d’un algorithme de filtrage adapté de l’algorithme proposé par Régim (1996). Une fois que toutes les commandes ont été réinsérées, un type de véhicule est affecté à chaque tournée avec un algorithme d’affectation heuristique. La gestion des types de véhicules au cours de la résolution est détaillée en Section 4.3.3. Troisièmement, un problème de partitionnement d’ensemble (SPP) est résolu à intervalles réguliers durant l’exécution du LNS afin de recombinaison les tournées produites au cours des différentes itérations. L’utilisation du SPP comme une technique de post-optimisation est reportée dans (Mancini 2017; Gschwind et Drexl 2019). Son hybridation avec des heuristiques est reportée dans de nombreuses publications (Prescott-Gagnon et al. 2009; Groër et al. 2011; Mendoza et Villegas 2013; Subramanian et al. 2013; Parragh et Schmid 2013; Yildirim et Çatay 2015; Grangier et al. 2017; Tellez et al. 2018; Dumez et al. 2021a).

4.3.1 Description de l’algorithme LNS-SPP

La méthode de résolution proposée est décrite par l’Algorithme 1.

Algorithme 1 : Algorithme LNS-SPP

Paramètres : Σ^-, Σ^+ : Opérateurs de destruction et de réparation
Entrées : s : Solution initiale

```

1:  $\Omega \leftarrow \text{routes}(s)$ 
2:  $s^* \leftarrow s$ 
3: tant que critère d’arrêt non vérifié faire
4:    $\sigma^- \leftarrow \text{ChoisirOperateurRetrait}(\Sigma^-)$ 
5:    $\sigma^+ \leftarrow \text{ChoisirOperateurInsertion}(\Sigma^+)$ 
6:    $\phi \leftarrow U(\phi_{min}, \phi_{max})$  // Choix de la taille du retrait
7:    $s' \leftarrow \sigma^+(\sigma^-(s, \phi))$ 
8:    $lb \leftarrow \text{BorneInferieure}(s')$ 
9:   si  $\text{CritereAcceptation}(lb, s^*)$  alors
10:     $s' \leftarrow \text{AffectationTournéesVehicules}(s')$ 
11:    si  $\text{CritereAcceptation}(z(s'), s^*)$  alors
12:       $\Omega \leftarrow \Omega \cup \text{routes}(s')$ 
13:       $s \leftarrow s'$ 
14:      si  $z(s) < z(s^*)$  alors
15:         $s^* \leftarrow s$ 
16:      fin
17:    fin
18:  fin
19:  si  $|\Omega| \geq \eta$  alors
20:     $s^* \leftarrow \text{SPP}(\Omega)$ 
21:     $\Omega \leftarrow \text{routes}(s^*)$ 
22:     $s \leftarrow s^*$ 
23:  fin
24: fin
25: retourner  $s^*$ 

```

Une solution initiale s est obtenue en appliquant l’opérateur d’insertion *Farthest List Insertion* sur une solution vide.

La banque de tournées Ω est initialisée avec les tournées contenues dans la solution initiale (ligne 1) et la meilleure solution s^* est initialisée comme une copie de la solution initiale s (ligne 2). La boucle principale (lignes 3–24) est répétée jusqu’à ce qu’un critère d’arrêt soit satisfait. Ce critère d’arrêt peut être déterminé par un temps d’exécution maximal, un nombre d’itérations maximal, ou un nombre maximal d’itérations sans

amélioration de la meilleure solution. À chaque itération, un opérateur de retrait et un opérateur d'insertion sont choisis aléatoirement avec une roulette biaisée (lignes 4 et 5). Un poids attribué à chaque opérateur de retrait et d'insertion détermine leur probabilité d'être utilisé à cette étape. Une taille de retrait ϕ est également choisie aléatoirement selon une loi uniforme dans l'intervalle $[\phi_{min}, \phi_{max}]$ (ligne 6). Les opérateurs de retrait et d'insertion sont ensuite appliqués à la solution courante s et le résultat est sauvegardé dans une copie s' (ligne 7). Pendant cette phase, toutes les insertions dans une tournée r sont évaluées en faisant l'hypothèse que la tournée r sera effectuée par le véhicule le moins cher compatible avec r après l'insertion. Avant d'appeler l'heuristique d'affectation, une borne inférieure lb est calculée (ligne 8). Cette borne est obtenue en affectant à chaque tournée son type de véhicule compatible le moins cher, sans tenir compte du nombre limité de véhicules de chaque type ou de la limite transporteur. On vérifie ensuite si la borne inférieure lb passe le *critère d'acceptation*. Le critère d'acceptation (lignes 9 et 11) est un test rapide basé sur le principe du «record-to-record» (Dueck 1993). Si la fonction objectif $z(s')$ de la nouvelle solution s' ne constitue pas une dégradation de plus de $\mathcal{T}\%$ par rapport à la meilleure solution s^* , alors s' est acceptée comme la nouvelle solution courante. Comme suggéré dans Dumez et al. (2021a), \mathcal{T} est périodiquement mis à jour afin de maintenir le ratio des solutions acceptées entre 4% et 14%. Si le taux de solutions acceptées lors des dernières itérations est inférieur à 4%, \mathcal{T} est multiplié par 1,5. S'il est supérieur à 14%, \mathcal{T} est divisé par 1,5.

Si la borne inférieure lb passe ce critère, alors un type de véhicule est affecté à chaque tournée de s' de sorte que toutes les contraintes du VRP-FLC soient satisfaites. L'heuristique d'affectation sera détaillée dans la section 4.3.3. Si cette solution complète passe le critère d'acceptation, alors les routes dans s' sont ajoutées à Ω (ligne 12) et la solution actuelle et, si besoin, la meilleure solution sont mises à jour (lignes 13–16).

Les lignes 19–23 correspondent à la résolution du SPP. Lorsque la taille de la banque de tournées Ω atteint un nombre prédéterminé de tournées η , une instance de SPP contenant toutes les tournées de la banque est résolue. Au lieu de résoudre une formulation SPP, nous résolvons une formulation de type Set Covering Problem (SCP). Yıldırım et Çatay (2015) montrent que la résolution d'un SCP au lieu d'un SPP réduit légèrement le temps de résolution. En contrepartie, la solution obtenue peut occasionnellement servir plusieurs fois certains clients, auquel cas une procédure gloutonne est appliquée afin de «réparer» la solution.

Le SCP est résolu par un solveur MIP avec une limite de temps de 30 secondes. Ensuite,

le banque de tournées est vidée et réinitialisée avec les tournées de la meilleure solution trouvée.

4.3.2 Opérateurs du LNS

L'efficacité du LNS repose sur le choix d'un ensemble d'opérateurs de retrait et d'insertion. Nous avons implémenté douze opérateurs de retrait et dix opérateurs d'insertion. Après des expérimentations préliminaires sur un ensemble représentatif d'instances, cet ensemble a été réduit aux six opérateurs de retrait présentés dans la sous-section 4.3.2.1 et aux cinq opérateurs d'insertion présentés dans la sous-section 4.3.2.2. La sélection de ces opérateurs est basée sur des expérimentations avec le package *irace* (López-Ibáñez et al. 2016), en laissant le poids de chaque opérateur prendre une valeur entière entre zéro et deux. Seuls les opérateurs ayant un poids strictement positif ont été conservés. Après cette première sélection, un second réglage avec *irace* a été effectué pour déterminer des poids entre 1 et 4. Les valeurs obtenues, reportées dans le Tableau 4.1, sont utilisées dans une roulette biaisée à chaque itération pour déterminer la probabilité de sélection de chaque opérateur.

4.3.2.1 Opérateurs de retrait

A chaque itération, les opérateurs de retrait sont paramétrés avec une taille de destruction ϕ tirée aléatoirement selon une loi uniforme dans l'intervalle $[\phi_{min}, \phi_{max}] \subset [0, 1]$. Pour tous les opérateurs de destruction énumérés ci-dessous, à l'exception du *string removal* et du *random removal*, les commandes sont triées en fonction d'un certain critère et les $\lceil \phi \times |\mathcal{N}| \rceil$ premières commandes sont retirées de la solution actuelle et placés dans une *banque de requêtes*.

Les opérateurs de retrait utilisés dans la ligne 4 de l'Algorithme 1 sont les suivants :

- *Random Removal* (Ropke et Pisinger 2006a) : cet opérateur retire des commandes aléatoirement.
- *(Split) String Removal* (Christiaens et Vanden Berghe 2020) : cet opérateur sélectionne plusieurs séquences de livraisons dans différentes tournées proches les unes des autres. Pour chaque séquence sélectionnée, soit elle est enlevée de la solution, soit elle est préservée et les livraisons en début et fin de tournée sont enlevées. La méthode a été implémentée telle que décrite par Christiaens et Vanden Berghe (2020) avec les paramètres suggérés à l'exception de \bar{c} , le nombre moyen de

- commandes à retirer, qui est défini par $[\phi \times |\mathcal{N}|]$ à chaque application de l'opérateur.
- *Least LTL Cost Removal* : cet opérateur retire les commandes ayant un coût d'expédition en LTL le plus faible.
 - *Distance Related Removal* (Shaw 1998) : cet opérateur choisit une commande aléatoirement et la retire ainsi que les commandes les plus proches géographiquement.
 - *Largest Request Removal* : cet opérateur retire les commandes les plus grandes, en fonction de leur nombre de palettes.
 - *Worst-Time Removal* (Demir et al. 2012) : cet opérateur retire les commandes pour lesquelles le véhicule arrive avec le plus d'avance par rapport à la fenêtre de temps choisie pour la livraison.

4.3.2.2 Opérateurs d'insertion

Comme toutes les commandes peuvent être expédiées en mode LTL, il est toujours possible (et facile) de reconstruire des solutions réalisables. Nous utilisons des opérateurs de réparation rapide repris ou inspirés de ceux proposés par Christiaens et Vanden Berghé (2020). Ils consistent à trier la banque de requêtes une fois au début de la phase de reconstruction, puis, en fonction de ce tri, toutes les commandes de la banque de requêtes sont séquentiellement insérées dans la solution à leur meilleure position possible. Un autre avantage de l'utilisation de ces heuristiques est que le nombre d'évaluations des coûts d'insertion est d'un ordre de grandeur moindre par rapport aux heuristiques traditionnelles du LNS telles que les insertions gloutonnes ou basées sur des regrets (Ropke et Pisinger 2006b).

Pour réduire davantage le temps consacré au calcul des coûts d'insertion, nous avons conçu la fonction d'évaluation des coûts de manière à accepter une liste de paramètres qui sont calculés de manière incrémentale. Les paramètres qui doivent être calculés dépendent des types de coûts de l'instance en question. Pour les coûts trouvés dans les instances de la littérature, seules la charge, la longueur et la durée de la tournée suffisent. D'autres informations, telles que les types de produits ou la liste des nœuds visités, sont calculées uniquement si des fonctions de coût plus complexes le nécessitent.

Les opérateurs d'insertion utilisés dans la ligne 5 de l'Algorithme 1 sont les suivants :

- *Random* : la séquence d'insertion des commandes est aléatoire ;
- *Request Size* : les commandes comportant le plus grand nombre de palettes sont insérées en premier ;
- *Farthest* : les commandes dont le lieu de livraison est le plus éloigné du dépôt sont

- insérées en premier ;
- *Nearest* : les commandes dont le lieu de livraison est le plus proche du dépôt sont insérées en premier ;
- *Time Window Width* : les commandes dont la fenêtre de temps est la plus étroite sont insérées en premier.

Comme dans Christiaens et Vanden Berghe (2020), lors de l'évaluation du coût d'insertion d'une commande à différentes positions dans une tournée, chaque position a une probabilité p^{blink} de ne pas être considérée.

Dans l'étape de réparation du LNS (ligne 7 de l'Algorithme 1), l'insertion d'une commande dans une solution incomplète nécessite d'évaluer l'augmentation de coût générée par l'utilisation de chaque type de véhicule à chaque position d'insertion. Vérifier la faisabilité de l'insertion d'une commande dans une tournée $r \in \Omega$ revient à vérifier que, après insertion (i) la tournée est toujours valide par rapport aux fenêtres de temps, (ii) la liste des types de véhicules compatibles avec r n'est pas vide et (iii) les contraintes (4.3)–(4.4) sont respectées. La réalisabilité d'une insertion vis à vis des fenêtres de temps peut être vérifiée en temps constant en maintenant des *forward time slack* (Savelsbergh 1985). Pour les instances comportant des fenêtres de temps multiples, nous maintenons également des *backward time slack* et utilisons la méthode proposée par Hoogeboom et al. (2020) afin de vérifier la faisabilité d'une insertion et de calculer l'augmentation de la durée de la tournée. On notera que tous les véhicules se déplacent à la même vitesse, ce calcul n'est donc pas dépendant du véhicule qui effectue la tournée.

L'évaluation du coût lié à une insertion se base sur le véhicule le moins cher qui reste compatible avec la tournée après l'insertion. C'est pourquoi dans le cas d'une flotte limitée, une attention particulière est portée à la gestion des types de véhicules compatibles avec les tournées, comme expliqué en sous-section 4.3.3.

4.3.3 Gestion de la flotte limitée

Cette sous-section détaille la vérification de la faisabilité de l'insertion de commandes vis à vis des contraintes de la flotte hétérogène limitée et de la limite transporteur. L'exemple 1 illustre le cas où une certaine affectation des types de véhicules aux tournées est rendue impossible par la combinaison de la taille de la flotte et de la limite transporteur.

Exemple 1 Soit deux transporteurs k_1 et k_2 . Le transporteur k_1 possède une flotte de 10

véhicules : 9 de type A et 1 de type B. Le transporteur k_2 possède une flotte de 4 véhicules : 2 de type C et 2 de type D, mais il n'a que 2 chauffeurs disponibles ; par conséquent, seuls 2 véhicules peuvent être utilisés.

	Transporteurs			
	k_1 (10)	k_2 (2)		
Types de véhicules	A	B	C	D
# véhicules disponibles	9	1	2	2

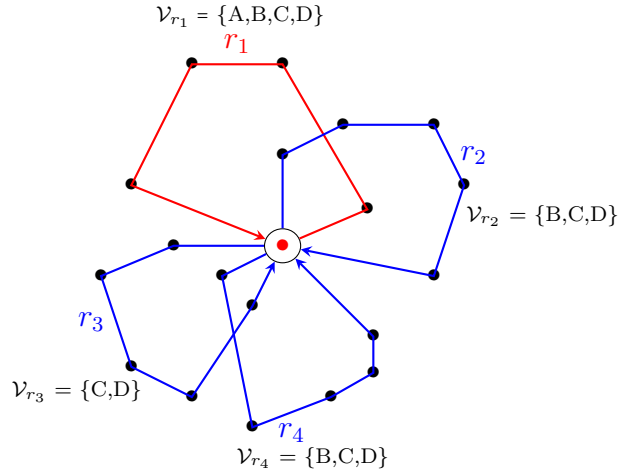


FIGURE 4.1 – Exemple d’une phase de réparation. La liste de véhicules compatibles avec r_1 peut être filtrée et restreinte à $\{A\}$

Considérons la solution incomplète représentée sur la Figure 4.1, avec quatre tournées désignées r_1, r_2, r_3 et r_4 . Les commandes non servies, placées dans la banque de requêtes, ne sont pas représentées. La tournée r_1 peut être effectuée par tous les types de véhicules ($\mathcal{V}_{r_1} = \{A, B, C, D\}$). Les autres tournées ont un ensemble limité de véhicules compatibles : $\mathcal{V}_{r_2} = \{B, C, D\}$, $\mathcal{V}_{r_3} = \{C, D\}$ et $\mathcal{V}_{r_4} = \{B, C, D\}$. Étant donné que les tournées r_2, r_3 et r_4 ne sont compatibles qu’avec les types de véhicules B, C et D, et qu’au maximum trois véhicules de ces types peuvent être utilisés, ces tournées utiliseront tous les véhicules B, C et D disponibles. Par conséquent, même si la tournée r_1 est compatible avec tous les types de véhicules, elle ne peut être effectuée que par un véhicule de type A. La tournée r_1 doit donc être affectée à un véhicule de type A.

Dans la sous-section 4.3.3.1, nous décrivons l’algorithme de filtrage utilisé pour restreindre les types de véhicules possibles pour chaque tournée. Dans la sous-section 4.3.3.2, nous détaillons l’algorithme heuristique utilisé pour attribuer un type de véhicule à chaque tournée.

4.3.3.1 Algorithme de filtrage

Comme illustré dans l'exemple précédent, un certain filtrage des listes de compatibilité des tournées peut être déduit pour chaque tournée, en fonction des autres tournées de la solution, de la taille des flottes et de la limite transporteur. L'objectif de ce filtrage est double : premièrement, il permet d'éviter les insertions qui aboutiraient à une solution infaisable (dans l'exemple, effectuer une insertion qui supprimerait A de \mathcal{V}_{r_1}), et deuxièmement, il permet une meilleure estimation du coût des insertions au cours de la réparation et du calcul de la borne inférieure avant de résoudre le problème d'affectation.

Soit s une solution incomplète durant la phase de réparation de l'Algorithme 1 définie par un ensemble de tournées $\Omega(s)$ et une banque de requêtes contenant les commandes à réinsérer.

Nous introduisons la notation \mathcal{V}_r^s pour définir la *liste de compatibilité filtrée* d'une tournée r dans la solution s , avec $\mathcal{V}_r^s \subseteq \mathcal{V}_r$.

Cette liste est obtenue par un algorithme de filtrage adapté de Régim (1996) afin de contraindre conjointement le nombre de véhicules de chaque type affecté aux tournées et le nombre maximum de véhicules affrétés à chaque transporteur. Régim (1996) propose un algorithme de filtrage pour la contrainte dite de *cardinalité globale* qui impose qu'un nombre minimal et maximal d'occurrences d'un ensemble de valeurs puisse être attribué à un ensemble de variables. Notre algorithme de filtrage repose sur un graphe $G = (X, E)$, où X contient trois couches de sommets représentant respectivement les transporteurs $t \in T$, les types de véhicules $v \in \mathcal{V}$ et les tournées $r \in \Omega(s)$. Il comprend également un nœud source \mathbf{o} et un nœud puits \mathbf{o}' . L'ensemble d'arcs E contient un arc (\mathbf{o}, k) de capacité ub_k pour chaque transporteur $k \in \mathcal{K}$, un arc (k, v) de capacité \mathcal{Q}_v pour chaque type de véhicule v d'un transporteur k , et des arcs (v, r) de capacité 1 entre les tournées et les types de véhicules compatibles dans la solution ($(v, r) \in E \iff v \in \mathcal{V}_r^s$). Les nœuds correspondant aux tournées sont reliés au nœud puits par des arcs (r, \mathbf{o}') de capacité 1. Comparativement à Régim (1996), le graphe proposé comporte une couche additionnelle pour les transporteurs.

Exemple 2 *Le graphe de filtrage correspondant au cas présenté dans l'Exemple 1 est représenté par la Figure 4.2.*

Soit \mathbf{f} un flot maximal allant de la source au puits dans le graphe G . Comme les nœuds correspondant aux tournées sont reliés par un arc de capacité 1 au puits \mathbf{o}' , la valeur de \mathbf{f} est inférieure ou égale au nombre de tournées. Si cette valeur est égale au

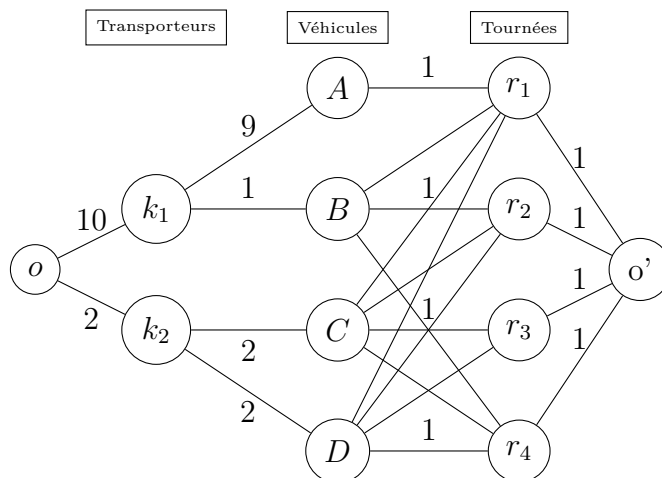


FIGURE 4.2 – Graphe représentant les attributions possibles de types de véhicules aux tournées

nombre de tournées, alors, par construction, chaque tournée est associée à un seul type de véhicule. Par conséquent, les arcs utilisés dans le flux entre les sommets des véhicules et des tournées définissent une affectation réalisable des tournées aux types de véhicules.

En effet, de gauche à droite dans l'exemple, la première et la deuxième couche d'arcs garantissent respectivement que les limites transporteurs et les contraintes de flotte limitée sont vérifiées. La troisième couche garantit que chaque tournée est affectée à un véhicule de sa liste de compatibilité. Étant donné un flot maximal \mathbf{f} , nous construisons un graphe résiduel $\mathcal{R}(\mathbf{f})$. Il s'agit d'un graphe orienté dont l'ensemble des sommets est le même que celui du graphe G et où les capacités des arcs sont les quantités par lesquelles le flux \mathbf{f} peut être soit augmenté soit diminué. Régim (1996) montre la propriété suivante :

Propriété 1 *Si le flot sur un arc reliant deux sommets est nul et si ces deux sommets n'appartiennent pas à la même composante fortement connexe dans $\mathcal{R}(\mathbf{f})$, alors il ne peut exister un autre flot maximal avec un flot non nul sur cet arc.*

Selon la Propriété 1, si pour un type de véhicule $k \in \mathcal{K}$ et une tournée $r \in \Omega(s)$ il existe un flot maximal \mathbf{f} avec $\mathbf{f}(k, u) = 0$ et si k et u n'appartiennent pas à la même composante fortement connexe de $\mathcal{R}(\mathbf{f})$, alors il n'existe pas d'affectation de toutes les tournées à un type de véhicule lorsque la tournée r est affectée au type de véhicule k . La méthode de filtrage consiste à calculer un flot maximal ainsi que son graphe résiduel et à identifier les composantes fortement connexes sur le graphe résiduel. Ensuite, pour chaque tournée $r \in \Omega(s)$ et chaque type de véhicule $v \in \mathcal{V}_r$, on vérifie si la Propriété 1 est vraie pour l'arc

(v, r) . Si c'est le cas, v est retiré de la liste de compatibilité \mathcal{V}_r^s de la tournée r . La Figure 4.3 montre un exemple de flot maximal qui correspond à l'affectation d'un véhicule de type A (resp. B, C, D) aux tournées r_1 (resp. r_2, r_3, r_4).

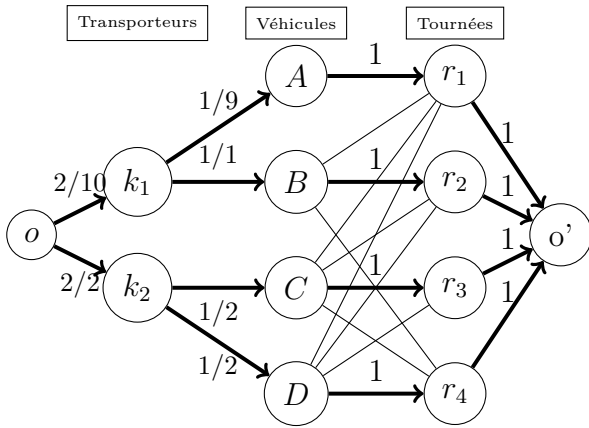


FIGURE 4.3 – Flot maximal correspondant à une affectation réalisable

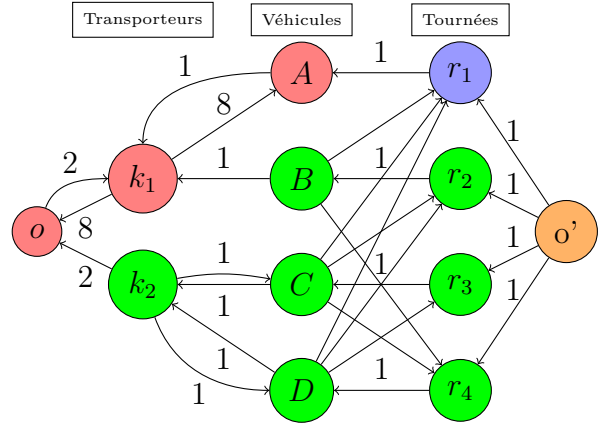


FIGURE 4.4 – Graphe résiduel et composantes fortement connexes d'un flot maximal

La Figure 4.4 montre le graphe résiduel de ce flot. Il possède quatre composantes fortement connexes $\{o, k_1, A\}$, $\{k_2, B, C, D, r_2, r_3, r_4\}$, $\{r_1\}$ et $\{o'\}$, qui sont indiquées par des couleurs différentes sur le graphe. Puisque r_1 et B (resp. C et D) n'appartiennent pas à la même composante fortement connexe dans $\mathcal{R}(\mathbf{f})$ et que le flux sur les arcs les reliant est nul dans \mathbf{f} , r_1 ne peut être attribué qu'au type de véhicule A.

4.3.3.2 Affectation des types de véhicules aux tournées

Après avoir réinséré toutes les commandes de la banque de requêtes, une solution complète est obtenue en affectant un type de véhicule à chaque tournée de la solution s' (ligne 10 de l'Algorithme 1). Cela revient à résoudre le *VRP-FLC* (4.1)-(4.6) avec Ω restreint à l'ensemble des tournées $\Omega(s')$.

Avant de calculer cette affectation, nous vérifions qu'une borne inférieure sur la valeur de l'affectation passe le critère d'acceptation. Cette borne est obtenue en résolvant le *VRP-FLC* sans tenir compte des contraintes de flotte limitée et de la limite transporteur (4.3) et (4.4). Dans une telle solution, chaque tournée est donc affectée à son type de véhicule compatible le moins cher. Ensuite, les tournées sont sélectionnées si et seulement si leur coût ne dépasse pas la somme des coûts de livraison LTL de ses commandes.

Si cette borne passe le critère d'acceptation, alors nous résolvons le problème d'affectation généralisé avec l'heuristique de regret MTHG (Martello 1990) pour obtenir une solution complète réalisable du *VRP-FLC*. Nous définissons le regret associé à une tournée comme la différence de coût de la tournée entre les deux véhicules compatibles les moins chers. Si une route peut être affectée à un seul type de véhicule, son regret est défini comme la différence entre le coût d'expédition en LTL de toutes ses commandes et le coût de la tournée avec son seul type de véhicule compatible. De manière itérative, nous affectons la route avec le plus grand regret à son type de véhicule le moins cher. Lorsqu'un type de véhicule ne peut plus être attribué en raison de sa disponibilité limitée ou de la limite transporteur, les regrets pour les tournées qui dépendaient de ce type de véhicule sont recalculés.

4.4 Expérimentations numériques

Cette section présente les résultats des expérimentations numériques effectuées sur un serveur de calcul fonctionnant avec Ubuntu 20.04.2 LTS et utilisant un processeur Intel(R) Xeon(R) Gold 6230 @ 2.10GHz. IBM Ilog CPLEX 20.1.0 est utilisé comme solveur MIP, avec un réglage standard et le paramètre `CPXPARAM_Threads` fixé à la valeur 1.

La sous-section 4.4.1 résume la valeur des paramètres utilisés dans l'Algorithme 1. Comme il n'existe aucune instance du problème VRP-FLC dans la littérature, nous évaluons l'efficacité de notre approche sur des instances de problèmes très proches. La sous-section 4.4.2 relate les expérimentations sur le HFF-VRP-TW. La sous-section 4.4.3 présente les résultats sur le HFF-VRP-TW-PC. La revue de la littérature pour ces différents problèmes a été présentée au Chapitre 3 en section 3.1. La sous-section 4.4.4 présente de nouvelles instances générées pour le VRP-FLC. Ces instances sont résolues par notre LNS et permettent de tirer des premiers enseignements sur le recours combiné aux flottes FTL et LTL.

4.4.1 Paramètres de l'algorithme

Le Tableau 4.1 présente les valeurs des différents paramètres utilisés dans l'Algorithme 1. Les poids des opérateurs de retrait et d'insertion ont été obtenus à l'aide du package *irace* sur un ensemble représentatif d'instances.

	Paramètre	Valeur
Paramètres généraux	critère d'acceptation initial	1%
	ϕ_{min}	10%
	ϕ_{max}	20%
	p^{blink}	0.01
	μ	300
Fréquence d'appel au SPP	η	3000
Poids des opérateurs de retrait	Random Removal	2
	(Split) String Removal	4
	Least LTL Cost Removal	1
	Distance Related Removal	1
	Largest Request Removal	1
	Greatest Waiting Time Removal	2
Poids des opérateurs d'insertion	Random	3
	Request Size	1
	Farthest	2
	Nearest	2
	Time Window Width	2

TABLEAU 4.1 – Liste et valeurs des paramètres du LNS

4.4.2 Résultats sur les instances de HFF-VRP-TW

Paraskevopoulos et al. (2008) proposent des instances pour le HFF-VRP-TW à partir d'un sous-ensemble des instances FSM-VRP-TW de Liu et Shen (1999). Ce sous-ensemble est constitué des 4 premières instances de chaque catégorie des instances de Liu et Shen (1999) avec des coûts fixes élevés pour les véhicules. La taille de la flotte est fixée comme étant égale à celle trouvée dans les meilleures solutions connues du FSM-VRP-TW.

L'objectif à minimiser est la somme des coûts fixes des véhicules et des durées des tournées. Le temps de service, qui est constant pour toute solution, n'est pas compté dans l'objectif. Koç et al. (2015) présentent des résultats de leur méthode Hybrid Evolutionary Algorithm (HEA) sur ces instances. Ces résultats sont déclinés selon deux fonctions objectifs : la minimisation du temps et minimisation de la distance.

Le Tableau 4.2 compare nos résultats avec ceux obtenus par Koç et al. (2015) pour la minimisation de la distance.

La première colonne contient les noms des instances. Pour chaque méthode, les colonnes intitulées **Mix** et z^{best} décrivent le nombre de véhicules de chaque type utilisés dans la meilleure solution trouvée, et la valeur de la fonction objectif. La colonne intitu-

Instance	HEA Koç et al. (2015)			LNS-SPP		
	Mix	z^{best}	Temps (min)	Mix	z^{best}	gap (%)
R101A	$B^{10}C^{11}D^1$	4355.41	5.19	$A^1B^{11}C^{10}D^1$	4333.36	-0.51
R102A	$B^4C^{13}D^2$	4356.44	6.24	$A^1B^4C^{14}D^1$	4187.93	-3.87
R103A	B^6C^{15}	4080.16	6.57	B^7C^{14}	4038.72	-1.02
R104A	B^7C^{14}	3954.72	5.89	B^7C^{14}	3952.75	-2.58
C101A	B^{10}	8828.94	4.25	B^{10}	8828.94	0.0
C102A	A^{19}	7080.17	3.97	A^{19}	7080.17	0.14
C103A	A^{19}	7079.21	3.99	A^{19}	7079.21	0.0
C104A	A^{19}	7075.06	2.98	A^{19}	7075.06	0.0
RC101A	$A^4B^7C^7$	5162.28	6.41	$A^4B^7C^7$	5162.25	0.0
RC102A	$A^2B^6C^8$	5018.05	5.24	$A^4B^5C^8$	4986.61	-0.63
RC103A	$A^{10}B^2C^8$	4926.55	4.39	$A^{10}B^2C^8$	4897.08	-0.6
RC104A	$A^2B^{13}C^3D^1$	4995.91	4.88	$A^1B^{13}C^3D^1$	4950.62	-0.91
R201A	A^5	3448.76	6.74	A^5	3463.27	0.42
R202A	A^5	3308.16	8.13	A^5	3314.89	0.2
R203A	A^4B^1	3382.39	7.49	A^4B^1	3392.79	0.31
R204A	A^5	3018.14	5.47	A^5	3022.23	0.14
C201A	A^4B^1	6082.38	4.21	A^4B^1	6082.38	0.0
C202A	A^1C^3	7618.62	3.69	A^1C^3	7618.62	0.0
C203A	C^2D^1	7303.37	3.67	C^2D^1	7303.37	0.0
C204A	A^5	5677.66	5.11	A^5	5677.66	0.0
RC201A	C^1E^3	5344.47	6.72	C^1E^3	5294.8	-0.93
RC202A	$A^1C^1D^1E^2$	4856.02	6.48	$A^1C^1D^1E^2$	4865.98	0.21
RC203A	$A^1B^1C^5$	4246.25	6.93	$A^1B^1C^5$	4212.44	-0.8
RC204A	$A^{14}B^2$	4195.32	6.17	$A^{14}B^2$	4198.68	0.08
Average		5224.77	5.45		5209.16	-0.43
Runs		10			10	
Processor		Xe 2.6 GHz			Xe Gold 6230 2.10GHz	

TABLEAU 4.2 – Résultats du LNS-SPPs sur les instances HFF-VRP-TW avec minimisation de la distance

lée **gap (%)** donne l'écart relatif entre l'algorithme LNS-SPP proposé et les meilleures solutions connues pour chaque instance. Un écart négatif indique une amélioration par rapport à la meilleure solution connue.

D'après ces expérimentations, nous constatons que, avec des temps de résolution plus courts, le LNS-SPP surpasse HEA sur les instances avec des tournées courtes (groupes d'instances R1, C1 et RC1) et qu'il est très proche des résultats de HEA sur les instances avec des tournées longues. De meilleures solutions sont trouvées pour 10 instances sur 24 ; Des solutions identiques sont trouvées pour 8 instances et des résultats légèrement moins bons ont été obtenus pour 6 instances.

Le Tableau 4.3 compare nos résultats avec ceux obtenus par Paraskevopoulos et al. (2008) et par Koç et al. (2015) pour la minimisation du temps.

Instance	ReVNTS		HEA		LNS-SPP		gap (%)
	Paraskevopoulos et al. (2008)		Koç et al. (2015)				
	Mix	z	Mix	z^{best}	Mix	z^{best}	
R101A	$B^{10}C^{11}D^1$	4583.99	$B^{10}C^{11}D^1$	4588.76	A^1, B^9, C^{11}, D^1	4626.66	0.931
R102A	$B^3C^{14}D^2$	4420.680	$A^1B^4C^{13}D^2$	4376.54	A^1, B^4, C^{14}, D^1	4394.16	0.403
R103A	B^6C^{15}	4195.05	B^6C^{15}	4201.71	B^7, C^{14}	4149.94	-1.075
R104A	B^8C^{14}	4065.52	B^9C^{13}	4027.69	B^7, C^{14}	3970.2	-1.427
C101A	B^{10}	8828.93	B^{10}	8828.93	B^{10}	8828.94	0.0
C102A	A^{19}	7137.79	A^{19}	7153.13	A^{19}	7119.35	-0.258
C103A	A^{19}	7143.88	A^{19}	7122.57	A^{19}	7102.86	-0.277
C104A	A^{19}	7104.96	A^{19}	7083.74	A^{19}	7081.51	-0.031
RC101A	$A^4B^7C^7$	5279.92	$A^4B^7C^7$	5266.36	A^4, B^7, C^7	5221.81	-0.846
RC102A	$A^4B^5C^8$	5149.95	$A^4B^5C^8$	5099.55	A^4, B^5, C^8	5043.57	-1.098
RC103A	$A^{10}B^2C^8$	5002.41	$A^{10}B^2C^8$	4991.29	A^{10}, B^2, C^8	4924.08	-1.347
RC104A	$A^2B^{13}C^3D^1$	5024.25	$A^2B^{13}C^3D^1$	5016.97	A^1, B^{13}, C^3, D^1	4952.61	-1.283
R201A	A^5	3779.12	A^5	3782.49	A^5	3839.82	1.606
R202A	A^5	3578.91	A^5	3583.92	A^5	3556.41	-0.629
R203A	A^4B^1	3582.54	A^4B^1	3553.92	A^4, B^1	3550.46	-0.097
R204A	A^5	3143.68	A^5	3081.80	A^5	3079.58	-0.072
C201A	A^4B^1	6140.64	A^4B^1	6140.64	A^4, B^1	6140.64	0.0
C202A	A^1C^3	7752.88	A^1C^3	7623.96	A^1, C^3	7623.96	0.0
C203A	C^2D^1	7303.37	C^2D^1	7303.37	C^2, D^1	7303.37	0.0
C204A	A^5	5721.09	A^5	5680.46	A^5	5680.46	0.0
RC201A	C^1E^3	5523.15	C^1E^3	5534.59	C^1, E^3	5536.76	0.246
RC202A	$A^1C^1D^1E^2$	5132.08	$A^1C^1D^1E^2$	5150.23	A^1, C^1, D^1, E^2	5189.71	1.123
RC203A	$A^1B^1C^5$	4508.27	$A^1B^1C^5$	4471.92	A^1, B^1, C^5	4466.48	-0.122
RC204A	$A^{14}B^2$	4252.87	$A^{14}B^2$	4241.83	A^{14}, B^2	4229.25	-0.297
Average		<i>5348.16</i>		<i>5326.74</i>		<i>5317.19</i>	<i>-0,19</i>
Average time (min)		20		5.61		2	
Runs		1		10		10	
Processor		PIV 1.5GHz		Xe 2.6GHz		Xe Gold 6230 2.10GHz	

TABLEAU 4.3 – Résultats du LNS-SPP sur les instances HFF-VRP-TW avec minimisation du temps

Là encore, le LNS-SPP obtient des résultats compétitifs par rapport à Paraskevopoulos et al. (2008) et Koç et al. (2015). Les meilleures solutions connues ont été améliorées pour

14 des 24 instances ; des solutions identiques sont trouvées pour 4 instances et des résultats légèrement moins bons ont été obtenus pour 6 instances.

4.4.3 Résultats sur les instances de HFF-VRP-TW-PC

Nous comparons les résultats de la méthode LNS-SPP avec les travaux les plus récents sur le HFF-VRP-TW-PC. Baller et al. (2020) proposent des instances pour le HFF-VRP-TW-PC basées sur les instances de Dabia et al. (2019) et séparées en deux catégories : une première avec des coûts LTL élevés et une seconde avec des coûts LTL faibles. Les définitions des flottes sont identiques à celles des instances de Liu et Shen (1999) et sont déclinées en trois types de flottes A, B, ou C selon que les coûts fixes soient respectivement élevés, moyens ou faibles. En revanche, chaque véhicule n'est disponible qu'en trois exemplaires. De plus petites instances sont également considérées en ne prenant que les 25 ou 50 premiers clients de chaque instance. Le Tableau 4.4 compare nos résultats aux résultats exacts proposés par Baller et al. (2020) sur le HFF-VRP-TW-PC. Chaque ligne contient des résultats agrégés sur un ensemble d'instances. La première colonne indique le type de flotte (A, B ou C) et la taille des instances (25, 50 ou 100 clients).

Chaque catégorie est composée de 56 instances. La colonne intitulée # indique le nombre d'instances de chaque catégorie que les auteurs ont pu résoudre de manière optimale dans un délai d'une heure. Ce nombre ne tient pas compte des solutions optimales consistant à expédier toutes les commandes en LTL. Les résultats présentés dans le Tableau 4.4 ne concernent que ce sous-ensemble d'instances. Les résultats complets pour les instances avec $n = 100$ clients sont disponibles en Annexe (Tableaux A.1 et A.2)¹

LNS-SPP est exécuté cinq fois sur chaque instance. La colonne intitulée #*opt* indique le nombre d'instances pour lesquelles la solution optimale a été trouvée sur au moins une des exécutions. La colonne 5 (z_{moy}^{best}) représente la moyenne de la meilleure fonction objectif (sur 5 exécutions) obtenue sur chaque instance. La colonne intitulée gap_{moy}^{best} indique l'écart d'optimalité associé à la valeur z_{moy}^{best} tandis que la colonne gap_{moy} indique l'écart moyen sur l'ensemble des exécutions de l'algorithme. Le temps de calcul (colonne 8) a été fixé à 30, 60 et 300 secondes respectivement pour les instances de 25, 50 et 100 clients.

Ce tableau montre que l'algorithme LNS-SPP est capable de trouver des solutions optimales ou quasi optimales pour la plupart des instances dans toutes les catégories, y

1. Suite à un échange avec les auteurs et un examen attentif de nos solutions optimales respectives, la valeur -0.002 s'explique par une procédure d'arrondi différente dans Dabia et al. (2019) et dans ce travail. Nous n'avons pas été en mesure de trouver l'origine précise de cette différence.

Instances	BCP			LNS-SPP				
	#	z_{moy}^*	t_{moy} (s)	# opt	z_{moy}^{best}	gap_{moy}^{best} (%)	gap_{moy} (%)	temps (s)
<i>Coûts LTL élevés</i>								
A/25	22	1233.03	10	22	1233.03	0	0.06	30
A/50	15	2402.88	185	14	2402.81	-0.002	0.41	60
A/100	5	5050.96	149	3	5051.52	0.01	0.14	300
B/25	34	586.36	9	34	586.36	0	0.26	30
B/50	25	1115.96	262	24	1116.35	0.037	0.28	60
B/100	11	2597.60	420	7	2599.27	0.07	0.25	300
C/25	37	470.84	9	34	471.36	0.11	0.14	30
C/50	28	890.25	330	26	890.82	0.06	0.15	60
C/100	10	2129.36	127	8	2129.80	0.02	0.24	300
<i>Coûts LTL faibles</i>								
A/25	13	1012.79	2	12	1012.86	0.0	0.25	30
A/50	12	2107.9	153	7	2109.85	0.094	0.47	60
A/100	3	4297.67	228	1	4323.60	0.6	1.06	300
B/25	25	585.23	4	25	585.23	0	0.19	30
B/50	19	1096.34	179	16	1096.91	0.05	0.22	60
B/100	6	2243.65	532	4	2244.70	0.05	0.25	300
C/25	29	483.61	23	28	483.72	0.023	0.29	30
C/50	20	920.57	289	19	921.125	0.060	0.23	60
C/100	6	1949.58	562	4	1951.60	0.12	0.45	300

TABLEAU 4.4 – Comparaison des résultats du LNS-SPP sur les instances HFF-VRP-TW-PC aux résultats exacts du branch-and-cut-and-price de Dabia et al. (2019)

compris pour la moitié des instances avec 100 clients.

4.4.4 Instances et résultats sur le VRP-FLC

Nous proposons de nouvelles instances de VRP-FLC adaptées de celles de Dabia et al. (2019), en considérant deux nouvelles caractéristiques du VRP-FLC : la présence de plusieurs transporteurs et une limitation du nombre total de véhicules affrétés à chaque transporteur. Les flottes de véhicules sont composées de 3 à 6 types de véhicules avec des coûts fixes élevés (resp. moyens, faibles) pour les instances de type A (resp. B, C) et une taille de flotte fixée à 3. Nous affectons la première moitié des types de véhicules (arrondie à la valeur entière la plus proche) à un transporteur et les types de véhicules restants à un second transporteur. Les limites transporteurs sont fixées comme suit : les transporteurs qui ont 2 types de véhicules peuvent utiliser au maximum 5 de leurs 6 véhicules ; les transporteurs qui ont 3 types de véhicules peuvent utiliser au maximum 7 de leurs 9

véhicules. Une instance VRP-FLC est générée pour chaque instance HFF-VRP-TW-PC, avec un coût d'expédition LTL élevé ou faible, soit un total de 336 instances.





































Inst	HFF-VRP-TW-PC			VRP-FLC			
	z_{min}	z_{moy}	Répartition des coûts Fixe Var LTL	z_{min}	Δ (%)	z_{moy}	Répartition des coûts Fixe Var LTL
R1A	4426.8	4456		4622.3	4.4	4648.2	
R1B	1967.6	2001.8		2463.2	25.2	2491.9	
R1C	1661.5	1692.1		2194.2	32.1	2221.9	
RC1A	4987.8	5000.4		5090.4	2.1	5098.2	
RC1B	2685.3	2696.5		3086.2	14.9	3096.9	
RC1C	2400.2	2410.8		2830.6	17.9	2842	
C1A	3620	3620		3620	0.0	3620	
C1B	2519.5	2528.7		2584.2	2.6	2598.4	
C1C	1783.7	1794.7		1882.3	5.5	1896.3	
R2A	3425.2	3463.6		3434.1	0.3	3466.9	
R2B	1413.2	1450.9		1415	0.1	1455.8	
R2C	1154.8	1187		1154.6	0.0	1186.4	
RC2A	4160.7	4211.3		4222.4	1.5	4283.2	
RC2B	1705.7	1729		1708	0.1	1733	
RC2C	1365.1	1389.1		1364	-0.1	1385.5	
C2A	3620	3620		3620	0.0	3620	
C2B	1648.7	1668.7		1648.5	0.0	1668.9	
C2C	1167.9	1184.1		1168.4	0.0	1185	

TABLEAU 4.5 – Résultats sur les instances avec coûts LTL élevés

Les Tableaux 4.5 et 4.6 résument les résultats sur ces nouvelles instances, pour les instances avec coûts LTL élevés et avec coûts LTL faibles, respectivement. Dans ces tableaux, nous comparons les résultats de notre approche LNS sur les deux variantes du VRPPC : les instances originales de HFF-VRP-TW-PC (colonnes 2-4), et les instances de VRP-FLC incluant la limite transporteur (colonnes 5-8). Pour chaque catégorie d'ins-

Inst	HFF-VRP-TW-PC			VRP-FLC			Répartition des coûts Fixe Var LTL
	z_{min}	z_{moy}	Répartition des coûts Fixe Var LTL	z_{min}	Δ (%)	z_{moy}	
R1A	4230	4249.7		4239	0.2	4253.6	
R1B	1897.2	1919.4		2144.3	13.0	2161.7	
R1C	1592.5	1612.5		1872.5	17.6	1891.9	
RC1A	3448	3448		3448	0.0	3448	
RC1B	2194.2	2200.5		2313.2	5.4	2319	
RC1C	1907	1913.1		2061.6	8.1	2068.9	
C1A	905	905		905	0.0	905	
C1B	905	905		905	0.0	905	
C1C	905	905		905	0.0	905	
R2A	3307.4	3363.9		3306.1	0.0	3359.8	
R2B	1412.7	1456.3		1413.8	0.1	1453	
R2C	1153.4	1188.1		1153.2	0.0	1189.7	
RC2A	3448	3448		3448	0.0	3448	
RC2B	1682	1722		1684.1	0.1	1725.8	
RC2C	1342.6	1377.8		1343.7	0.1	1370.2	
C2A	905	905		905	0.0	905	
C2B	905	905		905	0.0	905	
C2C	905	905		905	0.0	905	

TABLEAU 4.6 – Résultats sur les instances avec coûts LTL faibles

tance, nous reportons la valeur de la meilleure fonction objectif z_{min} trouvée sur cinq exécutions et la valeur moyenne z_{moy} . Les colonnes 4 et 8 montrent la répartition des coûts en trois catégories : de gauche à droite le pourcentage des coûts fixes de la flotte FTL, le pourcentage des coûts variables qui dépendent de la distance parcourue et le pourcentage des coûts de transport en LTL.

La colonne 6 indique l'écart relatif entre les meilleures solutions trouvées dans les deux cas. Cet indicateur montre l'impact des limite transporteur sur les coûts de transport.

Lorsque les coûts fixes des véhicules sont élevés (flottes de type A), la majorité des solutions optimales n'utilisent pas la totalité des flottes des transporteurs, la limite transporteur a donc un impact modéré. Les solutions du VRP-FLC expédient plus de commandes en LTL mais ce coût est amorti par les coûts fixes élevés des véhicules qui ne sont plus utilisés.

Quand les coûts fixes des véhicules sont faibles (flottes de type C), la flotte est entièrement utilisée et la majorité des coûts sont des coûts variables, la limite transporteur a alors beaucoup plus d'impact.

Cette limite a peu d'impact sur les instances clusterisées et même aucun lorsque les coûts LTL sont faibles puisque la totalité des marchandises est expédiée en LTL.

4.5 Cas d'étude : combinaison d'expéditions FTL et LTL

Dans cette section, nous étudions les potentielles économies que peut obtenir un chargeur en combinant des modes d'expédition FTL et LTL en comparant cette stratégie à des approches purement LTL ou FTL

4.5.1 Description

Nous avons généré un ensemble d'instances sur la base de l'expérience de CRC Services pour refléter les situations typiques rencontrées par leurs clients. Nous considérons la répartition d'un chargeur sur un territoire de 18000 km² en France. Les instances contiennent 60 commandes par jour en moyenne, réparties sur plus de 25 site clients. La demande totale journalière représente environ 300 palettes par jour. Cette quantité correspond à l'utilisation d'un peu moins de neuf camions complets, mais plus de véhicules sont nécessaires pour livrer les marchandises, car la combinaisons de fenêtres de temps

serrées et de différentes dates de disponibilité ne permet pas de livrer efficacement les commandes.

Dans ces instances nous considérons un unique transporteur pour lequel nous faisons varier la limite transporteur. Nous utilisons des tarifs LTL et FTL réalistes et deux types de véhicules de capacité 33 et 19 palettes.

Le coût d'une tournée FTL est définie par le modèle suivant : (i) une tournée peut servir au plus trois clients, (ii) un coût de livraison FTL est attribué à chaque client, (iii) le coût d'une tournée est égal au plus grand coût de livraison parmi les clients visités, plus un coût *d'ouverture de porte* pour chaque arrêt supplémentaire dans la tournée, (iv) la longueur d'une tournée est égale à la distance parcourue depuis le dépôt jusqu'au dernier client visité (on considère des tournées ouvertes), (v) les tournées sont soumises à un détour maximal (différence entre la longueur de la tournée et la distance du dépôt au dernier client livré) de 50 km.

Le tarif LTL est une fonction croissante du nombre de palettes à expédier et définie pour chaque client.

Les distances et durées de trajet entre chaque paire de sites ont été obtenues avec OSRM (Open Source Routing Machine, Luxen et Vetter (2011)) avec un profil « camion » pour les véhicules.

4.5.2 Expérimentations

Nous comparons trois stratégies différentes : expédier l'ensemble des commandes en LTL, en FTL, ou en autorisant une combinaison des deux modes de transport. Les résultats sont synthétisés par la Figure 4.5.

Les meilleures solutions obtenues avec une stratégie mixte sur les instances sans limiter la flotte emploient jusqu'à 13 véhicules. Nous résolvons ensuite les mêmes instances en considérant une limite transporteur variant de 1 à 13 véhicules. Cette limite est indiquée sur l'axe horizontal. Nous considérons également deux scénarios extrêmes : à gauche l'expédition de toutes les commandes en LTL et à droite l'expédition de toutes les commandes en FTL.

L'axe vertical représente les coûts de distribution. Le coût avec une limite transporteur fixée à 13 est utilisé comme coût de référence (100%) et tous les autres coûts sont exprimés comme des pourcentages par rapport au coût de référence.

L'expédition de toutes les commandes en LTL (à gauche) entraîne une hausse de 51% des coûts d'expédition. L'expédition de toutes les commandes en FTL (à droite) force

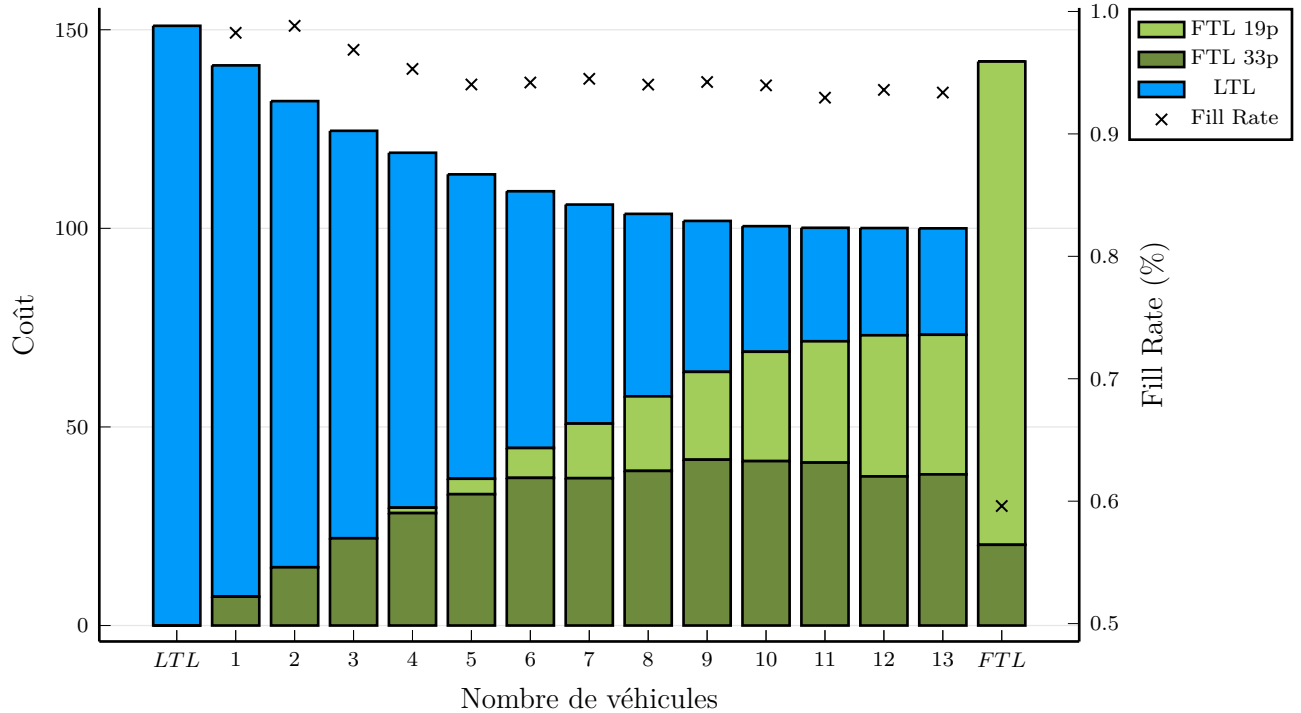


FIGURE 4.5 – Évolution du coût total et du taux de remplissage des tournées FTL selon la stratégie employée et la limite transporteur. Le coût est exprimé comme un pourcentage par rapport au scénario de coût minimal (avec une limite de 13 véhicules).

l'utilisation d'un grand nombre de véhicules afin de respecter les contraintes de fenêtres de temps et de détour maximal. Avec cette stratégie, entre 20 et 24 véhicules sont utilisés chaque jour, dont 88% sont de capacité 19 palettes. Cette stratégie entraîne une hausse de 42% des coûts d'expédition et un mauvais taux de remplissage des véhicules (60%).

Lorsque l'on autorise la combinaison d'expédition en LTL et en FTL sans limiter le nombre de véhicules, le nombre de véhicules utilisé pour les tournées FTL tombe à 11.5 en moyenne, le pourcentage de véhicules 19 palettes descend à 66% et le taux de remplissage moyen des véhicules FTL atteint 94%.

Pour chaque scénario, la partie bleue représente les coûts d'expédition LTL, celle en vert foncé représente les coûts FTL avec des véhicules 33 palettes et celle en vert clair les coûts FTL avec des véhicules 19 palettes.

Sur des données réalistes, intégrant des fenêtres de temps et des coûts et contraintes classiques d'un transport sous-traité, ces expérimentations montrent que la combinaison

d'expéditions en FTL et LTL est critique en terme de minimisation des coûts. On observe une certaine robustesse du coût vis à vis du mix optimal entre les deux modes de transport : faire varier la limite entre 10 et 13 véhicules a peu d'impact sur le coût total des solutions. En dessous de 10 véhicules, le pourcentage d'utilisation de véhicules 19 palettes est drastiquement réduit et le chargeur doit utiliser des expéditions en LTL beaucoup plus chères.

4.6 Conclusions

Dans ce chapitre, nous avons étudié une extension d'un problème de tournées de véhicules avec flotte hétérogène que l'on retrouve dans les réseaux de distribution, lorsque les chargeurs travaillent avec différents transporteurs FTL ou LTL. Nous avons défini le VRP-FLC et l'avons formulé comme une variante du VRPPC dans laquelle la flotte privée est composée de flottes hétérogènes appartenant à plusieurs transporteurs FTL. À chaque transporteur est associée une limite sur le nombre total de véhicules utilisés.

Pour résoudre le VRP-FLC, nous avons proposé une matheuristique basée sur la combinaison d'une métaheuristique de type LNS avec la résolution d'un problème de partitionnement. Afin de gérer les contraintes liées aux véhicules et aux transporteurs, un point clé de cet algorithme était de maintenir la liste des types de véhicules compatibles avec chaque route de la solution courante pendant la phase de reconstruction du LNS. Les expériences numériques montrent que notre algorithme produit de bons résultats sur plusieurs variantes du VRP avec une flotte hétérogène. Il montre que la combinaison des expéditions FTL et LTL dans un plan de distribution surpasse clairement les stratégies de transport purement FTL ou LTL.

Notre approche est également capable d'évaluer l'impact de la contrainte de limite du transporteur et peut donc aider les expéditeurs à se mettre d'accord avec les transporteurs sur le bon nombre de camions à affréter.

Ces travaux peuvent être étendus de plusieurs façons. Tout d'abord, une extension naturelle serait de considérer plusieurs transporteurs LTL et d'intégrer des tournées LTL dans la limite des transporteurs.

Deuxièmement, si on considère un cas comportant de multiples commodités, livrer plusieurs commandes vers un même client en LTL permet d'obtenir un tarif moindre que la somme des coûts LTL pour chaque commande, à condition que les fenêtres de temps de ces commandes soient compatibles. Ce coût pourrait être calculé par le biais de la fonction

« boîte noire » en modélisant l'expédition LTL par des tournées limitées à un point.

Une autre extension réaliste concerne un réseau logistique avec de multiples expéditeurs et clients, avec la possibilité de livrer les commandes à des plates-formes intermédiaires où le flux entrant provenant de chaque dépôt d'expéditeur pourrait être consolidé. Cette organisation introduit la possibilité d'expédier les commandes en deux étapes successives qui peuvent être soit en FTL, soit en LTL.

Dans un plan de transport combinant expédition LTL et FTL, l'expédition en LTL shipment peut servir à déléguer la livraison de commandes « gênantes », par exemple vers des clients éloignés ou avec des fenêtres de temps ne permettant pas de consolider les flux avec d'autres clients. Il serait instructif de connaître les types de clients ou commandes typiquement livrés en FTL ou LTL.

PROBLÈME DE COLLECTES ET LIVRAISONS AVEC PLATEFORMES INTERMÉDIAIRES

5.1	Modélisation du problème de collectes et livraisons avec plateformes inter- médiaires	80
5.1.1	Coûts et réalisabilité des tournées	82
5.1.2	Modélisation mathématique du PDPTF avec options	83
5.2	Small and Large Neighborhood Search	85
5.2.1	Opérateurs du SLNS	87
5.2.2	Évaluation des options	90
5.2.3	Problème de partitionnement d'ensemble	91
5.3	Expérimentations numériques	92
5.3.1	Vehicle Routing Problem with Time Windows and Transshipment Facilities	93
5.3.2	Generalized Vehicle Routing Problem with Time Windows	95
5.3.3	Vehicle Routing Problem with Multiple Time Windows	97
5.3.4	Commodity-Constrained Split-Delivery Vehicle Routing Problem	98
5.3.5	Discussion	100
5.4	Conclusion	100

Dans ce chapitre, nous étudions un problème de collectes et livraisons dans un réseau de grande distribution comprenant de multiples sites d'expédition et des plateformes intermédiaires. Comme présenté en section 3.2, ce cas représente une stratégie de sous-traitance dans laquelle la livraison des derniers kilomètres peut être effectuée par des transporteurs régionaux aussi appelés dégroupes. Ces derniers sont souvent spécialisés dans la distribution sur leur territoire et sont à même de proposer des tarifs attractifs

pour la livraison des derniers kilomètres. Dans cette thèse, nous nous intéressons particulièrement à réaliser la sous-traitance à un transporteur régional lorsqu'elle est utile. On se garde donc la possibilité de livrer des clients en direct au sein des tournées.

Dans ce contexte, nous considérons un ensemble de commandes de clients qui peuvent être livrées depuis différents sites d'expédition en fonction des produits demandés. Pour une même commande, il est possible de choisir parmi différents sites d'expédition.

L'originalité première du problème provient de l'utilisation de plateformes intermédiaires qui représentent les distributeurs régionaux. On considère alors qu'une commande client peut être livrée soit directement chez le client depuis le site d'expédition sélectionné, soit depuis une plateforme intermédiaire. Dans ce dernier cas, la commande devra arriver à temps à la plateforme pour pouvoir ensuite arriver chez le client pour sa date finale de livraison. Le transport depuis les points de collecte est sous-traité à des transporteurs proposant diverses structures de coût, FTL ou LTL.

La conception d'un plan de transport implique de décider pour chaque commande : le site chargeur d'où partira la commande, le site où elle sera livrée (au client ou sur une plateforme intermédiaire) et de définir un ensemble de tournées minimisant les coûts de transport et les coûts de distribution depuis les plateformes. Pour chaque commande, nous modélisons le choix du site de départ et le passage ou non par une plateforme intermédiaire comme différentes *options*.

La section 5.1 décrit cette modélisation et introduit le problème de collectes et livraisons avec plateformes Intermédiaires (PDPTF : Pickup and Delivery Problem with Transshipment Facilities). La section 5.2 décrit la méthode de résolution implémentée pour résoudre ce problème. Des expérimentations numériques permettant de valider la performance de la méthode sur des instances proches de la littérature sont proposées en section 5.3.

5.1 Modélisation du problème de collectes et livraisons avec plateformes intermédiaires

Le problème étudié est défini dans un réseau comprenant quatre types de sites : un dépôt de véhicules, un ensemble \mathcal{P} de sites chargeurs (pickup points), un ensemble \mathcal{H} de plateformes (hubs) et un ensemble \mathcal{D} de clients (deliveries).

Soit \mathcal{N} un ensemble de commandes devant être livrées sur un horizon de temps donné pouvant aller de un à cinq jours. Chaque commande $i \in \mathcal{N}$ est caractérisée par un nombre

de palettes et un poids, et doit être acheminée depuis un point de collecte vers sa destination $d_i \in \mathcal{D}$. Selon la nature des produits demandés et une éventuelle affectation des sites clients aux sites chargeurs, plusieurs points de collecte \mathcal{P}^i sont possibles pour une même commande. Ces collectes doivent se faire dans l'une des multiples fenêtres de temps du site chargeur et peuvent tenir compte d'une date de disponibilité pour la commande.

Après avoir collecté les marchandises sur un site chargeur, la commande peut être livrée de deux manières.

- *En livraison directe* : La commande est livrée sans rupture de charge sur le site du client d_i , parmi un ensemble de fenêtres horaires proposées par le client.
- *Via une plateforme* : La commande est déposée sur une des plateformes du réseau et un transport régional se charge de la distribution de la commande au client en échange d'une rémunération. Le prix demandé par le transporteur régional dépend de la distance séparant la plateforme du client et de la quantité de marchandises à livrer. Il peut aussi inclure des coûts de stockage, si la commande est livrée à la plateforme avec plusieurs jours d'avance. La livraison sur la plateforme doit se faire dans l'une des fenêtres horaires de la plateforme et suffisamment à l'avance pour que le client puisse ensuite être livré à temps.

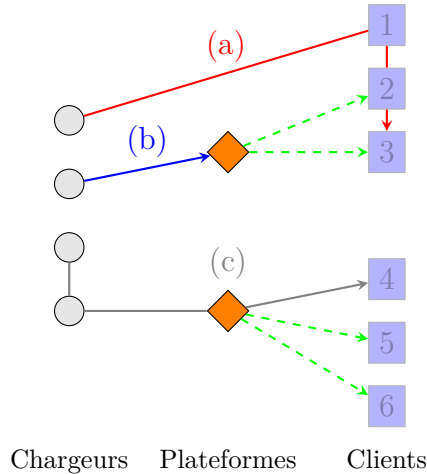


FIGURE 5.1 – Différents schémas de tournée possibles. En rouge (a), une tournée directe livrant les clients 1, 2 et 3 directement depuis un site chargeur. En bleu (b), les commandes des clients 2 et 3 sont acheminées jusqu'à une plateforme. La distribution des commandes aux clients 2 et 3 depuis cette plateforme est assurée par un transporteur régional. En gris (c), les commandes des clients 4, 5 et 6 sont collectées sur deux sites chargeurs. Les commandes des clients 5 et 6 sont déposées sur une plateforme et le client 4 est directement livré sans rupture de charge.

La Figure 5.1 illustre différents types de tournées possibles, comprenant la possibilité de collecter des marchandises sur plusieurs sites chargeurs et de livrer une partie des commandes en livraison directe et une autre partie via plateforme dans la même tournée. On note que la livraison d'un client avant une plateforme ou la livraison de plusieurs plateformes dans une même tournée ne sont pas représentées mais sont autorisées.

5.1.1 Coûts et réalisabilité des tournées

En plus de choisir le site de départ et livraison pour l'ensemble de ses commandes, le chargeur doit déterminer son plan de transport, c'est-à-dire un ensemble de tournées de véhicules. Nous considérons des tournées de collectes et livraisons où toutes les collectes sont effectuées avant les livraisons. Les livraisons, sur les plateformes et les sites clients, sont effectuées depuis le dernier site de collecte visité.

Chaque tournée doit respecter un certain nombre de contraintes horaires : fenêtres de temps pour l'expédition et date de disponibilité des marchandises sur les points de collecte, et fenêtres de temps pour la réception sur les plateforme logistique, ou pour la livraison chez les clients. De plus, on considère des temps de chargement et de déchargement dépendant du nombre de palettes chargées/déchargées et des temps fixes d'accès aux différents sites visités.

Nous considérons des flottes hétérogènes de véhicules. Les véhicules de ces flottes partageant souvent des caractéristiques identiques (capacité, contraintes, prix...), on les regroupe par type. Une flotte est donc composée de plusieurs types de véhicules. On suppose que chaque type de véhicule est disponible en une quantité infinie.

Chaque type de véhicule a ses propres contraintes sur les tournées qu'il peut réaliser, ces contraintes incluent une capacité maximale en palettes et/ou en poids, un nombre d'arrêts maximum, une durée de tournée maximum, une distance maximale par tournée, des sites non-livrables, un détour maximum. Certains types de véhicules effectuent des tournées ouvertes, d'autres des tournées fermées.

Pour un type de véhicule donné, le coût d'une tournée est une fonction boîte noire dépendant de la tournée (durée, distance, sites visités) et des commandes livrées (nombre de palettes). Contrairement à ce qui a été présenté au Chapitre 4 pour le VRP-FLC, l'expédition en LTL n'est pas modélisée comme un profit mais comme un type de véhicule avec un tarif dépendant du nombre de palettes ou du poids livré, et contraint à ne visiter qu'un seul site chargeur et à ne livrer qu'un seul site. Cette approche permet de modéliser plus finement les coûts d'expédition LTL lorsque plusieurs commandes sont livrées sur

une même plateforme ou dans le cas où l'on considère de multiples commodités livrées à un client dans une même tournée.

5.1.2 Modélisation mathématique du PDPTF avec options

Nous proposons une modélisation du PDPTF dans laquelle chaque commande $i \in \mathcal{N}$ est associée à un ensemble d'options \mathcal{O}^i . Une option o est définie par :

- Un site chargeur $p_o \in \mathcal{P}$ et une fenêtre de temps $[a_{p_o}, b_{p_o}]$ dans laquelle la commande pourra être collectée.
- Un magasin ou une plateforme $d_o \in \mathcal{D} \cup \mathcal{H}$ et une fenêtre de temps $[a_{d_o}, b_{d_o}]$ dans laquelle la commande devra être livrée.
- Un coût d'option c^o .

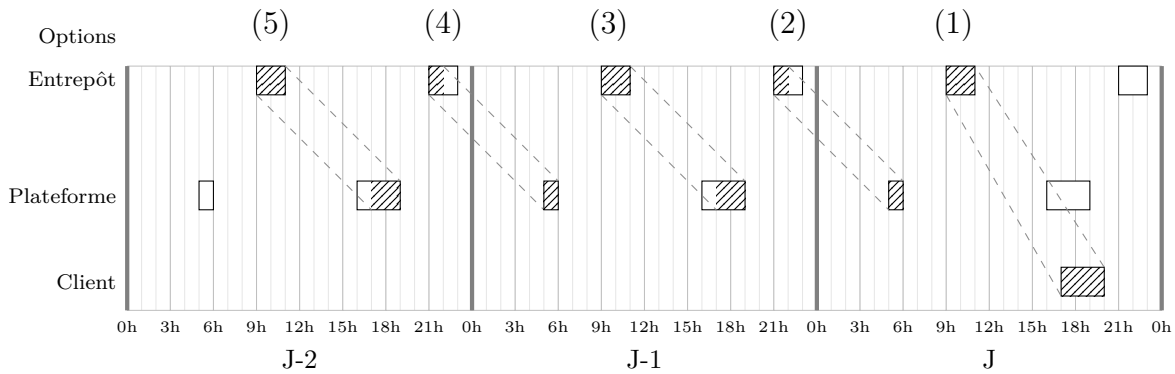
Choisir une option pour chaque commande revient donc à décider : du site chargeur d'où partiront les marchandises, du choix du mode de livraison, direct ou via plateforme, et permet de prendre en compte des fenêtres de temps multiples, sur les sites chargeurs et le site livré. Le coût associé à l'option est nul si elle a pour destination le site du client et est égal au coût de stockage et de distribution si l'option a pour destination une plateforme.

Exemple 3 *On considère une commande i devant être livrée à sa destination d_i entre 18h et 20h le jour J . La commande peut être livrée en direct ou via une plateforme proposant deux fenêtres de livraison par jour : le matin entre 5h et 6h ou le soir entre 16h et 19h. Le site chargeur dispose de deux fenêtres horaires pour la collecte, le matin entre 9h00 et 11h00 ou le soir entre 21h et 23h. On suppose une durée de tournée maximale de 12h.*

La Figure 5.2 et 5.2b illustre les différentes options associées à la commande i .

La Figure 5.2a présente en abscisse un horizon de temps continu sur trois jours, du jour $J-2$ jusqu'au jour J où le client attend la livraison de sa commande. Les sites concernés par la commande sont présentés en ordonnée. En face de chaque site, les fenêtres de temps d'ouverture du site sont indiquées par un rectangle. Les pointillés indiquent les différents itinéraires possibles pour la commande et correspondent aux différentes options numérotées (1) à (5). Un pré-traitement permet d'ajuster les heures de départ au plus tard ou heures d'arrivées au plus tôt pour les options. Les fenêtres horaires associées aux options, représentées par des zones hachurées, peuvent donc être différentes des fenêtres horaires d'ouverture des sites.

L'option 1 consiste à livrer la marchandise chez le client ; le coût de cette option est



(a) Différentes options en fonction des fenêtres de temps sur le site de collecte, la plateforme et le client

Option	Collecte		Livraison			Coût d'option
	Site	Fenêtre	Site	Fenêtre		
1	Entrepôt	[9h-11h] J	Client	[17h-20h] J	0	
2	Entrepôt	[21h-22h] J-1	Plateforme	[5h-6h] J	150	
3	Entrepôt	[8h-11h] J-1	Plateforme	[16h-19h] J-1	200	
4	Entrepôt	[21h-22h] J-2	Plateforme	[5h-6h] J-1	200	
5	Entrepôt	[8h-11h] J-2	Plateforme	[16h-19h] J-2	250	

(b) Options créées pour chaque itinéraire et paire de fenêtres de temps compatible

FIGURE 5.2 – Options d'une commande selon différents itinéraires et différentes fenêtres de temps

donc nul. L'option 2 consiste à livrer le client par une plateforme, en livrant la plateforme plus tôt dans la journée. Le coût associé à cette option est le coût de distribution proposé par le transporteur régional pour cette commande. L'heure de départ au plus tard pour cette commande est fixée à 22h afin de respecter l'heure d'arrivée au plus tard sur la plateforme.

Les options 3, 4 et 5 consistent à livrer la plateforme 1 à 2 jours en avance. Le coût de ces options incluent les coûts de distribution et des coûts de stockage. On suppose ici des coûts de stockage pour la plateforme égaux à 50 par jour.

Le tableau 5.2b résume les différentes options créées pour cette commande et indique pour chaque option le site de collecte, de livraison, les fenêtres horaires pour la collecte et la livraison et le coût de l'option.

5.2 Small and Large Neighborhood Search

Cette section décrit la méta-heuristique Small And Large Neighborhood Search (SLNS) que nous proposons pour résoudre le PDPTF.

Le SLNS est une extension de la recherche à voisinage large introduite par Dumez et al. (2021a) et sous le nom SLNS par Le Colleter et al. (2022).

De la même manière que dans un LNS, l'exploration de l'espace des solutions repose sur l'utilisation de différents opérateurs de retrait (destruction) et d'insertion (reconstruction).

Les opérateurs de retrait, présentés en sous-section 5.2.1.2 retirent une partie des commandes d'une solution et les placent dans une banque de requêtes. Les opérateurs de réparation sont des heuristiques qui réinsèrent les commandes de la banque de requêtes dans la solution afin d'obtenir une nouvelle solution réalisable.

La particularité du SLNS est de combiner deux tailles de destruction différentes. Des destructions de petite taille, permettant d'effectuer rapidement un grand nombre d'itérations et de compenser l'absence de recherche locale, sont combinées avec des destructions plus larges dont l'objectif est de diversifier la recherche et d'échapper au minima locaux. En fonction de la taille de la destruction, différents opérateurs de réparation sont utilisés. Ces opérateurs sont présentés en sous-section 5.2.1.1.

Une autre caractéristique de la méthode proposée est l'attribution de scores aux différentes options d'une commande en fonction de leur utilité. Le calcul de ces scores et leur utilisation dans les opérateurs de réparation sont présentés en sous-section 5.2.2.

Le SLNS proposé est combiné avec la résolution à intervalles réguliers d'un problème de partitionnement d'ensemble (SPP) sur les tournées obtenues au cours de la résolution. Ce composant est décrit en sous-section 5.2.3.

La méthode SLNS proposée est décrite par l'Algorithme 2.

Une solution initiale s est obtenue avec l'opérateur d'insertion 2-regret appliqué à une solution vide. La meilleure solution s^* est initialisée par une copie de la solution initiale (ligne 2). La banque de tournées Ω utilisée pour résoudre le SPP est initialisée avec les tournées de la solution s (ligne 1). Le compteur μ utilisé pour déterminer s'il faut déclencher une destruction large est initialisé à 0 (ligne 3).

La boucle principale (lignes 4-32) est exécutée jusqu'à ce que le critère d'arrêt soit satisfait. Ce critère peut être un nombre d'itérations maximal, un temps de calcul donné, ou un nombre maximal d'itérations sans amélioration de la meilleure solution. À chaque itération, un opérateur de retrait est choisi avec une roulette biaisée parmi un ensemble

Algorithme 2 : Algorithme Small & Large Neighborhood Search

Paramètres : $\Sigma^-, \Sigma_{small}^+, \Sigma_{large}^+$

Entrées : s

```

1:  $s^* \leftarrow s$ 
2:  $\Omega \leftarrow \text{routes}(s)$ 
3:  $\mu \leftarrow 0$ 
4: tant que critère d'arrêt non vérifié faire
5:    $\sigma^- \leftarrow \text{ChoisirOperateurDestruction}(\Sigma^-)$ 
6:   si  $\mu < \omega$  alors
7:      $\sigma^+ \leftarrow \text{ChoisirOperateurReparation}(\Sigma_{small}^+)$ 
8:      $\phi \leftarrow U(\phi_{min}^{small}, \phi_{max}^{small})$ 
9:   sinon
10:     $\sigma^+ \leftarrow \text{ChoisirOperateurReparation}(\Sigma_{large}^+)$ 
11:     $\phi \leftarrow U(\phi_{large}^{min}, \phi_{large}^{max})$ 
12:   fin
13:    $s' \leftarrow \sigma^+(\sigma^-(s, \phi))$ 
14:   si CritèreAcceptation( $z(s'), s^*$ ) alors
15:     si  $\mu > \omega$  alors
16:        $\mu \leftarrow 0$ 
17:     fin
18:      $s \leftarrow s'$ 
19:      $\Omega \leftarrow \Omega \cup \text{routes}(s)$ 
20:     si  $z(s) < z(s^*)$  alors
21:        $s^* \leftarrow s$ 
22:        $\mu \leftarrow 0$ 
23:     sinon
24:        $\mu \leftarrow \mu + 1$ 
25:     fin
26:   fin
27:   si  $|\Omega| \geq \eta$  alors
28:      $s^* \leftarrow SPP(\Omega)$ 
29:      $\Omega \leftarrow \text{routes}(s^*)$ 
30:      $s \leftarrow s^*$ 
31:   fin
32: fin
33: retourner  $s^*$ 

```

d'opérateurs Σ^- (ligne 5). La taille de la destruction et l'opérateur d'insertion utilisé dépendent du compteur μ et du paramètre ω . Si μ est inférieur à ω , une petite taille de destruction est choisie dans l'intervalle $[\phi_{min}^{small}, \phi_{max}^{small}]$ et un opérateur de réparation est choisi parmi l'ensemble Σ_{small}^+ (lignes 7 et 8). Si μ est supérieur à ω , une grande taille de destruction est choisie dans l'intervalle $[\phi_{min}^{large}, \phi_{max}^{large}]$ et un opérateur de réparation est choisi parmi l'ensemble Σ_{large}^+ (lignes 10 et 11).

L'ensemble d'opérateurs de retrait utilisé reste identique pour les petites destructions comme pour les larges.

Une nouvelle solution s' est obtenue en appliquant successivement les opérateurs de retrait et d'insertion (ligne 13). Elle est acceptée comme nouvelle solution courante s si elle passe un critère d'acceptation. Le critère utilisé est le critère Record-To-Record (Dueck 1993). Il consiste à accepter la solution si son coût ne dégrade pas le coût de la meilleure solution s^* de plus de $\mathcal{T}\%$. Lorsqu'une solution est acceptée, les tournées qui la constituent sont ajoutées à la banque de tournées Ω . La meilleure solution connue est mise à jour si besoin en ligne 20.

Le compteur μ est réinitialisé si la solution obtenue est une nouvelle meilleure solution (22), ou si une destruction large a permis de trouver une nouvelle solution s' qui passe le critère d'acceptation (16). Ainsi, les destructions larges sont répétées jusqu'à obtenir une nouvelle solution de bonne qualité.

Un problème de partitionnement d'ensemble, détaillé en sous-section 5.2.3, est résolu lorsque la taille de la banque de tournées Ω dépasse un seuil η (lignes 27 - 31).

5.2.1 Opérateurs du SLNS

Cette section décrit les différents opérateurs de destruction et de réparation utilisés dans le SLNS. Dans les problèmes traités en pratique le nombre de points de collecte est grandement inférieur au nombre de clients et de plateformes : plusieurs commandes sont typiquement collectées sur le même site et les temps de chargement et d'accès sur site restreignent le nombre de points de collecte visités. Par conséquent, les opérateurs utilisés se focalisent principalement sur les points de livraison, plateformes ou clients, et non sur les points de collecte.

5.2.1.1 Opérateurs de destruction

Les opérateurs de destruction sont paramétrés par une taille de destruction ϕ et enlèvent $[\phi \times |\mathcal{N}|]$ commandes de la solution pour les placer dans la banque de requêtes. À chaque itération, la taille de destruction ϕ est tirée aléatoirement dans un intervalle $[\phi_{min}^{small}, \phi_{max}^{small}]$ pour les petites destructions et $[\phi_{min}^{large}, \phi_{max}^{large}]$ pour les grandes destructions.

- *Random Removal* (Ropke et Pisinger 2006a) : suppression aléatoire de commandes.
- *Worst-Time Removal* (Demir et al. 2012) : Suppression des commandes pour lesquelles le véhicule arrive avec le plus d’avance par rapport à la fenêtre de temps de l’option livrée.
- *Random Removal in Most Expensive Route* : Cet opérateur consiste à supprimer aléatoirement une commande dans la tournée la plus chère de la solution. Après chaque suppression, le coût de la tournée affectée est recalculé.
- *Random Removal in Cheapest Route* : Cet opérateur consiste à supprimer des commandes dans la tournée la moins chère. Si la tournée ne contient pas plus de commandes qu’il ne reste de commandes à enlever, elle est détruite entièrement, sinon, une partie des commandes est retirée aléatoirement de la tournée.
- *Historical Node-Pair Removal* (Pisinger et Ropke 2007) : Cet opérateur consiste à supprimer des commandes qui étaient mieux placées dans des solutions précédemment trouvées au cours de la résolution. On maintient pour cela un score $g(o_1, o_2)$ pour chaque paire d’options $(o_1, o_2) \in \mathcal{O}^2$. Le score $g(o_1, o_2)$ est égal à la valeur de la fonction objectif de la meilleure solution trouvée au cours de la résolution dans laquelle la livraison de l’option o_1 précède la livraison de l’option o_2 . Pour une solution donnée, le score d’une commande livrée via une option o est égal à la somme des scores $g(o_{prec}, o)$ et $g(o, o_{suiv})$ où o_{prec} et o_{suiv} sont les options livrées respectivement avant et après l’option o dans la tournée. Les commandes sont triées selon leur score et supprimées itérativement en privilégiant les commandes ayant le plus grand score. L’indice de la commande à enlever est déterminé par $U([0, 1])^{0.3 \times |K|}$.
- (Split) String removal : Proposé par Christiaens et Vanden Berghe (2020), cet opérateur consiste à supprimer des chaînes de livraisons consécutives dans plusieurs tournées proches. Pour chaque tournée, un tirage aléatoire sélectionne tout d’abord une chaîne de livraisons. Ensuite, soit cette chaîne de livraison est supprimée (*string removal*), soit la chaîne est conservée et le reste de la tournée supprimée (*split string removal*). L’opérateur a été implémenté avec les paramètres suggérés par les auteurs et en ne considérant que les livraisons de la tournée et pas les visites

correspondant aux points de collecte.

5.2.1.2 Heuristiques de réparation

Nous proposons d'utiliser deux ensembles d'heuristiques de réparation différents en fonction de la taille de la destruction effectuée. De la même manière qu'au chapitre précédent, nous maintenons une liste des différents types de véhicules compatibles avec chaque tournée. Le calcul du coût d'insertion d'une option dans une tournée prend en compte l'éventuel changement de véhicule nécessaire si le véhicule le moins cher n'est plus compatible avec la tournée après insertion.

Les petites destructions sont combinées avec des heuristiques de réparation rapides, reprises ou inspirées des heuristiques proposées par Christiaens et Vanden Berghe (2020). Elles consistent à trier la banque de requêtes selon un critère, puis à réinsérer chaque commande selon cet ordre. Elles s'apparentent donc à des heuristiques de liste. Christiaens et Vanden Berghe (2020) montrent que ces heuristiques permettent d'effectuer un grand nombre d'itérations en un temps de calcul très faible, et ainsi obtenir de bons résultats sans ajouter de recherche locale au LNS. Le principe général d'une insertion est le suivant : pour la commande sélectionnée, on évalue le coût et la réalisabilité de l'ensemble des positions d'insertion de chacune de ses options dans les tournées de la solution courante. L'option réalisable ayant le meilleur coût est insérée à sa meilleure position. Afin de réduire le nombre d'options testées à l'insertion d'une commande, nous proposons un mécanisme reposant sur l'attribution d'un score aux options. Ce mécanisme est détaillé en sous-section 5.2.2.

- Random List Insertion : Les commandes sont réinsérées dans un ordre aléatoire.
- Request Size List Insertion : Les commandes les plus grandes (en nombre de palettes) sont insérées en premier.
- Farthest List Insertion : Les commandes sont triées en fonction de la distance qui sépare le client du site chargeur. Les commandes les plus éloignées sont insérées en premier.
- Hardest Requests List Insertion : Les commandes les plus dures à insérer sont insérées en premier. On maintient pour cela un compteur, pour chaque commande, du nombre de fois où aucune insertion n'a pas été réalisable au cours de la résolution.

Pour chacune de ces heuristiques, le coût d'insertion d'une option à une position donnée a une probabilité p^{blink} de ne pas être considérée.

Les destructions larges sont combinées avec des heuristiques donnant des solutions de

meilleure qualité mais au prix d'un temps de calcul plus long.

- Meilleure insertion : Le coût de l'insertion générant la plus petite augmentation de coût est calculé pour chaque commande. Les commandes les moins chères à insérer sont insérées en premier.
- 2-regret Insertion : On définit le regret d'une commande comme la différence entre le coût de son insertion dans la meilleure tournée, et le coût de son insertion dans la seconde meilleure. Les commandes avec le plus grand regret sont insérées en premier.

Afin de ne pas recalculer tous les coûts d'insertion à chaque itération de ces heuristiques, une liste contenant le coût d'insertion de chaque commande dans chaque tournée est maintenue. Après l'insertion d'une commande dans une tournée, seuls les coûts d'insertion des autres commandes dans cette tournée sont recalculés.

5.2.2 Évaluation des options

La réinsertion d'une commande repose sur l'évaluation de chacune de ses options. Savoir quelles options sont susceptibles d'apparaître ou non dans une solution optimale permet de réparer une solution plus rapidement en ne considérant que les options pertinentes pour chaque commande. Nous proposons pour cela d'attribuer un poids aux options au cours de la résolution. Les poids des options sont normalisés entre 0 et 1 et sont tous initialisés à 1 au début de l'algorithme.

Soit γ un paramètre compris entre 0 et 1. Lorsque l'on évalue la réinsertion d'une commande dans un opérateur de réparation, un nombre aléatoire est généré. Si celui-ci est inférieur à γ , seules les options ayant un poids supérieur à 0.20 sont considérées par l'opérateur de réparation. Dans le cas contraire, toutes les options sont considérées.

Mise à jour des poids

L'ajustement automatique du poids des options se fait via un mécanisme adaptatif similaire à celui proposé par Ropke et Pisinger (2006b) pour ajuster le poids des opérateurs dans un ALNS.

Des scores sont attribués aux options au cours de séquences de μ itérations consécutives du SLNS que l'on appelle des segments. Au début de chaque segment, le score π_o de chaque option $o \in \mathcal{O}$ est initialisé à zéro.

Après chaque itération, si la solution obtenue passe le critère d'acceptation, le score de

toutes les options sélectionnées dans la solution est incrémenté de σ_1, σ_2 ou σ_3 en fonction de la qualité de la solution, avec $\sigma_1 > \sigma_2 > \sigma_3$.

Les scores sont incrémentés de :

- σ_1 si la solution obtenue améliore la meilleure solution connue ;
- σ_2 si la solution obtenue améliore la solution courante ;
- σ_3 si la solution obtenue n'améliore pas la solution courante mais passe le critère d'acceptation.

À la fin de chaque segment, les poids des options sont mis à jour en fonction de leur performance sur le dernier segment, et de leur poids sur le segment précédent.

On note $w_{o,j}$ le poids d'une option o au cours d'un segment j .

Le poids de l'option o au cours du segment $j + 1$ est déterminé par :

$$w_{o,j+1} = \lambda w_{o,j} + (1 - \lambda) \frac{\pi_o}{\mu},$$

avec $0 < \lambda < 1$.

5.2.3 Problème de partitionnement d'ensemble

Lorsque la banque de tournées Ω contient plus de η tournées, nous résolvons le problème de partitionnement d'ensemble décrit par le modèle (5.1)–(5.3). On note Ω^o l'ensemble des tournées parmi Ω livrant l'option o , et c_r le coût d'une tournée, réalisée par le type de véhicule compatible le moins cher. Le coût d'une tournée c_r inclut le coût des options sélectionnées et livrées par la tournée r . On définit les variables binaires x_r , valant 1 si la tournée $r \in \mathcal{R}$ est effectuée, et 0 sinon.

$$\min \sum_{r \in \mathcal{R}} c_r x_r \tag{5.1}$$

s.c.

$$\sum_{o \in \mathcal{O}^i} \sum_{r \in \mathcal{R}^o} x_r = 1 \quad \forall i \in \mathcal{N} \tag{5.2}$$

$$x_r \in \{0, 1\} \quad \forall r \in \mathcal{R} \tag{5.3}$$

FIGURE 5.3 – Modèle mathématique sur les tournées pour le PDPTF

La fonction objectif (5.1) minimise la somme du coût des tournées. Les contraintes (5.2) assurent que pour chaque commande, une seule option est choisie et est prise en

charge par une unique tournée.

Le modèle est résolu avec une limite de temps fixée à 20 secondes. La banque de tournées Ω est alors vidée et réinitialisée avec les tournées de la nouvelle solution obtenue par la résolution du problème de partitionnement d'ensemble. Tout comme dans le chapitre précédent, nous formulons en pratique le problème comme un problème de couverture d'ensemble et un algorithme glouton répare la solution en éliminant les commandes livrées plusieurs fois.

Le seuil η pour déclencher la résolution du SPP est ajusté dynamiquement au cours de la résolution. Si la solution obtenue n'améliore pas la meilleure solution connue, η est incrémenté de 20%. Si le modèle n'a pas pu être résolu dans la limite de temps imparti, η est décrémenté de 20%.

5.3 Expérimentations numériques

Afin de valider la performance de la méthode développée, nous avons effectué des expérimentations numériques sur des instances de la littérature pour des problèmes proches du PDPTF. La sous-section 5.3.1 décrit les expérimentations numériques réalisées sur le Vehicle Routing Problem with Time Windows and Transshipment Facilities avec flotte homogène (VRPTWTF) et avec flotte hétérogène (FSMTWTF).

Des résultats sur les problèmes de tournées de véhicules généralisé avec fenêtres de temps (Generalized Vehicle Routing Problem with Time Windows, GVRPTW) et sur le Commodity-Constrained Split Delivery Vehicle Routing Problem (C-SDVRP) sont présentés en sous-section 5.3.2 et 5.3.4.

Le Tableau 5.1 décrit les différents paramètres de la méthode SLNS utilisés en fonction du type d'instance considéré. Ces paramètres ont été obtenus à l'aide du package *irace* (López-Ibáñez et al. 2016) et ajustés manuellement pour homogénéiser certains paramètres sur l'ensemble des instances considérées.

Paramètre	VRPTWTF	FSMTWTF	GVRPTW	VRPMTW	C-SDVRP
p^{blink}	0.02	0.02	0.03	0.01	0.02
$\phi_{min}^{small}, \phi_{max}^{small}$	(0.15, 0.20)	(0.15, 0.25)	(0.10, 0.30)	(0.10, 0.20)	(0.05, 0.15)
\mathcal{T}	0.05	0.01	0.05	0.01	0.01

TABLEAU 5.1 – Paramètres du SLNS spécifiques aux variantes traitées

Les paramètres utilisés quelque soit le type d'instance considéré et les poids utilisés

pour déterminer la probabilité d'utilisation des opérateurs de retrait et d'insertion sont présentés par le Tableau 5.2.

Paramètre généraux	$\phi_{min}^{large}, \phi_{max}^{large}$	(0.30, 0.50)
	ω	1000
	η	10000
Évaluation des options	γ	0.6
	λ	0.33
	$\sigma_1, \sigma_2, \sigma_3$	(3, 2, 1)
Poids des opérateurs de retrait	Random Removal	2
	Greatest Waiting Time Removal	2
	Random Removal in Most Expensive Route	1
	Random Removal in Cheapest Route	1
	Historical Node-Pair Removal	1
	(Split) String Removal	4
Poids des opérateurs d'insertion après une petite destruction	Random List Insertion	3
	Request Size List Insertion	1
	Farthest List Insertion	1
	Hardest Request List Insertion	1
Poids des opérateurs d'insertion après une destruction large	Meilleure insertion	1
	2-regret Insertion	2

TABLEAU 5.2 – Poids des opérateurs de retrait et d'insertion

5.3.1 Vehicle Routing Problem with Time Windows and Transshipment Facilities

Le VRPTWTF, proposé par Friedrich et Elbert (2022), est un cas particulier du PDPTF dans lequel toutes les commandes ont pour origine un unique site chargeur. Les auteurs proposent un ensemble d'instances adaptées des instances de VRPTW de Solomon (1987) en ajoutant cinq plateformes intermédiaires. Les commandes peuvent être livrées chez le client ou déposées sur une plateforme. Le coût de livraison d'une commande si elle est déposée sur une plateforme est déterminé par la distance séparant la plateforme du site client.

Les auteurs proposent également un ensemble d'instances pour la variante du VRPTWTF avec flotte hétérogène (FSMTWTF). La flotte est constituée des types de véhicules tels que proposés par Liu et Shen (1999) en utilisant la définition de la flotte de type A, qui correspond à des coûts fixes élevés pour chaque véhicule.

Les Tableaux 5.3 et 5.4 comparent les résultats de la méthode SLNS (colonnes 5 à

8) aux résultats obtenus par l’ALNS de Friedrich et Elbert (2022) (colonnes 2 à 4). Les algorithmes ALNS et SLNS ont été exécutés 10 fois sur chaque instance. Les Tableaux 5.3 et 5.4 reportent les résultats agrégés sur chaque groupe d’instance (C1, C2, R1, R2, RC1 et RC2). Les instances de type R1 et R2 sont caractérisées par des clients placés aléatoirement. Ils sont clusterisés dans les instances de type C1 et C2. Les instances de type RC1 et RC2 présentent un mix de clients placés aléatoirement et clusterisés. Dans les instances de type C2, R2 et RC2, les capacités des véhicules et la largeur des fenêtres de temps permet de livrer un grand nombre de clients au sein d’une même tournée. Pour chaque groupe d’instance, les colonnes 2 et 5 (\bar{z}) indiquent le coût moyen des meilleures solutions obtenues sur les 10 exécutions. Les colonnes 3 et 6 ($\sum r$) indiquent le nombre total de tournées obtenues et les colonnes 4 et 7 (\bar{t}) indiquent le temps moyen de résolution. Le nombre de véhicules utilisé est donné à titre indicatif mais n’est pas un objectif à minimiser. Si c’était le cas, les solutions optimales consisteraient à livrer la totalité ou presque des commandes sur plateforme. La colonne 8 (gap (%)) indique l’écart relatif entre les solutions obtenues par le SLNS et les solutions obtenues par l’ALNS de Friedrich et Elbert (2022).

Sur les instances avec flotte homogène, le SLNS trouve 14 nouvelles meilleures solutions et 25 solutions identiques à celles rapportées par Friedrich et Elbert (2022). Sur les instances avec flotte hétérogène, le SLNS trouve 2 nouvelles meilleures solutions et 17 solutions identiques. L’écart moyen au global est de 0,28%. Des résultats complets pour le VRPTWTF et le FSMTWTF sont présentés dans les tableaux B.1 et B.2.

Groupe d’instances	ALNS			SLNS			
	\bar{z}	$\sum r$	t	\bar{z}	$\sum r$	t	gap (%)
C1 (9)	828.4	90	84.7	828.4	90	180	0.00
C2 (8)	589.9	24	132.0	589.9	24	180	0.00
R1 (12)	1080.6	143	173.9	1077.1	129	180	-0.34
R2 (11)	930.6	30	312.5	934.0	30	180	0.40
RC1 (8)	1271.4	92	122.0	1272.1	86	180	0.02
RC2 (8)	1081.8	26	247.3	1080.9	26	180	-0.09
Total	967.9	405	183.9	967.8	385	180	-0.01

TABLEAU 5.3 – Comparaison des résultats du SLNS aux résultats de la méthode ALNS proposée par Friedrich et Elbert (2022) sur les instances VRPTWTF

Groupe d'instances	ALNS			SLNS			
	\bar{z}	$\sum r$	t	\bar{z}	$\sum r$	t	gap (%)
C1 (9)	7079.0	171	265.1	7079.0	171	370	0.0
C2 (8)	5686.8	40	396.0	5686.8	40	370	0.0
R1 (12)	4012.8	229	314.0	4017.7	232	370	0.13
R2 (11)	3146.4	55	427.9	3158.1	55	370	0.37
RC1 (8)	4859.9	122	400.2	4910.5	133	370	1.05
RC2 (8)	4209.7	89	430.3	4218.0	91	370	0.20
Total	4723.7	706	369.2	4735.5	722	370	0.28

TABLEAU 5.4 – Comparaison des résultats du SLNS aux résultats de la méthode ALNS proposée par Friedrich et Elbert (2022) sur les instances FSMTWTF

5.3.2 Generalized Vehicle Routing Problem with Time Windows

Le GVRPTW est une généralisation du VRPTW dans lequel chaque client doit être livré sur un site parmi un ensemble de sites possibles. La livraison des clients doit se faire dans une fenêtre horaire donnée pour chaque site. Le GVRPTW peut être vu comme un cas particulier du VRPTWTF dans lequel chaque site représente une option et où toutes les options ont un coût nul.

Nous évaluons la performance du SLNS sur les instances de GVRPTW de Moccia et al. (2012) et la comparons aux résultats obtenus récemment par la méthode CGBH (Column Based Generation Heuristic) de Yuan et al. (2021) et par la méthode SLNS proposée par Dumez et al. (2021b).

Les instances sont nommées $i-n-v_{min}-v_{max}$ où n est le nombre de commandes et v_{min} et v_{max} sont des bornes sur le nombre d'options générées pour chaque commande. Dans ces instances les fenêtres de temps associées à tous les sites d'une même commande sont identiques.

Tout comme Dumez et al. (2021b), nous ajustons les matrices de distances et de durées afin que celles-ci respectent l'inégalité triangulaire. Yuan et al. (2021) ne mentionnent pas avoir appliqué une telle procédure, ce qui explique certaines différences dans les solutions optimales rapportées. De plus, Yuan et al. (2021) transforment la contrainte de durée maximale des tournées en une fenêtre de temps sur le dépôt d'une largeur égale à la durée maximale. Leur problème est donc plus contraint.

Les résultats obtenus sur les instances de GVRPTW sont donnés par le Tableau 5.5. La

Instances	BKS	Yuan et al. (2021) CBGH		Dumez et al. (2021b) SLNS		SLNS		
		z^*	t	z^*	t	z^*	t	gap (%)
i-030-04-08	3497	3498	8.47	3497	5	3497	5	0
i-030-08-12	2796	2797	15.37	2796	5	2796	5	0
i-040-04-08	3811	3811	6.46	3811	8	3811	8	0
i-040-08-12	3759	3768	7.80	3768	8	3759	8	0
i-050-04-08	5439	5439	10.08	5439	11	5439	11	0
i-050-08-12	4034	4054	12.35	4034	11	4034	11	0
i-060-04-08	5908	5908	49.10	5908	14	5908	14	0
i-060-08-12	4303	4303	26.26	4303	14	4303	14	0
i-070-04-08	6205	6228	24.78	6224	22	6202	22	-0.05
i-070-08-12	4644	4694	45.27	4644	22	4644	22	0
i-080-04-08	7390	7420	179.33	7390	27	7353	27	-0.50
i-080-08-12	5611	5613	43.65	5661	27	5575	27	-0.64
i-090-04-08	7108	7108	328.34	7182	32	7104	32	-0.08
i-090-08-12	5808	5893	307.66	5808	32	5807	32	-0.02
i-100-04-08	7295	7339	85.78	7295	38	7300	38	0.07
i-100-08-12	6546	6788	419.38	6546	38	6407	38	-2.12
i-110-04-08	8618	8618	128.12	8687	45	8614	45	-0.84
i-110-08-12	6249	6343	120.46	6249	45	6178	45	-1.14
i-120-04-08	8344	8455	190.67	8344	60	8344	60	0
i-120-08-12	6772	6772	151.67	6774	60	6770	60	-0.06
Moyenne	5706.85	5742.45	108.05	5718	26.2	5692.25	26.2	-0.27

TABLEAU 5.5 – Résultats des méthodes CBGH de Yuan et al. (2021), SLNS de Dumez et al. (2021a) et du SLNS sur les instances de GVRPTW de Moccia et al. (2012)

colonne 2 (BKS) indique la meilleure solution connue pour chaque instance. Les colonnes 3 et 4 indiquent les résultats obtenus et le temps d'exécution pour une exécution de la méthode CBGH de Yuan et al. (2021). Dumez et al. (2021b) proposent des résultats pour le GVRPTW en exécutant 4 configurations différentes de leur méthode SLNS : avec ou sans combinaison avec la résolution d'un SPP et avec ou sans recherche locale avec un voisinage de Balas-Simonetti. Chaque configuration est exécutée 5 fois. On remarque que nos méthodes diffèrent dans l'utilisation des destructions larges. On remarquera que dans le SLNS proposé par Dumez et al. (2021b), après avoir effectué une destruction large puis réparé la solution, celle-ci est systématiquement acceptée comme nouvelle solution courante.

La colonne 5 contient la meilleure solution parmi ces 20 exécutions. La colonne 6 indique le budget de temps alloué à chaque exécution. Les résultats de notre méthode

sont reportés en colonnes 7, 8 et 9. La colonne 7 indique la valeur de la meilleure solution trouvée sur 10 exécutions du SLNS. Les valeurs en gras indiquent une nouvelle meilleure solution trouvée par notre méthode. La colonne 8 indique le budget de temps alloué à chaque exécution, défini comme identique au budget de temps alloué pour le SLNS de Dumez et al. (2021b). La colonne 9 indique l'écart relatif par rapport à la meilleure solution connue.

En moyenne, le SLNS améliore les meilleures solutions connues de 0.27%. Sur 20 instances, le SLNS améliore la meilleure solution connue de 9 instances, et l'égalise sur 10 instances.

5.3.3 Vehicle Routing Problem with Multiple Time Windows

De la même manière que Dumez et al. (2021b), nous considérons le VRPMTW comme un VRPTW avec options, où pour chaque client, une option à coût nul est associée à chacune des fenêtres de temps.

Afin de valider la performance de notre méthode sur ce type de problème, nous évaluons notre méthode sur les instances de VRPMTW de Belhaiza et al. (2014), adaptées des instances de Solomon (1987). L'objectif considéré pour ces instances est dans un premier temps la minimisation du nombre de véhicules utilisés, puis la minimisation de la somme de la durée des tournées. Nous exécutons notre méthode 10 fois sur chaque instance avec un budget de temps fixé à 10 minutes pour chaque instance. La moitié du temps d'exécution est utilisée dans une première phase visant à réduire le nombre de véhicules : le nombre de véhicules autorisés est réduit d'une unité à chaque fois qu'une solution réalisable est obtenue.

Le Tableau 5.6 compare les résultats obtenus par notre méthode à ceux de la méthode ALNS de Larsen et Pacino (2019), qui proposent des expérimentations dans des conditions identiques.

Chaque ligne représente un résultat agrégé pour un ensemble d'instances du même type.

La colonne 2 indique la moyenne des meilleures solutions connues pour chaque groupe d'instances. Les colonnes 3, 4 et 5 indiquent respectivement le nombre total de véhicules utilisés, la valeur moyenne de la fonction objectif, et l'écart relatif aux meilleures solutions connues pour l'ALNS de Larsen et Pacino (2019). Les colonnes 6, 7 et 8 indiquent ces mêmes information pour l'algorithme SLNS.

Notre méthode trouve une flotte égale à celle des meilleures solutions connues sur

Instances	BKS	ALNS (Larsen & Pacino)			SLNS		
		Σr	z^*	gap (%)	Σr	z^*	gap (%)
CM1	12276.3	86	12288.9	0.10	88	12338.5	0.50
CM2	13182.8	37	13191.4	0.07	37	13290.8	0.81
RCM1	4208.7	82	4248.1	0.92	82	4254.0	1.06
RCM2	3740.5	16	3746.7	0.17	16	3767.9	0.73
RM1	3710.4	73	3743.7	0.91	73	3755.2	1.21
RM2	3694.0	16	3706.5	0.34	16	3694.2	0.00
Total général	6802.1	310	6820.9	0.42	312	6850.1	0.72

TABLEAU 5.6 – Comparaison des résultats du SLNS avec la méthode ALNS Larsen et Pacino (2019)

l'ensemble des instances à l'exception de deux instances de catégorie CM1. L'écart moyen aux meilleures solutions connues est de 0.72% et nous trouvons trois nouvelles meilleures solutions, pour des instances de la catégorie RM2.

5.3.4 Commodity-Constrained Split-Delivery Vehicle Routing Problem

Le C-SDVRP est un problème de tournées de véhicules introduit par Archetti et al. (2014a) dans lequel les clients demandent des produits de types différents. Différents produits peuvent voyager dans le même véhicule et la demande totale d'un client peut être servie par plusieurs véhicules. En revanche la demande d'un client pour un produit spécifique doit être livrée en une fois.

Cette caractéristique est également présente dans les réseaux des clients de CRC Services. Les magasins peuvent passer plusieurs commandes, de produits homogènes ou non, et la livraison de chaque commande doit être effectuée en une fois. Nous souhaitons donc évaluer la performance de notre méthode lorsque plusieurs commandes pour un même site peuvent être combinées.

Archetti et al. (2014a) proposent un ensemble d'instances pour le C-SDVRP en adaptant les instances C101 et R101 de Solomon (1987). Des instances sont générées pour $m = 2$ et 3 types de produits différents. La demande d'un client pour un produit est générée aléatoirement dans un intervalle $\Delta = [1, 100]$ ou $[40, 60]$. Les clients ont une probabilité p fixée à 0.6 ou 1 d'exprimer une demande pour chaque type de produit. La capacité des véhicules dépend d'un paramètre $\alpha \in (1.1, 1.5, 2, 2.5)$ et est déterminée par

αd_{max} où d_{max} est la demande totale la plus grande sur l'ensemble des produits pour un client. Cinq instances sont générées pour chaque combinaison des paramètres (m, Δ, p, α) et pour chaque instance C101 ou R101, ce qui fait un total de 320 instances.

type	m	p	Δ	ALNS		SLNS			
				\bar{z}	t	\bar{z}^*	$gap(\%)$	$\overline{z_{moy}}$	$gap_{moy}(\%)$
C101	2	0.6	[1, 100]	1635.3	354.2	1633.9	-0.06	1640.5	0.37
C101	2	0.6	[40, 60]	2709.8	305.9	2703.6	-0.23	2709.6	0.04
C101	2	1	[1, 100]	2391.8	581.6	2393.5	0.09	2410.6	0.85
C101	2	1	[40, 60]	3900.3	556.7	3896.4	-0.05	3909.5	0.37
C101	3	0.6	[1, 100]	1820.2	598.3	1820.9	0.07	1837.1	0.98
C101	3	0.6	[40, 60]	2566.1	516.7	2567.8	0.09	2584.5	0.78
C101	3	1	[1, 100]	2486.9	1138.7	2502.4	0.64	2526.9	1.63
C101	3	1	[40, 60]	3745.6	1073.8	3771.5	0.82	3800.9	1.64
R101	2	0.6	[1, 100]	1458.2	346.9	1455.6	-0.16	1463.5	0.45
R101	2	0.6	[40, 60]	2333.0	300.0	2328.1	-0.22	2330.6	-0.07
R101	2	1	[1, 100]	2080.4	566.8	2074.5	-0.25	2090.1	0.59
R101	2	1	[40, 60]	3301.8	548.7	3295.0	-0.20	3300.1	0.01
R101	3	0.6	[1, 100]	1597.9	580.3	1598.3	0.05	1613.4	1.05
R101	3	0.6	[40, 60]	2194.4	515.7	2189.9	-0.17	2199.0	0.32
R101	3	1	[1, 100]	2154.2	1087.5	2163.8	0.49	2187.6	1.63
R101	3	1	[40, 60]	3185.2	1069.9	3189.6	0.30	3212.0	1.15
Moyenne				2472.6	633.8	2474.1	0.08	2488.5	0.74

TABLEAU 5.7 – Comparaison des résultats du SLNS et de l'ALNS de Gu et al. (2019) sur les instances de C-SDVRP de Archetti et al. (2014a)

Le Tableau 5.7 compare les résultats obtenus par notre méthode à la méthode ALNS proposée par Gu et al. (2019). Les résultats rapportés par Gu et al. (2019) sont ceux obtenus après une unique exécution de leur méthode sur chaque instance. Nous exécutons notre méthode 10 fois sur chaque instance avec une limite de temps de 180 secondes, et rapportons le meilleur résultat et la moyenne sur l'ensemble des exécutions de l'algorithme.

Chaque ligne donne les résultats agrégés pour chaque groupe de 20 instances selon le paramètre α . Les colonnes 1 à 4 indiquent les autres paramètres utilisés pour générer les instances.

Les colonnes 5 et 6 indiquent le coût moyen des solutions et le temps moyen pour une exécution de la méthode ALNS de Gu et al. (2019). Les colonnes 7 et 8 indiquent le coût et l'écart à la meilleure solution connue pour la meilleure solution obtenue parmi les 10 exécutions du SLNS. Les colonnes 9 et 10 indiquent le coût moyen et l'écart moyen à la meilleure solution connue pour l'ensemble des exécutions du SLNS.

Avec un budget de temps deux fois plus long si l'on considère les 10 exécutions de notre méthode par instance, nous parvenons à trouver des résultats d'une qualité similaires à ceux trouvés par la méthode ALNS de Gu et al. (2019), qui contient plusieurs éléments de recherche locale spécifiquement développés pour le C-SDVRP.

Sur l'ensemble des 320 instances, notre méthode permet de trouver de nouvelles meilleures solutions dans 155 cas, des solutions équivalentes à la meilleure solution dans 12 cas et de moins bonnes solutions dans 153 cas. On remarque que les solutions de moins bonne qualité sont obtenues sur les instances générées avec $m = 3$ et $p = 1$, c'est à dire les instances contenant 300 commandes. On peut déduire que la limite des 3 minutes par instance n'est probablement pas suffisante compte tenu de la taille de ces instances.

5.3.5 Discussion

Les différentes expérimentations numériques présentées dans cette section nous ont permis de valider notre méthode sur différents benchmarks de la littérature.

Le SLNS obtient d'excellents résultats en particulier sur les problèmes comportant explicitement plusieurs lieux possibles de livraison pour une même commande (VRPTWTF, GVRPTW). Les résultats obtenus sur la variante avec flotte hétérogène (FSMTWTF) suggèrent que la gestion de la flotte peut encore être améliorée, en particulier sur un groupe spécifique d'instances.

Enfin, nous obtenons de très bons résultats sur des instances présentant de multiples commodités ou de multiples fenêtres de temps, sans avoir apporté de modification ou d'opérateurs spécifiques à ces caractéristiques.

Ces résultats nous ont permis d'être confiants quant à l'application de la méthode pour résoudre différents cas industriels au sein de la société CRC Services. Ces applications sont présentées dans le Chapitre 6.

5.4 Conclusion

Dans ce chapitre, nous avons introduit le PDPTF : un problème de collectes et livraisons dans lequel les marchandises peuvent être distribuées depuis différents sites d'expédition et livrées directement depuis le site d'expédition ou par un transporteur régional. Le transport est sous-traité à différents transporteurs proposant des structures de coût variées, FTL ou LTL. Nous avons modélisé ce problème comme un problème de collectes

et livraisons avec options, ce qui permet de modéliser un grand nombre de cas rencontrés par la société CRC Services (choix de site d'expédition, du site livré, prise en compte de fenêtres de temps multiples, multiples commodités).

Pour résoudre ce problème nous avons proposé une heuristique Small and Large Neighborhood Search (SLNS) combinée à la résolution d'un SPP. Les destructions de petite taille permettent d'intensifier la recherche autour de solutions prometteuses et d'effectuer un grand nombre d'itérations. Les destructions de grande taille, à intervalles réguliers, permettent d'échapper aux minima locaux. Les opérateurs d'insertion sont couplés avec un mécanisme adaptatif consistant à attribuer un score aux différentes options. Ce score permet de ne pas évaluer l'ensemble des options lors de la réinsertion d'une commande et de concentrer la recherche sur des options pertinentes.

Le PDPTF généralise un grand nombre de variantes de problèmes de tournées présentant des caractéristiques que l'on retrouve dans les réseaux de grande distribution : livraison en direct ou via un transporteur régional, multiples commodités ou encore fenêtres de temps multiples.

Des expérimentations numériques montrent que notre algorithme est capable de traiter ces différentes caractéristiques et est performant sur ces variantes de problèmes de tournées.

Parmi les perspectives de recherche et développement, nous envisageons de proposer un ensemble d'instances pour le PDPTF combinant les différentes variantes de problèmes de tournées présentées dans les expérimentations et incluant de multiples points de collecte pour les commandes. Il serait notamment intéressant d'étudier différentes stratégies de consolidation entre plusieurs chargeurs : via un transporteur régional et/ou via des tournées effectuant des collectes sur plusieurs sites chargeurs. De plus, le coût sur les options pourrait être utilisé pour modéliser des problématiques que nous n'avons pas encore abordées telles que des pénalités de retard en associant des coûts différents aux options livrant le client sur site, ou des coûts de production ou d'achat différents en associant des coûts différents aux options selon leur point de collecte.

Dans ce chapitre, la distribution des marchandises depuis les plateformes des transporteurs régionaux n'est pas pilotée par le chargeur ou CRC Services. Le PDPTF pourrait être étendu pour considérer la livraison des marchandises via différents CRC. La distribution des marchandises depuis ces CRC pourrait alors être pilotée, en faisant converger ces travaux avec la résolution du problème VRP-FLC décrit au chapitre précédent. Cela permettrait de résoudre des problèmes de collectes et livraisons avec cross-docking encore

peu résolu dans la littérature dans le cas de plateformes multiples. Une approche possible serait d'adapter les méthodes itératives de Absi et al. (2015) et Chitsaz et al. (2019) pour faire le lien entre les deux algorithmes en mettant à jour les coûts des options en fonction du coût effectif de distribution des options confiées aux plateformes entre les itérations.

Le PDPTF concerne la planification court terme des transports à une échelle nationale. Cette planification doit impérativement se faire quelques jours en avance de manière à réserver les véhicules nécessaires. Or, la totalité de la demande n'est pas encore connue, ou peut être imprécise. Une perspective de recherche est donc de modéliser et résoudre le PDPTF avec des commandes clients incertaines.

APPLICATIONS INDUSTRIELLES

6.1	Introduction	103
6.2	Mise en œuvre des algorithmes développés au sein de CRC Services	105
6.2.1	Données d'entrée	105
6.2.2	Données de sortie	106
6.2.3	Intégration dans CoLivRi	106
6.2.4	Perspectives	108
6.3	Cas d'étude	109
6.3.1	Projet Auptilait	109
6.3.2	Optimisation de la distribution de flux mutualisés depuis un CRC vers des grandes surfaces spécialisées	111
6.3.3	Optimisation du transport de marchandises pour un acteur de l'ultra- frais	112
6.4	Conclusion	113

6.1 Introduction

Ce chapitre a pour but de présenter la dimension industrielle des travaux réalisés au cours de cette thèse.

Avant le démarrage de mes travaux de thèse en 2019, j'ai pu bénéficier d'une expérience de développeur en tant qu'ingénieur chez CRC Services, entre Septembre 2018 et Mai 2019. Ma mission principale était le développement de la plateforme CoLivRi, au sein d'une équipe de développeurs d'une entreprise de services du numérique (ESN) travaillant pour CRC Services. Ces neuf mois d'expérience m'ont permis d'acquérir une bonne connaissance de la plateforme CoLivRi, des modèles de données utilisés, des règles métier et du vocabulaire lié au secteur du transport. J'ai également été responsable de

l'implémentation et de l'intégration dans CoLivRi de plusieurs algorithmes d'optimisation, basés sur la résolutions de PLNE.

Au cours de la thèse, CRC Services a pris la décision de poursuivre le développement de la plateforme CoLivRi avec sa propre équipe de développeurs. J'ai été responsable de la formation des nouveaux développeurs. Ces expériences m'ont été utiles pour la définition des problèmes de recherche comme pour les différentes applications des algorithmes au sein de CRC Services et leur intégration sur la plateforme CoLivRi.

Les algorithmes développés au cours de cette thèse ont été fréquemment utilisés dès leurs premières versions au sein de la société CRC Services afin de mener différentes études pour des clients ou des prospects. En effet, l'offre de service proposée par CRC Services propose des accompagnements ou missions courtes de type « Preuve de concept ». Pour ces missions, l'objectif est de mesurer les enjeux d'un changement de l'organisation transport pour un périmètre restreint, sur un plan économique, environnemental et niveau de service. Les résultats d'une mission permettent aux clients d'évaluer la pertinence d'évolutions dans leur organisation transport et de l'utilisation des algorithmes d'optimisation afin de construire leur stratégie transport ou gérer les tâches opérationnelles.

Une première étape de l'accompagnement d'un client consiste à faire l'état des lieux de sa stratégie transport courante et à identifier des axes d'amélioration. Afin de répondre aux problématiques identifiées, CRC Services propose différentes stratégies ou transformations des processus stratégiques, tactiques et opérationnels du client. Ces transformations peuvent consister à utiliser de nouveaux schémas de transport (groupage, tournées directes, mutualisation, rail-route etc...), ou à utiliser de nouvelles plateformes logistiques, par exemple, en mutualisant ses flux avec un autre chargeur depuis une plateforme de passage à quai. Enfin, les travaux peuvent concerner le choix des fréquences de livraison et les jours de cadencement, notamment dans la mise en place de réseau de transport mutualisés où les cadencements doivent être retravaillés. Afin d'évaluer les enjeux liés à une nouvelle organisation, les algorithmes d'optimisation sont alimentés par des historiques de commandes ou de flux ainsi qu'un certain nombre d'hypothèses (règles de calcul des coûts, schémas de transport possibles) et de contraintes sur les commandes et les tournées imposées par le client ou ses prestataires logistiques. En sortie, ils proposent des plans de transport optimisés combinant les différents schémas de transports que le client veut tester. Les solutions obtenues sont alors comparées à l'organisation actuelle et permettent de déterminer la pertinence et les enjeux liés à la nouvelle stratégie testée.

La suite de ce chapitre illustre les différentes applications dans la société CRC Services

des algorithmes développés au cours de cette thèse. La section 6.2 décrit l'utilisation des algorithmes développés et leur intégration dans la plateforme CoLivRi. En section 6.3, nous décrivons différents projets menés avec des clients de CRC Services.

6.2 Mise en œuvre des algorithmes développés au sein de CRC Services

Dans cette section, nous décrivons d'abord le format des données d'entrée permettant d'utiliser les algorithmes développés, et le format des données de sortie. Nous décrivons ensuite une première intégration des algorithmes sur la plateforme CoLivRi, consistant à importer la sortie des algorithmes afin de visualiser les résultats et de pouvoir les analyser. Nous concluons cette section par les perspectives concernant l'intégration complète des méthodes développées sur CoLivRi.

6.2.1 Données d'entrée

Lors de mission de conseil, les algorithmes de CRC Services, en particulier la lecture et le pré-traitement des données, sont adaptés afin de s'approcher au plus près des problématiques clients. L'enjeu est de pouvoir exprimer les données d'un problème de manière à ce qu'il puisse être résolu par un des algorithmes d'optimisation déployé.

La lecture des données d'entrée est réalisée à partir d'un fichier Excel fourni par les clients. Bien que la lecture de ces fichiers soit spécifique à chaque étude menée, ceux-ci suivent une logique récurrente. Un premier onglet présente le réseau logistique : les différents sites chargeurs, CRC et clients, leurs coordonnées géographiques, fenêtres horaires de réception et d'expédition. Un second onglet décrit l'historique de flux : dates, tailles, origines et destinations des commandes, types de produit, etc. Un dernier onglet décrit le référentiel transport : les transporteurs, leurs types de véhicules, leurs tarifs et leurs contraintes.

Un pré-traitement spécifique à chaque étude permet de créer un ensemble d'instances découpées à la journée ou à la semaine à partir des données fournies. Ce pré-traitement permet également de modéliser la problématique en fonction de l'algorithme d'optimisation choisi. Par exemple, un pré-traitement peut être nécessaire pour calculer les différentes options de livraison d'une commande en fonction de la date de disponibilité de la commande, des créneaux de livraison possibles chez le client, des heures d'ouverture

d'éventuelles plateformes intermédiaires ou d'autres contraintes spécifiées par le chargeur. Ce pré-traitement peut aussi être nécessaire afin de recalculer des données manquantes. Par exemple, si on considère la distribution depuis un CRC mais que le chargeur ne dispose que des heures de départ de ses commandes depuis ses sites d'expédition, différentes règles peuvent être appliquées afin de reconstituer les heures à partir desquelles les commandes peuvent être expédiées depuis le CRC.

6.2.2 Données de sortie

Les données de sortie des algorithmes sont exportées sous la forme d'un fichier Excel contenant deux onglets. Un premier onglet « Tournées » présente le détail des tournées réalisées pour chaque journée d'expédition considérée : sites visités, commandes livrées sur chaque site, heures de départ et de passage à chaque point, etc. Un second onglet « Commandes » retranscrit l'historique de flux associé au mode de transport de chaque commande, entre autres la tournée livrant la commande, son site de départ et de livraison ou le coût de livraison de la commande au sein de la tournée selon différents pro-ratas (poids, nombre de palettes, poids \times distance parcourue). Ce fichier est ensuite retravaillé afin de faire ressortir différents indicateurs à présenter aux clients : tournées les plus fréquentes, coûts de livraison au global, par région ou par site, taux de remplissage, transporteurs ou modes de transport sélectionnés, ou encore options choisies lorsque les commandes peuvent être livrées par des prestataires logistiques externes ou déposées sur des plateformes intermédiaires.

6.2.3 Intégration dans CoLivRi

Une fois exportées au format Excel, les données résultant d'une optimisation sont importées sur CoLivRi afin d'analyser les résultats et comparer différents scénarios.

Les Figures 6.1 et 6.2 montrent différents écrans d'analyse paramétrables, un premier affichant différents indicateurs au cours du temps (à la semaine, au mois ou à l'année) et un second affichant des résultats agrégés pour l'ensemble de l'optimisation. Différents indicateurs peuvent également être sélectionnés et projetés sur une carte, comme illustré par la Figure 6.3. Ces indicateurs peuvent être combinés et incluent le nombre de livraisons par site, le nombre moyen d'unités logistiques pour chaque livraison, le coût total ou par unité logistique ou encore l'économie générée par l'optimisation par rapport à une situation initiale.



FIGURE 6.1 – Onglet « Focus » alimenté par les résultats d’une optimisation

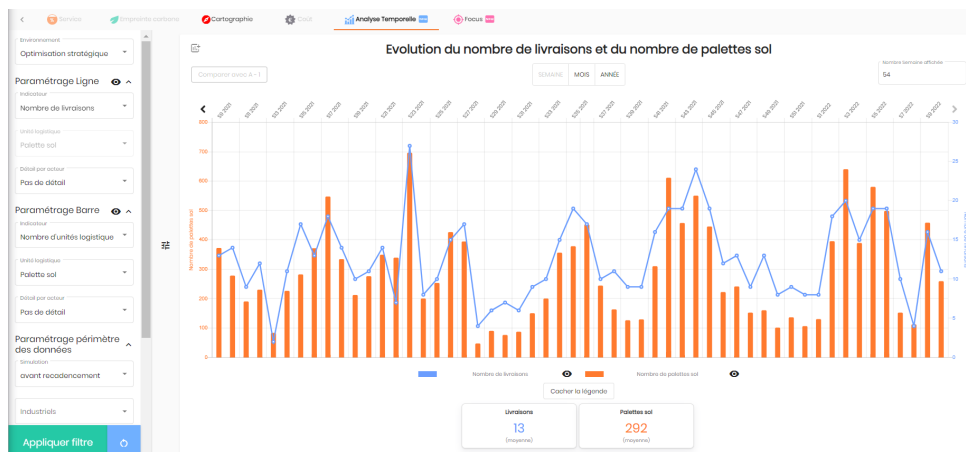


FIGURE 6.2 – Onglet « Analyse temporelle » alimenté par les résultats d’une optimisation

L’intérêt de la visualisation dans CoLivRi est de pouvoir naviguer de manière dynamique dans l’ensemble des données et d’appliquer des filtres d’une richesse importante pour pouvoir calculer des indicateurs métiers sur un grand nombre de simulations. Par exemple, dans l’optimisation d’un réseau mutualisé, la comparaison entre la distribution optimisée d’un ensemble de chargeurs et l’optimisation chargeur par chargeur est une forme récurrente d’analyse. CoLivRi permet de développer ces comparaisons par point de livraison, par période, par CRC, etc.

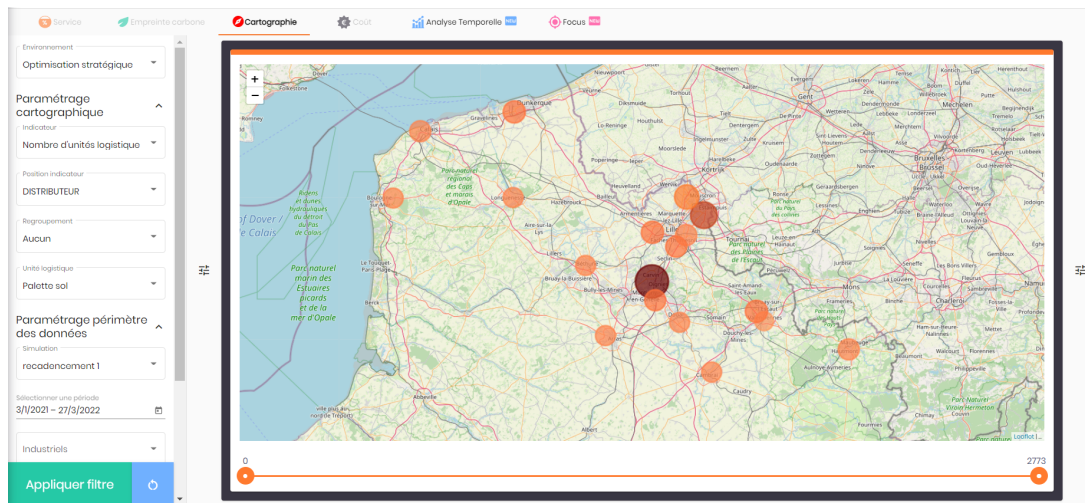


FIGURE 6.3 – Projection des données de sortie sur une carte

6.2.4 Perspectives

L'objectif à court terme est d'intégrer les algorithmes développés au cours de cette thèse avec les autres algorithmes déjà présents sur la plateforme CoLivRi afin de permettre aux utilisateurs de les utiliser de façon autonome, pour des usages stratégiques ou opérationnels.

Les algorithmes ont été développés dans le but d'être le plus génériques possible afin de pouvoir traiter un grand nombre de cas d'usage, sans devoir développer un nouvel algorithme de A à Z à chaque nouveau cas client. Toutefois, il n'est pas envisageable de laisser les clients de CoLivRi paramétrer l'ensemble des données nécessaires aux algorithmes. Ce procédé serait très lourd, et les utilisateurs ne sont pas des praticiens de la recherche opérationnelle. L'enjeu est alors de trouver le bon équilibre au sein du parcours utilisateur afin qu'il soit suffisamment simple d'accès tout en laissant assez de liberté pour permettre aux utilisateurs de tester différentes stratégies.

L'architecture envisagée pour intégrer les algorithmes développés sur la plateforme CoLivRi est illustrée par la Figure 6.4. Elle consiste à proposer un parcours utilisateur accessible aux clients de CoLivRi et à «cacher» une partie de la complexité de la préparation des données dans des algorithmes de pré-traitement, développés au cas par cas pour les clients. Cette approche permet de ne pas (ou peu) modifier les algorithmes implémentés tout en s'adaptant aux différentes problématiques des clients. La sortie des algorithmes étant générique, les résultats peuvent être visualisés sur CoLivRi sans nécessiter de nouveaux développements après l'intégration d'un nouveau client.

Ces travaux d'intégration débuteront à l'issue de cette thèse, dès mi-2022.

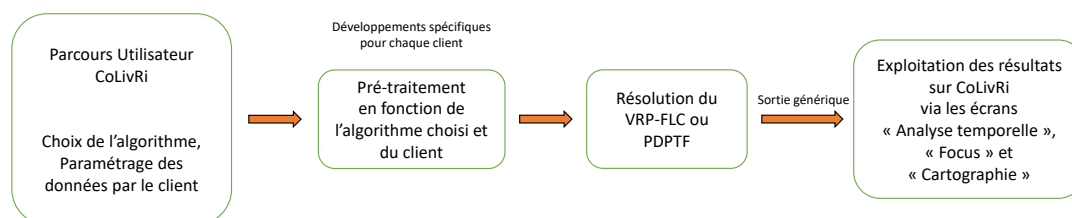


FIGURE 6.4 – Architecture prévue pour l'intégration des algorithmes développés sur la plateforme CoLivRi

6.3 Cas d'étude

Cette section présente plusieurs cas d'étude qui sont ou ont été menés au sein de CRC Services à l'aide des algorithmes développés au cours de cette thèse. Certains cas ont été légèrement modifiés afin de respecter des engagements de confidentialité auprès des clients de CRC Services.

6.3.1 Projet Auptilait

Lauréat de l'appel à projet TRANSLOG 2020 de l'Agence De l'Environnement et de la Maîtrise de l'Énergie (ADEME), le projet *Auptilait* (Auchan Optimisation Lait) vise à améliorer l'approvisionnement des filières « Laits » du distributeur Auchan. Il implique le distributeur Auchan Retail France (ARF), la société CRC Services et un industriel fournissant plusieurs produits de la gamme Laits. Le projet a démarré en 2021.

Les conditions d'approvisionnement des magasins Auchan se sont détériorées au cours des dernières années, du fait de l'atomisation de la gamme des filières « Laits », liée aux orientations stratégiques prises par l'entreprise. 20% des flux des filières « Laits » livrés directement aux magasins le sont par 10 palettes ou moins, selon un fournisseur de la filière, alors que l'approvisionnement par camions complets devrait être la norme. La problématique pour ARF consiste donc à rechercher et expérimenter, avec ses fournisseurs et ses magasins, une solution numérique, s'appuyant sur des algorithmes de routage de flux,

pour optimiser l’approvisionnement magasins des filières « Laits ». La solution proposée doit améliorer la disponibilité linéaire de cette gamme de produits tout en réduisant les coûts logistiques ainsi que les émissions de GES. Par ailleurs, le processus de réapprovisionnement en magasin doit être simplifié et la démarche devra être applicable à d’autres filières dans le futur.

Le réseau de distribution, illustré par la Figure 6.5 est constitué de différents magasins Auchan (Hyper, Drive), de plusieurs plateformes logistiques Auchan, et de différents sites fournisseurs d’où partent les produits.

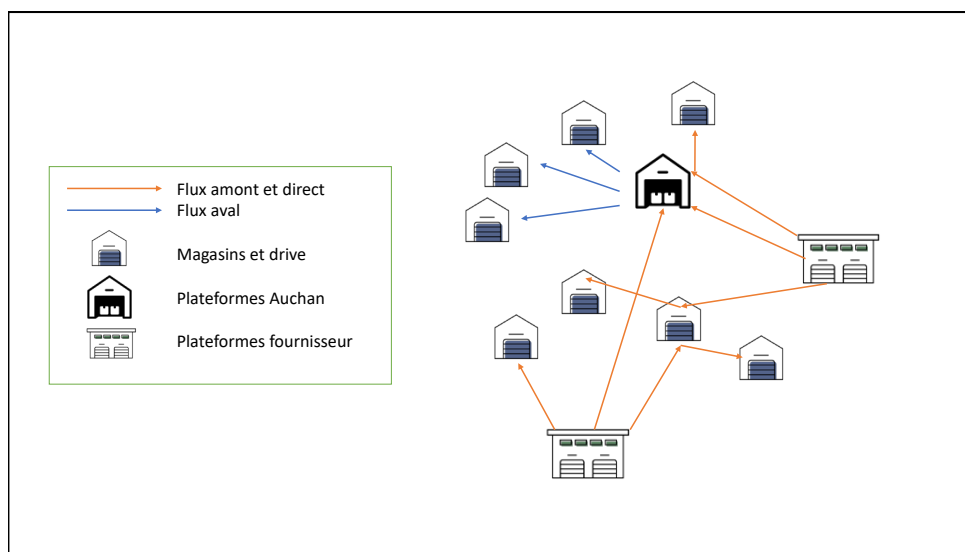


FIGURE 6.5 – Réseau de distribution et schémas de flux pour l’approvisionnement « Laits » des magasins Auchan

Aujourd’hui, les commandes suivent deux circuits de réapprovisionnement en fonction du magasin passant la commande : les plus petits magasins sont systématiquement livrés via une des plateformes Auchan (qui passe alors elle-même la commande au fournisseur), tandis que d’autres sont systématiquement livrés en direct depuis une des plateformes fournisseur. De plus, les magasins et les plateformes sont incités à passer des commandes pour des quantités proches de celle d’un camion complet. Le fournisseur propose en effet différents seuils de remise dits « BQ » (Barème Quantitatif) sur une partie des produits en fonction de la quantité totale de produits demandée par le magasin ou la plateforme. En fonction de leurs besoins et de leurs jours de passage de commande, certains magasins ne peuvent pas bénéficier du meilleur BQ et risquent soit un surstock en commandant plus que leurs besoins, soit une rupture de stock sur certains produits s’ils attendent trop

pour passer commande.

L'objectif de cette étude est de remettre en cause le caractère figé des schémas de transport actuels. La décision de livrer une commande par plateforme ou en direct est donc un choix dynamique, modélisé grâce au formalisme des options dans le PDPTF présenté au Chapitre 5. Nous considérons également les gains potentiels à livrer des plateformes et des magasins au sein d'une même tournée. La méthode de résolution développée pour le PDPTF nous permet de modéliser les coûts de transport FTL et LTL que paye le fournisseur et également de prendre en compte l'impact de la remise BQ sur les coûts d'achat des magasins ou plateformes Auchan. Une des prochaines étapes du projet est d'identifier des clusters de magasins et de travailler sur le cadencement des jours de passage de commande pour chaque cluster afin de mieux consolider les tournées de livraison.

6.3.2 Optimisation de la distribution de flux mutualisés depuis un CRC vers des grandes surfaces spécialisées

CRC Services a été contacté par un distributeur d'une enseigne de grandes surfaces spécialisées. Ce client souhaitait étudier la mise en place d'un réseau de transport mutualisé entre ses fournisseurs et ses magasins. Le projet, démarré en 2020, a été mené en partenariat avec un transporteur sélectionné pour effectuer le rôle de CRC depuis différentes plateformes et pour prendre la distribution des commandes depuis les CRC vers les magasins.

Une première étape a consisté à étudier la distribution des marchandises depuis un CRC vers un ensemble de magasins. Plusieurs itérations nous ont permis de modéliser au plus proche les contraintes du transporteur (flotte hétérogène, contraintes sur les tournées), des magasins (fenêtres horaires, jours de passage de commande) et les coûts du transporteur. Le problème a été formulé comme un VRPPC et résolu avec la méthode LNS-SPP présentée au Chapitre 4 sur un ensemble d'instances générées à partir d'un historique de commandes s'étalant sur une période de 12 mois. Une fois ces différents éléments bien établis, le modèle a été appliqué à l'ensemble des CRC considérés, évitant au transporteur le long processus pour chiffrer ses coûts depuis chaque CRC et s'assurer de sa rentabilité.

Cette étude a permis de convaincre le transporteur de l'intérêt du projet et les CRC ont été déployés sur toute la France. Les tournées de véhicules préconisées par l'algorithme sont celles utilisées aujourd'hui par le transporteur. Afin de suivre la performance du transport

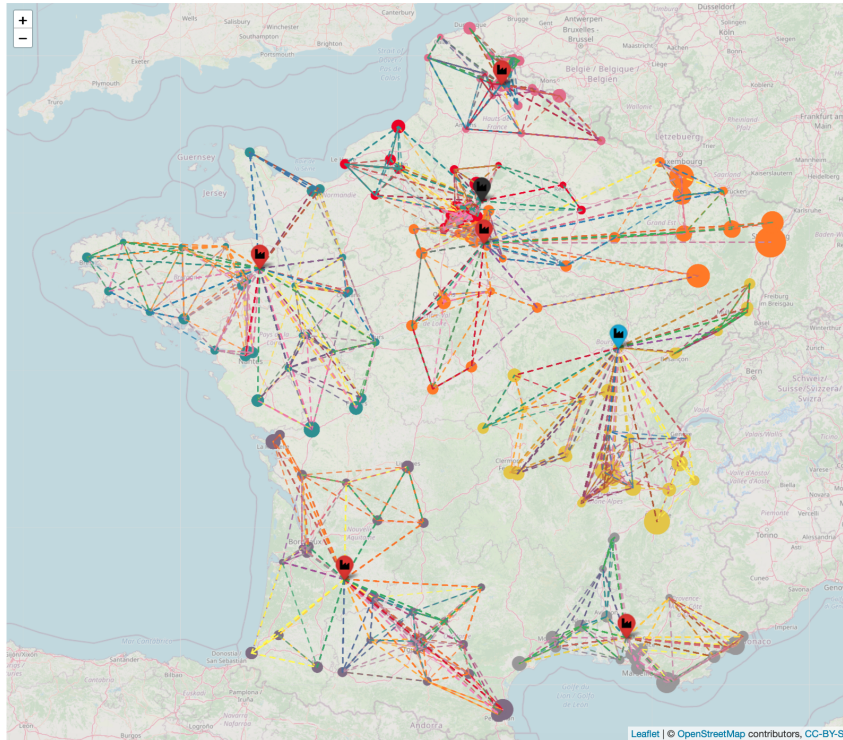


FIGURE 6.6 – Réseau de mutualisation déployé autour de six CRC

de ce réseau, CoLivRi est déployé chez l'ensemble des acteurs de la mutualisation pour tracer et évaluer la performance de ce nouveau réseau mutualisé.

6.3.3 Optimisation du transport de marchandises pour un acteur de l'ultra-frais

Nous décrivons dans cette sous-section un projet mené par CRC Services pour un de ses clients au cours de l'année 2021.

Le client est un industriel produisant des marchandises dans le secteur de l'ultra-frais et les expédiant à plusieurs clients en France. Les clients de cet industriel sont majoritairement des enseignes de la GMS (Grandes et Moyennes Surfaces) et comportent quelques points de vente du secteur CHD (Consommation Hors Domicile). L'industriel dispose de plusieurs sites de production et regroupe l'essentiel de ses expéditions sur une unique plateforme.

Le secteur de l'ultra-frais fonctionne en flux tendu : les clients passent le plus souvent commande au jour J pour une livraison en J+1. Dans ce contexte, le plan de transport de

l'industriel est relativement figé. Il travaille avec différents transporteurs selon les zones livrées et selon différents schémas de transport : groupage, traction-distribution avec des transporteurs régionaux, et quelques tournées directes pré-établies à des tarifs négociés au cas par cas avec différents transporteurs. Certains clients sont systématiquement livrés en traction-distribution, mais auraient des flux suffisant pour être, périodiquement, livrés en direct.

L'industriel souhaitait évaluer les gains potentiels à utiliser un algorithme d'optimisation pour modifier son plan de transport en fonction de ses volumes et à repenser sa stratégie transport : utilisation de tournées qui ne sont pas pré-établies, livraison en direct de clients habituellement livrés en traction-distribution, tournées mixtes comprenant certaines commandes livrées en traction-distribution et d'autres en direct.

Après avoir identifié la problématique de l'industriel, un premier travail a consisté à recueillir et traiter un historique des flux expédiés sur les deux dernières années et à identifier ses différentes contraintes relatives au transport et à sa tarification.

Ensuite, en modélisant le problème comme un PDPTF, nous avons pu étudier différentes hypothèses comme la livraison de certains clients en direct ou en traction-distribution, en fonction des opportunités de consolidation. Pour chaque client, les différents schémas de transport ont été modélisés comme autant d'options de livraison.

Notre méthode a permis de restituer au client des résultats sous la forme de potentiels de gain et de présenter ce que pourrait être un plan de transport type sur quelques semaines. Ces résultats ont été comparés à la stratégie de transport actuelle du client et les résultats proposés sont actuellement à l'étude chez le client.

Ce type d'approche, basée des arguments chiffrés, permet à un chargeur d'estimer les enjeux qu'il peut avoir en adoptant une organisation plus agile de son transport. Cette étape est nécessaire pour convaincre de la mise en place de nouvelles stratégies de transport parfois difficile à changer.

6.4 Conclusion

Dans ce Chapitre, nous avons présenté le volet industriel de la thèse, à savoir l'utilisation chez CRC Services des algorithmes de recherche opérationnelle développés au cours de cette thèse. Ces algorithmes ont servi à la réalisation de projets concrets auprès de clients de l'entreprise, et ont été en partie intégrés à la plateforme CoLivRi. Une intégration complète, impliquant la conception et le développement d'un parcours utilisateur, est

prévue à la suite de cette thèse.

La généricité des modèles proposés nous a permis de pouvoir résoudre un grand nombre de cas. De plus, les méthodes basées sur des recherches à voisinage large se sont montrées utiles, car elles ne sont pas sujettes à des structures de coûts ou des contraintes de tournées particulières.

CONCLUSION

Cette thèse a été guidée par les besoins de la société CRC Services qui souhaite développer des outils afin d’accompagner ses clients dans la transformation ou l’amélioration de leur stratégie de transport. Les clients de CRC Services sont des chargeurs, achetant des services transport à différents prestataires logistiques afin d’acheminer des marchandises dans des réseaux de la grande distribution. Comme présenté au Chapitre 1, ces chargeurs ont en effet des problématiques qui leur sont propres, liées au transport sous-traité, aux différents schémas de livraison possibles et aux différentes contraintes imposées par les transporteurs, le réseau, ou les commandes clients.

Au cours de cette thèse, nous avons formulé deux problèmes de tournées de véhicules répondant à la problématique chargeur : le Problème de Tournées de Véhicules avec Transporteurs FTL et LTL (VRP-FLC) et le Problème de Collectes et Livraisons avec Plateformes Intermédiaires (PDPTF).

Le VRP-FLC, introduit au Chapitre 4, concerne la distribution de marchandises depuis un unique site d’expédition qui peut être un site fournisseur ou un CRC mutualisant les flux de plusieurs chargeurs. Le transport de ces marchandises vers les clients est sous-traité à différents transporteurs proposant des tarifs FTL ou LTL. Ces transporteurs disposent de flottes de véhicules hétérogènes limitées et ne peuvent s’engager que sur un nombre restreint de véhicules, appelé *limite transporteur*. Pour résoudre ce problème, nous avons proposé une heuristique combinant une Recherche à Voisinage Large et la résolution d’un problème de partitionnement d’ensemble. La principale contribution de ce chapitre est l’utilisation d’une méthode de filtrage issue de la programmation par contraintes au cours de la réparation d’une solution.

Le PDPTF, introduit dans le Chapitre 5, est un problème de collectes et livraisons dans un réseau comprenant plusieurs sites de production et plusieurs plateformes intermédiaires. Ce problème intègre le schéma de transport *traction-distribution*, dans lequel les marchandises sont d’abord acheminées vers une plateforme intermédiaire appartenant à des transporteurs régionaux. Ces transporteurs régionaux assurent alors la distribution des marchandises jusqu’au client. Nous avons modélisé ce problème en utilisant le formalisme des options. Une option détermine pour une commande son site de départ, le mode

de transport et les fenêtres de temps choisies pour la collecte et la livraison. Pour résoudre ce problème, nous avons proposé une matheuristique de type Small and Large Neighborhood Search, combinée à la résolution d'un problème de partitionnement d'ensemble. De petites destructions sont utilisées pour intensifier la recherche et effectuer un grand nombre d'itérations tandis que des destructions plus importantes servent à diversifier la recherche et échapper aux minima locaux.

Un point remarquable de cette thèse est qu'elle a permis d'identifier, modéliser et résoudre deux problèmes originaux de tournées de véhicules : le VRP-FLC et le PDPTF. Ces modèles ne sont à priori pas des problèmes centraux dans la communauté académique, et encore moins dans le monde des logiciels d'optimisation de tournées de véhicules. Pourtant, d'un point de vue chargeur, ils permettent de modéliser plusieurs cas concrets dans des contextes variés.

Un second point notable est que pour concevoir les algorithmes de résolution pour ces deux problèmes, nous nous sommes largement appuyés sur des benchmarks de la littérature. Ces benchmarks nous ont permis de valider la très bonne performance des algorithmes proposés. Ce travail, bien que très académique dans sa nature, a renforcé la confiance de l'entreprise dans les solutions proposées. Il faut également considérer l'apport de ces benchmarks au cours du processus de conception de nos algorithmes : ils ont été utiles pour la validation de la version finale mais également pour l'amélioration des performances, notamment la détection d'instances difficiles, pour le paramétrage des algorithmes et pour la détection de bugs. Nous avons pu en cela bénéficier de précieux échanges avec plusieurs chercheurs ayant développé des méthodes d'optimisation pour les benchmarks cités aux Chapitres 4 et 5.

Les travaux réalisés au cours de cette thèse ont servi à résoudre plusieurs cas d'étude pour des clients de CRC Services. Ces applications ont été détaillées dans le Chapitre 6. Nous y décrivons également l'intégration progressive des algorithmes sur la plateforme CoLivRi. Ces applications ont été l'occasion pour moi de rencontrer directement les clients et de m'impliquer dans les différents projets menés par CRC Services. Certains des cas d'étude menés ont aussi fortement guidé la définition des problèmes étudiés.

Des travaux visant à intégrer pleinement les algorithmes sur la plateforme CoLivRi sont prévus et débiteront à l'issue de cette thèse. Les utilisateurs de CoLivRi pourront alors eux-même utiliser les algorithmes afin de mener des études stratégiques ou pour un usage opérationnel.

Les perspectives de recherche sont nombreuses. D'un point de vue méthodologique,

nous aimerions étudier plus en détail l'impact des différents éléments du Large Neighborhood Search et leurs paramétrages. Beaucoup de ces éléments ont en effet un impact fortement dépendant des autres paramètres, rendant un paramétrage global de la méthode difficile. La variante SLNS est encore une variante récente qui mériterait d'être mieux analysée, en particulier sur un plus grand nombre de benchmarks de la littérature.

Les algorithmes développés au cours de cette thèse pourraient être appliqués dans d'autres contextes, tels que celui de la logistique urbaine ou du transport inter-modal, en considérant des schémas de transport rail-route ou fluviaux. En effet, en pratique, un schéma de transport rail-route revient à faire appel à un prestataire spécialisé pour ce type de distribution. Dans certains cas, ce mode peut alors être incorporé soit comme une liaison LTL vers un CRC, soit comme une plateforme intermédiaire qui se charge de la livraison du client final. Le formalisme des options permet d'envisager une intégration relativement aisée de ce mode de transport pour peu que la fréquence associée à chaque liaison reste raisonnable à l'intérieur d'une journée. Les modèles présentés aux Chapitres 4 et 5 pourraient également être considérés dans des contextes multi-objectifs, en minimisant l'impact environnemental, par exemple, ou dans un contexte de données incertaines (prévision de flux, temps de trajet aléatoires). Le laboratoire commun CRC Lab soutenu par l'ANR et démarré en 2021, a justement pour but d'étudier la modélisation d'évènement aléatoires dans les réseaux de transports mutualisés et leur intégration dans des algorithmes d'optimisation.

BIBLIOGRAPHIE

- Absi, Nabil, Claudia Archetti, Stéphane Dauzère-Pérès et Dominique Feillet (2015), « A Two-Phase Iterative Heuristic Approach for the Production Routing Problem », in : *Transportation Science* 49.4, p. 784-795.
- Alcaraz, Juan J, Luis Caballero-Arnaldos et Javier Vales-Alonso (2019), « Rich vehicle routing problem with last-mile outsourcing decisions », in : *Transportation Research Part E : Logistics and Transportation Review* 129, p. 263-286.
- Archetti, Claudia, Nicola Bianchessi et M Grazia Speranza (2015), « A branch-price-and-cut algorithm for the commodity constrained split delivery vehicle routing problem », in : *Computers & Operations Research* 64, p. 1-10.
- Archetti, Claudia, Ann Melissa Campbell et M Grazia Speranza (2014a), « Multicommodity vs. single-commodity routing », in : *Transportation Science* 50.2, p. 461-472.
- Archetti, Claudia, M. Grazia Speranza et Daniele Vigo (2014b), « Chapter 10 : Vehicle Routing Problems with Profits », in : *Vehicle Routing*, sous la dir. de Paolo Toth et Daniele Vigo, Philadelphia, PA : Society for Industrial et Applied Mathematics, p. 273-297.
- Bakir, Ilke, Alan Erera et Martin Savelsbergh (2021), « Motor Carrier Service Network Design », in : *Network Design with Applications to Transportation and Logistics*, sous la dir. de Teodor Gabriel Crainic, Michel Gendreau et Bernard Gendron, Springer, p. 427-467.
- Balas, Egon et Neil Simonetti (2001), « Linear time dynamic-programming algorithms for new classes of restricted TSPs : A computational study », in : *INFORMS journal on Computing* 13.1, p. 56-75.
- Baldacci, Roberto, Sandra Ulrich Ngueveu et Roberto Wolfler Calvo (2017), « The vehicle routing problem with transshipment facilities », in : *Transportation Science* 51.2, p. 592-606.
- Baller, Annelieke C., Said Dabia, Wout E. H. Dullaert et Daniele Vigo (2020), « The Vehicle Routing Problem with Partial Outsourcing », in : *Transportation Science* 54.4, p. 1034-1052.

-
- Ballot, Eric, Benoit Montreuil et Russell Meller (2014), *The physical internet*, La Documentation Française.
- Beheshti, Ali Kourank, Seyed Reza Hejazi et Mehdi Alinaghian (2015), « The vehicle routing problem with multiple prioritized time windows : A case study », in : *Computers & Industrial Engineering* 90, p. 402-413.
- Belhaiza, Slim, Pierre Hansen et Gilbert Laporte (2014), « A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows », in : *Computers & Operations Research* 52, p. 269-281.
- Belhaiza, Slim et Rym M'Hallah (2016), « A Pareto non-dominated solution approach for the vehicle routing problem with multiple time windows », in : *2016 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, p. 3515-3524.
- Belhaiza, Slim, Rym M'Hallah et Ghassen Ben Brahim (2017), « A new hybrid genetic variable neighborhood search heuristic for the vehicle routing problem with multiple time windows », in : *2017 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, p. 1319-1326.
- Belhaiza, Slim, Rym M'Hallah, Ghassen Ben Brahim et Gilbert Laporte (2019), « Three multi-start data-driven evolutionary heuristics for the vehicle routing problem with multiple time windows », in : *Journal of Heuristics* 25.3, p. 485-515.
- Bolduc, Marie-Claude, Jacques Renaud et Fayez Boctor (2007), « A heuristic for the routing and carrier selection problem », in : *European Journal of Operational Research* 183.2, p. 926-932.
- Bolduc, Marie-Claude, Jacques Renaud, Fayez Boctor et Gilbert Laporte (2008), « A perturbation metaheuristic for the vehicle routing problem with private fleet and common carriers », in : *Journal of the Operational Research Society* 59.6, p. 776-787.
- Bulhões, Teobaldo, Minh Hoàng Hà, Rafael Martinelli et Thibaut Vidal (2018), « The vehicle routing problem with service level constraints », in : *European Journal of Operational Research* 265.2, p. 544-558.
- Ceschia, Sara, Luca Di Gaspero et Andrea Schaerf (2011), « Tabu search techniques for the heterogeneous vehicle routing problem with time windows and carrier-dependent costs », in : *Journal of Scheduling* 14.6, p. 601-615.
- Ceselli, Alberto, Giovanni Righini et Matteo Salani (2009), « A column generation algorithm for a rich vehicle-routing problem », in : *Transportation Science* 43.1, p. 56-69.

-
- Chitsaz, Masoud, Jean-François Cordeau et Raf Jans (2019), « A Unified Decomposition Matheuristic for Assembly, Production, and Inventory Routing », in : *INFORMS Journal on Computing* 31.1, p. 134-152.
- Christiaens, Jan et Greet Vanden Berghe (2020), « Slack induction by string removals for vehicle routing problems », in : *Transportation Science* 54.2, p. 417-433.
- Chu, Ching-Wu (2005), « A heuristic algorithm for the truckload and less-than-truckload problem », in : *European Journal of Operational Research* 165.3, p. 657-667.
- Côté, Jean-François et Jean-Yves Potvin (2009), « A tabu search heuristic for the vehicle routing problem with private fleet and common carrier », in : *European Journal of Operational Research* 198.2.
- Crainic, Teodor Gabriel et Mike Hewitt (2021), « Service network design », in : *Network Design with Applications to Transportation and Logistics*, sous la dir. de Teodor Gabriel Crainic, Michel Gendreau et Bernard Gendron, Springer, p. 347-382.
- Croxton, Keely L, Bernard Gendron et Thomas L Magnanti (2003), « A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems », in : *Management Science* 49.9, p. 1268-1273.
- Dabia, Said, David Lai et Daniele Vigo (2019), « An Exact Algorithm for a Rich Vehicle Routing Problem with Private Fleet and Common Carrier », in : *Transportation Science* 53.4, p. 986-1000.
- Demir, Emrah, Tolga Bektaş et Gilbert Laporte (2012), « An adaptive large neighborhood search heuristic for the pollution-routing problem », in : *European Journal of Operational Research* 223.2, p. 346-359.
- Desaulniers, Guy (2010), « Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows », in : *Operations Research* 58.1, p. 179-192.
- Dragomir, Alina G, Tom Van Woensel et Karl F Doerner (2022), « The pickup and delivery problem with alternative locations and overlapping time windows », in : *Computers & Operations Research*, p. 105758.
- Dueck, Gunter (1993), « New optimization heuristics : The great deluge algorithm and the record-to-record travel », in : *Journal of Computational physics* 104.1, p. 86-92.
- Dumez, Dorian, Fabien Lehuédé et Olivier Péton (2021a), « A large neighborhood search approach to the vehicle routing problem with delivery options », in : *Transportation Research Part B : Methodological* 144, p. 103-132.
- Dumez, Dorian, Christian Tilk, Stefan Irnich, Fabien Lehuédé et Olivier Péton (2021b), « Hybridizing large neighborhood search and exact methods for generalized vehicle

-
- routing problems with time windows », in : *EURO Journal on Transportation and Logistics* 10, p. 100040.
- Euchi, Jalel (2017), « The vehicle routing problem with private fleet and multiple common carriers : Solution with hybrid metaheuristic algorithm », in : *Vehicular Communications* 9, p. 97-108.
- Fachini, Ramon Faganello et Vinícius Amaral Armentano (2020), « Logic-based Benders decomposition for the heterogeneous fixed fleet vehicle routing problem with time windows », in : *Computers & Industrial Engineering* 148, p. 106641.
- Favaretto, Daniela, Elena Moretti et Paola Pellegrini (2007), « Ant colony system for a VRP with multiple time windows and multiple visits », in : *Journal of Interdisciplinary Mathematics* 10.2, p. 263-284.
- Frangioni, Antonio et Bernard Gendron (2021), « Piecewise Linear Cost Network Design », in : *Network Design with Applications to Transportation and Logistics*, sous la dir. de Teodor Gabriel Crainic, Michel Gendreau et Bernard Gendron, Springer, p. 167-185.
- Friedrich, Christian et Ralf Elbert (2022), « Adaptive large neighborhood search for vehicle routing problems with transshipment facilities arising in city logistics », in : *Computers & Operations Research* 137, p. 105491.
- Gahm, Christian, Christian Brabänder et Axel Tuma (2017), « Vehicle routing with private fleet, multiple common carriers offering volume discounts, and rental options », in : *Transportation Research Part E : Logistics and Transportation Review* 97, p. 192-216.
- Goeke, Dominik, Timo Gschwind et Michael Schneider (2019), « Upper and lower bounds for the vehicle-routing problem with private fleet and common carrier », in : *Discrete Applied Mathematics* 264, p. 43-61.
- Golden, Bruce, Arjang Assad, Larry Levy et Filip Gheysens (1984), « The fleet size and mix vehicle routing problem », in : *Computers & Operations Research* 11, p. 49-66.
- Grangier, Philippe, Michel Gendreau, Fabien Lehuédé et Louis-Martin Rousseau (2017), « A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking », in : *Computers & Operations Research* 84, p. 116-126.
- Groër, Chris, Bruce Golden et Edward Wasil (2011), « A parallel algorithm for the vehicle routing problem », in : *INFORMS Journal on Computing* 23.2, p. 315-330.
- Gschwind, Timo et Michael Drexl (2019), « Adaptive large neighborhood search with a constant-time feasibility test for the dial-a-ride problem », in : *Transportation Science* 53.2, p. 480-491.

-
- Gu, Wenjuan, Diego Cattaruzza, Maxime Ogier et Frédéric Semet (2019), « Adaptive large neighborhood search for the commodity constrained split delivery VRP », in : *Computers & Operations Research* 112, p. 104761.
- Hall, Randolph W. (1985), « Dependence between Shipment Size and Mode in Freight Transportation », in : *Transportation Science* 19.4, p. 436-444.
- Hoogeboom, Maaïke, Wout Dullaert, David Lai et Daniele Vigo (2020), « Efficient neighborhood evaluations for the vehicle routing problem with multiple time windows », in : *Transportation Science* 54.2, p. 400-416.
- Koç, Çağrı, Tolga Bektaş, Ola Jabali et Gilbert Laporte (2015), « A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows », in : *Computers & Operations Research* 64, p. 11-27.
- Koç, Çağrı, Tolga Bektaş, Ola Jabali et Gilbert Laporte (2016), « Thirty years of heterogeneous vehicle routing », in : *European Journal of Operational Research* 249.1, p. 1-21.
- Larsen, Rune et Dario Pacino (2019), « Fast delta evaluation for the vehicle routing problem with multiple time windows », in : *arXiv preprint arXiv :1905.04114*.
- Le Colleter, Théo, Dorian Dumez, Fabien Lehuédé et Olivier Péton (2022), « Small and Large Neighborhood Search for the Park-and-Loop Routing Problem with Parking Selection », working paper or preprint.
- Lee, Young Hae, Jung Woo Jung et Kyong Min Lee (2006), « Vehicle routing scheduling for cross-docking in the supply chain », in : *Computers & Industrial Engineering* 51.2, p. 247-256.
- Lindsey, Kathleen A, Alan L Erera et Martin WP Savelsbergh (2013), « A pickup and delivery problem using crossdocks and truckload lane rates », in : *EURO Journal on Transportation and Logistics* 2.1-2, p. 5-27.
- Liu, Fuh-Hwa et Sheng-Yuan Shen (1999), « The fleet size and mix vehicle routing problem with time windows », in : *Journal of the Operational Research society* 50.7, p. 721-732.
- Liu, Ran, Zhibin Jiang, Xiao Liu et Feng Chen (2010), « Task selection and routing problems in collaborative truckload transportation », in : *Transportation Research Part E : Logistics and Transportation Review* 46.6, p. 1071-1085.
- López-Ibañez, Manuel, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Mauro Birattari et Thomas Stützle (2016), « The irace package : Iterated racing for automatic algorithm configuration », in : *Operations Research Perspectives* 3, p. 43-58.

-
- Luxen, Dennis et Christian Vetter (2011), « Real-time routing with OpenStreetMap data », in : *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, Chicago, Illinois : ACM, p. 513-516.
- Mancini, Simona (2017), « A combined multistart random constructive heuristic and set partitioning based formulation for the vehicle routing problem with time dependent travel times », in : *Computers & Operations Research* 88, p. 290-296.
- Mancini, Simona et Margaretha Gansterer (2021), « Vehicle routing with private and shared delivery locations », in : *Computers & Operations Research* 133, p. 105361.
- Martello, Silvano (1990), « Knapsack problems : algorithms and computer implementations », in : *Wiley-Interscience series in discrete mathematics and optimization*.
- Medina, Juliette (2016), « Modèles et méthodes d'optimisation pour la mutualisation des chaînes logistiques », thèse de doct., Ecole des Mines de Nantes.
- Medina, Juliette, Mike Hewitt, Fabien Lehuédé et Olivier Péton (2019), « Integrating long-haul and local transportation planning : The service network design and routing problem », in : *EURO Journal on Transportation and Logistics* 8.2, p. 119-145.
- Mendoza, Jorge E et Juan G Villegas (2013), « A multi-space sampling heuristic for the vehicle routing problem with stochastic demands », in : *Optimization Letters* 7.7, p. 1503-1516.
- Moccia, Luigi, Jean-François Cordeau et Gilbert Laporte (2012), « An incremental tabu search heuristic for the generalized vehicle routing problem with time windows », in : *Journal of the Operational Research Society* 63.2, p. 232-244.
- Montreuil, Benoit, Russell D Meller et Eric Ballot (2013), « Physical internet foundations », in : *Service orientation in holonic and multi agent manufacturing and robotics*, Springer, p. 151-166.
- Nakao, Yoshitaka et Hiroshi Nagamochi (2007), « A DP-based heuristic algorithm for the discrete split delivery vehicle routing problem », in : *Journal of Advanced Mechanical Design, Systems, and Manufacturing* 1.2, p. 217-226.
- Ozbaygin, Gizem, Oya Ekin Karasan, Martin Savelsbergh et Hande Yaman (2017), « A branch-and-price algorithm for the vehicle routing problem with roaming delivery locations », in : *Transportation Research Part B : Methodological* 100, p. 115-137.
- Paraskevopoulos, Dimitris C, Panagiotis P Repoussis, Christos D Tarantilis, George Ioannou et Gregory P Prastacos (2008), « A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows », in : *Journal of Heuristics* 14.5, p. 425-455.

-
- Parragh, Sophie N et Verena Schmid (2013), « Hybrid column generation and large neighborhood search for the dial-a-ride problem », in : *Computers & Operations Research* 40.1, p. 490-497.
- Pisinger, David et Stefan Ropke (2007), « A general heuristic for vehicle routing problems », in : *Computers & operations research* 34.8, p. 2403-2435.
- Pisinger, David et Stefan Ropke (2019), « Large neighborhood search », in : *Handbook of metaheuristics*, sous la dir. de Michel Gendreau et Jean-Yves Potvin, Springer, p. 99-127.
- Potvin, Jean-Yves et Marc-André Naud (2011), « Tabu search with ejection chains for the vehicle routing problem with private fleet and common carrier », in : *Journal of the Operational Research Society* 62.2, p. 326-336.
- Prescott-Gagnon, Eric, Guy Desaulniers et Louis-Martin Rousseau (2009), « A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows », in : *Networks* 54.4, p. 190-204.
- Prins, Christian (2009), « Two memetic algorithms for heterogeneous fleet vehicle routing problems », in : *Engineering Applications of Artificial Intelligence* 22.6, p. 916-928.
- Régin, Jean-Charles (1996), « Generalized arc consistency for global cardinality constraint », in : *American Association for Artificial Intelligence (AAAI'96)*, p. 209-215.
- Reyes, Damián, Martin Savelsbergh et Alejandro Toriello (2017), « Vehicle routing with roaming delivery locations », in : *Transportation Research Part C : Emerging Technologies* 80, p. 71-91.
- Ropke, Stefan et David Pisinger (2006a), « A unified heuristic for a large class of vehicle routing problems with backhauls », in : *European Journal of Operational Research* 171.3, p. 750-775.
- Ropke, Stefan et David Pisinger (2006b), « An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows », in : *Transportation science* 40.4, p. 455-472.
- Santos, Fernando Afonso, Geraldo Robson Mateus et Alexandre Salles da Cunha (2013), « The Pickup and Delivery Problem with Cross-Docking », in : *Computers & Operations Research* 40.4, p. 1085-1093.
- Savelsbergh, Martin WP (1985), « Local search in routing problems with time windows », in : *Annals of Operations research* 4.1, p. 285-305.
- Savelsbergh, Martin WP (1992), « The vehicle routing problem with time windows : Minimizing route duration », in : *ORSA journal on computing* 4.2, p. 146-154.

-
- Savelsbergh, Martin WP et Marc Sol (1995), « The general pickup and delivery problem », in : *Transportation science* 29.1, p. 17-29.
- Schrumpf, Gerhard, Johannes Schneider, Hermann Stamm-Wilbrandt et Gunter Dueck (2000), « Record breaking optimization results using the ruin and recreate principle », in : *Journal of Computational Physics* 159.2, p. 139-171.
- Shaw, Paul (1998), « Using constraint programming and local search methods to solve vehicle routing problems », in : *International conference on principles and practice of constraint programming*, Springer, p. 417-431.
- Sitek, Pawel et Jarosław Wikarek (2019), « Capacitated vehicle routing problem with pick-up and alternative delivery (CVRPPAD) : model and implementation using hybrid approach », in : *Annals of Operations Research* 273.1, p. 257-277.
- Solomon, Marius M (1987), « Algorithms for the vehicle routing and scheduling problems with time window constraints », in : *Operations research* 35.2, p. 254-265.
- Stenger, Andreas, Daniele Vigo, Steffen Enz et Michael Schwind (2013), « An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping », in : *Transportation Science* 47.1, p. 64-80.
- Subramanian, Anand, Eduardo Uchoa et Luiz Satoru Ochi (2013), « A hybrid algorithm for a class of vehicle routing problems », in : *Computers & Operations Research* 40.10, p. 2519-2531.
- Taillard, E. D. (1999), « A heuristic column generation method for the heterogeneous fleet VRP », in : *RAIRO - Operations Research* 33.1, p. 1-14.
- Tellez, Oscar, Samuel Vercraene, Fabien Lehuédé, Olivier Péton et Thibaud Monteiro (2018), « The fleet size and mix dial-a-ride problem with reconfigurable vehicle capacity », in : *Transportation Research Part C : Emerging Technologies* 91, p. 99-123.
- The Shift Project (2022), *Assurer le fret dans un monde fini*, https://theshiftproject.org/wp-content/uploads/2022/03/Fret_rapport-final_ShiftProject_PTEF.pdf.
- Tilk, Christian, Katharina Olkis et Stefan Irnich (2021), « The last-mile vehicle routing problem with delivery options », in : *OR Spectrum* 43.4, p. 877-904.
- Toth, Paolo et Daniele Vigo (2014), *Vehicle routing : problems, methods, and applications*, SIAM.
- Tricoire, Fabien, Martin Romauch, Karl F Doerner et Richard F Hartl (2010), « Heuristics for the multi-period orienteering problem with multiple time windows », in : *Computers & Operations Research* 37.2, p. 351-367.

-
- Vidal, Thibaut, Teodor Gabriel Crainic, Michel Gendreau et Christian Prins (2014), « A unified solution framework for multi-attribute vehicle routing problems », in : *European Journal of Operational Research* 234.3, p. 658-673.
- Vidal, Thibaut, Nelson Maculan, Luiz Satoru Ochi et Puca Huachi Vaz Penna (2015), « Large Neighborhoods with Implicit Customer Selection for Vehicle Routing Problems with Profits », in : *Transportation Science*, p. 16.
- Wen, M, J Larsen, J Clausen, J-F Cordeau et G Laporte (2009), « Vehicle routing with cross-docking », in : *Journal of the Operational Research Society* 60.12, p. 1708-1718.
- Yıldırım, Umman Mahir et Bülent Çatay (2015), « An ant colony-based matheuristic approach for solving a class of vehicle routing problems », in : *International Conference on Computational Logistics (ICCL)*, Springer, p. 105-119.
- Yuan, Yuan, Diego Cattaruzza, Maxime Ogier, Frédéric Semet et Daniele Vigo (2021), « A column generation based heuristic for the generalized vehicle routing problem with time windows », in : *Transportation Research Part E : Logistics and Transportation Review* 152, p. 102391.
- Zhou, Lin, Roberto Baldacci, Daniele Vigo et Xu Wang (2018), « A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution », in : *European Journal of Operational Research* 265.2, p. 765-778.

RÉSULTATS DE LA MÉTHODE LARGE NEIGHBORHOOD SEARCH

A.1 Résultats détaillés de la méthode LNS sur les instances VRPPC et VRPFLC

Instance	z^*	HFF-VRP-TW-PC					VRP-FLC				
		z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL	z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL
R101A	5053.9	5054.2	5060.2	$A^3B^3C^3D^3E^2$	2560	1390	5138.9	5153.0	$A^1B^3C^3D^3E^2$	2460	1695
R102A		4585.5	4608.8	$A^2B^3C^3D^3E^3$	3010	365	4683.1	4717.7	$A^1B^3C^3D^2E^3$	2710	1005
R103A		4353.1	4394.4	$A^3B^3C^3D^3E^3$	3060	190	4546.4	4571.6	$A^1B^3C^3D^2E^3$	2710	1005
R104A		4267.7	4296.3	$A^3B^3C^3D^3E^3$	3060	125	4519.0	4549.1	$A^1B^3C^3D^2E^3$	2710	990
R105A		4622.0	4657.6	$A^2B^3C^3D^3E^3$	3010	430	4753.4	4782.0	$A^1B^3C^3D^2E^3$	2710	1400
R106A		4403.4	4442.8	$A^2B^3C^3D^3E^3$	3010	270	4610.1	4635.0	$A^1B^3C^3D^2E^3$	2710	1000
R107A		4298.0	4333.2	$A^3B^3C^3D^3E^3$	3060	155	4532.0	4551.2	$A^1B^3C^3D^2E^3$	2710	1005
R108A		4242.7	4271.8	$A^3B^3C^3D^3E^3$	3060	115	4509.4	4529.6	$A^1B^3C^3D^2E^3$	2710	1000
R109A		4393.8	4427.9	$A^3B^3C^3D^3E^3$	3060	160	4562.1	4615.8	$A^1B^3C^3D^2E^3$	2710	1005
R110A		4327.5	4360.3	$A^3B^3C^3D^3E^3$	3060	155	4550.7	4577.6	$A^1B^3C^3D^2E^3$	2710	1005
R111A		4311.3	4336.8	$A^3B^3C^3D^3E^3$	3060	155	4531.0	4543.3	$A^1B^3C^3D^2E^3$	2710	990
R112A		4262.1	4282.4	$A^3B^3C^3D^3E^3$	3060	115	4502.0	4540.4	$A^1B^3C^3D^2E^3$	2710	1010
R201A		3668.6	3726.3	A^2B^2	2300	310	3716.2	3775.4	A^2B^2	2300	300
R202A		3526.5	3601.2	A^2B^2	2300	295	3507.3	3584.0	A^2B^2	2300	290
R203A		3436.7	3476.7	A^2B^2	2300	295	3412.4	3468.9	A^2B^2	2300	10
R204A		3285.6	3317.2	A^1B^3	2550	10	3300.7	3321.9	A^1B^3	2550	5
R205A		3520.4	3551.5	A^2B^2	2300	305	3471.1	3528.6	A^2B^2	2300	305
R206A		3409.8	3462.2	A^2B^2	2300	290	3438.3	3473.9	A^2B^2	2300	15
R207A		3375.3	3399.4	A^1B^3	2550	0	3363.5	3391.0	A^1B^3	2550	10
R208A		3264.9	3280.6	A^1B^3	2550	5	3270.5	3276.1	A^1B^3	2550	5
R209A		3438.1	3452.2	A^1B^3	2550	5	3404.6	3452.5	A^2B^2	2300	10
R210A		3436.4	3489.6	A^2B^2	2300	300	3436.1	3486.4	A^2B^2	2300	15
R211A		3314.5	3342.3	A^1B^3	2550	5	3309.9	3328.3	A^1B^3	2550	5
RC101A	5161.7	5161.7	5168.1	$A^3B^3C^3D^3$	2880	1144.5	5186.5	5190.3	$A^2B^3C^3D^2$	2370	1977.5
RC102A	5012.2	5016.5	5025.0	$A^3B^3C^3D^3$	2880	1123.5	5115.4	5126.2	$A^2B^3C^3D^2$	2370	1949.5
RC103A		4954.5	4967.2	$A^3B^3C^3D^3$	2880	1120	5073.4	5082.5	$A^2B^3C^3D^2$	2370	1949.5

HFF-VRP-TW-PC											VRP-FLC				
Instance	z^*	z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL	z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL				
RC104A		4908.0	4927.6	$A^3B^3C^3D^3$	2880	1113	5043.2	5050.1	$A^2B^3C^2D^3$	2520	1792				
RC105A	5046.1	5046.1	5059.3	$A^3B^3C^3D^3$	2880	1144.5	5131.2	5137.4	$A^2B^3C^3D^2$	2370	1956.5				
RC106A	4980.9	4980.9	4995.2	$A^3B^3C^3D^3$	2880	1113	5084.2	5093.4	$A^2B^3C^3D^2$	2370	1946				
RC107A		4930.4	4944.1	$A^3B^3C^3D^3$	2880	1113	5050.1	5060.9	$A^2B^3C^2D^3$	2520	1771				
RC108A		4904.3	4916.9	$A^3B^3C^3D^3$	2880	1113	5040.1	5048.3	$A^2B^3C^2D^3$	2520	1778				
RC201A		4385.9	4423.9	$A^3B^2C^3$	2800	451.5	4484.9	4547.4	$A^2B^2C^3$	2650	451.5				
RC202A		4239.4	4299.6	$A^3B^2C^3$	2800	458.5	4370.1	4427.2	$A^1B^3C^3$	2850	462				
RC203A		4087.9	4142.7	$A^2B^3C^3$	3000	108.5	4173.9	4237.7	$A^3B^2C^2D^1$	3050	115.5				
RC204A		3983.6	4033.4	$A^2B^3C^3$	3000	101.5	4046.0	4112.8	$A^3B^2C^2D^1$	3050	115.5				
RC205A		4262.6	4332.2	$A^3B^2C^3$	2800	451.5	4365.3	4424.5	$A^1B^3C^3$	2850	455				
RC206A		4210.3	4261.3	$A^2B^3C^3$	3000	101.5	4281.9	4363.4	$A^1B^3C^3$	2850	444.5				
RC207A		4135.3	4186.2	$A^2B^3C^3$	3000	101.5	4210.9	4308.1	$A^1B^3C^3$	2850	444.5				
RC208A		3980.7	4010.8	$A^2B^3C^3$	3000	91	4108.5	4156.8	$A^1B^3C^3$	2850	448				
C101A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620				
C102A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620				
C103A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620				
C104A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620				
C105A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620				
C106A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620				
C107A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620				
C108A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620				
C109A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620				
C201A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620				
C202A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620				
C203A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620				
C204A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620				
C205A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620				
C206A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620				
C207A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620				
C208A		3620.0	3620.0		0	3620	3620.0	3620.0		0	3620				
R101B		2647.7	2660.6	$A^3B^3C^3D^3E^3$	612	840	3170.9	3176.6	$A^1B^3C^3D^3E^2$	492	1650				
R102B		2129.7	2155.3	$A^3B^3C^3D^3E^3$	612	220	2515.6	2542.7	$A^1B^3C^3D^2E^3$	542	1025				
R103B		1886.0	1926.8	$A^3B^3C^3D^3E^3$	612	180	2374.4	2408.0	$A^1B^3C^3D^2E^3$	542	1000				
R104B		1794.1	1812.9	$A^3B^3C^3D^3E^3$	612	115	2332.5	2360.4	$A^1B^3C^3D^2E^3$	542	995				
R105B		2163.7	2200.6	$A^3B^3C^3D^3E^3$	612	225	2598.7	2619.5	$A^1B^3C^3D^2E^3$	542	1060				
R106B		1951.8	1994.2	$A^3B^3C^3D^3E^3$	612	185	2433.8	2456.8	$A^1B^3C^3D^2E^3$	542	995				
R107B		1832.0	1874.2	$A^3B^3C^3D^3E^3$	612	140	2360.5	2387.2	$A^1B^3C^3D^2E^3$	542	1005				
R108B		1779.6	1821.1	$A^3B^3C^3D^3E^3$	612	135	2332.2	2359.4	$A^1B^3C^3D^2E^3$	542	1005				
R109B		1927.1	1957.8	$A^3B^3C^3D^3E^3$	612	165	2424.9	2443.3	$A^1B^3C^3D^2E^3$	542	1005				
R110B		1862.3	1906.1	$A^3B^3C^3D^3E^3$	612	125	2394.0	2409.4	$A^1B^3C^3D^2E^3$	542	1005				
R111B		1844.7	1870.0	$A^3B^3C^3D^3E^3$	612	135	2356.2	2370.7	$A^1B^3C^3D^2E^3$	542	995				
R112B		1792.6	1841.8	$A^3B^3C^3D^3E^3$	612	130	2348.7	2366.7	$A^1B^3C^3D^2E^3$	542	1000				
R201B		1725.1	1740.5	A^3B^2	550	30	1716.1	1732.5	A^3B^2	550	30				
R202B		1562.0	1595.9	A^3B^2	550	30	1561.0	1587.6	A^1B^3	510	30				
R203B		1412.8	1450.4	A^1B^3	510	0	1415.7	1451.4	A^1B^3	510	10				

HFF-VRP-TW-PC											VRP-FLC				
Instance	z^*	z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL	z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL				
R204B		1266.5	1307.3	A^3C^1	510	5	1251.3	1290.6	A^1B^3	510	5				
R205B		1490.5	1536.1	A^1B^3	510	15	1489.6	1517.8	A^1B^3	510	20				
R206B		1399.9	1429.8	A^1B^3	510	15	1410.8	1435.9	A^1B^3	510	15				
R207B		1337.0	1376.3	A^3C^1	510	0	1339.6	1375.7	A^3C^1	510	10				
R208B		1250.4	1293.1	A^3C^1	510	5	1230.6	1279.6	A^1B^3	510	5				
R209B		1384.5	1427.1	A^1B^3	510	15	1379.0	1405.9	A^1B^3	510	10				
R210B		1440.0	1471.9	A^1B^3	510	15	1438.9	1468.9	A^1B^3	510	15				
R211B		1276.8	1331.3	A^1B^3	510	5	1274.1	1305.8	A^1B^3	510	20				
RC101B		2857.7	2865.4	$A^3B^3C^3D^3$	576	1144.5	3211.1	3213.6	$A^2B^3C^2D^3$	504	1802.5				
RC102B		2710.7	2715.3	$A^3B^3C^3D^3$	576	1116.5	3109.5	3117.7	$A^2B^3C^2D^3$	504	1767.5				
RC103B		2650.5	2658.6	$A^3B^3C^3D^3$	576	1127	3059.5	3070.5	$A^2B^3C^2D^3$	504	1764				
RC104B		2607.4	2622.2	$A^3B^3C^3D^3$	576	1116.5	3027.7	3034.3	$A^2B^3C^2D^3$	504	1785				
RC105B		2749.6	2760.0	$A^3B^3C^3D^3$	576	1116.5	3123.8	3134.7	$A^2B^3C^2D^3$	504	1785				
RC106B		2678.0	2702.3	$A^3B^3C^3D^3$	576	1113	3076.5	3088.5	$A^2B^3C^2D^3$	504	1774.5				
RC107B		2630.4	2636.5	$A^3B^3C^3D^3$	576	1120	3044.1	3049.9	$A^2B^3C^2D^3$	504	1785				
RC108B		2598.1	2611.3	$A^3B^3C^3D^3$	576	1109.5	3023.1	3032.0	$A^2B^3C^2D^3$	504	1771				
RC201B		1952.8	1991.3	$A^3B^3C^3$	630	0	1972.0	1993.1	$A^2B^3C^2D^1$	650	17.5				
RC202B		1794.0	1819.2	$A^2B^3C^2D^1$	650	0	1787.2	1815.8	$A^1B^2C^3D^1$	660	0				
RC203B		1635.2	1648.5	$A^1B^2C^3D^1$	660	0	1632.1	1656.8	C^2D^3	700	0				
RC204B		1512.5	1522.5	C^2D^3	700	0	1512.5	1532.4	C^2D^3	700	0				
RC205B		1845.5	1863.4	$A^3B^3C^3$	630	0	1839.3	1880.9	$A^2B^3C^2D^1$	650	73.5				
RC206B		1748.0	1788.4	$A^1B^2C^3D^1$	660	7	1750.6	1789.8	$A^1C^3D^2$	680	0				
RC207B		1667.3	1690.0	$A^1B^2C^3D^1$	660	0	1660.8	1695.4	$A^1B^2C^3D^1$	660	0				
RC208B		1490.1	1508.4	C^2D^3	700	0	1491.9	1510.5	C^2D^3	700	0				
C101B		2609.8	2609.8	$A^3B^3C^3$	1470	400	2695.0	2695.2	$A^2B^3C^3$	1410	640				
C102B		2541.7	2550.0	$A^3B^3C^3$	1470	260	2609.7	2615.9	$A^2B^3C^3$	1410	480				
C103B		2471.4	2484.2	$A^3B^3C^3$	1470	160	2533.8	2544.0	$A^2B^3C^3$	1410	360				
C104B		2413.2	2429.5	$A^3B^3C^3$	1470	160	2424.1	2472.5	$A^2B^3C^3$	1410	280				
C105B		2548.8	2548.8	$A^3B^3C^3$	1470	340	2630.3	2630.3	$A^2B^3C^3$	1410	540				
C106B		2570.1	2570.9	$A^3B^3C^3$	1470	340	2651.6	2651.6	$A^2B^3C^3$	1410	560				
C107B		2537.7	2551.0	$A^3B^3C^3$	1470	280	2616.0	2618.4	$A^2B^3C^3$	1410	500				
C108B		2527.2	2531.7	$A^3B^3C^3$	1470	260	2575.5	2588.8	$A^2B^3C^3$	1410	460				
C109B		2455.7	2482.0	$A^3B^3C^3$	1470	160	2512.7	2525.6	$A^2B^3C^3$	1410	340				
C201B		1663.0	1675.8	A^2B^2	960	20	1660.8	1677.7	A^2B^2	960	20				
C202B		1656.1	1661.0	A^2B^2	960	40	1655.4	1673.4	A^2B^2	960	40				
C203B		1637.7	1646.6	A^2B^2	960	20	1638.4	1646.2	A^2B^2	960	20				
C204B		1634.6	1657.6	A^2B^2	960	20	1632.2	1642.6	A^2B^2	960	40				
C205B		1653.5	1663.6	A^2B^2	960	40	1654.9	1676.3	A^2B^2	960	20				
C206B		1652.1	1679.9	A^2B^2	960	20	1652.8	1667.4	A^2B^2	960	20				
C207B		1643.3	1685.5	A^2B^2	960	20	1643.3	1643.6	A^2B^2	960	20				
C208B		1649.5	1679.3	A^2B^2	960	20	1649.5	1665.2	A^2B^2	960	20				
R101C	2341.7	2341.7	2344.3	$A^3B^3C^3D^3E^3$	306	840	2901.7	2908.1	$A^1B^3C^3D^2E^3$	271	1645				
R102C		1825.8	1855.1	$A^3B^3C^3D^3E^3$	306	255	2241.2	2269.6	$A^1B^3C^3D^2E^3$	271	1015				
R103C		1585.6	1629.2	$A^3B^3C^3D^3E^3$	306	165	2103.9	2129.2	$A^1B^3C^3D^2E^3$	271	1005				
R104C		1492.7	1518.0	$A^3B^3C^3D^3E^3$	306	130	2067.8	2092.3	$A^1B^3C^3D^2E^3$	271	990				

HFF-VRP-TW-PC							VRP-FLC				
Instance	z^*	z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL	z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL
R105C		1845.1	1884.3	$A^3B^3C^3D^3E^3$	306	230	2326.9	2362.2	$A^1B^3C^3D^2E^3$	271	1040
R106C		1643.8	1679.8	$A^3B^3C^3D^3E^3$	306	190	2151.4	2177.8	$A^1B^3C^3D^2E^3$	271	1005
R107C		1516.5	1551.6	$A^3B^3C^3D^3E^3$	306	110	2092.2	2125.8	$A^1B^3C^3D^2E^3$	271	1000
R108C		1478.9	1505.6	$A^3B^3C^3D^3E^3$	306	140	2067.2	2075.7	$A^1B^3C^3D^2E^3$	271	995
R109C		1637.5	1663.3	$A^3B^3C^3D^3E^3$	306	135	2128.6	2166.5	$A^1B^3C^3D^2E^3$	271	1005
R110C		1551.6	1593.6	$A^3B^3C^3D^3E^3$	306	150	2116.5	2134.2	$A^1B^3C^3D^2E^3$	271	1010
R111C		1528.0	1555.5	$A^3B^3C^3D^3E^3$	306	140	2087.9	2110.9	$A^1B^3C^3D^2E^3$	271	995
R112C		1491.2	1524.6	$A^3B^3C^3D^3E^3$	306	135	2069.7	2095.0	$A^1B^3C^3D^2E^3$	271	1005
R201C		1444.5	1460.3	A^3B^2	275	30	1439.1	1457.8	A^3B^2	275	15
R202C		1293.9	1327.8	A^3B^2	275	30	1296.4	1314.9	A^3B^2	275	30
R203C		1156.8	1187.4	A^1B^3	255	10	1177.7	1204.5	A^3B^2	275	10
R204C		1001.4	1041.7	A^1B^3	255	5	1010.8	1037.7	A^1B^3	255	5
R205C		1227.0	1264.6	A^1B^3	255	15	1230.2	1243.6	A^3B^2	275	15
R206C		1136.9	1160.2	A^3C^1	255	15	1153.8	1178.3	A^1B^3	255	15
R207C		1085.3	1126.4	A^1B^3	255	0	1078.1	1120.8	A^1B^3	255	0
R208C		1004.4	1022.2	A^1C^2	285	5	999.8	1024.5	A^3C^1	255	5
R209C		1127.4	1171.1	A^1B^3	255	20	1127.9	1149.1	A^1B^3	255	0
R210C		1190.2	1208.4	A^1B^3	255	15	1180.6	1191.1	A^1B^3	255	20
R211C		1035.2	1086.6	A^1B^3	255	15	1027.8	1051.7	A^1B^3	255	10
RC101C	2569.7	2569.7	2574.0	$A^3B^3C^3D^3$	288	1144.5	2959.1	2962.4	$A^2B^3C^2D^3$	252	1802.5
RC102C	2420.2	2422.7	2431.3	$A^3B^3C^3D^3$	288	1116.5	2861.4	2867.3	$A^2B^3C^2D^3$	252	1778
RC103C		2361.5	2378.4	$A^3B^3C^3D^3$	288	1113	2810.2	2817.2	$A^2B^3C^2D^3$	252	1774.5
RC104C		2321.2	2329.5	$A^3B^3C^3D^3$	288	1109.5	2773.0	2781.8	$A^2B^3C^2D^3$	252	1774.5
RC105C	2454.1	2461.8	2471.2	$A^3B^3C^3D^3$	288	1127	2871.8	2880.0	$A^2B^3C^2D^3$	252	1785
RC106C	2388.9	2404.2	2420.2	$A^3B^3C^3D^3$	288	1113	2822.7	2832.9	$A^2B^3C^2D^3$	252	1785
RC107C		2341.6	2348.9	$A^3B^3C^3D^3$	288	1120	2781.3	2796.1	$A^2B^3C^2D^3$	252	1778
RC108C		2319.2	2333.2	$A^3B^3C^3D^3$	288	1113	2770.9	2776.7	$A^2B^3C^2D^3$	252	1767.5
RC201C		1625.8	1645.8	$A^1B^2C^3D^1$	330	21	1637.8	1651.9	$A^2B^3C^2D^1$	325	10.5
RC202C		1444.1	1471.4	$A^1B^2C^3D^1$	330	17.5	1457.3	1480.9	$A^1B^2C^3D^1$	330	17.5
RC203C		1280.4	1300.2	C^2D^3	350	0	1280.4	1299.9	C^2D^3	350	0
RC204C		1163.6	1185.9	$C^3D^1E^1$	355	0	1163.6	1188.5	$C^3D^1E^1$	355	0
RC205C		1505.9	1523.5	$A^1B^2C^3D^1$	330	28	1506.3	1532.7	$A^1B^2C^3D^1$	330	28
RC206C		1418.2	1447.3	$A^1C^3D^2$	340	24.5	1410.6	1442.2	$A^1C^3D^2$	340	24.5
RC207C		1336.4	1351.5	$A^1C^3D^2$	340	0	1337.5	1357.5	$A^1C^3D^2$	340	7
RC208C		1146.2	1186.8	C^2D^3	350	0	1147.1	1186.6	C^2D^3	350	0
C101C	1874.8	1874.8	1874.8	$A^3B^3C^3$	735	400	1990.0	1990.9	$A^2B^3C^3$	705	640
C102C		1806.7	1820.4	$A^3B^3C^3$	735	260	1904.7	1910.7	$A^2B^3C^3$	705	460
C103C		1736.4	1754.9	$A^3B^3C^3$	735	160	1837.2	1854.2	$A^2B^3C^3$	705	380
C104C		1670.7	1703.4	$A^3B^3C^3$	735	140	1758.5	1776.5	$A^2B^3C^3$	705	300
C105C	1813.8	1813.8	1813.8	$A^3B^3C^3$	735	340	1925.3	1925.3	$A^2B^3C^3$	705	540
C106C	1835.1	1835.1	1835.1	$A^3B^3C^3$	735	340	1946.6	1946.6	$A^2B^3C^3$	705	560
C107C	1802.7	1802.7	1810.2	$A^3B^3C^3$	735	280	1911.0	1919.0	$A^2B^3C^3$	705	500
C108C	1792.6	1792.2	1794.1	$A^3B^3C^3$	735	260	1887.0	1890.7	$A^2B^3C^3$	705	460
C109C		1720.7	1746.0	$A^3B^3C^3$	735	160	1807.7	1821.9	$A^2B^3C^3$	705	340
C201C		1183.0	1183.6	A^2B^2	480	20	1182.1	1184.5	A^2B^2	480	40

Instance	z^*	HFF-VRP-TW-PC					VRP-FLC				
		z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL	z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL
C202C		1172.1	1193.3	A^3C^1	500	20	1172.1	1175.4	A^3C^1	500	40
C203C		1157.7	1176.3	A^2B^2	480	20	1158.4	1163.5	A^2B^2	480	20
C204C		1148.5	1159.4	A^2B^2	480	20	1151.1	1158.3	A^2B^2	480	20
C205C		1174.9	1179.7	A^2B^2	480	20	1174.9	1176.2	A^2B^2	480	20
C206C		1173.7	1194.7	A^2B^2	480	20	1172.1	1174.3	A^2B^2	480	20
C207C		1163.3	1202.2	A^2B^2	480	20	1163.3	1163.6	A^2B^2	480	20
C208C		1169.5	1184.0	A^2B^2	480	20	1169.5	1174.7	A^2B^2	480	20

TABLEAU A.1 – Résultats du LNS sur les instances HFF-VRP-TW-PC et VRP-FLC avec coûts LTL élevés

HFF-VRP-TW-PC											VRP-FLC				
Instance	z^*	z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL	z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL				
R101A	4376.8	4378.8	4384.3	$A^3B^3C^3D^3$	1560	2187.5	4409.5	4422.9	$A^1B^3C^3D^3$	1460.0	2387				
R102A		4283.9	4291.5	$A^2B^3C^3D^3E^1$	2010	1585.5	4312.5	4326.0	$A^1B^3C^3D^3E^1$	1960.0	1008				
R103A		4198.9	4218.4	$A^2B^3C^3D^3E^2$	2510	906.5	4221.9	4238.2	$A^1B^3C^3D^3E^2$	2460.0	987				
R104A		4189.0	4200.4	$A^3B^3C^3D^3E^2$	2560	773.5	4194.3	4214.9	$A^1B^3C^3D^3E^2$	2460.0	980				
R105A		4309.6	4317.1	4334.6	$A^3B^3C^3D^3E^1$	2060	1494.5	4345.8	4365.2	$A^1B^3C^3D^3E^1$	1960.0	1683.5			
R106A			4227.7	4265.2	$A^2B^3C^3D^3E^2$	2510	882	4263.3	4278.9	$A^1B^3C^3D^3E^2$	2460.0	990.5			
R107A			4201.5	4214.0	$A^3B^3C^3D^3E^2$	2560	787.5	4210.3	4233.7	$A^1B^3C^3D^3E^2$	2460.0	987			
R108A		4206.6	4167.9	4191.2	$A^3B^3C^3D^3E^2$	2560	773.5	4191.0	4217.3	$A^1B^3C^3D^3E^2$	2460.0	973			
R109A			4231.8	4262.1	$A^3B^3C^3D^3E^2$	2560	787.5	4246.9	4273.6	$A^1B^3C^3D^3E^2$	2460.0	980			
R110A			4197.8	4225.9	$A^2B^3C^3D^3E^2$	2510	885.5	4229.2	4244.2	$A^1B^3C^3D^3E^2$	2460.0	987			
R111A			4195.9	4213.5	$A^3B^3C^3D^3E^2$	2560	784	4212.7	4224.8	$A^1B^3C^3D^3E^2$	2460.0	990.5			
R112A			4169.2	4194.8	$A^2B^3C^3D^3E^2$	2510	882	4192.5	4213.1	$A^1B^3C^3D^3E^2$	2460.0	980			
R201A			3518.8	3605.5	A^3B^1	2050	560	3535.5	3631.7	A^3B^1	2050.0	567			
R202A		3390.0	3463.0	A^3B^1	2050	560	3463.3	3512.8	A^3B^1	2050.0	556.5				
R203A		3318.7	3358.1	A^2B^2	2300	210	3320.1	3422.3	A^3B^1	2050.0	556.5				
R204A		3187.8	3250.9	A^2B^2	2300	203	3213.3	3251.3	A^2B^2	2300.0	206.5				
R205A		3406.7	3500.9	A^3B^1	2050	560	3383.4	3423.9	A^3B^1	2050.0	217				
R206A		3296.0	3368.2	A^3B^1	2050	556.5	3286.1	3359.1	A^2B^2	2300.0	553				
R207A		3253.5	3276.2	A^3B^1	2050	556.5	3239.3	3295.0	A^2B^2	2300.0	220.5				
R208A		3186.5	3220.6	A^2B^2	2300	210	3155.1	3193.6	A^2B^2	2300.0	203				
R209A		3303.4	3351.0	A^2B^2	2300	206.5	3294.4	3361.4	A^2B^2	2300.0	210				
R210A		3305.0	3363.2	A^2B^2	2300	206.5	3327.6	3395.3	A^2B^2	2300.0	553				
R211A		3214.9	3245.5	A^2B^2	2300	203	3187.2	3220.6	A^2B^2	2300.0	206.5				
RC101A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448				
RC102A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448				
RC103A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448				
RC104A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448				
RC105A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448				
RC106A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448				
RC107A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448				
RC108A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448				
RC201A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448				
RC202A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448				
RC203A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448				
RC204A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448				
RC205A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448				
RC206A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448				
RC207A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448				
RC208A		3448.0	3448.0		0	3448	3448.0	3448.0		0.0	3448				
C101A		905.0	905.0		0	905	905.0	905.0		0.0	905				
C102A		905.0	905.0		0	905	905.0	905.0		0.0	905				
C103A		905.0	905.0		0	905	905.0	905.0		0.0	905				
C104A		905.0	905.0		0	905	905.0	905.0		0.0	905				
C105A		905.0	905.0		0	905	905.0	905.0		0.0	905				
C106A		905.0	905.0		0	905	905.0	905.0		0.0	905				
C107A		905.0	905.0		0	905	905.0	905.0		0.0	905				

HFF-VRP-TW-PC											VRP-FLC				
Instance	z^*	z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL	z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL				
C108A		905.0	905.0		0	905	905.0	905.0		0.0	905				
C109A		905.0	905.0		0	905	905.0	905.0		0.0	905				
C201A		905.0	905.0		0	905	905.0	905.0		0.0	905				
C202A		905.0	905.0		0	905	905.0	905.0		0.0	905				
C203A		905.0	905.0		0	905	905.0	905.0		0.0	905				
C204A		905.0	905.0		0	905	905.0	905.0		0.0	905				
C205A		905.0	905.0		0	905	905.0	905.0		0.0	905				
C206A		905.0	905.0		0	905	905.0	905.0		0.0	905				
C207A		905.0	905.0		0	905	905.0	905.0		0.0	905				
C208A		905.0	905.0		0	905	905.0	905.0		0.0	905				
R101B		2393.3	2393.8	$A^3B^3C^3D^3E^3$	612	602	2655.0	2656.8	$A^1B^3C^3D^3E^2$	492.0	1274				
R102B		2044.4	2060.9	$A^3B^3C^3D^3E^3$	612	220.5	2200.4	2243.5	$A^1B^3C^3D^2E^3$	542.0	749				
R103B		1843.8	1869.9	$A^3B^3C^3D^3E^3$	612	154	2083.8	2103.7	$A^1B^3C^3D^2E^3$	542.0	703.5				
R104B		1756.4	1777.5	$A^3B^3C^3D^3E^3$	612	122.5	2042.9	2061.8	$A^1B^3C^3D^2E^3$	542.0	696.5				
R105B		2083.8	2102.9	$A^3B^3C^3D^3E^3$	612	241.5	2273.3	2295.9	$A^1B^3C^3D^2E^3$	542.0	801.5				
R106B		1906.9	1934.6	$A^3B^3C^3D^3E^3$	612	140	2129.4	2144.9	$A^1B^3C^3D^2E^3$	542.0	717.5				
R107B		1786.9	1822.5	$A^3B^3C^3D^3E^3$	612	101.5	2062.6	2088.7	$A^1B^3C^3D^2E^3$	542.0	700				
R108B		1742.6	1754.9	$A^3B^3C^3D^3E^3$	612	91	2037.3	2053.7	$A^1B^3C^3D^2E^3$	542.0	696.5				
R109B		1863.7	1886.3	$A^3B^3C^3D^3E^3$	612	168	2099.0	2129.3	$A^1B^3C^3D^2E^3$	542.0	717.5				
R110B		1817.0	1845.7	$A^3B^3C^3D^3E^3$	612	143.5	2081.8	2096.3	$A^1B^3C^3D^2E^3$	542.0	703.5				
R111B		1793.8	1817.5	$A^3B^3C^3D^3E^3$	612	119	2045.1	2079.0	$A^1B^3C^3D^2E^3$	542.0	703.5				
R112B		1734.1	1766.6	$A^3B^3C^3D^3E^3$	612	108.5	2047.9	2068.5	$A^1B^3C^3D^2E^3$	542.0	707				
R201B		1710.2	1757.0	A^3B^2	550	73.5	1718.0	1728.9	A^3B^2	550.0	63				
R202B		1542.9	1601.1	A^1B^3	510	45.5	1544.6	1572.8	A^1B^3	510.0	73.5				
R203B		1409.9	1443.0	A^3C^1	510	10.5	1414.1	1441.6	A^3C^1	510.0	49				
R204B		1259.5	1278.2	A^3C^1	510	3.5	1267.8	1315.6	A^1B^3	510.0	3.5				
R205B		1513.0	1556.5	A^3C^1	510	45.5	1477.8	1508.9	A^1B^3	510.0	24.5				
R206B		1399.3	1453.0	A^3C^1	510	21	1405.3	1435.0	A^1B^3	510.0	31.5				
R207B		1336.0	1400.0	A^1B^3	510	14	1327.1	1365.4	A^1B^3	510.0	0				
R208B		1255.9	1296.9	A^3C^1	510	3.5	1258.7	1300.1	A^1B^3	510.0	3.5				
R209B		1389.4	1426.5	A^1B^3	510	14	1378.6	1410.5	A^1B^3	510.0	14				
R210B		1442.6	1466.0	A^1B^3	510	42	1433.8	1456.1	A^1B^3	510.0	38.5				
R211B		1281.3	1340.6	A^3C^1	510	10.5	1273.1	1298.4	A^1B^3	510.0	21				
RC101B		2344.1	2345.6	$A^3B^3C^3D^3$	576	744	2420.2	2424.4	$A^2B^3C^2D^3$	504.0	1086				
RC102B		2221.1	2223.4	$A^3B^3C^3D^3$	576	680	2344.1	2350.0	$A^2B^3C^2D^3$	504.0	1034				
RC103B		2164.9	2175.7	$A^3B^3C^3D^3$	576	670	2294.9	2301.8	$A^2B^3C^2D^3$	504.0	1016				
RC104B		2122.0	2128.3	$A^3B^3C^3D^3$	576	650	2263.7	2270.9	$A^2B^3C^2D^3$	504.0	1026				
RC105B		2241.5	2246.9	$A^3B^3C^3D^3$	576	678	2355.5	2357.9	$A^2B^3C^2D^3$	504.0	1036				
RC106B		2198.2	2204.8	$A^3B^3C^3D^3$	576	674	2308.5	2318.9	$A^2B^3C^2D^3$	504.0	1042				
RC107B		2144.3	2151.0	$A^3B^3C^3D^3$	576	664	2267.9	2276.3	$A^2B^3C^2D^3$	504.0	1024				
RC108B		2117.6	2128.6	$A^3B^3C^3D^3$	576	658	2253.6	2264.7	$A^2B^3C^2D^3$	504.0	1022				
RC201B		1918.7	1954.9	$A^1B^1C^2D^2$	640	178	1908.9	1935.4	$B^2C^3D^1$	630.0	120				
RC202B		1756.5	1778.5	$A^1B^1C^3D^1$	590	268	1739.8	1779.7	$B^2C^3D^1$	630.0	162				
RC203B		1604.4	1637.4	C^3D^2	650	58	1601.6	1618.1	C^3D^2	650.0	104				
RC204B		1497.5	1533.3	C^3D^2	650	66	1476.4	1503.5	C^3D^2	650.0	64				

HFF-VRP-TW-PC											VRP-FLC				
Instance	z^*	z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL	z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL				
RC205B		1792.9	1851.9	$A^1B^3C^3$	570	280	1788.0	1824.4	$A^2B^1C^3D^1$	620.0	140				
RC206B		1768.0	1794.3	C^3D^2	650	60	1749.4	1769.2	$B^2C^3D^1$	630.0	74				
RC207B		1641.1	1695.6	$B^2C^3D^1$	630	70	1628.0	1673.4	$B^2C^3D^1$	630.0	72				
RC208B		1476.5	1529.9	C^3D^2	650	66	1477.5	1501.1	C^3D^2	650.0	52				
C101B		905.0	905.0		0	905	905.0	905.0		0.0	905				
C102B		905.0	905.0		0	905	905.0	905.0		0.0	905				
C103B		905.0	905.0		0	905	905.0	905.0		0.0	905				
C104B		905.0	905.0		0	905	905.0	905.0		0.0	905				
C105B		905.0	905.0		0	905	905.0	905.0		0.0	905				
C106B		905.0	905.0		0	905	905.0	905.0		0.0	905				
C107B		905.0	905.0		0	905	905.0	905.0		0.0	905				
C108B		905.0	905.0		0	905	905.0	905.0		0.0	905				
C109B		905.0	905.0		0	905	905.0	905.0		0.0	905				
C201B		905.0	905.0		0	905	905.0	905.0		0.0	905				
C202B		905.0	905.0		0	905	905.0	905.0		0.0	905				
C203B		905.0	905.0		0	905	905.0	905.0		0.0	905				
C204B		905.0	905.0		0	905	905.0	905.0		0.0	905				
C205B		905.0	905.0		0	905	905.0	905.0		0.0	905				
C206B		905.0	905.0		0	905	905.0	905.0		0.0	905				
C207B		905.0	905.0		0	905	905.0	905.0		0.0	905				
C208B		905.0	905.0		0	905	905.0	905.0		0.0	905				
R101C	2087.3	2087.3	2090.0	$A^3B^3C^3D^3E^3$	306	602	2397.2	2399.1	$A^1B^3C^3D^2E^3$	271.0	1214.5				
R102C		1740.7	1753.1	$A^3B^3C^3D^3E^3$	306	238	1923.5	1942.8	$A^1B^3C^3D^2E^3$	271.0	724.5				
R103C		1532.4	1559.6	$A^3B^3C^3D^3E^3$	306	126	1802.0	1819.8	$A^1B^3C^3D^2E^3$	271.0	714				
R104C		1443.3	1462.2	$A^3B^3C^3D^3E^3$	306	98	1763.0	1785.0	$A^1B^3C^3D^2E^3$	271.0	696.5				
R105C	1758.5	1776.0	1789.7	$A^3B^3C^3D^3E^3$	306	297.5	2009.9	2021.0	$A^1B^3C^3D^2E^3$	271.0	742				
R106C		1580.6	1609.9	$A^3B^3C^3D^3E^3$	306	178.5	1850.3	1879.0	$A^1B^3C^3D^2E^3$	271.0	717.5				
R107C		1480.9	1518.2	$A^3B^3C^3D^3E^3$	306	87.5	1783.7	1809.0	$A^1B^3C^3D^2E^3$	271.0	696.5				
R108C		1447.1	1458.0	$A^3B^3C^3D^3E^3$	306	98	1763.8	1780.9	$A^1B^3C^3D^2E^3$	271.0	700				
R109C		1577.8	1596.6	$A^3B^3C^3D^3E^3$	306	133	1824.6	1852.8	$A^1B^3C^3D^2E^3$	271.0	710.5				
R110C		1513.2	1536.9	$A^3B^3C^3D^3E^3$	306	126	1805.2	1821.8	$A^1B^3C^3D^2E^3$	271.0	714				
R111C		1482.4	1514.4	$A^3B^3C^3D^3E^3$	306	101.5	1769.8	1803.1	$A^1B^3C^3D^2E^3$	271.0	707				
R112C		1448.2	1461.2	$A^3B^3C^3D^3E^3$	306	126	1758.3	1777.5	$A^1B^3C^3D^2E^3$	271.0	700				
R201C		1432.9	1474.9	A^3B^2	275	108.5	1441.8	1451.8	A^3B^2	275.0	84				
R202C		1277.3	1303.9	A^3B^2	275	45.5	1272.4	1303.2	A^3B^2	275.0	45.5				
R203C		1159.5	1191.5	A^1B^3	255	28	1156.4	1170.2	A^3B^2	275.0	28				
R204C		1003.2	1039.7	A^1B^3	255	10.5	995.9	1040.6	A^1B^3	255.0	3.5				
R205C		1227.5	1259.0	A^3C^1	255	14	1219.7	1246.7	A^3B^2	275.0	21				
R206C		1138.7	1195.0	A^1B^3	255	21	1143.2	1182.0	A^1B^3	255.0	10.5				
R207C		1087.8	1120.9	A^1B^3	255	7	1084.9	1110.6	A^1B^3	255.0	28				
R208C		1001.6	1022.2	A^1C^2	285	3.5	997.0	1019.2	A^1C^2	285.0	7				
R209C		1134.9	1163.9	A^1B^3	255	7	1122.7	1136.0	A^1B^3	255.0	14				
R210C		1177.6	1206.5	A^1B^3	255	10.5	1180.6	1198.2	A^1B^3	255.0	49				
R211C		1046.8	1091.0	A^1B^3	255	14	1034.5	1062.6	A^1B^3	255.0	24.5				
RC101C	2056.1	2056.1	2057.0	$A^3B^3C^3D^3$	288	744	2166.5	2171.5	$A^2B^3C^2D^3$	252.0	1086				

HFF-VRP-TW-PC							VRP-FLC				
Instance	z^*	z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL	z_{min}	z_{moy}	Flotte	Coût Fixe	Coût LTL
RC102C	1933.1	1934.4	1938.1	$A^3B^3C^3D^3$	288	676	2091.9	2096.0	$A^2B^3C^2D^3$	252.0	1046
RC103C		1877.7	1884.3	$A^3B^3C^3D^3$	288	652	2042.9	2046.0	$A^2B^3C^2D^3$	252.0	1030
RC104C		1839.0	1846.5	$A^3B^3C^3D^3$	288	672	2008.3	2017.7	$A^2B^3C^2D^3$	252.0	1032
RC105C	1953.5	1953.5	1957.8	$A^3B^3C^3D^3$	288	678	2103.5	2105.6	$A^2B^3C^2D^3$	252.0	1036
RC106C	1909.0	1910.2	1921.5	$A^3B^3C^3D^3$	288	674	2056.5	2066.8	$A^2B^3C^2D^3$	252.0	1042
RC107C		1855.4	1859.2	$A^3B^3C^3D^3$	288	648	2019.6	2027.1	$A^2B^3C^2D^3$	252.0	1042
RC108C		1829.6	1840.0	$A^3B^3C^3D^3$	288	652	2008.4	2012.7	$A^2B^3C^2D^3$	252.0	1022
RC201C		1585.8	1607.1	$A^1B^3C^2D^1$	310	126	1598.4	1613.6	$B^2C^3D^1$	315.0	126
RC202C		1433.9	1460.9	C^3D^2	325	108	1417.7	1438.0	C^3D^2	325.0	144
RC203C		1260.1	1318.2	C^3D^2	325	58	1266.2	1291.6	C^2D^3	350.0	58
RC204C		1146.8	1163.6	C^3D^2	325	54	1159.0	1174.0	$C^1D^1E^2$	355.0	70
RC205C		1469.7	1488.1	$A^2B^1C^2D^2$	335	78	1476.1	1495.0	$A^2B^1C^2D^2$	335.0	58
RC206C		1392.9	1467.1	C^3D^2	325	88	1408.2	1439.9	$B^2C^2D^2$	340.0	12
RC207C		1310.4	1351.0	C^3D^2	325	78	1319.2	1340.0	$B^2C^2D^2$	340.0	98
RC208C		1141.5	1166.4	C^2D^3	350	6	1140.7	1165.3	C^2D^3	350.0	6
C101C		905.0	905.0		0	905	905.0	905.0		0.0	905
C102C		905.0	905.0		0	905	905.0	905.0		0.0	905
C103C		905.0	905.0		0	905	905.0	905.0		0.0	905
C104C		905.0	905.0		0	905	905.0	905.0		0.0	905
C105C		905.0	905.0		0	905	905.0	905.0		0.0	905
C106C		905.0	905.0		0	905	905.0	905.0		0.0	905
C107C		905.0	905.0		0	905	905.0	905.0		0.0	905
C108C		905.0	905.0		0	905	905.0	905.0		0.0	905
C109C		905.0	905.0		0	905	905.0	905.0		0.0	905
C201C		905.0	905.0		0	905	905.0	905.0		0.0	905
C202C		905.0	905.0		0	905	905.0	905.0		0.0	905
C203C		905.0	905.0		0	905	905.0	905.0		0.0	905
C204C		905.0	905.0		0	905	905.0	905.0		0.0	905
C205C		905.0	905.0		0	905	905.0	905.0		0.0	905
C206C		905.0	905.0		0	905	905.0	905.0		0.0	905
C207C		905.0	905.0		0	905	905.0	905.0		0.0	905
C208C		905.0	905.0		0	905	905.0	905.0		0.0	905

TABLEAU A.2 – Résultats du LNS sur les instances HFF-VRP-TW-PC et VRP-FLC avec coûts LTL faibles

RÉSULTATS DE LA MÉTHODE SMALL AND LARGE NEIGHBORHOOD SEARCH

B.1 Résultats détaillés des instances VRPTWTF

Instance	Friedrich et Elbert (2022) ALNS			SLNS		
	z^*	#r	t	z^*	gap (%)	#r
C101	828.94	10	86.4	828.94	0.00	10
C102	828.94	10	89.8	828.94	0.00	10
C103	828.06	10	90.1	828.06	0.00	10
C104	824.78	10	84.2	824.78	0.00	10
C105	828.94	10	81.3	828.94	0.00	10
C106	828.94	10	78.4	828.94	0.00	10
C107	828.94	10	81.4	828.94	0.00	10
C108	828.94	10	82.3	828.94	0.00	10
C109	828.94	10	88.3	828.94	0.00	10
C201	591.56	3	122.4	591.56	0.00	3
C202	591.56	3	133.5	591.56	0.00	3
C203	591.17	3	134.6	591.17	0.00	3
C204	590.6	3	128.4	590.6	0.00	3
C205	588.88	3	145.3	588.88	0.00	3
C206	588.49	3	130.9	588.49	0.00	3
C207	588.29	3	132.5	588.29	0.00	3
C208	588.32	3	128.7	588.32	0.00	3
R101	1400.9	19	174.3	1402.78	0.13	15
R102	1242.24	17	183.1	1243	0.06	13
R103	1094.32	13	176.2	1095.96	0.15	12
R104	958.28	9	171.8	961.66	0.35	9
R105	1215.11	14	170.2	1216.99	0.15	12
R106	1130.07	12	177.7	1133.06	0.26	10
R107	1008.94	10	169.5	1011.34	0.24	10
R108	920.21	9	164.2	920.21	0.00	9
R109	1071.95	11	180.9	1071.95	0.00	11
R110	1005.22	10	174.8	1005.22	0.00	10
R111	996.55	10	173.7	996.55	0.00	10
R112	923.64	9	170.1	923.64	0.00	9
R201	1197.53	4	303.1	1197.53	0.00	4
R202	1139.98	3	318.1	1132.54	-0.65	3
R203	900.56	3	310.8	905.77	0.58	3

Suite du Tableau B.1

Instance	Friedrich et Elbert (2022) ALNS			SLNS		
	z^*	#r	t	z^*	gap (%)	#r
R204	779.21	2	321	787.31	1.04	2
R205	994.43	3	294.6	1023.97	2.97	3
R206	900.59	3	300.6	900.59	0.00	3
R207	874.28	2	318.9	873.09	-0.14	2
R208	725.42	2	309.8	735.65	1.41	2
R209	909.16	3	333	915.33	0.68	3
R210	929.7	3	314.8	927.34	-0.25	3
R211	885.71	2	313	885.77	0.01	2
RC101	1502.96	14	119.9	1502.96	0.00	12
RC102	1346.32	12	122.5	1346.12	-0.01	12
RC103	1221.27	11	118	1227.68	0.52	10
RC104	1121.73	10	116.4	1122.23	0.04	10
RC105	1368.01	13	123.9	1370.84	0.21	12
RC106	1303.2	11	127.4	1303.2	0.00	11
RC107	1195.13	11	126.3	1195.13	0.00	11
RC108	1112.55	10	121.9	1112.55	0.00	10
RC201	1333.35	4	234.3	1341.99	0.65	4
RC202	1284.22	3	242.4	1288.65	0.34	3
RC203	1033.27	3	238.5	1028.1	-0.50	3
RC204	798.46	3	240.6	801.03	0.32	3
RC205	1232.56	4	227.3	1238.16	0.45	4
RC206	1129.83	3	249.5	1129.83	0.00	3
RC207	1024.08	3	256.2	1024.08	0.00	3
RC208	818.27	3	289.6	818.27	0.00	3

TABLEAU B.1 – Résultats de la méthode SLNS sur les instances VRPTWTF de Friedrich et Elbert (2022)

B.2 Résultats détaillés des instances FSMTWTF

Instance	Friedrich et Elbert (2022) ALNS			SLNS-SPP			Flotte
	z^*	#r	t	z^*	gap (%)	#r	
C101	7085.43	19	301.2	7085.43	0.00	19	A^{19}
C102	7076.65	19	270.5	7076.65	0.00	19	A^{19}
C103	7075.66	19	295.1	7075.69	0.00	19	A^{19}
C104	7071.2	19	276.4	7071.2	0.00	19	A^{19}
C105	7085.43	19	253.2	7085.43	0.00	19	A^{19}
C106	7081.3	19	233.6	7081.3	0.00	19	A^{19}
C107	7082.95	19	224.2	7083.21	0.00	19	A^{19}
C108	7077.39	19	286.6	7077.39	0.00	19	A^{19}
C109	7075.03	19	244.9	7075.03	0.00	19	A^{19}
C201	5695.02	5	427.2	5695.02	0.00	5	A^5
C202	5685.24	5	345.5	5685.24	0.00	5	A^5
C203	5681.55	5	378.5	5681.55	0.00	5	A^5
C204	5677.66	5	372.8	5677.66	0.00	5	A^5
C205	5691.36	5	415.9	5691.36	0.00	5	A^5
C206	5689.32	5	414.3	5689.32	0.00	5	A^5
C207	5687.35	5	422.1	5687.35	0.00	5	A^5
C208	5686.5	5	391.8	5686.5	0.00	5	A^5
R101	4230.29	19	310.6	4218.42	-0.28	20	B^4C^{16}
R102	4142.25	19	336.7	4093.56	-1.18	19	A^1C^{18}
R103	3994.49	19	324.6	4006.46	0.30	19	A^1C^{18}
R104	3936	19	281.3	3952.48	0.42	19	$B^3C^{15}D^1$
R105	4096.17	20	290.5	4097.05	0.02	20	B^4C^{16}
R106	4026.28	19	322.6	4043.36	0.42	19	$B^3C^{15}D^1$
R107	3964.29	19	304.9	3976.17	0.30	19	A^1C^{18}
R108	3926.31	19	319.2	3929.06	0.07	19	$B^3C^{15}D^1$
R109	3996.48	19	304	4011.43	0.37	21	B^7C^{14}
R110	3959.76	19	314.9	3966.42	0.17	19	A^1C^{18}
R111	3963.68	19	333.1	3978.34	0.37	19	A^1C^{18}
R112	3917.88	19	325.3	3940.16	0.57	19	B^2C^{17}
R201	3426.73	5	427.8	3449.14	0.65	5	A^5
R202	3295.1	5	430.4	3303.25	0.25	5	A^5
R203	3128.06	5	426.1	3130.41	0.08	5	A^5
R204	3014.24	5	386.1	3015.08	0.03	5	A^5
R205	3218.97	5	443.4	3248.38	0.91	5	A^5
R206	3146.34	5	437.6	3158.87	0.40	5	A^5
R207	3074.01	5	410.9	3087.5	0.44	5	A^5
R208	2997.24	5	420.8	3007.59	0.35	5	A^5
R209	3119.56	5	428.9	3136.04	0.53	5	A^5
R210	3170.41	5	442.4	3175.18	0.15	5	A^5
R211	3019.93	5	452.9	3027.25	0.24	5	A^5
RC101	5076.28	17	340.7	5085.84	0.19	19	$A^5B^8C^6$
RC102	4929.53	16	406.8	4992.4	1.28	19	$A^5B^8C^6$
RC103	4799.78	14	403.9	4830.54	0.64	15	$A^2B^4C^9$
RC104	4717.63	12	399.4	4788.29	1.50	16	$A^4B^3C^9$
RC105	4961.39	16	389.4	4995.68	0.69	16	$A^1B^8C^7$
RC106	4901.37	16	414.5	4940.99	0.81	16	$A^2B^6C^8$

Suite du Tableau B.2

Instance	Friedrich et Elbert (2022) ALNS			SLNS-SPP			Flotte
	z^*	#r	t	z^*	gap (%)	#r	
RC107	4784	16	432.7	4885.11	2.11	17	$A^3B^7C^7$
RC108	4708.85	15	414.4	4765.17	1.20	15	$A^2B^4C^9$
RC201	4374.09	12	414.6	4376.25	0.05	14	$A^{10}B^4$
RC202	4244.63	13	432.6	4249.7	0.12	13	A^8B^5
RC203	4170.17	11	453.6	4178.89	0.21	10	$A^4B^4C^2$
RC204	4087.11	9	438.5	4090.74	0.09	9	$A^4B^1C^4$
RC205	4291.93	12	419.3	4297.2	0.12	13	A^8B^5
RC206	4251.88	11	446.4	4258.17	0.15	12	A^6B^6
RC207	4182.44	12	415.3	4210.82	0.68	11	$A^5B^5C^1$
RC208	4075.04	9	421.9	4082.61	0.19	9	$A^3B^3C^3$

TABLEAU B.2 – Résultats de la méthode SLNS sur les instances FSMTWTF de Friedrich et Elbert (2022)

B.3 Résultats détaillés des instances CSDVRP

Type	Instance					Gu et al. (2019) ALNS		SLNS		
	m	p	Δ	α	id	z^*	t	z^*	$gap(\%)$	$gap_{moy}(\%)$
C101	2	0.6	[1, 100]	1.1	1.1	2016.5051	260.68	2002.0842	-0.72	-0.42
C101	2	0.6	[1, 100]	1.1	1.1	2170.969	270.33	2169.081	-0.09	-0.08
C101	2	0.6	[1, 100]	1.1	1.1	2076.9958	236.59	2065.4363	-0.56	-0.50
C101	2	0.6	[1, 100]	1.1	1.1	2133.8412	268.38	2134.331	0.02	0.23
C101	2	0.6	[1, 100]	1.1	1.1	2016.1057	270.64	2016.0394	0.00	0.17
C101	2	0.6	[1, 100]	1.5	1.5	1571.839	325.56	1559.21	-0.80	-0.06
C101	2	0.6	[1, 100]	1.5	1.5	1690.1594	322.81	1691.8133	0.10	0.44
C101	2	0.6	[1, 100]	1.5	1.5	1579.4423	326.38	1579.1904	-0.02	0.43
C101	2	0.6	[1, 100]	1.5	1.5	1593.0295	295.26	1594.0426	0.06	0.34
C101	2	0.6	[1, 100]	1.5	1.5	1542.7069	327.96	1543.2152	0.03	0.37
C101	2	0.6	[1, 100]	2	2	1484.428	443.9	1483.7584	-0.05	0.46
C101	2	0.6	[1, 100]	2	2	1594.3828	390.17	1591.8955	-0.16	0.09
C101	2	0.6	[1, 100]	2	2	1551.0808	358.75	1570.1478	1.23	1.74
C101	2	0.6	[1, 100]	2	2	1532.8325	409.17	1531.3797	-0.09	0.28
C101	2	0.6	[1, 100]	2	2	1537.2621	345.92	1529.1479	-0.53	0.61
C101	2	0.6	[1, 100]	2.5	2.5	1290.4392	509.62	1291.6086	0.09	0.45
C101	2	0.6	[1, 100]	2.5	2.5	1378.622	407.99	1372.9094	-0.41	0.38
C101	2	0.6	[1, 100]	2.5	2.5	1318.8051	429.45	1318.8399	0.00	0.73
C101	2	0.6	[1, 100]	2.5	2.5	1319.5941	475.72	1325.8096	0.47	0.96
C101	2	0.6	[1, 100]	2.5	2.5	1307.123	408.3	1308.7991	0.13	0.78
C101	2	0.6	[40, 60]	1.1	1.1	3686.3063	212.18	3684.6792	-0.04	-0.04
C101	2	0.6	[40, 60]	1.1	1.1	3923.7703	211.63	3916.2126	-0.19	-0.19
C101	2	0.6	[40, 60]	1.1	1.1	3720.0138	247.43	3714.9997	-0.13	-0.13
C101	2	0.6	[40, 60]	1.1	1.1	3815.5816	271.71	3806.7984	-0.23	-0.23
C101	2	0.6	[40, 60]	1.1	1.1	3645.8466	230.77	3644.4631	-0.04	-0.04
C101	2	0.6	[40, 60]	1.5	1.5	2670.5453	259.21	2650.3267	-0.76	-0.41
C101	2	0.6	[40, 60]	1.5	1.5	2780.6093	269.9	2771.8017	-0.32	-0.31
C101	2	0.6	[40, 60]	1.5	1.5	2606.9999	278.21	2606.8752	0.00	0.00
C101	2	0.6	[40, 60]	1.5	1.5	2703.7315	306.65	2686.6892	-0.63	-0.63
C101	2	0.6	[40, 60]	1.5	1.5	2606.6952	249.35	2606.6952	0.00	0.10
C101	2	0.6	[40, 60]	2	2	2364.7218	331.61	2356.298	-0.36	-0.36
C101	2	0.6	[40, 60]	2	2	2489.1959	358.79	2465.9236	-0.93	0.02
C101	2	0.6	[40, 60]	2	2	2401.3033	343.9	2400.4822	-0.03	0.46
C101	2	0.6	[40, 60]	2	2	2437.5545	353.23	2438.9735	0.06	0.35
C101	2	0.6	[40, 60]	2	2	2440.5193	339.58	2420.0584	-0.84	-0.21
C101	2	0.6	[40, 60]	2.5	2.5	1963.9711	368.04	1964.25	0.01	0.70
C101	2	0.6	[40, 60]	2.5	2.5	2027.4671	367.94	2029.733	0.11	0.50
C101	2	0.6	[40, 60]	2.5	2.5	1950.7756	406.74	1949.8181	-0.05	0.54
C101	2	0.6	[40, 60]	2.5	2.5	1987.3101	343.06	1986.4954	-0.04	0.37
C101	2	0.6	[40, 60]	2.5	2.5	1973.9611	367.4	1969.6089	-0.22	0.29
C101	2	1	[1, 100]	1.1	1.1	3096.022	493.02	3109.7769	0.44	0.96
C101	2	1	[1, 100]	1.1	1.1	3047.0941	459.72	3046.2171	-0.03	0.29
C101	2	1	[1, 100]	1.1	1.1	3341.0044	530.95	3339.2927	-0.05	0.45
C101	2	1	[1, 100]	1.1	1.1	3293.8945	508.36	3282.9892	-0.33	0.18
C101	2	1	[1, 100]	1.1	1.1	3025.7796	459.09	3023.7165	-0.07	0.22

Suite du Tableau B.3

Type	Instance					Gu et al. (2019) ALNS		SLNS		
	m	p	Δ	α	id	z^*	t	z^*	gap(%)	gap _{moy} (%)
C101	2	1	[1, 100]	1.5	1.5	2334.4763	526.05	2337.6973	0.14	0.72
C101	2	1	[1, 100]	1.5	1.5	2301.507	566.56	2308.2855	0.29	1.11
C101	2	1	[1, 100]	1.5	1.5	2518.5285	554.75	2522.7496	0.17	0.88
C101	2	1	[1, 100]	1.5	1.5	2479.4547	534.61	2485.8364	0.26	0.68
C101	2	1	[1, 100]	1.5	1.5	2297.1414	527.3	2284.6349	-0.54	0.48
C101	2	1	[1, 100]	2	2	2124.8932	603.94	2121.3512	-0.17	0.61
C101	2	1	[1, 100]	2	2	2169.8922	629.32	2181.2605	0.52	1.21
C101	2	1	[1, 100]	2	2	2245.4843	617.32	2256.0053	0.47	1.05
C101	2	1	[1, 100]	2	2	2262.6622	554.71	2256.4388	-0.28	0.61
C101	2	1	[1, 100]	2	2	2156.8393	523.47	2154.9946	-0.09	0.79
C101	2	1	[1, 100]	2.5	2.5	1753.7203	693.48	1755.3687	0.09	1.58
C101	2	1	[1, 100]	2.5	2.5	1811.6854	756.15	1817.8996	0.34	2.13
C101	2	1	[1, 100]	2.5	2.5	1871.8435	637.95	1869.8775	-0.11	0.72
C101	2	1	[1, 100]	2.5	2.5	1895.2543	716.49	1907.624	0.65	1.25
C101	2	1	[1, 100]	2.5	2.5	1808.1843	738.02	1808.8838	0.04	1.15
C101	2	1	[40, 60]	1.1	1.1	5536.0546	433.77	5527.1203	-0.16	-0.16
C101	2	1	[40, 60]	1.1	1.1	5504.4821	454.4	5477.0715	-0.50	-0.50
C101	2	1	[40, 60]	1.1	1.1	5679.6959	424.29	5674.9309	-0.08	-0.08
C101	2	1	[40, 60]	1.1	1.1	5645.4455	463.91	5642.7972	-0.05	-0.05
C101	2	1	[40, 60]	1.1	1.1	5479.9888	469.36	5459.9957	-0.36	-0.36
C101	2	1	[40, 60]	1.5	1.5	3849.0963	507.87	3851.4236	0.06	0.28
C101	2	1	[40, 60]	1.5	1.5	3867.0017	482.07	3860.7464	-0.16	0.08
C101	2	1	[40, 60]	1.5	1.5	3959.2347	502.91	3942.3154	-0.43	-0.22
C101	2	1	[40, 60]	1.5	1.5	3898.8312	583.69	3898.1419	-0.02	0.01
C101	2	1	[40, 60]	1.5	1.5	3839.8436	598.34	3817.4775	-0.58	-0.42
C101	2	1	[40, 60]	2	2	3328.8877	613.39	3327.2282	-0.05	0.50
C101	2	1	[40, 60]	2	2	3391.5541	609.4	3374.0935	-0.51	0.50
C101	2	1	[40, 60]	2	2	3402.1518	593.52	3394.7748	-0.22	0.35
C101	2	1	[40, 60]	2	2	3446.1003	604.41	3444.9778	-0.03	0.75
C101	2	1	[40, 60]	2	2	3384.4928	628.07	3383.4155	-0.03	0.79
C101	2	1	[40, 60]	2.5	2.5	2717.2918	680.71	2736.2264	0.70	1.28
C101	2	1	[40, 60]	2.5	2.5	2776.1339	596.97	2793.7838	0.64	1.09
C101	2	1	[40, 60]	2.5	2.5	2761.2009	616.17	2762.0914	0.03	0.64
C101	2	1	[40, 60]	2.5	2.5	2786.8485	589.1	2788.7344	0.07	1.38
C101	2	1	[40, 60]	2.5	2.5	2751.4713	680.74	2769.8511	0.67	1.46
C101	3	0.6	[1, 100]	1.1	1.1	2199.3967	447.71	2204.6803	0.24	1.11
C101	3	0.6	[1, 100]	1.1	1.1	2357.4534	509.65	2352.6432	-0.20	0.68
C101	3	0.6	[1, 100]	1.1	1.1	2487.822	505.21	2488.5177	0.03	0.43
C101	3	0.6	[1, 100]	1.1	1.1	2266.5132	448.84	2257.9253	-0.38	0.56
C101	3	0.6	[1, 100]	1.1	1.1	2509.7369	471.85	2507.861	-0.07	0.73
C101	3	0.6	[1, 100]	1.5	1.5	1698.9351	449.14	1700.2765	0.08	0.91
C101	3	0.6	[1, 100]	1.5	1.5	1807.9754	579.95	1801.6577	-0.35	0.73
C101	3	0.6	[1, 100]	1.5	1.5	1893.4346	521.23	1891.0933	-0.12	0.54
C101	3	0.6	[1, 100]	1.5	1.5	1745.0116	568.39	1741.5992	-0.20	0.69
C101	3	0.6	[1, 100]	1.5	1.5	1930.7389	576.69	1927.5545	-0.16	0.30
C101	3	0.6	[1, 100]	2	2	1626.3891	633.94	1633.7562	0.45	1.47
C101	3	0.6	[1, 100]	2	2	1652.1369	677.73	1654.3981	0.14	1.11
C101	3	0.6	[1, 100]	2	2	1684.7036	648.54	1684.3722	-0.02	1.74
C101	3	0.6	[1, 100]	2	2	1677.4419	584.07	1677.3094	-0.01	0.64

Suite du Tableau B.3

Type	Instance					Gu et al. (2019) ALNS		SLNS		
	m	p	Δ	α	id	z^*	t	z^*	gap(%)	gap _{moy} (%)
C101	3	0.6	[1, 100]	2	2	1714.1925	600.65	1716.7392	0.15	1.22
C101	3	0.6	[1, 100]	2.5	2.5	1411.6091	681.11	1410.5057	-0.08	0.61
C101	3	0.6	[1, 100]	2.5	2.5	1424.9209	810.87	1434.1699	0.65	1.24
C101	3	0.6	[1, 100]	2.5	2.5	1428.427	712.86	1429.0011	0.04	1.81
C101	3	0.6	[1, 100]	2.5	2.5	1442.4131	780.48	1455.5563	0.91	1.41
C101	3	0.6	[1, 100]	2.5	2.5	1444.4045	756.13	1448.6164	0.29	1.69
C101	3	0.6	[40, 60]	1.1	1.1	3257.003	401.09	3257.5802	0.02	0.20
C101	3	0.6	[40, 60]	1.1	1.1	3489.208	435.08	3475.6577	-0.39	0.07
C101	3	0.6	[40, 60]	1.1	1.1	3582.1768	399.24	3583.7103	0.04	0.53
C101	3	0.6	[40, 60]	1.1	1.1	3484.3878	446.43	3487.5233	0.09	0.47
C101	3	0.6	[40, 60]	1.1	1.1	3615.4109	496.11	3615.5432	0.00	0.44
C101	3	0.6	[40, 60]	1.5	1.5	2407.4264	419.67	2409.382	0.08	0.62
C101	3	0.6	[40, 60]	1.5	1.5	2523.9967	558.51	2525.7645	0.07	0.23
C101	3	0.6	[40, 60]	1.5	1.5	2600.9273	452.04	2607.103	0.24	0.49
C101	3	0.6	[40, 60]	1.5	1.5	2525.7262	468.16	2532.5256	0.27	0.82
C101	3	0.6	[40, 60]	1.5	1.5	2640.8231	479.02	2637.3717	-0.13	0.89
C101	3	0.6	[40, 60]	2	2	2265.3978	482.4	2269.4271	0.18	0.71
C101	3	0.6	[40, 60]	2	2	2326.5017	643.35	2331.6047	0.22	1.70
C101	3	0.6	[40, 60]	2	2	2313.2239	540.93	2317.6935	0.19	1.00
C101	3	0.6	[40, 60]	2	2	2352.4767	574.32	2353.0321	0.02	1.43
C101	3	0.6	[40, 60]	2	2	2321.6905	560.01	2319.8832	-0.08	0.36
C101	3	0.6	[40, 60]	2.5	2.5	1857.8147	501.48	1856.3517	-0.08	1.09
C101	3	0.6	[40, 60]	2.5	2.5	1948.0161	643.45	1952.1352	0.21	1.68
C101	3	0.6	[40, 60]	2.5	2.5	1917.8823	606.26	1926.8775	0.47	1.26
C101	3	0.6	[40, 60]	2.5	2.5	1972.7209	618.31	1975.9728	0.16	0.94
C101	3	0.6	[40, 60]	2.5	2.5	1918.972	607.4	1921.8193	0.15	0.75
C101	3	1	[1, 100]	1.1	1.1	3378.7279	952.16	3399.7573	0.62	2.52
C101	3	1	[1, 100]	1.1	1.1	3245.1446	914.82	3283.6508	1.19	1.88
C101	3	1	[1, 100]	1.1	1.1	3265.0504	865.51	3285.8119	0.64	1.21
C101	3	1	[1, 100]	1.1	1.1	3370.6534	825.87	3374.9051	0.13	1.14
C101	3	1	[1, 100]	1.1	1.1	3162.0426	900.59	3170.1989	0.26	1.04
C101	3	1	[1, 100]	1.5	1.5	2561.2706	972.71	2564.3767	0.12	0.70
C101	3	1	[1, 100]	1.5	1.5	2459.3592	1024.42	2466.7524	0.30	1.09
C101	3	1	[1, 100]	1.5	1.5	2476.9681	1083.84	2482.22	0.21	1.42
C101	3	1	[1, 100]	1.5	1.5	2565.6834	923.07	2580.4504	0.58	1.41
C101	3	1	[1, 100]	1.5	1.5	2404.8723	968	2424.0701	0.80	1.29
C101	3	1	[1, 100]	2	2	2280.2733	1211.44	2309.7325	1.29	2.16
C101	3	1	[1, 100]	2	2	2192.9203	1153.59	2228.0296	1.60	2.59
C101	3	1	[1, 100]	2	2	2253.2807	1142.33	2263.1672	0.44	1.82
C101	3	1	[1, 100]	2	2	2390.4263	1099.96	2405.0787	0.61	1.93
C101	3	1	[1, 100]	2	2	2231.88	1240.23	2243.7259	0.53	1.65
C101	3	1	[1, 100]	2.5	2.5	1914.9262	1348.23	1927.4869	0.66	1.59
C101	3	1	[1, 100]	2.5	2.5	1856.8834	1606.09	1872.4286	0.84	2.21
C101	3	1	[1, 100]	2.5	2.5	1867.447	1519.55	1886.8716	1.04	1.82
C101	3	1	[1, 100]	2.5	2.5	1985.1207	1396.51	1989.1221	0.20	1.23
C101	3	1	[1, 100]	2.5	2.5	1875.578	1625.37	1890.4004	0.79	1.93
C101	3	1	[40, 60]	1.1	1.1	5196.4247	906.42	5162.4077	-0.65	0.13
C101	3	1	[40, 60]	1.1	1.1	5059.3108	1016.59	5082.1525	0.45	0.89
C101	3	1	[40, 60]	1.1	1.1	5103.2213	1054.92	5114.2925	0.22	0.79

Suite du Tableau B.3

Type	Instance					Gu et al. (2019) ALNS		SLNS		
	m	p	Δ	α	id	z^*	t	z^*	gap(%)	gap _{moy} (%)
C101	3	1	[40, 60]	1.1	1.1	5243.8349	987.19	5248.0937	0.08	0.61
C101	3	1	[40, 60]	1.1	1.1	5114.0431	978.28	5120.773	0.13	0.61
C101	3	1	[40, 60]	1.5	1.5	3755.7431	829.41	3772.7067	0.45	1.06
C101	3	1	[40, 60]	1.5	1.5	3733.9771	944.05	3732.3722	-0.04	1.20
C101	3	1	[40, 60]	1.5	1.5	3732.6484	971.9	3760.0228	0.73	1.28
C101	3	1	[40, 60]	1.5	1.5	3770.5935	897.98	3799.8267	0.78	1.29
C101	3	1	[40, 60]	1.5	1.5	3740.0157	1020.43	3779.6394	1.06	1.90
C101	3	1	[40, 60]	2	2	3331.3655	1145.1	3372.574	1.24	1.90
C101	3	1	[40, 60]	2	2	3307.6401	1081.25	3325.5594	0.54	1.98
C101	3	1	[40, 60]	2	2	3297.9859	1044.56	3328.9219	0.94	2.12
C101	3	1	[40, 60]	2	2	3382.5992	1254.62	3440.4399	1.71	2.59
C101	3	1	[40, 60]	2	2	3330.949	1240.97	3375.4598	1.34	2.46
C101	3	1	[40, 60]	2.5	2.5	2785.0606	1190.8	2819.2388	1.23	2.00
C101	3	1	[40, 60]	2.5	2.5	2735.8088	1259.03	2781.7011	1.68	2.49
C101	3	1	[40, 60]	2.5	2.5	2754.3082	1255.07	2797.9533	1.58	2.15
C101	3	1	[40, 60]	2.5	2.5	2789.5742	1251.01	2843.2989	1.93	2.82
C101	3	1	[40, 60]	2.5	2.5	2746.4348	1145.49	2773.3006	0.98	2.50
R101	2	0.6	[1, 100]	1.1	1.1	1887.495	279.09	1883.8071	-0.20	-0.15
R101	2	0.6	[1, 100]	1.1	1.1	2152.571	300.47	2144.1997	-0.39	-0.32
R101	2	0.6	[1, 100]	1.1	1.1	1920.3454	265.37	1917.1439	-0.17	-0.16
R101	2	0.6	[1, 100]	1.1	1.1	1956.0302	302.81	1956.5361	0.03	0.35
R101	2	0.6	[1, 100]	1.1	1.1	1828.4456	306.71	1819.1154	-0.51	-0.49
R101	2	0.6	[1, 100]	1.5	1.5	1482.2547	319.35	1480.9105	-0.09	0.36
R101	2	0.6	[1, 100]	1.5	1.5	1670.7013	307.37	1669.5622	-0.07	0.36
R101	2	0.6	[1, 100]	1.5	1.5	1517.8384	296.24	1512.3851	-0.36	0.45
R101	2	0.6	[1, 100]	1.5	1.5	1541.2523	307.34	1534.6774	-0.43	-0.02
R101	2	0.6	[1, 100]	1.5	1.5	1464.6001	312	1458.2644	-0.43	0.36
R101	2	0.6	[1, 100]	2	2	1213.6448	355.57	1213.6448	0.00	0.76
R101	2	0.6	[1, 100]	2	2	1351.5177	378.88	1347.4458	-0.30	0.71
R101	2	0.6	[1, 100]	2	2	1249.8766	370.13	1241.937	-0.64	0.52
R101	2	0.6	[1, 100]	2	2	1260.8517	418.11	1266.6831	0.46	1.21
R101	2	0.6	[1, 100]	2	2	1200.0221	325.71	1197.5244	-0.21	0.80
R101	2	0.6	[1, 100]	2.5	2.5	1059.0853	385.07	1060.6888	0.15	1.40
R101	2	0.6	[1, 100]	2.5	2.5	1169.7717	453.09	1169.953	0.02	0.38
R101	2	0.6	[1, 100]	2.5	2.5	1084.9342	414.91	1082.6993	-0.21	1.17
R101	2	0.6	[1, 100]	2.5	2.5	1102.6288	409.88	1102.9471	0.03	0.70
R101	2	0.6	[1, 100]	2.5	2.5	1050.7626	430.69	1052.2383	0.14	0.70
R101	2	0.6	[40, 60]	1.1	1.1	3333.0578	263.31	3326.1535	-0.21	-0.21
R101	2	0.6	[40, 60]	1.1	1.1	3553.3423	250.22	3546.7668	-0.19	-0.19
R101	2	0.6	[40, 60]	1.1	1.1	3327.4991	223.6	3327.4991	0.00	0.00
R101	2	0.6	[40, 60]	1.1	1.1	3386.6904	275.68	3386.6904	0.00	0.00
R101	2	0.6	[40, 60]	1.1	1.1	3244.8496	236.62	3244.8496	0.00	0.00
R101	2	0.6	[40, 60]	1.5	1.5	2420.4671	261.82	2415.3828	-0.21	-0.21
R101	2	0.6	[40, 60]	1.5	1.5	2598.2538	293.5	2583.9033	-0.55	-0.55
R101	2	0.6	[40, 60]	1.5	1.5	2414.9036	274.4	2414.9036	0.00	0.01
R101	2	0.6	[40, 60]	1.5	1.5	2481.8594	274.12	2477.0981	-0.19	-0.19
R101	2	0.6	[40, 60]	1.5	1.5	2414.1367	313.44	2391.1005	-0.95	-0.95
R101	2	0.6	[40, 60]	2	2	1895.5429	326.88	1891.1228	-0.23	0.07
R101	2	0.6	[40, 60]	2	2	1992.3865	330	1987.4174	-0.25	-0.14

Suite du Tableau B.3

Type	Instance					Gu et al. (2019) ALNS		SLNS		
	m	p	Δ	α	id	z^*	t	z^*	gap(%)	gap _{moy} (%)
R101	2	0.6	[40, 60]	2	2	1881.4183	326.29	1869.1301	-0.65	-0.56
R101	2	0.6	[40, 60]	2	2	1906.1707	353.72	1904.8823	-0.07	0.17
R101	2	0.6	[40, 60]	2	2	1852.4127	324.98	1848.6343	-0.20	-0.20
R101	2	0.6	[40, 60]	2.5	2.5	1577.0403	344.39	1577.8706	0.05	0.54
R101	2	0.6	[40, 60]	2.5	2.5	1659.9154	330.66	1657.3528	-0.15	0.27
R101	2	0.6	[40, 60]	2.5	2.5	1570.492	343.04	1567.3426	-0.20	-0.11
R101	2	0.6	[40, 60]	2.5	2.5	1591.3215	348.6	1591.1414	-0.01	0.75
R101	2	0.6	[40, 60]	2.5	2.5	1558.7095	305.21	1553.7024	-0.32	0.03
R101	2	1	[1, 100]	1.1	1.1	2756.7887	531.18	2740.7655	-0.58	0.02
R101	2	1	[1, 100]	1.1	1.1	2827.3512	426.29	2815.8242	-0.41	-0.11
R101	2	1	[1, 100]	1.1	1.1	2985.2441	507.52	2981.8978	-0.11	0.13
R101	2	1	[1, 100]	1.1	1.1	3031.0145	533.16	3010.44	-0.68	-0.34
R101	2	1	[1, 100]	1.1	1.1	2784.0475	496.69	2768.6104	-0.55	-0.11
R101	2	1	[1, 100]	1.5	1.5	2122.9233	530.35	2114.4298	-0.40	0.42
R101	2	1	[1, 100]	1.5	1.5	2183.9167	578.51	2173.777	-0.46	0.21
R101	2	1	[1, 100]	1.5	1.5	2295.0886	501.07	2295.1955	0.00	0.62
R101	2	1	[1, 100]	1.5	1.5	2297.4593	582.93	2290.7361	-0.29	0.25
R101	2	1	[1, 100]	1.5	1.5	2137.8987	507.42	2138.3234	0.02	0.74
R101	2	1	[1, 100]	2	2	1698.3738	551.73	1693.7309	-0.27	1.03
R101	2	1	[1, 100]	2	2	1722.1094	589.96	1723.3676	0.07	0.79
R101	2	1	[1, 100]	2	2	1813.5162	547.78	1799.9646	-0.75	0.44
R101	2	1	[1, 100]	2	2	1827.3575	582.61	1829.5698	0.12	0.88
R101	2	1	[1, 100]	2	2	1704.9316	512.4	1695.5815	-0.55	0.68
R101	2	1	[1, 100]	2.5	2.5	1452.8434	682.7	1441.6077	-0.77	0.92
R101	2	1	[1, 100]	2.5	2.5	1460.9983	705.62	1458.7044	-0.16	1.20
R101	2	1	[1, 100]	2.5	2.5	1533.0524	633.6	1537.9231	0.32	1.20
R101	2	1	[1, 100]	2.5	2.5	1541.2999	702.58	1536.6845	-0.30	1.12
R101	2	1	[1, 100]	2.5	2.5	1431.3942	631.85	1442.2701	0.76	1.69
R101	2	1	[40, 60]	1.1	1.1	4815.842	477.78	4808.5511	-0.15	-0.15
R101	2	1	[40, 60]	1.1	1.1	4854.3103	533.46	4854.3103	0.00	0.00
R101	2	1	[40, 60]	1.1	1.1	4928.0506	396.24	4926.0663	-0.04	-0.04
R101	2	1	[40, 60]	1.1	1.1	4886.1829	357.43	4883.5109	-0.05	-0.05
R101	2	1	[40, 60]	1.1	1.1	4778.5328	390.36	4767.2859	-0.24	-0.24
R101	2	1	[40, 60]	1.5	1.5	3489.745	602.15	3478.9113	-0.31	-0.13
R101	2	1	[40, 60]	1.5	1.5	3544.2844	554.11	3533.8308	-0.29	-0.29
R101	2	1	[40, 60]	1.5	1.5	3551.5178	609.51	3540.8962	-0.30	-0.20
R101	2	1	[40, 60]	1.5	1.5	3546.3662	512.74	3533.6454	-0.36	-0.35
R101	2	1	[40, 60]	1.5	1.5	3472.3544	586.66	3449.3681	-0.66	-0.54
R101	2	1	[40, 60]	2	2	2622.657	636.79	2618.2983	-0.17	0.19
R101	2	1	[40, 60]	2	2	2657.6846	485.8	2649.8047	-0.30	-0.02
R101	2	1	[40, 60]	2	2	2665.3118	578.84	2664.161	-0.04	0.01
R101	2	1	[40, 60]	2	2	2695.0898	620.18	2683.2182	-0.44	-0.11
R101	2	1	[40, 60]	2	2	2641.4628	550.42	2617.9546	-0.89	-0.65
R101	2	1	[40, 60]	2.5	2.5	2141.4426	644.84	2145.1834	0.17	0.63
R101	2	1	[40, 60]	2.5	2.5	2189.4736	571.24	2178.6372	-0.49	0.17
R101	2	1	[40, 60]	2.5	2.5	2202.7464	675.78	2209.9792	0.33	0.80
R101	2	1	[40, 60]	2.5	2.5	2203.0546	650.99	2206.0933	0.14	0.69
R101	2	1	[40, 60]	2.5	2.5	2149.326	538.39	2149.6148	0.01	0.41
R101	3	0.6	[1, 100]	1.1	1.1	2082.4706	448.5	2072.0507	-0.50	0.28

Suite du Tableau B.3

Type	Instance					Gu et al. (2019) ALNS		SLNS		
	m	p	Δ	α	id	z^*	t	z^*	gap(%)	gap _{moy} (%)
R101	3	0.6	[1, 100]	1.1	1.1	2221.3245	534.88	2224.964	0.16	0.89
R101	3	0.6	[1, 100]	1.1	1.1	2134.4081	447.11	2131.1624	-0.15	0.25
R101	3	0.6	[1, 100]	1.1	1.1	2131.5677	458.8	2130.209	-0.06	0.33
R101	3	0.6	[1, 100]	1.1	1.1	2183.8628	478.3	2170.7835	-0.60	0.15
R101	3	0.6	[1, 100]	1.5	1.5	1628.8003	507.92	1629.3262	0.03	1.02
R101	3	0.6	[1, 100]	1.5	1.5	1726.5884	564.75	1728.7249	0.12	1.48
R101	3	0.6	[1, 100]	1.5	1.5	1678.2108	563.18	1695.3901	1.02	1.82
R101	3	0.6	[1, 100]	1.5	1.5	1683.4501	502.74	1675.697	-0.46	0.71
R101	3	0.6	[1, 100]	1.5	1.5	1698.7518	508.7	1699.9912	0.07	1.02
R101	3	0.6	[1, 100]	2	2	1333.3018	579.85	1334.3219	0.08	0.53
R101	3	0.6	[1, 100]	2	2	1409.0886	638.09	1405.6694	-0.24	0.78
R101	3	0.6	[1, 100]	2	2	1366.0804	554.83	1375.229	0.67	1.32
R101	3	0.6	[1, 100]	2	2	1380.4631	628.49	1384.7823	0.31	1.72
R101	3	0.6	[1, 100]	2	2	1372.9908	612.43	1380.4314	0.54	1.93
R101	3	0.6	[1, 100]	2.5	2.5	1144.9686	703.87	1147.3854	0.21	1.54
R101	3	0.6	[1, 100]	2.5	2.5	1211.4598	739.74	1205.462	-0.50	0.90
R101	3	0.6	[1, 100]	2.5	2.5	1183.2168	707.54	1187.4392	0.36	1.78
R101	3	0.6	[1, 100]	2.5	2.5	1188.4567	750.52	1189.1562	0.06	1.38
R101	3	0.6	[1, 100]	2.5	2.5	1199.0058	676.18	1198.4367	-0.05	1.16
R101	3	0.6	[40, 60]	1.1	1.1	3027.5465	455.96	3013.2015	-0.47	-0.46
R101	3	0.6	[40, 60]	1.1	1.1	3244.0645	444.34	3237.4613	-0.20	-0.16
R101	3	0.6	[40, 60]	1.1	1.1	3030.9153	451	3023.2649	-0.25	-0.24
R101	3	0.6	[40, 60]	1.1	1.1	3115.1466	432.35	3109.0796	-0.19	-0.06
R101	3	0.6	[40, 60]	1.1	1.1	3112.9713	469.27	3105.3741	-0.24	-0.23
R101	3	0.6	[40, 60]	1.5	1.5	2224.7829	447.71	2224.3339	-0.02	0.60
R101	3	0.6	[40, 60]	1.5	1.5	2368.894	531.41	2361.1624	-0.33	0.11
R101	3	0.6	[40, 60]	1.5	1.5	2254.6679	528.31	2241.9773	-0.56	0.13
R101	3	0.6	[40, 60]	1.5	1.5	2313.743	459.05	2300.6904	-0.56	-0.55
R101	3	0.6	[40, 60]	1.5	1.5	2356.9995	463.99	2342.7353	-0.61	0.09
R101	3	0.6	[40, 60]	2	2	1787.6283	468.94	1784.5292	-0.17	0.55
R101	3	0.6	[40, 60]	2	2	1898.5402	573.18	1892.8283	-0.30	0.53
R101	3	0.6	[40, 60]	2	2	1776.213	560.2	1778.3489	0.12	0.66
R101	3	0.6	[40, 60]	2	2	1832.6458	615.95	1832.3726	-0.01	0.76
R101	3	0.6	[40, 60]	2	2	1850.7579	539.93	1850.1375	-0.03	0.30
R101	3	0.6	[40, 60]	2.5	2.5	1489.3193	475.82	1486.1101	-0.22	0.77
R101	3	0.6	[40, 60]	2.5	2.5	1587.9639	639.35	1597.7612	0.62	1.13
R101	3	0.6	[40, 60]	2.5	2.5	1505.6898	489.36	1508.3233	0.17	0.80
R101	3	0.6	[40, 60]	2.5	2.5	1555.8633	651.61	1557.2769	0.09	0.82
R101	3	0.6	[40, 60]	2.5	2.5	1554.1869	616.64	1550.2854	-0.25	0.90
R101	3	1	[1, 100]	1.1	1.1	3041.7396	893.51	3037.557	-0.14	1.04
R101	3	1	[1, 100]	1.1	1.1	2934.6564	837.63	2922.4492	-0.42	1.10
R101	3	1	[1, 100]	1.1	1.1	2964.3072	934.43	2974.5535	0.35	1.17
R101	3	1	[1, 100]	1.1	1.1	3079.849	813.04	3082.8521	0.10	0.65
R101	3	1	[1, 100]	1.1	1.1	2852.7976	944.08	2854.967	0.08	1.20
R101	3	1	[1, 100]	1.5	1.5	2325.9298	930.92	2336.5286	0.46	1.50
R101	3	1	[1, 100]	1.5	1.5	2250.625	945.72	2270.6179	0.89	2.28
R101	3	1	[1, 100]	1.5	1.5	2296.8218	1068.96	2325.5699	1.25	2.07
R101	3	1	[1, 100]	1.5	1.5	2365.1241	1020.65	2378.9664	0.59	1.27
R101	3	1	[1, 100]	1.5	1.5	2185.2957	956.33	2201.9572	0.76	1.85

Suite du Tableau B.3										
Instance						Gu et al. (2019) ALNS		SLNS		
Type	m	p	Δ	α	id	z^*	t	z^*	gap(%)	gap _{moy} (%)
R101	3	1	[1, 100]	2	2	1847.8667	1070.1	1863.0345	0.82	1.64
R101	3	1	[1, 100]	2	2	1800.5916	1023.86	1812.3865	0.66	1.47
R101	3	1	[1, 100]	2	2	1813.6036	1117.82	1838.3989	1.37	2.19
R101	3	1	[1, 100]	2	2	1871.2735	1052.49	1876.7997	0.30	1.74
R101	3	1	[1, 100]	2	2	1763.3915	1277.7	1766.9616	0.20	1.22
R101	3	1	[1, 100]	2.5	2.5	1561.1608	1208.35	1579.4633	1.17	2.41
R101	3	1	[1, 100]	2.5	2.5	1522.9296	1498.51	1523.7163	0.05	1.71
R101	3	1	[1, 100]	2.5	2.5	1527.6379	1474.97	1543.2072	1.02	2.45
R101	3	1	[1, 100]	2.5	2.5	1584.9913	1285.72	1597.8947	0.81	2.01
R101	3	1	[1, 100]	2.5	2.5	1494.2903	1395.11	1487.8288	-0.43	1.71
R101	3	1	[40, 60]	1.1	1.1	4605.8439	887.44	4570.1315	-0.78	-0.63
R101	3	1	[40, 60]	1.1	1.1	4510.2998	820.83	4484.1997	-0.58	-0.27
R101	3	1	[40, 60]	1.1	1.1	4509.015	848.02	4481.31	-0.61	-0.56
R101	3	1	[40, 60]	1.1	1.1	4643.9582	821.14	4614.3692	-0.64	-0.59
R101	3	1	[40, 60]	1.1	1.1	4518.3114	900.92	4495.5253	-0.50	-0.38
R101	3	1	[40, 60]	1.5	1.5	3379.7642	1001.82	3406.8775	0.80	1.76
R101	3	1	[40, 60]	1.5	1.5	3362.6067	1095.28	3375.5976	0.39	0.97
R101	3	1	[40, 60]	1.5	1.5	3385.7239	1056.57	3383.5237	-0.06	0.63
R101	3	1	[40, 60]	1.5	1.5	3408.6962	1021.9	3403.7073	-0.15	0.39
R101	3	1	[40, 60]	1.5	1.5	3374.5175	1022.52	3393.6124	0.57	1.10
R101	3	1	[40, 60]	2	2	2626.6402	1149.13	2631.5377	0.19	1.35
R101	3	1	[40, 60]	2	2	2604.937	1033.69	2631.2339	1.01	1.95
R101	3	1	[40, 60]	2	2	2582.7011	1123.78	2597.6205	0.58	1.53
R101	3	1	[40, 60]	2	2	2636.4516	1344.94	2641.1783	0.18	1.43
R101	3	1	[40, 60]	2	2	2620.8533	1138.19	2633.104	0.47	1.61
R101	3	1	[40, 60]	2.5	2.5	2195.1808	1226.16	2209.2057	0.64	2.42
R101	3	1	[40, 60]	2.5	2.5	2170.1547	1264.51	2191.8102	1.00	2.65
R101	3	1	[40, 60]	2.5	2.5	2179.0868	1255.9	2211.6611	1.49	3.00
R101	3	1	[40, 60]	2.5	2.5	2220.6163	1227.96	2249.57	1.30	2.75
R101	3	1	[40, 60]	2.5	2.5	2167.7968	1157.51	2185.3058	0.81	1.86

TABLEAU B.3 – Résultats de la méthode SLNS sur les instances C-SDVRP de Archetti et al. (2014a)

B.4 Résultats détaillés des instances VRPMTW

Instance	BKS	Larsen et Pacino (2019) ALNS		SLNS		
		r	z^*	r	z^*	gap (%)
cm101	12253.2	10	12253.2	10	12264.9	0.10
cm102	12382.4	12	12467.5	12	12613.94	1.87
cm103	12506.0	11	12506.0	12	12676.94	1.37
cm104	12911.9	13	12911.9	14	12940.98	0.23
cm105	12023.1	10	12038.1	10	12060.86	0.31
cm106	12043.6	10	12043.6	10	12057.67	0.12
cm107	12106.2	10	12106.2	10	12106.21	0.00
cm108	11984.3	10	11984.3	10	11986.37	0.02
cm201	13444.9	5	13472.0	5	13654.46	1.56
cm202	14020.2	6	14021.3	6	14121.48	0.72
cm203	13465.5	5	13465.5	5	13579.66	0.85
cm204	13323.9	5	13323.9	5	13448.19	0.93
cm205	12827.8	4	12852.1	4	12928.23	0.78
cm206	12713.2	4	12729.8	4	12786.29	0.57
cm207	12937.0	4	12937.0	4	13033.42	0.75
cm208	12729.8	4	12729.8	4	12774.88	0.35
rcm101	4056.8	10	4079.6	10	4080.6	0.59
rcm102	4178.4	10	4178.4	10	4186.4	0.19
rcm103	4132.5	10	4144.6	10	4156.0	0.57
rcm104	4128.3	10	4137.0	10	4155.0	0.65
rcm105	4114.9	10	4188.4	10	4195.5	1.96
rcm106	4167.6	10	4175.4	10	4175.4	0.19
rcm107	4410.1	11	4516.5	11	4517.7	2.44
rcm108	4480.6	11	4565.2	11	4565.2	1.89
rcm201	3733.8	2	3733.8	2	3767.7	0.91
rcm202	3756.1	2	3756.1	2	3781.9	0.69
rcm203	3716.5	2	3716.5	2	3797.5	2.18
rcm204	3691.7	2	3699.8	2	3697.1	0.15
rcm205	3709.7	2	3731.8	2	3735.0	0.68
rcm206	3725.0	2	3744.7	2	3739.6	0.39
rcm207	3859.1	2	3859.1	2	3886.4	0.71
rcm208	3731.7	2	3731.7	2	3737.9	0.17
rm101	4005.5	10	4014.0	10	4053.9	1.21
rm102	3729.9	9	3729.9	9	3754.0	0.65
rm103	3676.6	9	3700.6	9	3704.5	0.76
rm104	3682.5	9	3700.6	9	3705.5	0.62
rm105	3625.6	9	3686.6	9	3692.5	1.85
rm106	3622.6	9	3708.0	9	3708.0	2.36
rm107	3653.3	9	3689.9	9	3701.2	1.31
rm108	3687.1	9	3719.9	9	3722.1	0.95
rm201	3815.0	2	3815.0	2	3826.8	0.31
rm202	3706.8	2	3725.9	2	3725.6	0.51
rm203	3691.4	2	3696.1	2	3665.7	-0.70
rm204	3671.7	2	3681.5	2	3672.1	0.01
rm205	3668.1	2	3678.7	2	3661.1	-0.19
rm206	3672.6	2	3690.2	2	3679.6	0.19

Suite du Tableau B.4

Instance	BKS	Larsen et Pacino (2019) ALNS		SLNS		
		r	z^*	r	z^*	gap (%)
rm207	3662.4	2	3688.3	2	3667.4	0.14
rm208	3663.6	2	3676.2	2	3655.3	-0.23

TABLEAU B.4 – Résultats de la méthode SLNS sur les instances VRPMTW de Belhaiza et al. (2014)

Titre : Optimisation de la distribution de marchandises avec sous-traitance du transport : une problématique chargeur

Mot clés : Optimisation des transports, tournées de véhicules, recherche à voisinage large, métaheuristique

Résumé : Cette thèse a pour but d'optimiser le transport de marchandises dans un réseau logistique incluant des chargeurs et un ensemble d'entrepôts ou points de vente de la grande distribution. Nous étudions en particulier la problématique des chargeurs qui achètent des services de transport à différents prestataires logistiques proposant des tarifs FTL ou LTL. Dans une première partie, nous résolvons un problème de tournées de véhicules depuis un dépôt unique avec des flottes hétérogènes proposées par plusieurs transporteurs. Une caractéristique originale du problème est la formulation d'une contrainte d'engagement maximum des transporteurs en terme de nombre de véhicules.

Nous résolvons ce problème avec une métaheuristique de type Recherche à Voisinage Large intégrant un algorithme de filtrage issu de la programmation par contraintes. Dans une seconde partie, nous étudions un problème de collectes et de livraisons avec la possibilité de livrer les marchandises sur des plateformes intermédiaires. Des transporteurs régionaux assurent alors la distribution en aval de ces plateformes. Nous formulons ce problème comme un problème de collectes et livraisons avec options et proposons une métaheuristique alternant des destructions de petite et de grande taille. Nous concluons cette thèse par plusieurs cas d'étude menés au sein de la société CRC Services.

Title: Optimizing the delivery of goods with sub-contracted transport: the shipper's point of view

Keywords: Transport optimization, vehicle routing, large neighborhood search, metaheuristic

Abstract: This thesis aims to optimize the transportation of goods in a logistics network including a set of shippers and retail facilities. Specifically, we study the problem of shippers who purchase transportation services from different carriers offering FTL or LTL rates. In a first part, we solve a vehicle routing problem with a single depot and heterogeneous fleets offered by several carriers. An original feature of the problem is a constraint limiting the total number of vehicles offered by each carrier. We solve this problem with a Large Neighbor-

hood Search metaheuristic including a filtering algorithm arising from constraint programming. In a second part, we study a pickup and delivery problem with the possible use of intermediate platforms. The distribution from these platforms is ensured by regional carriers. This problem is formulated as a pickup and delivery problem with options that we solve with a metaheuristic alternating small and large destructions. We conclude this thesis with several case studies conducted within the CRC Services company.