



Optimisation de la logistique hospitalière : problème conjoint de tournées de véhicules et de planification des chauffeurs

Moise Aimé Noumbissi Tchoupo

► To cite this version:

Moise Aimé Noumbissi Tchoupo. Optimisation de la logistique hospitalière : problème conjoint de tournées de véhicules et de planification des chauffeurs. Gestion et management. Université de Technologie de Troyes, 2019. Français. NNT : 2019TROY0001 . tel-03613220

HAL Id: tel-03613220

<https://theses.hal.science/tel-03613220>

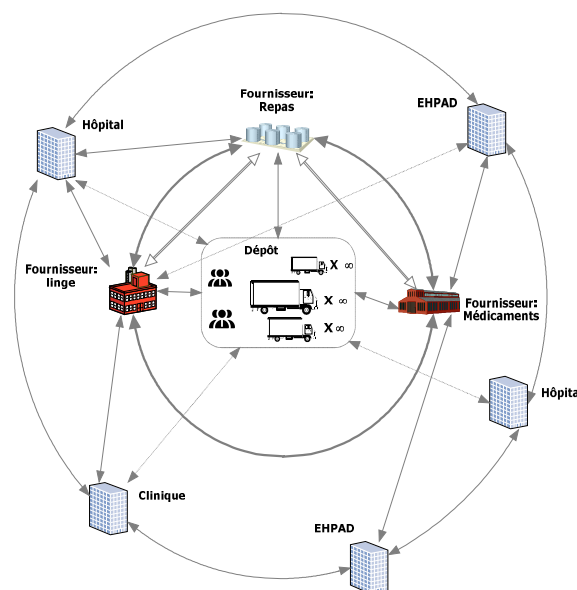
Submitted on 18 Mar 2022

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.

Moïse Aimé NOUMBISSI TCHOUPPO

Optimisation de la logistique hospitalière : problème conjoint de tournées de véhicules et de planification des chauffeurs



Champ disciplinaire :
Sciences pour l'Ingénieur

2019TROY0001

Année 2019

THESE
pour l'obtention du grade de
DOCTEUR
de l'UNIVERSITE DE TECHNOLOGIE DE TROYES
EN SCIENCES POUR L'INGENIEUR
Spécialité : OPTIMISATION ET SURETE DES SYSTEMES

présentée et soutenue par
Moïse Aimé NOUMBISSI TCHOUPPO
le 11 janvier 2019

**Optimisation de la logistique hospitalière : problème conjoint
de tournées de véhicules et de planification des chauffeurs**

JURY

Mme. O. BATTAIA	PROFESSEURE - HDR	Présidente
M. O. GRUNDER	MAITRE DE CONFERENCES - HDR	Rapporteur
M. A. NAKIB	MAITRE DE CONFERENCES - HDR	Rapporteur
M. E.-G. TALBI	PROFESSEUR DES UNIVERSITES	Examineur
M. F. YALAOUI	PROFESSEUR DES UNIVERSITES	Examinatuer
M. L. AMODEO	PROFESSEUR DES UNIVERSITES	Directeur de thèse
Mme A. YALAOUI	MAITRE DE CONFERENCES - HDR	Directrice de thèse

Personnalité invitée

Mme. P. FLORI	DIRECTRICE DES ACHATS ET DE LA LOGISTIQUE - CHT
---------------	---

Remerciements

Les travaux présentés dans cette thèse ont été effectués dans le cadre d'une convention CIFRE avec la société Centre Hospitalier de Troyes (CHT) en collaboration avec l'Université de Technologie de Troyes (UTT).

J'exprime toute ma gratitude aux rapporteurs Dr. Olivier GRUNDER et Dr. Amir NAKIB, qui me font l'honneur de participer à l'examen de cette thèse.

Je remercie également Dr. Olga BATTAIA et Pr. El-Ghazali TALBI pour l'honneur qu'ils me font d'être dans mon jury de thèse.

Il me sera très difficile de remercier tout le monde car c'est grâce à de nombreuses personnes que j'ai pu mener cette thèse à son terme.

Je remercie chaleureusement mes directeurs de thèse, Dr. Alice YALAOUI et Pr. Lionel AMODEO, pour toute leur aide. Je suis ravi d'avoir travaillé en leur compagnie car, outre leur appui scientifique et technique, ils ont toujours été là pour me soutenir et me conseiller.

Mes remerciements vont également à mes encadrants successifs M. Frederic LUTZ et Mme Pauline FLORI pour toute l'expertise technique et leurs conseils tout au long de cette thèse. Je les remercie aussi pour le temps précieux qu'ils m'ont accordé afin de définir et d'étudier les problèmes de la logistique hospitalière.

Je tiens à remercier grandement Pr. Farouk YALAOUI pour sa participation personnelle et scientifique dans toutes les étapes de cette thèse, pour son enthousiasme scientifique, ses encouragements, et pour avoir accepté de participer à mon jury de thèse.

Je remercie également Mr. Philippe BLUA, directeur des Hôpitaux Champagne-Sud, qui a su inspirer et accompagner ce sujet de thèse.

Je tiens à remercier particulièrement Mr. Eric THIVET pour toutes nos discussions et ses conseils. Son accent sur le facteur humain et la prise en compte du bien-être au travail m'ont particulièrement marqué.

Je ne saurais comment remercier mon feu père Victor TCHOUPPO, ma mère Cathérine NZETCHUENG, mon frère Romuald et mes sœurs (Annie, Gaëlle et Vanessa). J'espère un jour

pouvoir trouver les bons mots.

Je remercie toutes les personnes avec qui j'ai partagé mes études et notamment ces années de thèse.

Mes derniers remerciements vont à mon oncle Mr. Gabriel NCHULA qui m'a soutenu dans tout ce que j'ai entrepris.

Résumé

Nos travaux de recherche ont été motivés par une problématique réelle d'optimisation du transport logistique du Centre Hospitalier de Troyes (CHT). Le CHT s'inscrit actuellement dans la revue et l'amélioration des processus et l'implémentation des techniques de recherche opérationnelle afin d'apporter des solutions aux problèmes conjoints de transports de produits et planification des chauffeurs.

L'objectif de cette étude est de proposer des solutions efficaces afin d'assurer au quotidien le transport des chariots de repas, de linge, de produits pharmaceutiques et de produits magasiniers. Après la formalisation du problème, nous avons proposé une démarche de résolution composée de trois phases : la première traite du problème connu de collectes et livraisons avec fenêtre de temps et flotte homogène de véhicules, la seconde considère une flotte hétérogène de véhicules, et la troisième phase intègre l'affectation et le dimensionnement de l'équipe des chauffeurs. Nous avons abouti au développement et à la mise en place d'un outil d'aide à la décision. Nous avons suite à nos travaux, développé et mis en place, en partenariat avec la start-up OPTA-LP, un logiciel au CHT. Tout au long de nos travaux, nous avons développé des méthodes de résolution exactes et approchées, et aussi élaboré des techniques pour diminuer les temps de calculs dans une approche de résolution exacte, ou encore pour améliorer une solution donnée dans le cas d'une méthode approchée.

Abstract

Our research study was motivated by a real life problem of optimization of the logistic transportation system at the Champagne Sud Hospital (HCS). The today's challenge for the HCS, the Aube Territorial Hospital Group, and the Sézannais is to integrate the new shared territorial organization of logistics into a real coordination of all the resources, means, and technical tools. In addition to meet the requirements of the security and the continuity of supply chain, the challenges are to optimize logistics performance thanks to decision support tools considering the organization of logistics flows between hospitals, as well as the fleet management.

The objective of this study is to propose an effective solution to ensure the daily transport of food trolleys, linens, pharmaceutical and warehouse products. After formalizing the problem, a three-phase resolution approach was proposed: the first deals with the known problem of pickup and delivery problem with a time window and a homogeneous fleet, the second considers a heterogeneous fleet of vehicles, and the third phase integrates the planning of the drivers team. Finally, the development and implementation of a decision support tool was realized. Throughout our work, several exact and approximate resolution methods were developed. Some techniques were proposed, on the one hand to reduce resolution time of the proposed approaches, on the other hand, to improve a given solution in the case of approximate methods.

Table des matières

Remerciements	i
Résumé	iii
Abstract	v
Table des matières	vii
Liste des tableaux	xi
Liste des figures	xiii
Introduction	1
1 Contexte et problématique	7
1.1 Introduction	7
1.2 Le Centre Hospitalier de Troyes	8
1.2.1 Les Hôpitaux Champagne Sud	8
1.2.2 Le service Intérieur	9
1.2.3 La plate-forme logistique relais du CHT	10
1.2.4 La modernisation de la logistique au CHT	12
1.3 La recherche opérationnelle et le milieu hospitalier	13
1.4 Problématiques	16
1.5 Conclusion	19
2 État de l’art	21
2.1 Introduction	21
2.2 PDPTW Flotte homogène	21
2.2.1 Méthodes exactes	24
2.2.2 Méthodes approchées	24
2.3 PDPTW Flotte hétérogène	27
2.4 Problème conjoint de tournées de véhicules couplé à la planification du personnel	30
2.4.1 Problèmes conjoint de collecte et livraison avec fenêtre de temps et de planification du personnel	31
2.4.2 Problèmes conjoint de logistique hospitalière externe et de planification du personnel	34
2.5 Conclusion	35

3	Collecte et livraisons avec fenêtres de temps : flotte homogène	37
3.1	Introduction	37
3.2	Résolution exacte du problème	39
3.2.1	Modélisation par un programme linéaire en nombres mixtes	39
3.2.2	Bornes sur les fenêtres de temps	43
3.2.3	Réduction de l'espace de recherche	43
3.2.4	Inégalités valides	44
3.2.5	Résultats expérimentaux	47
3.3	Résolution par métaheuristique	48
3.3.1	Algorithme de colonie de fourmis	48
	Principe général	48
	Codage d'une solution	49
	Construction d'une solution	49
	Vérification de l'éligibilité	50
	Mise à jour des phéromones	51
3.3.2	Algorithmes de recherche locale	52
	Algorithme de recherche locale H1 : réduction du nombre de routes	52
	Algorithme de recherche locale H2 : réduction de la distance de la solution dans une tournée	53
	Algorithme de recherche locale H3 : réduction de la distance de la solution dans une solution	53
3.3.3	Expérimentations	55
	Analyse de sensibilité	55
	Résultats expérimentaux	56
3.4	Conclusion	57
4	Collectes et livraisons avec fenêtre horaires : flotte hétérogène	59
4.1	Introduction	59
4.2	Problème de collectes et livraisons avec fenêtres horaires et flotte hétérogène de véhicules : cas d'une flotte de taille finie	60
4.2.1	Programme linéaire en variables mixtes	61
	Modélisation mathématique	61
4.2.2	Décomposition de Benders	62
4.2.3	Expérimentations	64
	Expérimentations des algorithmes (PL4) et (Q1)	64
	Expérimentations des algorithmes (PL4) et (Q1) avec addition des techniques de réduction de l'espace de recherche	65
	Expérimentations des algorithmes (PL4) et (Q1) avec addition des techniques de réduction de l'espace de recherche et inégalités valides	65
	Conclusion	65
4.3	Problème de collecte et livraisons avec fenêtre horaires et flotte hétérogène de véhicules : cas d'une flotte de taille infinie	66
4.3.1	Modélisation	67
4.3.2	Algorithme de génération de colonnes	68
	Heuristique	70
	Formulation d'un problème de pricing pour un type de véhicule	71
	Algorithme à labels	74

4.3.3	Expérimentations	76
4.3.3	Algorithme hybride de colonie de fourmis	79
	Construction d'une solution	79
	Expérimentations	80
	Impact des algorithmes de recherche locale	84
4.4	Conclusion	89
5	Optimisation des tournées et planification des chauffeurs	91
5.1	Introduction	91
5.2	Description du problème	92
5.3	Modélisation mathématique	93
5.3.1	Règles sur le planning journalier des chauffeurs	95
5.3.2	Contraintes d'incompatibilité	95
5.3.3	Modèle mathématique	97
5.4	Méthode de résolution : Algorithme hybride de colonie de fourmis	100
5.4.1	Construction d'une solution	101
5.4.2	Initialisation et mise à jour des phéromones	102
5.4.3	Heuristiques de recherche locale	103
	Procédure d'insertion d'une requête dans un planning	103
	Heuristique de recherche locale H1 : diminution du nombre de chauffeur	104
	Heuristique de recherche locale H2 : diminution du coût d'acquisition de la flotte	105
	Heuristiques de recherche locale H3, H4 et H5 : diminution du coût dû à la distance parcourue	106
5.5	Expérimentations	109
5.5.1	Procédure de génération d'un benchmark	110
5.5.2	Expérimentations sur l'algorithme hybride de colonie de fourmis	111
	Procédure de fixation des valeurs des paramètres	112
	Impact des recherches locales et résultats des expérimentations	112
	Comparaison de l'HACO avec le modèle linéaire	113
5.5.3	Application sur un cas industrielle : Centre Hospitalier de Troyes	113
5.6	Conclusion	118
6	Outils d'aide à la décision	121
6.1	Introduction	121
6.2	Présentation de l'application : OPTA Logistics®	122
6.2.1	Accueil	124
6.2.2	Administration	124
6.2.3	Mises à jour des matrices de distances et de temps de parcours entre les sites	124
6.2.4	Lancer une optimisation	124
6.2.5	Planning	124
6.2.6	Gant	128
6.3	Conclusion	128
7	Annexe	135
	Bibliographie	139

Liste des tableaux

1.1	Nombre d'agents nécessaires à chaque poste pour chaque jour de la semaine en 2015.	10
2.1	Classification de quelques travaux.	23
2.2	Méthodes exactes de résolution du PDPTW.	25
2.3	Heuristiques et méta-heuristiques pour le PDPTW.	28
2.4	Synthèse sur le PDPTW avec une flotte hétérogène de véhicules.	29
2.5	Quelques travaux sur la planification du personnel (André [1]).	32
2.6	Publications sur les problèmes de la logistique hospitalière externe couplé au problème de planification du personnel (voir Volland et al. [2])	35
3.1	Notations	40
3.2	Résultats expérimentaux sur les instances réelles.	47
3.3	Résultats des expérimentations de l'HACO sur le Benchmark à 100 demandes de Li and Lim [3].	56
3.4	Résultats des expérimentations de l'HACO sur le Benchmark à 100 demandes de Li and Lim [3].	57
4.1	Notations	61
4.2	Instances réelles sans réduction de l'espace de recherche et sans coupes.	64
4.3	Instances réelles avec réduction de l'espace de recherche et sans coupes.	65
4.4	Instances réelles avec réduction de l'espace de recherche et inégalités valides.	66
4.5	Notations	67
4.6	Résultats obtenus sur les instances avec une distribution géographique groupée des demandes.	78
4.7	Résultats obtenus sur les instances avec une distribution géographique uniforme des demandes.	78
4.8	Résultats obtenus sur les instances avec une distribution géographique semi-groupée des demandes.	79
4.9	Paramètres de l'HACO	82
4.10	Résultats sur le benchmark de $T20C1$ et $T20C2$	83
4.11	Résultats sur les benchmarks $T20RC1$ et $T20RC2$	83
4.12	Résultats sur les benchmarks $T20R1$ et $T20R2$	84
4.13	Résumé des expérimentations sur l'étude de l'impact des recherches locales sur la qualité de la solution retournée par l'algorithme HACO.	85
4.14	Expérimentations ACO + H1 sur le benchmark FLT100R.	86
4.15	Expérimentations ACO + H1 sur le benchmark FLT100RC.	86
4.16	Expérimentations ACO + H1 sur le benchmark FLT100C.	86

4.17	Expérimentations ACO + H1 et H2, sur le benchmark FLT100RC.	87
4.18	Expérimentations ACO + H1 et H2, sur le benchmark FLT100C.	87
4.19	Expérimentations ACO + H1, H2 et H3 sur le benchmark FLT100RC.	87
4.20	Expérimentations ACO + H1, H2 et H3, sur le benchmark FLT100C.	88
4.21	Expérimentations ACO + H1, H2 et H3, sur le benchmark FLT100R.	88
4.22	Expérimentations ACO + H1 et H2, sur le benchmark FLT100R.	88
5.1	Notations du chapitre 5.	94
5.2	Paramètres pour modéliser les règles d'un planning de chauffeur.	96
5.3	Types de Véhicule	110
5.4	Différents types de journées	111
5.5	Paramètres et domaine des valeurs.	112
5.6	Expérimentations de l'HACO et du MILP sur des instances à 20 demandes. . .	113
5.7	Expérimentations ACO + H1 sur le benchmark T100C.	114
5.8	Expérimentations ACO + H1 sur le benchmark T100R.	114
5.9	Expérimentations ACO + H1 H1 sur le benchmark T100RC.	114
5.10	Expérimentations ACO + H1 et H2, sur le benchmark T100C.	115
5.11	Expérimentations ACO + H1 et H2, sur le benchmark T100R.	115
5.12	Expérimentations ACO + H1 et H2, sur le benchmark T100RC.	115
5.13	Expérimentations ACO + H1, H2 et H3, sur le benchmark T100C.	116
5.14	Expérimentations ACO + H1, H2 et H3, sur le benchmark T100R.	116
5.15	Expérimentations ACO + H1, H2 et H3, sur le benchmark T100RC.	116
5.16	Types de véhicules	117
5.17	Solution Industrielle VS Solution de l'HACO	118
7.1	Expérimentations HACO et MILP sur le benchmark à 20 demandes FLT20C1.	135
7.2	Expérimentations HACO et MILP sur le benchmark à 20 demandes FLT20C2.	135
7.3	Expérimentations HACO et MILP sur le benchmark à 20 demandes FLT20R1.	137
7.4	Expérimentations HACO et MILP sur le benchmark à 20 demandes FLT20R2.	137
7.5	Expérimentations HACO et MILP sur le benchmark à 20 demandes FLT20RC1.	137
7.6	Expérimentations HACO et MILP sur le benchmark à 20 demandes FLT20RC2.	138

Liste des figures

0.1	Hôpitaux Champagne Sud.	4
0.2	Démarche de résolution.	5
1.1	Répartition du budget des HCS	8
1.2	Répartition des investissements des HCS	9
1.3	Plate-forme logistique du CH de Troyes à Saint Julien Les Villas.	10
1.4	Entrée du nouveau bâtiment d'accueil du CHT.	12
1.5	Véhicules à guidage automatique	12
1.6	Exemple de chariot de repas.	13
1.7	Les différentes fonctions de la logistique selon Chow et al. [4]	15
1.8	Représentation de réseau logistique étudié.	17
1.9	Problématique selon le niveau de décision.	18
2.1	Exemple illustrant quatre formes de solution pour les problèmes de type $1-M-1/P-D/-$. (P = collecte, D = livraison, PD = collecte et livraison combinés). . .	23
3.1	Exemple d'une solution d'un problème de collectes et livraisons avec fenêtre de temps et flotte homogène de véhicules composé de 20 demandes et 3 véhicules.	38
3.2	Exemple d'éligibilité d'une demande de collecte et d'une demande de livraison	50
3.3	Exemple de suppression d'une tournée par l'algorithme H1.	53
3.4	Exemple de réduction du coût d'une tournée par l'algorithme H2.	54
3.5	Exemple de mouvement réalisé par l'algorithme H3	54
4.1	Exemple d'une solution de PDPTW avec flotte hétérogène.	60
4.2	Décomposition de Dantzig-Wolfe	69
4.3	Processus de détermination du problème de pricing.	70
4.4	Algorithme de génération de colonnes résolut avec le modèle linéaire en nombres mixtes.	75
4.5	Algorithme de génération de colonnes résolu avec l'algorithme à labels.	77
4.6	Impact des algorithmes de recherche locale sur le coût de la solution.	85
4.7	Impact des algorithmes de recherche locale sur le temps de résolution.	85
5.1	Exemple de solution du problème étudié.	92
5.2	Exemple de différents états de travail d'un chauffeur dans un planning.	95
5.3	Exemple d'utilisation d'un véhicule dans deux plannings de chauffeur.	96
5.4	Exemple d'une solution construite par une fourmi.	102
5.5	Suppression d'un planning dans une solution	105
5.6	Modification du type d'un véhicule dans une solution.	106
5.7	Suppression et réinsertion d'une requête dans une solution.	107

5.8	Suppression d'une tournée et réinsertion des requêtes dans une solution.	108
5.9	Réorganisation de l'ordre de réalisation des demandes d'une tournée.	109
6.1	Architecture d'OPTA Logistics.	123
6.2	Une image de l'écran d'accueil.	125
6.3	Une image de la page des différents types de journée.	126
6.4	Image de la page de mise à jour des des distances et des temps de parcours. . .	127
6.5	Image de la page de lancement d'une optimisation.	127
6.6	Image d'un planning de camion pour une journée donnée.	128
6.7	Image des cartes d'un chauffeur pour une journée.	129
6.8	Image du planning des chauffeurs pour une journée donné.	129
6.9	Image du planning des camions pour une journée donné.	130
7.1	Exemple d'une instance réelle (celle de dimanche).	136

Introduction générale

La garanti de la qualité des services supports dans un hôpital est un enjeu quotidien important que doivent considérer l'ensemble des services supports d'un centre hospitalier. S'agissant d'approvisionner différents services, par exemple, une pénurie de tenues propres et de champs stériles n'est pas envisageable dans un bloc opératoire, les repas pour les patients hospitalisés sont obligatoires, il en est de même pour le changement des draps pour minimiser les risques infectieux dans les services. La gestion des approvisionnements est donc un maillon important permettant de garantir le bon fonctionnement de nombreux services.

Les flux logistiques à l'hôpital obéissent donc à des contraintes de régularité des approvisionnements : en ce qui concerne la livraison de repas, ceux-ci doivent être assurés bien entendu quotidiennement, deux fois par jour. Du linge propre à destination des professionnels chaque jour, et ainsi que pour changer le linge du lit des patients. Les fournitures pharmaceutiques et non pharmaceutiques doivent être à disposition en quantité suffisante des équipes soignantes. A cette contrainte de régularité des livraisons, s'ajoute la contrainte de respect des délais de distribution, particulièrement forte à l'hôpital.

Les repas doivent en effet être livrés dans un intervalle de temps précis et restreint. Tout retard en matière de livraison de repas peut être source de dysfonctionnements pénalisant l'organisation des soins. Un retard du service du repas en service de maison de retraite de personnes âgées dépendantes, va reporter le moment de disponibilité des professionnels pour l'aide au coucher des résidents les plus dépendants. Ces enjeux de respect des délais de distribution sont complétés par des exigences de délais de désapprovisionnement.

L'organisation de la collecte du linge sale doit ainsi en milieu hospitalier éviter une stagnation trop longue du linge sale dans les services de soins, vecteur potentiel de contaminations, et source d'une éventuelle dégradation de la qualité du linge (risque d'incrustation des taches). Par ailleurs, une blanchisserie hospitalière peut fournir du linge propre aux services de soins à la condition qu'elle reçoive au préalable du linge sale de ces services. Le désapprovisionnement du linge sale et son retour en blanchisserie conditionnent donc les capacités de production de linge propre et d'approvisionnement des services.

De même, pour de nombreux flux logistiques, les différents containers servant à l'approvision-

nement servent également au désapprovisionnement.

Les flux logistiques hospitaliers s'inscrivent donc dans une chaîne continue d'exigences horaires, tant en approvisionnement des services de soins qu'en désapprovisionnement. Ces exigences sont fonction des contraintes géographiques, architecturales, et des choix organisationnels faits par l'établissement.

Dans un contexte de financement de l'activité hospitalière par la tarification à l'activité (T2A), et des recettes versées très majoritairement par l'assurance Maladie, la logistique hospitalière n'échappe pas à l'exigence d'optimisation économique.

D'après Ferebee [5], la logistique peut être définie comme la fourniture de biens et de services d'un point d'approvisionnement à un point de demande. Le maillon le plus coûteux dans une chaîne logistique est la livraison des produits aux clients (d'après Ball et al. [6]). Il existe de nombreux cas réels dans lesquels le personnel doit exécuter des tâches à différents endroits, ce qui nécessite des moyens de transport. Par exemple, les chauffeurs de camions qui acheminent les marchandises entre différents endroits, les infirmières qui rendent visite aux patients à domicile, les techniciens qui effectuent des réparations chez les clients, et les agents de sécurité qui font des rondes dans différents locaux. Ces problèmes combinent à la fois des problèmes de tournées de véhicules et d'affectation du personnel. L'objectif est d'assurer que les employés arrivent à l'heure à l'endroit où les tâches doivent être exécutées, et ceci à moindre coût.

La problématique traitée dans ce travail de thèse est un problème concret rencontré au Centre Hospitalier de Troyes (CHT). Les travaux menés pour apporter des solutions efficaces ont été réalisés en collaboration avec le laboratoire LOSI (Laboratoire d'Optimisation des Systèmes Industriels) de l'Université de Technologie de Troyes. Ceci a pu se réaliser grâce au dispositif Cifre (Conventions Industrielles de Formation par la Recherche) qui subventionne toute entreprise de droit français qui embauche un doctorant pour le placer au cœur d'une collaboration de recherche avec un laboratoire public.

En concert avec le plan hôpital 2002 (et 2012), et afin d'améliorer l'efficacité de ses diverses activités, le Centre Hospitalier de Troyes a entrepris la réalisation de projets tels que :

- la construction d'une plate-forme logistique (cuisine, blanchisserie, magasin),
- la construction d'un bâtiment pour l'IRM et le scanner,
- la mise en service d'un nouveau plateau technique à la disposition de la pharmacie, des blocs opératoires et de la chirurgie ambulatoire,
- la réalisation des travaux qui ont abouti à la restructuration du service des urgences ; mené dans le cadre de deux thèses CIFRE en collaboration avec le LOSI, dont une soutenue par Dr. Mohamed AFILAL en décembre 2017, intitulée : "Optimisation de la prévision et de la planification des activités d'un centre d'urgence hospitalier".
- les travaux d'extension du service de radiothérapie,

-
- la construction d'un bâtiment de 80 lits de maison de retraite à Nazareth.

Le CHT est un ensemble d'établissements de santé constitué de :

- l'hôpital des Simone Veil,
- la résidence Comte Henri,
- la maison de retraite Domaine de Nazareth.

Le périmètre des échanges (de flux), et donc de l'étude, s'est agrandi lorsque a été signé le 8 juillet 2016, la convention constitutive du Groupement Hospitalier de Territoire (GHT) de l'Aube et du Sézannais. Ce regroupement est constitué en tant que membres des Hôpitaux Champagne Sud (HCS) comprenant le CHT, et en tant que partenaires :

- L'établissement d'hébergement pour personnes âgées dépendantes (EHPAD) de la résidence Pierre d'Arcis
- L'EHPAD résidence Sainte Marthe
- L'EHPAD résidence Les Tilleuls
- L'EHPAD résidence La Belle Verrière
- L'ASIMAT
- Le Groupement de coopération Sanitaire composé du CHT, de la Mutualité Française de l'Aube et du Centre de Rééducation et de Réadaptation Fonctionnelle Pasteur.

Le CHT fait parti de groupement d'établissement de santé nommé Hôpitaux Champagne Sud (HCS). Ce groupement de six établissements (voir figure 0.1) agit chaque jour dans l'objectif d'assurer des soins de qualité et de bien accompagner les 340 000 habitants du département de l'Aube et du Sézannais dans leur parcours de santé. Les HCS sont formés :

- du Centre Hospitalier de Troyes,
- du Groupement Hospitalier Aube-Marne (GHAM) composé des établissements de Romilly-sur-Seine, Nogent-sur-Seine et Sézanne,
- de l'établissement publique de santé mentale de l'Aube (EPSMA) de Brienne-le-Château,
- du Centre Hospitalier de de Bar-sur-Aube,
- du Centre Hospitalier de de Bar-sur-Seine,
- de l'EHPAD résidence Cardinal de Loménie.

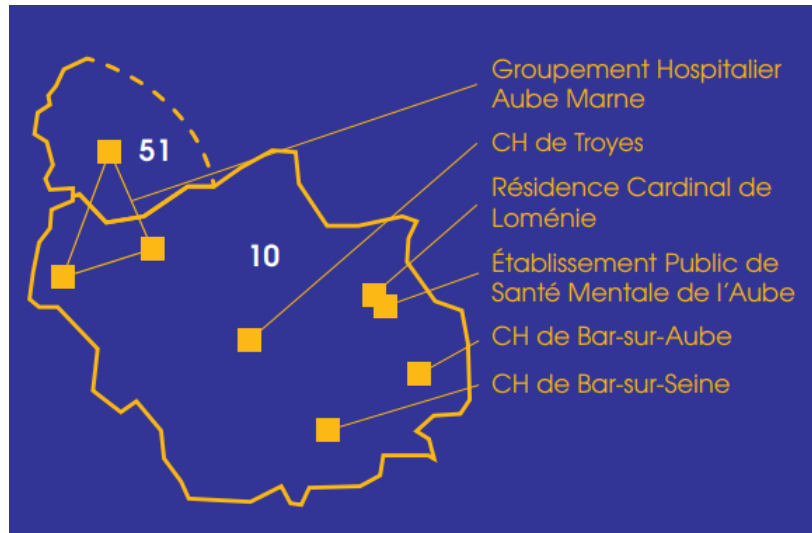


FIGURE 0.1 – Hôpitaux Champagne Sud.

Les activités de distribution et collecte en repas, linges, produits pharmaceutiques et produits magasiniers du CHT s'étendent sur une zone de plus de 50 km. cette activité est assurée par une équipe de chauffeurs, utilisant une flotte de véhicules acquis par location et dont le dépôt est situé à l'hôpital Simone Veil.

L'objectif de nos travaux qui ont débuté en 2015, a été de répondre de manière efficiente en matière de coût et de temps de réponse, aux questions :

- Quel est le meilleur dimensionnement de l'équipe des chauffeurs ?
- Quelle type de parc de véhicules a-t-on besoin ?
- Pour chaque demande, quel est le chauffeur qui la réalise et avec quel véhicule ?

La solution apportée à ces questions doit être en mesure de :

- Satisfaire toutes les demandes.
- Respecter la contrainte sur la fenêtre de temps pour débiter le service d'une demande.
- Respecter la contrainte de capacité des véhicules.
- Respecter le règlement sur les temps de travail, de conduite et de pause.
- Respecter les contraintes d'incompatibilité entre les différents types de produits, les contraintes d'incompatibilité entre types de produit et types de véhicules, et les contraintes d'incompatibilité entre types de véhicules et les sites de collecte et de livraison.

Les enjeux de cette thèse sont multiples, mais peuvent être regroupés en deux points :

- Assurer les approvisionnements des activités de soins du CHT et de ses différents partenaires.
- Répondre à l'exigence de performance économique qui concerne de manière générale les établissements publics de santé.

Pour faire face à ce double défi qualitatif et économique, une décomposition en sous-problèmes a été réalisée. Une solution globale a été construite à partir des solutions apportées à ces sous-problèmes. Nous avons dans un premier temps traité le problème de tournées de véhicules sous-jacent, et ensuite nous avons ajouté la planification des chauffeurs.

La démarche de résolution composée de trois étapes :

1. étude du problème de collectes et livraisons avec fenêtre de temps dans le cas d'une flotte homogène de véhicules,
2. étude du problème précédent étendu au cas d'une flotte hétérogène de véhicules,
3. étude du problème précédent combiné à la planification des chauffeurs.

L'approche proposée est schématisée par la figure 0.2.

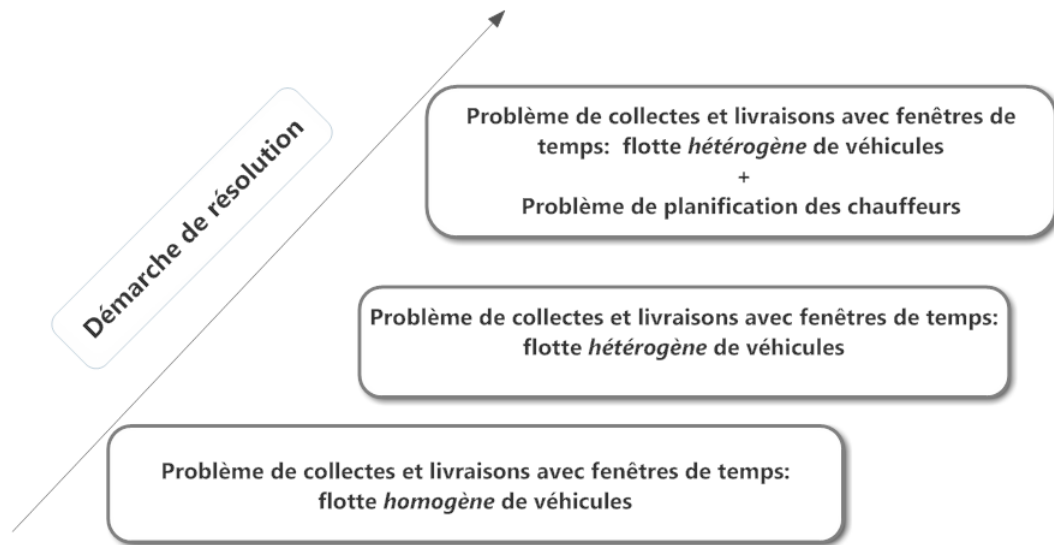


FIGURE 0.2 – Démarche de résolution.

Dans le prochain chapitre, nous verrons plus en détail le contexte et l'environnement qui ont été à l'origine des études menées sur cette problématique. Un état de l'art sur les problèmes abordés dans cette thèse est réalisé au chapitre 2. Ensuite, le problème de collectes et livraisons avec fenêtre de temps est traité au chapitre 3, dans le cas d'une flotte homogène de véhicules. Dans la continuité de notre démarche de résolution, le chapitre 4 présente les travaux menés dans le cas d'une flotte hétérogène. Le problème conjoint de collectes et livraisons avec fenêtre de temps dans le cas d'une flotte hétérogène couplé à la planification des chauffeurs est traité au chapitre 5. Un outil d'aide à la décision résultant de nos travaux et mis en place au CHT est présenté et décrit au chapitre 6.

Chapitre 1

Contexte et problématique

1.1 Introduction

L'année 2014 restera une année de forte mobilisation et d'implication de la direction générale de l'offre des soins (DGOS) sur les grands enjeux que connaît actuellement l'offre de soins en France. Le projet de loi porté en 2014 sur la modernisation, la simplification et l'innovation pour un égal accès de tous à des soins continus et de qualité ont facilité entre autres de mettre en place des groupements hospitaliers de territoire, la santé numérique, et la professionnalisation de la fonction achat.

D'après la DGOS, les résultats financiers des hôpitaux publics en France en 2016 ont été à l'équilibre ou en excédent pour 54%, et en déficit pour 46%. Mais derrière ces chiffres, se cache une autre réalité, car seulement 14 % des établissements concentrent 80% du déficit.

La Fédération hospitalière de France (FHF) dans son [rapport annuel 2017](#), a noté que la situation financière globale des hôpitaux publics en France s'est aggravé avec un déficit qui a doublé, avoisinant le milliard d'euros. En effet, en 2016 ce déficit était de 470 millions d'euros. Les solutions proposées par FHF pour faire des économies reposent sur quatre axes : la confiance envers les professionnels, l'équité entre les acteurs de santé, l'innovation et l'investissement.

Les travaux de Henning [7], Kowalski [8], Bourgeon et al. [9] ont porté sur l'évaluation de l'activité de la logistique dans le milieu hospitalier en France, et estiment la part de cette dernière entre 31% et 34%. Chow et al. [4] estime à 46% la part de l'activité logistique dans le budget d'exploitation d'un hôpital. Davis [10] estime cette part à 25%, et conclut qu'il est possible de réduire de 15% les dépenses liées à la logistique. Enfin, Davis [10] conclut qu'il est possible de réduire les coûts liés à la logistique hospitalière de 15%, ce qui démontre l'importance en terme d'économie, d'une réorganisation de l'activité logistique des hôpitaux.

C'est dans ce contexte que le Centre Hospitalier de Troyes (CHT) a établi une collaboration

avec l'Université de Technologie de Troyes, représentée par le laboratoire d'optimisation des systèmes industriels (LOSI), afin de mener dans le cadre d'une thèse CIFRE, des travaux d'optimisation de sa chaîne logistique.

Dans ce chapitre, nous décrivons l'environnement dans lequel ce travail de thèse s'inscrit. Ensuite, nous énumérerons des liens entre la recherche opérationnelle et des problématiques rencontrés dans le milieu hospitalier. Et enfin, nous présentons la problématique rencontrée au sein du CHT, et dont découlent nos travaux.

1.2 Le Centre Hospitalier de Troyes

Le centre hospitalier de Troyes est l'un des établissements des HCS. Ces établissements assurent la majorité des soins auxquels un patient peut avoir besoin. Ceci passe par leurs compétences en médecine, chirurgie, obstétrique, psychiatrie, soins à domicile, EHPAD, soins de longue durée, soins de suite et de réadaptation. Dans cette section, nous donnons quelques chiffres clés des HCS, et présentons les structures du CHT.

1.2.1 Les Hôpitaux Champagne Sud

En 2017, les moyens humains des HCS étaient composés de :

- 4 223 emplois, dont 275 médicaux et 3 947 non médicaux.
- 1 508 étudiants, dont 951 stagiaires paramédicaux, 183 stagiaires administratifs et techniques et 210 internes.

Le budget global des HCS s'élevait à 365,4 millions d'euros en 2017, et dont la répartition est donnée par la figure 1.1.

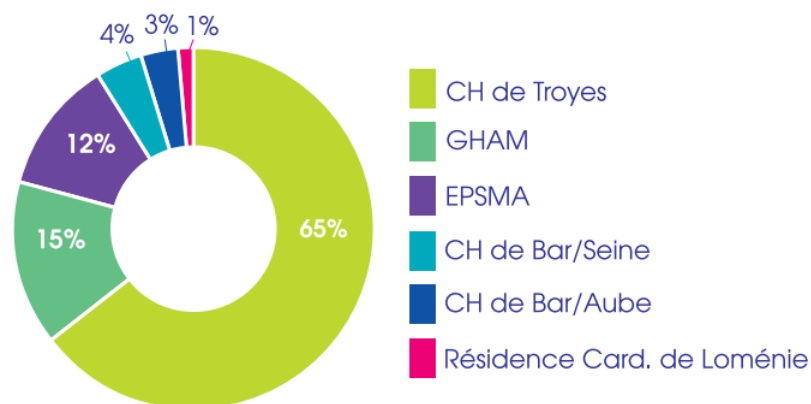


FIGURE 1.1 – Répartition du budget des HCS

La figure 1.2 donne la répartition des investissements sur l'ensemble des établissements des HCS en 2017, avec un montant global de : 16,5 millions d'euros.

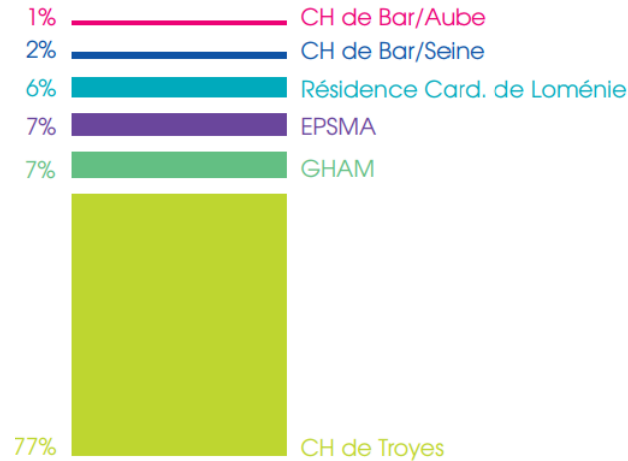


FIGURE 1.2 – Répartition des investissements des HCS

Ces moyens ont permis de prendre en charge 570 224 patients, avec 34 249 admissions.

Le CHT est un acteur clé des HCS avec en 2017, un investissement chiffré à hauteur de 11,2 millions d'euros, et des dépenses d'exploitation tous budgets confondus 237,2 millions. Le CHT est le premier employeur de l'Aube avec 203 médecins et 2 208 agents soignants, éducatifs, techniques, administratifs et médico-techniques.

Les principaux acteurs clés dans la logistique au CHT sont :

- le service intérieur,
- la plateforme logistique relais situé à Saint-Julien-les-Villas.

1.2.2 Le service Intérieur

Le Service intérieur est le service qui assure la gestion de la logistique au quotidien au CHT. Les missions qui lui sont assignées sont :

- assuré la collecte et la livraison des produits (repas, linge, médicaments et produits magasiniers) ;
- effectué les courses pour la distribution du courrier administratif.
- collecter et conditionner les déchets ;
- assuré la liaison entre les AGV et les services. Pour les bâtiments dont les AGV n'ont pas accès, le service intérieur achemine les chariots en utilisant des véhicules de manutention.

Afin d'assurer les missions qui lui sont confiées, le service intérieur dispose d'un certain nombre d'agents qui effectuent des roulements quotidiens sur 5 postes différents de travail (table 1.1).

Le service intérieur réalise le roulement des agents sur ces différents postes de travail de façon à respecter le statut spécial de la fonction publique hospitalière en France, mais aussi les

	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
Poste coupure	1	1	1	1	1	0	0
Poste de déchets	1	1	1	1	1	1	1
Poste intendant	2	2	2	2	2	0	0
Poste chauffeur	5	5	5	5	5	3	2
Poste Ucsa	0	1	0	0	1	0	0
Somme	9	10	9	9	10	5	4

TABLE 1.1 – Nombre d’agents nécessaires à chaque poste pour chaque jour de la semaine en 2015.

normes Européennes sur le temps de conduite, le temps de travail et le temps de repos d’un conducteur de véhicules de plus de 3,5T.

1.2.3 La plate-forme logistique relais du CHT

La plate-forme logistique a été inaugurée en 2009 à Saint-Julien-les-Villas (figure 1.3). Elle est composée de trois secteurs :

- la cuisine, qui produit les repas des patients, des résidents et du personnel de la majorité des HCS ;
- le magasin central, qui reçoit, stocke et livre le matériel et les produits alimentaires ;
- la blanchisserie, qui traite le linge (tenues de travail des agents, des chemises des patients, draps, torchons, ...).

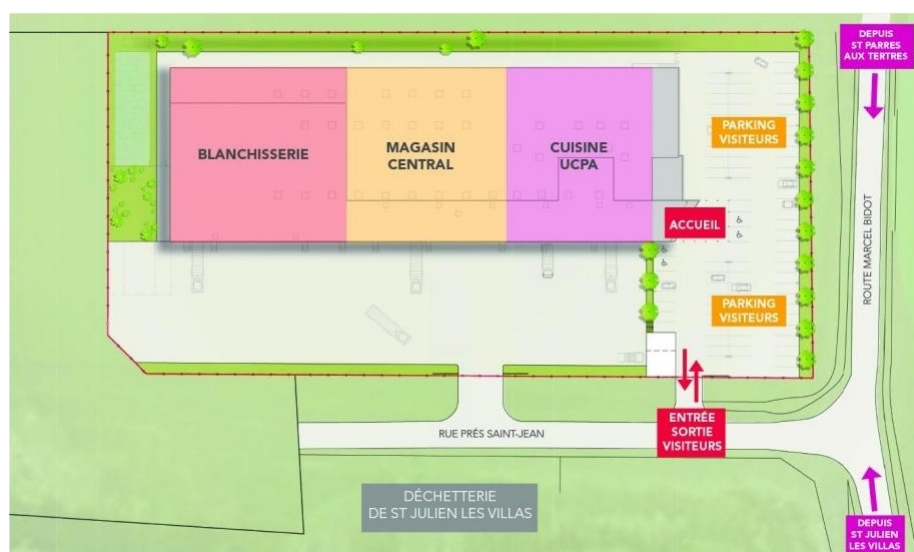


FIGURE 1.3 – Plate-forme logistique du CH de Troyes à Saint Julien Les Villas.

Le magasin central a à son actif, plus de 1 000 références de produits, et conditionne 5 200 mètres cubes de produits par an à destination des services de soins.

La cuisine a un panel de 132 profils alimentaires, avec une capacité de production de 5 000 repas par jour. Les établissements bénéficiant des repas confectionnés par la cuisine sont :

- l'hôpital Simone Veil,
- le centre de rééducation Pasteur,
- le centre psychiatrique de l'Aube,
- le Domaine de Nazareth,
- l'EPSMA de Brienne le Château,
- le Groupement Hospitalier Aube-Marne.

Le service blanchisserie de la plateforme logistique du CHT est composé de 29 professionnels : agents d'entretien qualifiés, ouvriers professionnels, maîtres d'ouvriers, agents de maîtrise et techniciens. L'objectif de ce service est quotidiennement de réceptionner et traiter le linge sale des patients et des résidents, tout en respectant les normes RABC visant à maîtriser les risques liés à la biocontamination. Les agents doivent s'assurer de la propreté bactériologique et visuelle avant de le restituer.

La blanchisserie assure le traitement du linge de plusieurs établissements du département de l'AUBE. Il s'agit :

- de l'hôpital Simone Veil,
- du Centre de rééducation Pasteur,
- du Centre psychiatrique de l'Aube,
- du Domaine de Nazareth,
- du CH de Bar-sur-Aube,
- du CH de Bar-sur-Seine,
- de l'EPSMA de Brienne le Château,
- de la résidence Pierre d'Arcis,
- de la résidence Delatour.

Cette activité du service de blanchisserie se traduit par 30 tonnes de linge lavé par semaine, et par jour :

- 5 000 draps et alèses lavés,
- 5 000 vêtements de travail et chemises d'isollements,
- 1 250 nids d'abeilles,
- 1 500 sacs/éponges.

1.2.4 La modernisation de la logistique au CHT

La politique de modernisation du CHT a permis en 2014, d'inaugurer un nouveau bâtiment d'hospitalisation (voir figure 1.4).



FIGURE 1.4 – Entrée du nouveau bâtiment d'accueil du CHT.

Au-delà de cette possibilité d'accueillir plus de patients, ce projet a permis de répondre au problème de troubles musculo-squelettiques dont souffrent aujourd'hui des agents dans la fonction logistique des hôpitaux. La réponse donnée a été la mise en service, d'un système logistique de 11 robots manutentionnaires (voir figure 1.5).



FIGURE 1.5 – Véhicules à guidage automatique

D'un coût de 70 000 euros l'unité avec le système nécessaire pour sa navigation, cette technologie permet d'assurer un transport automatique par chariots (figure 1.6) des denrées et des

produits (plateaux repas, armoires de linge, armoires produit pharmaceutique, ...) à partir de la plateforme logistique.



FIGURE 1.6 – Exemple de chariot de repas.

Ces véhicules sont guidés de façon automatique avec une technique reposant sur l'utilisation des lasers. Ils réalisent quotidiennement des transports entre la plateforme logistique et les étages des bâtiments en passant par le sous-sol, et en prenant l'ascenseur tout seul.

La mise en place de cette technologie permet :

- de maintenir une bonne fiabilité dans la livraison ponctuelle des marchandises,
- de maintenir un niveau élevé de sécurité des biens et des personnes,
- de réduire les coûts logistiques internes,
- de tracer et suivre automatiquement le matériel.

Au CHT, cette technologie a permis aussi de libérer des tâches de manutentions 18 agents hospitalier, notamment des aides soignants.

1.3 La recherche opérationnelle et le milieu hospitalier

La commission logistique hospitalière de l'ASLOG (Association française pour la logistique) définit la logistique hospitalière ainsi : "Il s'agit de la gestion des flux de patients, de produits, de matières, des services et d'informations qui s'y rapportent, pour assurer la qualité et la sécurité à un niveau défini de performance et d'efficience, depuis le fournisseur jusqu'au patient et, selon le cas, au destinataire final". La logistique hospitalière est donc l'étude et l'optimisation de deux types de flux physiques hospitaliers : les flux de personnes (qui comprennent les

mouvements des patients depuis leur entrée jusqu'à leur sortie de l'établissement, mais aussi ceux du personnel et des visiteurs), et les flux de matières. Ces derniers sont très variés au cœur d'une unité de soins avec les flux entrants : dossier patient, produits sanguins labiles, médicaments et dispositifs médicaux stériles et non stériles, repas, linge propre, consommables généraux, courrier et colis ; et les flux sortants : prélèvements, dispositifs à stériliser, vaisselle et linge sale, déchets, courrier et colis.

Concernant la logistique des flux physiques, Sampieri-Teissier [11] met en évidence cinq phases de développement potentiel de logistique hospitalière :

1. La logistique embryonnaire, comprenant la logistique d'intendance (restauration, blanchisserie, nettoyage, ...) et la logistique de distribution (médicaments, produits à usage unique, ...) est fondée sur le principe de gestion en flux poussés.
2. La logistique intégrée au sein de l'hôpital suppose une gestion en flux tirés, le niveau des stocks étant ajusté en fonction des consommations réelles.
3. Le soutien logistique intégré permet d'aller plus loin dans l'intégration de la logistique aux autres fonctions de l'hôpital, en le prenant en compte dès la construction d'un bâtiment. Ce type de logistique se développe de plus en plus de nos jours avec la construction des nouveaux hôpitaux dans le cadre « Hôpital 2007 et Hôpital 2012 ».
4. L'éco logistique concerne la gestion des produits usagés (recyclage, déchets contaminés, ...), à laquelle il est bon d'adjoindre une réponse architecturale ou organisationnelle.
5. La logistique de type supply chain (chaîne d'approvisionnement) élargit l'intégration aux structures extérieures qui interviennent dans la chaîne logistique, comme les prestataires extérieurs et les structures sanitaires coopératrices.

Quel que soit le pays, le milieu hospitalier se heurte à des défis de dimensionnement et d'organisation afin de satisfaire les besoins des patients. Ces besoins ont un caractère urgent et une nécessité de qualité. Dans ce milieu, on retrouve des systèmes complexes auxquels s'intéressent de plus en plus le domaine de la recherche opérationnelle afin d'apporter des solutions respectant un ensemble de contraintes tout en minimisant les coûts.

Les activités de la logistique hospitalière d'après Chow et al. [4] sont réparties en trois catégories :

- L'approvisionnement : processus d'achat et de gestion des stocks des différents produits.
- La production : processus de transformation tel que le traitement du linge, la fabrication des repas ou la stérilisation des instruments.
- La distribution (ou réapprovisionnement) : transport des produits transformés dans des contenants depuis les unités de production jusqu'aux zones de stockage avant l'utilisation auprès du patient.

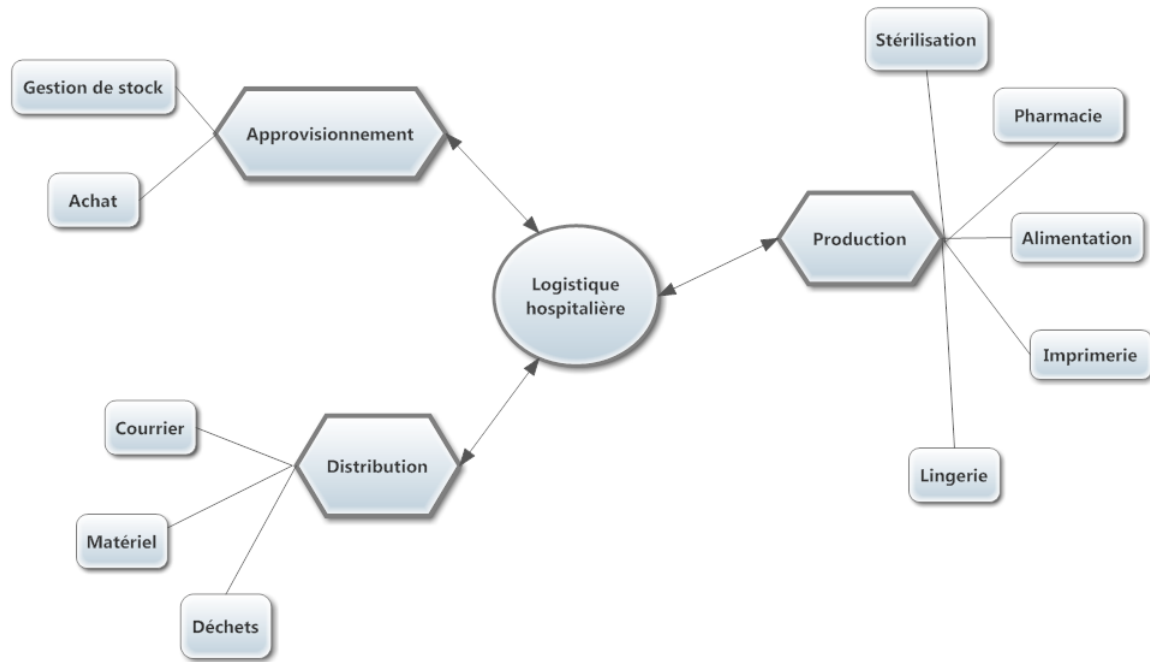


FIGURE 1.7 – Les différentes fonctions de la logistique selon Chow et al. [4]

Aux différentes fonctions de la logistique de la figure 1.7, nous pouvons rajouter au sein des composantes de la distribution : la distribution des flux d'alimentaire, de linge, et de produits pharmaceutiques.

Dans tous les pays, les milieux hospitaliers présentent de nombreux problèmes d'organisation, de planification et de dimensionnement. Ces problèmes peuvent être résolus par des techniques scientifiques. La recherche opérationnelle (RO), fournit les outils informatiques d'aide à la décision qui permettent une bonne gestion des ressources. La recherche opérationnelle propose des méthodes scientifiques pour la prise de bonnes décisions en rationalisant, en simulant, ou encore en optimisant les systèmes de productions ou organisations. C'est une discipline à l'intersection des domaines des mathématiques, de l'informatique et des sciences économiques. Cependant, les applications de la RO, ont essentiellement porté sur le milieu de l'industrie. En France, la recherche opérationnelle n'est impliquée que depuis peu dans le milieu hospitalier, alors que dans d'autres pays, essentiellement anglophones, cette activité existe depuis plus de 40 ans. Par exemple, le groupe O.R.A.H.S. (Operational Research Applied to Health Services) est un groupe européen formé en 1975 qui traite des problèmes rencontrés en milieu hospitalier. D'après Kergosien et al. [12], la difficulté du développement de la recherche opérationnelle en France tient probablement à la particularité du système économique des hôpitaux.

Les applications de la recherche opérationnelle dans le milieu hospitalier sont diverses. Pour un tour d'horizon plus complet dans ce domaine, il existe plusieurs états de l'art généraux. L'un des premiers fût celui de Fries [13]. Ce sujet a ensuite été traité dans les travaux de

Rosenhead [14], Boldy and O’Kane [15], Boldy [16] et de Wiers [17], Shi et al. [18]. Il existe aussi d’autres états de l’art dédiés à des problèmes récurrents dans le milieu hospitalier comme pour l’ordonnancement des blocs opératoires (Cardoen et al. [19]). La planification d’horaires de travail des infirmières (Burke et al. [20]), les critères de mesure de performance d’un système de santé (Li and Benton [21]), ou encore le processus d’analyse hiérarchique pour la prise de décisions dans le milieu médical (Liberatore and Nydick [22]).

En France plusieurs Centres Hospitaliers ont fait la démarche de réorganiser les flux de livraison et la collecte des produits. C’est le cas du CHU de Clermont Ferrand qui, en partenariat avec le laboratoire de recherche LIMOS (Laboratoire d’Informatique en Modélisation et Optimisations des Systèmes), a effectué des travaux de réorganisation de ses flux logistiques, dans le cadre d’une thèse soutenue par André Virginie en 2011 intitulée "Problème de livraison-collecte dans un environnement hospitalier : méthodes d’optimisation, modèle de simulation et couplages". Sa thèse porte sur la proposition de méthodes d’optimisation couplées avec un modèle de simulation dans laquelle elle planifie les horaires des chauffeurs. Un autre exemple d’une illustration de la mise en place des techniques de RO a été réalisé au centre hospitalier du CHRU de Tours par Kergosien [23]. Leurs travaux se sont articulés autour de l’étude de deux principaux problèmes : le transport de flux de produits et le transport de patients.

Le Centre hospitalier de Troyes en partenariat avec le Laboratoire d’Optimisation des Systèmes Industriels s’est proposé de se doter des moyens et outils pour organiser les flux de collectes et de livraisons des produits. Les problématiques rencontrés dans l’organisation

1.4 Problématiques

Dans le cadre de cette thèse, nous nous intéressons à la recherche de solutions efficaces afin de dimensionner l’équipe de chauffeurs et la flotte de camions pour satisfaire un ensemble de requêtes à moindre coût. Les coûts retenus dans notre étude sont :

- le coût induit par le nombre de postes de chauffeurs,
- le coût fixe d’acquisition de la flotte des véhicules,
- le coût de carburant utilisé.

La flotte de véhicules considérée dans notre problème est une flotte hétérogène de véhicules. L’aspect hétérogène est caractérisé par :

- la capacité,
- le type de véhicule pouvant être "frigorifique" ou "non frigorifique",
- le coût fixe d’acquisition du véhicule (coût de location ou coût d’achat),
- la consommation moyenne de carburant au km.

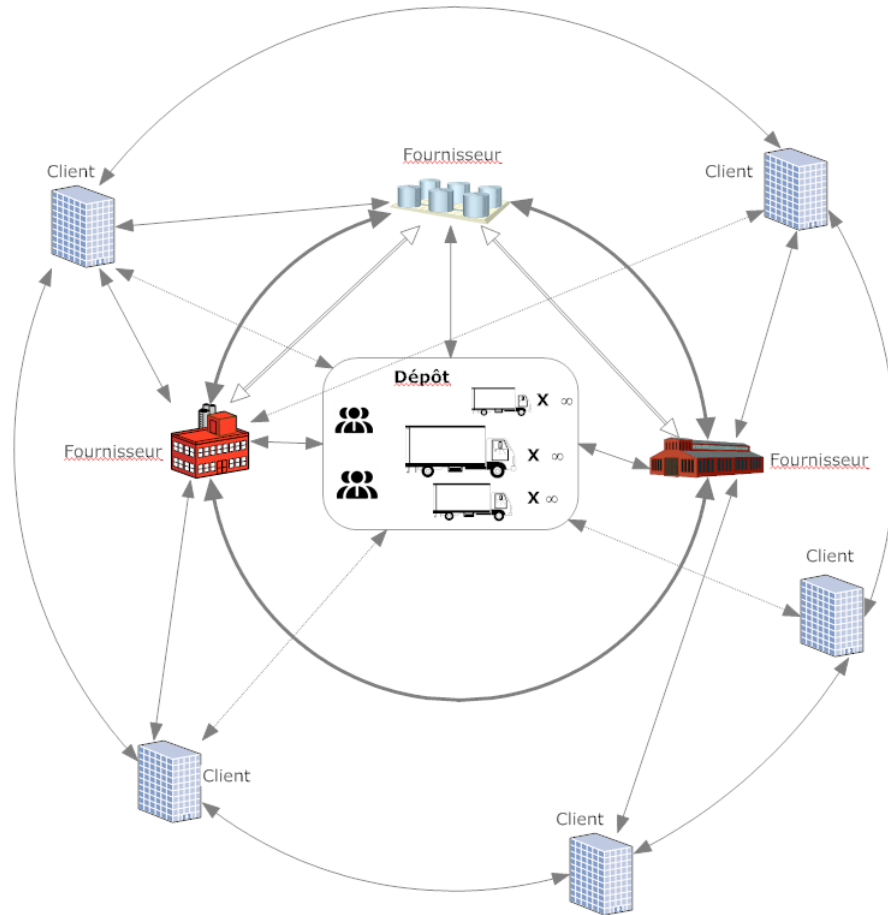


FIGURE 1.8 – Représentation de réseau logistique étudié.

La figure 1.8 décrit le type de réseau logistique à la base de nos travaux. On y retrouve un ensemble de clients dont les besoins sont satisfaits à partir d'un dépôt et d'un ensemble de fournisseurs. Les chauffeurs à partir du dépôt, utilisent une flotte hétérogène de véhicules afin de réaliser leurs plannings journaliers de collectes et livraisons de chariots de produits. Ainsi d'une part, la réglementation sur les temps de travail et de pause est respectée, et d'autre part, les besoins des clients sont tous satisfaits. La présence de différents types de fournisseurs sur la figure 1.8, marque la présence de différents types de produits et de la spécialisation des fournisseurs dans la production d'un ou de plusieurs types de produits.

Dans notre cas d'étude, les clients sont des centres hospitaliers, des maisons de retraite, et des hôpitaux psychiatriques. Ces clients ont besoin, afin d'assurer la qualité des soins des patients :

- des plateaux-repas prêts à temps,
- du linge pour les patients et le personnel soignant disponible en temps et en heure,
- des médicaments et d'autres produits qui participent de près ou de loin à améliorer la

qualité de service.

Le dimensionnement de l'équipe de chauffeurs est soumis d'une part à la réglementation européenne sur les temps de conduite et de pauses des chauffeurs de véhicules de plus de 3,5T, à la réglementation de temps de travail d'un agent de la fonction publique hospitalière en France et aux souhaits de l'établissement (exemple : les postes de travail doivent avoir une durée de 07h30 de temps de travail). Dans le planning d'un chauffeur, il peut arriver qu'il change de véhicule au dépôt, et dans ce cas, un temps de changement de véhicule est considéré.

Les camions au CHT sont obtenus par contrat de location (sur 5 ans), ce qui leur procure de la flexibilité quant à reconsidérer le type de flotte de véhicules en leur possession. À ce niveau, l'établissement cherche à apporter des réponses à plusieurs niveaux de décisions voir figure 0.2.

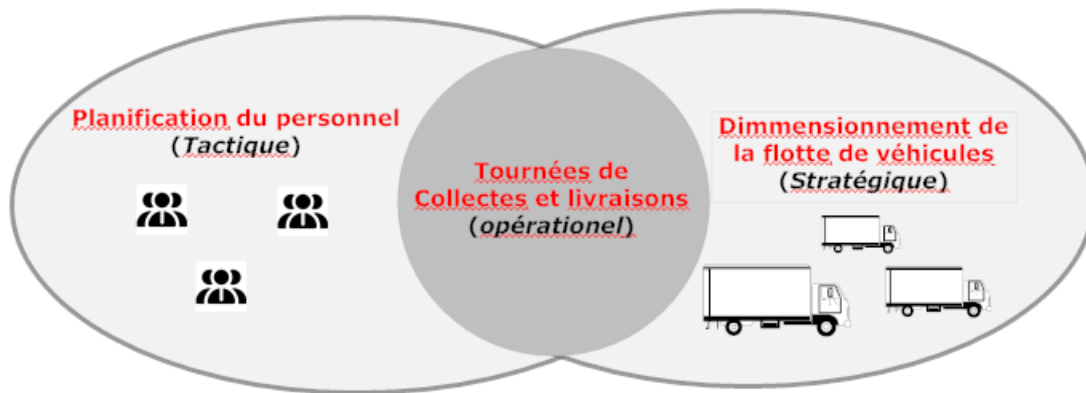


FIGURE 1.9 – Problématique selon le niveau de décision.

Dans le problème étudié, tous les besoins doivent être satisfaits, autrement dit, le taux de service est de 100%. Formellement, un besoin (ou **requête**) est défini comme une demande de collecte d'un nombre de chariots d'un type de produits sur un site et d'une demande de livraison de ce qui a été collecté à un autre site.

Chaque demande est caractérisée par :

- un type de produit,
- un nombre de chariot (associé à un type de produit),
- une fenêtre de temps dans laquelle le service doit débiter,
- une durée nécessaire pour effectuer le service (chargement, déchargement, nettoyage du véhicule).

Notre cas d'étude présente aussi des contraintes d'incompatibilité :

- incompatibilité type produit/type produit (certains produits ne peuvent être transportés ensemble),

- incompatibilité type produit/type camion (certains produits doivent être transportés dans des véhicules spécifiques),
- incompatibilité type de camion/site demande (certaines demandes sont sur des sites avec des accès spécifiques).

1.5 Conclusion

Dans ce chapitre, nous avons présenté l'environnement industriel dans lequel se sont réalisés nos travaux. Nous avons abordé les différents enjeux du milieu hospitalier en France, et les travaux qui déterminent la proportion des coûts liés à la logistique en milieu hospitalier. Nous avons présenté les différents établissements de santé dont la réorganisation des flux a été au cœur des travaux que nous avons mené. Dans la dernière partie de ce chapitre, nous avons présenté plus succinctement la problématique qui nous a été soumise, et dont les solutions apportées sont présentées dans les chapitres suivants. Mais avant tout, le prochain chapitre présente un état de l'art des travaux réalisés sur les problématiques voisines à celles que nous étudions.

Chapitre 2

État de l'art

2.1 Introduction

L'état de l'art est une partie centrale dans tout projet ou étude scientifique. En effet, c'est cette phase qui permet d'une part de faire un point sur les différents travaux qui ont été faits sur et/ou autour du sujet à étudier. D'autre part, elle permet de positionner le sujet étudié au sein des différentes études et de ce fait de donner une meilleure visibilité pour de futures recherches liées à la problématique traitée.

Dans la suite de ce chapitre, dans un premier temps, nous présenterons une classification et les travaux sur les problèmes de collectes et livraisons avec fenêtre de temps et flotte homogène de véhicules (PDPTW). Ensuite, par rapport à cette problématique de PDPTW, nous traitons un aspect supplémentaire sur le type de flotte de véhicules. Nous considérons à cet effet un parc hétérogène de véhicules. Et enfin, nous rajoutons d'autres aspects tels que : l'affectation et le dimensionnement de l'équipe des chauffeurs. Dans ce chapitre, nous présentons aussi des problèmes conjoints de logistique hospitalière externe et de planification du personnel.

2.2 PDPTW Flotte homogène

Dans cette section, nous présentons une classification des problèmes de collectes et livraisons, et les travaux menés sur ce sujet.

En 2007, Berbeglia et al. [24] ont proposées une classification des problèmes de collectes et de livraisons. Cette classification proposée est composée de trois champs : $|Structure|Visites|Véhicules|$. La signification de chaque champ est :

- *Structure* : précise le nombre de destinations ou d'origines des biens à transporter. Ses attributs sont :
 - "*Many-to-Many*" (M-M) : chaque type de biens a plusieurs origines et destinations.

- "*One-to-Many-to-One*" (1-M-1) : le flux de livraison va du dépôt vers les clients, et le flux de collecte va des clients vers le dépôt.
 - "*One-to-One*" (1-1) : chaque bien a exactement un sommet de collecte et un sommet de livraison.
- *Visites* : fournit des informations sur la façon dont les opérations de collecte et de livraison sont effectuées aux sites de chaque bien. On distingue les cas :
- *PD* : pour indiquer que chaque sommet du client est visité exactement une fois pour une opération collecte et de livraison,
 - *P-D* : si ces les deux opérations de collecte et de livraison peuvent être effectuées ensemble ou séparément.
 - *P/D* : pour le cas où chaque sommet du client possède soit une demande de collecte, soit une demande de livraison mais pas les deux, de sorte qu'une opération de collecte ou de livraison est effectuée à chaque sommet.
- En plus, la lettre *T* est utilisée pour exprimer que certains sommets peuvent être utilisés comme points de transfert.
- *Véhicules* : donne le nombre de véhicules utilisés dans la solution.

Chaque fois qu'un champ n'est pas défini, la notation "-" est utilisé.

De toute évidence, cette classification ne rend pas pleinement compte de toutes les complexités des problèmes de collectes et livraisons, telles que la qualité hétérogène ou pas des véhicules, les objectifs alternatifs, le nombre de type de produits, ... Mais en utilisant cette classification, on peut déjà avoir une bonne idée de la structure implicite du type problème de collectes et livraisons étudiées.

Utilisant cette classification on peut avoir une visibilité sur la direction prise par les travaux sur les problèmes de collectes et livraisons. Quelques exemples de travaux sont donnés par le tableau 2.1.

En 2007, Gribkovskaia et al. [31] ont étudié les problèmes de collectes et livraisons du type *1-M-1/P-D/-*, et ont identifié pour ce type de problème quatre types de solutions : *générale*, *hamiltonienne*, *lasso* et *double chemin*. Une illustration de ces différents types de solution est donné par la figure 2.1.

Le problème de collecte et livraison avec fenêtres de temps est un problème NP-difficile car il peut être vu comme un problème de voyageur de commerce pour certains jeux de données.

Durant ces 30 dernières années, plusieurs méthodes de résolution exactes du PDPTW ont été proposé. Mais le nombre de travaux traitant de cette problématique reste moindre par rapport au problème classique de tournées de véhicules avec fenêtres de temps ou avec contrainte de capacité. La vérification de l'efficacité des méthodes de résolution proposées a été réalisée sur

Structure	Classe	Travaux
M-M	M-M P-D 1	[25]
	M-M P/D 1	[26–30]
1-M-1	1-M-1 P-D -	[31–34, 34]
	1-M-1 P/D 1	[35–41]
	1-M-1 PD m	[42–49]
	1-M-1 P/D m	[43, 50–65]
1-1	1-1 P/D 1	[66, 67]
	1-1 P/D -	[68–70]
	1-1 P/D 1	[71, 71–80]
	1-1 P/D m	[62, 81–92]
	1-1 P/D-T -	[93, 94]

TABLE 2.1 – Classification de quelques travaux.

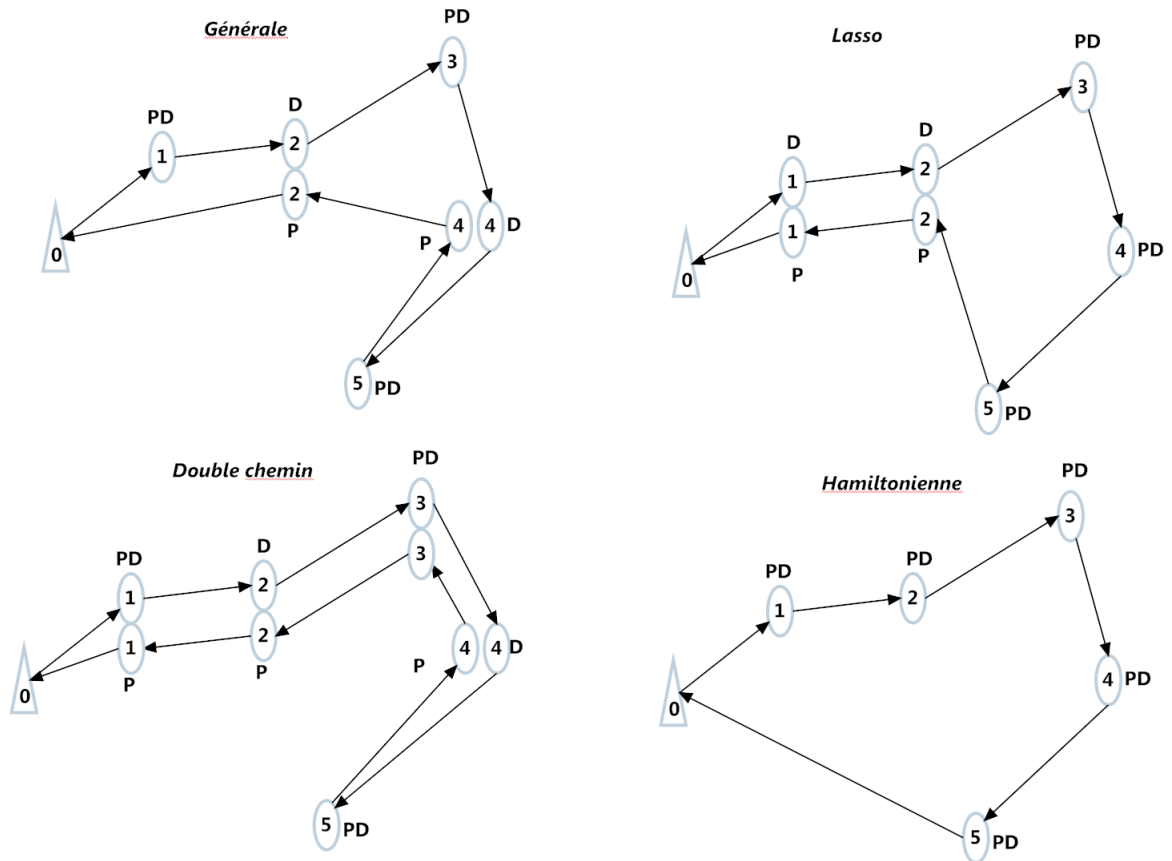


FIGURE 2.1 – Exemple illustrant quatre formes de solution pour les problèmes de type 1-M-1/P-D/-. (P = collecte, D = livraison, PD = collecte et livraison combinés).

différents benchmarks, ce qui rend la comparaison difficile entre les travaux de recherche sur ce sujet.

2.2.1 Méthodes exactes

La première méthode de résolution à l'exacte du problème de PDPTW a été proposée en 1986 par Desrosiers et al. [95]. Dans ces travaux, ils ont proposé un algorithme pour résoudre le problème de PDPTW avec un véhicule (1-vehicle PDPTW) à l'optimalité. L'algorithme proposé est un programme dynamique. Plus spécifiquement c'est un algorithme à labels intégrant des règles de dominance pour éliminer les labels qui ne sont pas intéressants, et de ce fait, diminuer le temps de résolution du problème. Cet algorithme a pu résoudre des instances à 80 demandes (40 requêtes).

Dans les premières années du 20^{ième} siècle, des méthodes de résolution se basant sur les algorithmes de générations de colonnes ont été proposé pour résoudre le problème de PDPTW. Dans ces méthodes, la recherche de(s) colonne(s) de coût réduit négatif a été résolue soit par des méthodes exactes, soit par des heuristiques. Les premiers travaux proposant des méthodes exactes de résolution du problème de PDPTW utilisant la méthode de génération de colonnes ont été proposés par Dumas et al. [81] en 1991. Pour ce faire, ils ont proposé une méthode de séparation et évaluation (Branch-and-bound) afin de résoudre à l'optimalité le problème.

En 2004, afin de résoudre le problème de PDPTW, Lu and Dessouky [96] ont proposé un algorithme qui utilise la méthode de séparation et évaluation pour parcourir efficacement l'espace des solutions, et la méthode des plans sécants pour diminuer le temps de résolution (algorithme de Branch-and-cut). Dans leurs travaux, ils ont proposé une modélisation du problème de PDPTW sous forme de programme linéaire en nombres entiers utilisant des variables à deux index. Le modèle proposé a un ensemble de variables et de contraintes de taille polynomiale. Ils ont renforcé le modèle avec des inégalités valides pour améliorer la qualité de la relaxation linéaire du modèle, et ont pu alors résoudre à l'optimalité, des instances ayant 30 demandes (15 requêtes). En 2006, Ropke et al. [84] ont proposé un algorithme de Branch-and-cut pour résoudre le problème de PDPTW. Et en 2009, dans leurs recherches Ropke and Cordeau [97], ils ont proposé un algorithme de Branch-and-cut-price.

2.2.2 Méthodes approchées

Nanry and Barnes [85] ont été les premiers à proposer une méta heuristique pour résoudre le problème de PDPTW en l'an 2000. Pour ce faire, ils ont proposé une méthode de recherche Tabou combinant plusieurs techniques de recherche locales. Pour tester l'efficacité de leur méthode de résolution, ils ont proposé un benchmark, en modifiant le benchmark proposé par Solomon [99] pour réaliser les tests sur le problème de tournées de véhicules avec fenêtres de temps (VRPTW). Chaque instance de ce benchmark est constituée de 100 demandes. Lau and Liang [86] ont aussi proposé une méthode de recherche Tabou pour résoudre le problème de PDPTW. Ils ont proposé un ensemble d'heuristiques pour construire une solution. Ils ont appliqué une technique d'optimisation appelée "Squeaky wheel optimization", combiné

Articles		Desrosiers et al. [95]	Dumas et al. [81]	Sigurd et al. [98]	Lu and Dessouky [96]	Ropke et al. [84]	Ropke and Cordeau [97]	Nos travaux(chapitre 3)
Type de flotte	1 véhicule Plusieurs véhicules	✓	✓	✓	✓	✓	✓	✓
Contraintes	Capacité Fenêtre de temps Précédence	✓ ✓ ✓	✓ ✓ ✓	✓ ✓ ✓	✓ ✓ ✓	✓ ✓ ✓	✓ ✓ ✓	✓ ✓ ✓
Ind-Per	Coût fixe de la flotte Distance	✓	✓	✓	✓	✓	✓	✓
Meth-Res	Branch-and-bound Branch-and-cut Branch-and-cut-and-price Décomposition Benders Programme dynamique	 ✓	✓ 	✓ 	✓ 	 ✓ 	 ✓	 ✓
Benchmark	[3] Nouveau (nb requêtes)	✓ ✓	 ✓ (19)	 ✓ (205)	 ✓ (25)	✓ ✓	✓ ✓	✓

Ind-Per := indicateur de performance ; Meth-Res := méthode de résolution.

TABLE 2.2 – Méthodes exactes de résolution du PDPTW.

avec des techniques de recherche locale. En 2003, A. Li *et* H. Lim [100] ont proposé une métaheuristique hybride pour résoudre le PDPTW. L'algorithme proposé est composé de deux métaheuristicues : un algorithme de Recherche Tabou, et un algorithme de Recuit simulé. Ils ont mené des expérimentations sur les 9 plus grands instances de Nanry and Barnes [85] et ont considéré 56 nouvelles instances obtenues à partir du benchmark de Solomon [99].

Créput et al. [101] ont proposé un algorithme évolutionnaire pour résoudre le PDPTW. En optimisation, les algorithmes évolutionnaires (ou évolutionnistes) sont des algorithmes qui font évoluer un ensemble (une "population") de solutions ("individus"). L'évolution se traduit par le fait que les individus les plus robustes ont plus de chances de transmettre une partie de leur structure au sein de la population. Dans ce type d'algorithme, la robustesse est mesurée grâce à une valeur ("fitness") attribuée à chaque individu. Les résultats des expérimentations menées

sur le benchmark de Li and Lim [3] à 100 demandes/instance, montrent que l'algorithme proposé par Créput et al. [101] est moins performant que l'algorithme de Recherche Tabou proposé par Li and Lim [3].

Un algorithme génétique a été proposé par Pankratz [102] pour résoudre le PDPTW. Rappelons que l'algorithme génétique fait partie de la famille des algorithmes évolutionnistes. Son principe est de débiter avec une population de base qui se compose le plus souvent de chaînes de caractères correspondants chacune à un chromosome. Une note est attribuée à chaque chromosome, et ensuite, itérativement, on effectue des opérations de sélection, croisements et mutations au sein de cette population pour avoir une nouvelle population. Dans les travaux de Pankratz [102], un chromosome correspond à une partition de l'ensemble des requêtes (ordonner ensuite pour former une route réalisable). Les tests menés sur les instances de Li and Lim [3] et de Nanry and Barnes [85] ont montré que cette approche est plus performante qu'une autre approche basée sur l'algorithme génétique (proposée par Créput et al. [101]).

En 2006, Bent and Van Hentenryck [87] appliquèrent une heuristique à deux niveaux pour résoudre le PDPTW et obtinrent de nouvelles meilleures solutions sur le benchmark proposé par Li and Lim [3]. Au premier niveau de leur algorithme, un algorithme de recuit simulé est utilisé pour décroître le nombre de véhicules utilisés dans la solution courante. Au second niveau, la distance totale parcourue est minimisée en utilisant un algorithme de recherche locale à grand voisinage (LNS). En optimisation, une méthode de recherche locale à grand voisinage est un algorithme de recherche locale dont la définition de voisinage est potentiellement de taille exponentielle. Afin de montrer l'efficacité de cette méthode, ils ont réalisé les tests sur le benchmark proposé par Li and Lim [3]. Cette méthode a montré son efficacité en matière de temps de résolution et de qualité des solutions par rapport aux algorithmes proposées avant cette dernière. Ropke and Pisinger [103] ont proposé un algorithme LNS pour résoudre le PDPTW. Dans leur cas d'étude, le réseau logistique est composé de plusieurs dépôts de véhicules. Ils ont réussi à améliorer des solutions dans le benchmark de Li and Lim [3].

En 2010, Nagata and Kobayashi [104] ont été les premiers à appliquer un algorithme de recherche à grand voisinage (couplé à des heuristiques de recherche locale) appelé "Guided ejection search" (GES) sur le problème de tournées de véhicules avec fenêtre de temps. Les expérimentations de cet algorithme ont été réalisées sur le benchmark de Li and Lim [3], et ont permis d'améliorer 105 solutions sur les 298 instances que compte le benchmark. En 2015, Blocho and Nalepa [105] ont proposé une parallélisation de l'algorithme GES permettant d'améliorer la qualité des résultats et d'accélérer la convergence des calculs. L'étude expérimentale de cet algorithme a permis d'obtenir 13 (22%) nouvelles solutions du benchmark de Li and Lim [3]. Nalepa and Blocho [106] ont proposé dans leurs travaux en 2016 sur le PDPTW, un algorithme de GES pour le résoudre.

L'objectif fixé dans leurs travaux a été de minimiser le nombre de véhicules utilisés. Les

améliorations proposées réduisent le temps de convergence du GES et améliorent la qualité des résultats obtenus sur le benchmark de Li and Lim [3]. Nalepa and Blocho [107] ont introduit en 2017 un algorithme mémétique pour résoudre le PDPTW. Dans cet algorithme, les processus parallèles coopèrent périodiquement et échangent les meilleures solutions trouvées afin d'orienter la recherche de manière efficiente. L'étude expérimentale réalisée pour étudier l'efficacité de cet algorithme sur le benchmark de Li and Lim [3] a permis d'obtenir des solutions de qualité, et de trouver solutions de meilleure qualité que celles obtenues dans les autres travaux dont l'objectif est de minimiser la distance totale.

Une synthèse des différentes heuristiques et métaheuristiques proposées pour résoudre le PDPTW est donnée dans la table 2.3.

2.3 PDPTW Flotte hétérogène

Qu and Bard [109] ont introduit en 2013 le PDPTW avec une flotte hétérogène de véhicules de taille fixe, la capacité des véhicules étant configurable. Ils ont proposé une modélisation en programme linéaire en nombres mixtes de ce problème, et ont développé une heuristique à deux phases pour le résoudre. Dans la première phase de cet algorithme, des procédures aléatoires sont utilisées pour obtenir un ensemble de solutions réalisables "intéressantes". Et dans la seconde phase, un algorithme de recherche locale à grands voisinages est utilisé pour améliorer chacune des solutions obtenues à la première phase. Cet algorithme a permis de résoudre des instances réelles à 100 demandes et 4 types de véhicules. Ceci a permis d'obtenir des solutions avec une différence de coût par rapport à celle qui était mise en place de l'ordre de 20 à 40%. En 2014, ces mêmes auteurs ont introduit dans Qu and Bard [110], le PDPTW avec une flotte hétérogène de véhicules de taille infinie, la capacité des véhicules étant configurable. Ils ont proposé une modélisation de ce problème en programme linéaire en nombres mixtes, et un algorithme de Branch-and-price-and-cut pour le résoudre. Dans le processus de génération de colonnes, le problème de pricing est résolu avec un algorithme à labels. En effet, le problème de pricing dans leur modélisation est un problème de plus court chemin. Ils ont proposé des inégalités valides et des critères de dominance pour diminuer le temps de résolution. Afin de tester l'efficacité de leur algorithme, ils ont généré un nouveau benchmark dont les instances sont composées de 50 demandes. Ils ont pu obtenir une solution optimale dans la majorité des cas.

La même année, Bettinelli et al. [111] ont introduit le PDPTW avec une flotte hétérogène de véhicules de taille fixe dans laquelle les fenêtres de temps peuvent être violées, mais avec un coût. Ce travail a été motivé par des problématiques rencontrées par de petites entreprises de transports opérant dans la zone urbaine de Milan. Pour résoudre ce problème, ils ont proposé un algorithme de Branch-and-price basé sur des techniques de programmation dynamique. Ils ont pu ainsi résoudre à l'optimalité des instances ayant 75 demandes. Après analyse des résul-

		Articles											
		Nanry and Barnes [85]	Lau and Liang [86]	Lim et al. [108]	Li and Lim [3]	Créput et al. [101]	Pankratz [102]	Bent and Van Hentenryck [87]	Ropke and Pisinger [103]	Nagata and Kobayashi [104]	Blocho and Nalepa [105]	Nalepa and Blocho [106]	Nalepa and Blocho [107]
		Nos travaux(chapitre 3)											
Contraintes	Capacité	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Fenêtre de temps	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Précédence	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ind-Per	Distance	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
	Durée des tournées			✓	✓				✓				✓
	Coût fixe de la flotte			✓	✓	✓	✓	✓				✓	✓
	RI								✓				
	Temps d'attente												✓
	Violation capacité	✓											✓
	Violation fenêtre	✓			✓								✓
Meth-Res	Algorithme génétique					✓	✓						
	Algorithme mémétique												
	Colonie de fourmis												
	Guided ejection search									✓	✓	✓	
	Recherche locale							✓	✓				
	Recherche tabou	✓	✓		✓								
	Recuit simulé				✓			✓					
	SWO			✓									✓
Benchmark	[3]			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	[85]	✓	✓		✓	✓	✓		✓				
	Nouveau							✓	✓	✓	✓		

Ind-Per :=indicateur de performance; Meth-Res :=méthode de résolution; RI :=nombre de requêtes insatisfaites; SWO :=Squeaky wheel optimization;

TABLE 2.3 – Heuristiques et méta-heuristiques pour le PDPTW.

tats, les auteurs ont conclu que la possibilité de violer les fenêtres de temps (avec pénalité), augmente le coût total des tournées de 5 à 15%.

La table 2.4 fait une synthèse des différents travaux sur les problèmes de collectes et livraisons avec fenêtre de temps et flotte hétérogène de véhicules. Nous pouvons remarquer qu'aucun de

ces travaux ne traite du cas du dimensionnement de la flotte des véhicules.

Articles		Qu and Bard [109]	Qu and Bard [110]	Bettinelli et al. [111]	Nos travaux (chapitre 4)	Nos travaux(chapitre 4)	Nos travaux(chapitre 4)
Configuration	Flotte hétérogène de taille finie	✓	✓	✓	✓		
	Flotte hétérogène de taille infinie					✓	✓
	Plusieurs fenêtre de temps		✓				
	VCM	✓	✓				
Contraintes	Capacité	✓	✓	✓	✓	✓	✓
	Fenêtre de temps	✓	✓	✓	✓	✓	✓
	Précédence	✓	✓	✓	✓	✓	✓
Ind-Per	Coût fixe de la flotte	✓	✓		✓	✓	✓
	Distance	✓	✓	✓	✓	✓	✓
	Durée des tournées		✓				
	Violation fenêtre de temps			✓			
Meth-Res	Branch-and-price-and-cut		✓	✓			
	Colonie de fourmis						✓
	Décomposition de Benders				✓		
	Génération de colonnes					✓	
	Heuristique à deux phases	✓					
Benchmark	Instances réelles	✓		✓	✓		
	Nouveau		✓		✓	✓	✓

VCM := Véhicule de capacité modifiable.

TABLE 2.4 – Synthèse sur le PDPTW avec une flotte hétérogène de véhicules.

En 2000, Irnich [112] ont étudié un nouveau problème de collectes et livraisons avec fenêtres de temps en intégrant plusieurs dépôts. Le cas d'étude est un cas spécifique de problème de collecte et livraison. En effet, ils ont considérés dans leurs travaux l'existence d'une plateforme logistique telle que toute requête (couple collecte-livraison) a soit sa collecte sur la plateforme, soit la livraison sur cette dernière. Ils ont aussi travaillé en faisant l'hypothèse que tout site de collecte et/ou de livraison est un dépôt pour les véhicules. Ils ont proposés une heuristique pour résoudre ce problème, et un nouveau benchmark pour expérimenter leur méthode de

résolution.

De nombreux problèmes rencontrés dans la vie réelle traitant des problèmes de tournées de véhicules, sont liés à la planification des chauffeurs. En effet, les tournées sont réalisées par ces derniers. De plus, pour de nombreuses entreprises, le coût de la main-d'œuvre est la principale composante des coûts. Et réduire ce coût par la mise en place d'une nouvelle organisation du personnel pourrait donc s'avérer très bénéfique.

Dans la suite, nous aborderons le problème conjoint de tournées de véhicules et de planification du personnel. Nous établirons une revue de la littérature (non exhaustive) des travaux sur le problème de planification du personnel. Ensuite, nous présenterons des travaux sur le problème conjoint de tournées de véhicules et de planification du personnel. Et enfin nous porterons un accent sur les travaux réalisés dans le secteur hospitalier.

2.4 Problème conjoint de tournées de véhicules couplé à la planification du personnel

Le problème de planification du personnel ou problème d'affectation du personnel peut-être défini comme le problème de conception des horaires de travail (ou planning), afin de réaliser un nombre d'activités donné. On rencontre ce type de problème dans les compagnies de téléphone, les compagnies aériennes, les hôpitaux, les services de police et les banques.

Au cours des dernières décennies, les problèmes de planification du personnel ont fait l'objet de nombreuses études. L'augmentation de l'attention accordée à la recherche pourrait être motivée par des considérations économiques.

Les premiers travaux sur cette problématique ont été réalisés par en 1954 par Edie [113]. Dans la littérature, des méthodes exactes et approchées ont été proposées pour résoudre ce problème. Le modèle de base souvent utilisé pour modéliser le problème sous forme de programme linéaire en nombres entiers est inspiré du modèle linéaire proposé par Dantzig [114]. Une comparaison des différentes modélisations a été proposé par Aykin [115]. Mais ces modèles ne prennent pas en considération la détermination des heures de pause. Dans les années 90, les travaux de Bechtold and Jacobs [116], Thompson [117] ont permis de palier ce manquement.

Dans l'industrie, la problématique de planification de personnel se pose à tous les niveaux de production. Un exemple est donné dans les travaux de Battaïa et al. [118] sur la planification des travailleurs sur les lignes d'assemblages.

Bechtold and Jacobs [116] ont travaillé sur le problème de planification du personnel sous les hypothèses suivantes :

- Pour chaque activité, le nombre d'agents affecté doit être supérieur ou égal au nombre d'agent nécessaire pour réaliser cette activité.

- Le planning est réalisé sur la journée (24 heures) ;
- Une pause est associée à chaque planning ;
- Les plannings ont tous la même durée de travail ;
- Dans chaque planning, la pause est réalisée dans un intervalle de temps donné.

En 1995, Thompson [117] ont travaillé sur une extension du problème d'écrit par Bechtold and Jacobs [116], en supposant que tous les plannings n'avaient pas la même durée de travail. Et en 2000, Aykin [115] ont proposé d'étendre le problème d'écrit par Bechtold and Jacobs [116], en supposant qu'un planning pourrait contenir plusieurs pauses.

Gärtner et al. [119], Di Gaspero et al. [120] vont encore plus loin en intégrant dans la construction d'un planning, les contraintes sur :

- la durée minimale de travail avant et après la pause dans la construction d'un planning,
- la durée minimale et maximale d'une pause.

Gärtner et al. [119] ont proposé une heuristique gloutonne pour générer une pause dans un planning journalier. Cette heuristique se déroule en trois phases :

1. Trouver la plage horaire légale de prise de pause ;
2. Trouver la meilleure date de début de la pause (en fonction du coût) ;
3. Déterminer une durée réalisable de la pause.

De nombreux types de problèmes de planification du personnel ont été abordés dans la littérature (voir les travaux de : Solomon [99], Baker [122], Miller et al. [123], Golembiewski and Proehl Jr [124], Cheang et al. [125], Alfares [126]). Une revue de la littérature des travaux menés en fonction des secteurs d'activité sur ce sujet a été proposée par Ernst et al. [127].

Dans la section suivante, nous nous intéressons aux travaux traitant du problème conjoint de planification de personnel et de tournées de véhicules.

2.4.1 Problèmes conjoint de collecte et livraison avec fenêtre de temps et de planification du personnel

En 1997, Brandao and Mercer [128] se sont intéressés au problème conjoint de tournées de véhicules avec fenêtre de temps et planification du personnel. Dans ce problème, chaque jour, un véhicule peut effectuer plus d'une tournée, les demandes ont des délais de livraison, les véhicules ont des capacités différentes en matière de volume et de poids ; et les plannings des chauffeurs doivent respecter des contraintes sur le temps de conduite, les temps de début et de fin de travail, et les temps de début et de fin de pause. Ils ont aussi pris en compte le temps de déchargement des chariots, et ont proposé un algorithme à trois phases :

1. affectation des demandes aux tournées en respectant les contraintes de fenêtre de temps et de capacité de véhicule,

Caractéristiques		Articles	Dantzig [114]	Bechtold and Jacobs [116]	Thompson [117]	Aykin [115]	Musliu et al. [121]	Di Gaspero et al. [120]	Gärtner et al. [119]
Hypothèses	Plannings cycliques						✓		
	Horizon de planification		1j	1j	1j	1j	7j	1j	7j
	Duré plannings imposés et égaux		✓	✓			✓		
	Duré plannings imposés et différents				✓	✓			
	Durées plannings à déterminer							✓	✓
	Dates de début des plannings imposées						✓		
	Nombre de pauses		1	1	1	3	0	>10	>10
	Date de début de pause imposée		✓						
	Date de début de pause à déterminer			✓	✓	✓		✓	✓
	Durée des pauses fixées		✓			✓			
	Durée des pauses à déterminer			✓	✓			✓	✓
Critères	Nombre d'employés affectés		✓						
	Coût d'affectation des employés					✓			
	Nombre de plannings distincts						✓	✓	✓
	Somme du nombre d'employés en surplus par rapport à la demande						✓	✓	✓
	Somme du nombre d'employés manquants par rapport à la demande						✓	✓	✓
	Écart avec le nombre moyen de permanence par employé et par semaine						✓		✓

TABLE 2.5 – Quelques travaux sur la planification du personnel (André [1]).

2. affectation des tournées aux chauffeurs en respectant les contraintes sur le planning,
3. réduction du coût de la solution obtenue.

Sur des instances réelles (de 45 à 70 demandes par instance), l'algorithme proposé par Brandao and Mercer [128] a permis de réduire des durées de travail. Les distances parcourues et le nombre de véhicules ont diminué de 20% en moyenne.

Eliyi et al. [129] ont proposé six heuristiques pour résoudre ce problème, et ont pu trouver de bonnes solutions pour des instances à 100 demandes.

Xu et al. [83] ont introduit un problème de tournée de véhicule avec planification du personnel composé :

- d'un problème de collecte et livraison (de type 1-M-1|PD|m) avec fenêtres de temps,
- d'une flotte hétérogène de véhicules,
- des contraintes de compatibilité entre type de véhicule et site (collecte/livraison), et entre type de produit et type de véhicule,
- des contraintes sur les règles de travail du conducteur prescrites par le ministère des Transports qui précisent les heures de travail légales d'un chauffeur.

Dans leurs travaux Xu et al. [83] n'ont pas considéré le dimensionnement de l'équipe des chauffeurs et aussi celui de la flotte de véhicules. De plus dans le problème qu'ils ont étudié, un chauffeur réalise une seule tournée dans son planning et avec un véhicule. Donc dans toute solution, il y a une bijection entre le nombre de chauffeurs et le nombre de tournées, et une bijection entre le nombre de chauffeurs et le nombre de véhicules utilisés. Dans le problème que nous étudions, ces aspects sont pris en compte.

Prenant en compte le dimensionnement de l'équipe des chauffeurs, Kergosien [23] ont mené des travaux sur la livraison des produits (conditionnés dans des contenants) depuis le quai de déchargement jusqu'aux unités de soins et pour collecter les contenants vides depuis les unités de soins jusqu'aux quais de chargement.

Parallèlement, André [1] ont étendu le problème traité par Brandao and Mercer [128] en prenant en compte des aspects supplémentaires tels que :

- Les ressources utilisées (les contenants, les quais de chargement ou de déchargement) ;
- Les contraintes de compatibilité entre les produits et les véhicules ;
- Les contraintes liées aux lignes de production et aux aires de nettoyage ;
- Les contraintes de précédence entre les demandes ;
- Les dates de début au plus tôt des demandes.

La solution proposée par André [1] est un algorithme itératif, chaque itération étant composé de trois phases :

- Évaluation par simulation du nombre maximal de chauffeurs nécessaires pour réaliser l'ensemble des activités de transport du jour comportant le plus grand nombre d'activités ;
- Construction par un modèle linéaire du planning journalier de chaque chauffeur ;
- Affectation et ordonnancement des activités en utilisant le couplage métaheuristique - modèle de simulation.

Par rapport au problème traité par André [1], nous ne prenons pas en compte les contraintes liées aux lignes de production et aux aires de nettoyage, mais nous considérons en plus :

- Un parc hétérogène de véhicules (infini mais dénombrable), ce qui induit :

- le dimensionnement de la flotte de véhicules,
- la prise en compte de la consommation de carburant de chaque type de véhicule,
- la possibilité pour un agent de changer de véhicule au cours de son planning.
- L'interdiction de la violation des fenêtres de temps.
- L'interdiction de la réalisation d'heures supplémentaires par un agent.

Une autre différence notable, est que dans l'étude de André [1], le problème de collecte et livraison sous-jacent est un problème de tournée de véhicule avec charge complète.

Dans la prochaine section, nous aborderons les problèmes conjoignent de tournées de véhicules et de planification du personnel dans le secteur hospitalier.

2.4.2 Problèmes conjoint de logistique hospitalière externe et de planification du personnel

De nombreux travaux se sont intéressés aux problèmes conjoints de tournées de véhicules et de planification du personnel dans le secteur hospitalier. Dans cette section nous faisons un état de l'art de ces différents travaux et positionnons notre problème par rapport à ces travaux.

En 2006, Bertels and Fahle [130] ont travaillé sur le problème de planification des tournées d'infirmières pour effectuer des soins aux domiciles des patients. Dans leurs travaux, les tournées des infirmières débutent et se terminent soit à leur domicile, soit dans un centre de santé. Les préférences des patients quant à l'heure de la visite sont respectées autant que possible. La planification des infirmières est soumise aux contraintes sur le nombre d'infirmières, la durée du temps de travail, la durée du temps de pause et la durée du temps de repos. Ils ont aussi considéré les cas où les soins réalisés par les infirmières ont des durées différentes, et le cas où des soins peuvent nécessiter plusieurs infirmières. Les soins de santé ont été abordés (voir Bertels and Fahle [130, 130], Begur et al. [131], Cheng and Rich [132, 132], Akjiratikarl et al. [133], Eveborn et al. [134]) principalement avec des méthodes hybrides combinant la programmation linéaire en nombres mixtes et une heuristique (voire une méta-heuristique). L'une des deux étant dédié à la réalisation des tournées et l'autre à la réalisation des plannings en utilisant les tournées générées. Des méthodes exactes (voir Barnhart et al. [135], Bredstrom and Rönnqvist [136], Kergosien et al. [137]) ont également été utilisées, en particulier l'algorithme de Branch-and-price. Le problème maître dans ce cas-là est très souvent une formulation en problème de partitionnement, et le problème de pricing est un problème de recherche de plus court chemin.

Volland et al. [2] ont proposé une revue de la littérature sur le transport de matériel dans la logistique hospitalière en 2017. Ils ont répertorié cinq articles qui traitent de l'optimisation de la logistique externe (transport inter-hôpitaux) couplée à la planification des chauffeurs. Ce travail permet de positionner (voir table 2.6) notre problème par rapport aux travaux de la littérature.

Publication	Description du problème	Fonction objective	Produits
Shih and Chang [138] (2001)	Planification des routes pour la collecte périodique des déchets dans un réseau logistique	Minimiser le coût du transport, Minimiser le nombre de kilomètres parcourus au quotidien	Déchets
Swaminathan [139] (2003)	Allocation des médicaments rares aux hôpitaux	Minimiser la valeur totale du budget des médicaments laissés, et maximiser la valeur totale du budget des médicaments alloué	Produits pharmaceutiques
Medaglia et al. [140] (2009)	Problème de choix d'installations pour le réseau de traitement des déchets hospitaliers	Minimiser le coût de transport, minimiser les employés affectés	Déchets
Bailey et al. [141] (2013)	Démonstration de la faisabilité d'un système de livraison de boîte de casier sans surveillance	Maximiser le nombre de commandes pouvant être stockées dans les boîtes de casiers	Articles urgents
Kergosien et al. [142] (2013)	Conception des flux de transport entre les unités d'un hôpital; dimensionnement des employés	Minimiser la somme des retards et minimiser le nombre nécessaire d'employés	Produits pharmaceutiques, consommables
Notre problème	Dimensionnement de l'équipe des chauffeurs; dimensionnement de la flotte des véhicules; contraintes d'incompatibilité entre types de produits, entre type de produit et type de camion, entre type de camion et site de collecte et/ou de livraison; contraintes sur les durées de temps de travail, de temps de conduite de temps de service et de temps de repos des chauffeurs; possibilité pour un chauffeur de changer de véhicule	Minimiser le coût dû au nombre de chauffeurs utilisés, le coût fixe d'utilisation des véhicules et le coût total de carburant	linge, médicaments, repas, et produits magasiniers

TABLE 2.6 – Publications sur les problèmes de la logistique hospitalière externe couplé au problème de planification du personnel (voir Volland et al. [2])

2.5 Conclusion

Le problème abordé dans cette thèse est un problème difficile, car c'est une extension du problème de collecte et livraison avec fenêtre de temps qui est un problème Np-difficile (voir Xu et al. [83]). Comme le montre la revue de la littérature faite dans ce chapitre, des travaux sur les problématiques voisines à la nôtre ont été abordés. Mais ces travaux n'ont pas considéré

tous les aspects qui sont présents dans notre étude. Donc les études et les solutions que nous présentons dans les prochains chapitres, sont les premières méthodes de résolution proposées pour résoudre ce problème. Comme noté dans ce chapitre, la démarche de résolution que nous avons entreprise se déroule en trois étapes, dont la première est la résolution du problème de collectes et livraisons avec fenêtres de temps et flotte hétérogène de véhicules que nous aborderons au prochain chapitre.

Chapitre 3

Collecte et livraisons avec fenêtres de temps : flotte homogène

3.1 Introduction

Le problème de collectes et livraisons avec fenêtres horaires et flotte homogène de véhicules peut être défini comme un problème de transport des personnes ou de biens par une flotte de véhicules ayant les mêmes caractéristiques. Chaque personne ou marchandise est transportée de son point de collecte, au point de livraison correspondant. L'heure de départ et l'heure d'arrivée sont contraintes chacune par une fenêtre de temps. Le transport d'une personne ou d'une marchandise entre son point de départ et son point d'arrivée est effectué par le même véhicule. Après qu'un camion ait satisfait une demande de collecte, si les contraintes temporelles le permettent, il peut effectuer d'autres demandes avant de réaliser la demande de livraison correspondante.

Dans la figure 3.1, un exemple d'une solution du problème de collectes et livraisons avec fenêtres horaires et flotte homogène de véhicules composées de 20 demandes (soit 10 requêtes) desservies par 3 véhicules est proposé. Dans chaque route, chaque couple de demande d'une requête $(p_i, d_i) \in \{1, \dots, 10\}$ est satisfait dans la même tournée. Dans chaque tournée, la demande de collecte p_i est satisfaite avant la demande de livraison correspondante d_i .

Les coûts retenus dans notre travail sont le coût fixe d'acquisition des véhicules et le coût de carburant nécessaire pour satisfaire l'ensemble des requêtes. On retrouve les problèmes de collectes et livraisons (PDP : Pickup and Delivery Problem) dans le transport ferroviaire, aérien, routier, et fluvial. Il faut noter que tout problème de tournée de véhicules est assimilable à un problème de collecte et livraison (en considérant le(s) dépôt(s) comme étant le point de livraison). Une synthèse des travaux menés sur ce sujet jusqu'en 2008 a été réalisée par Parragh et al. [143].

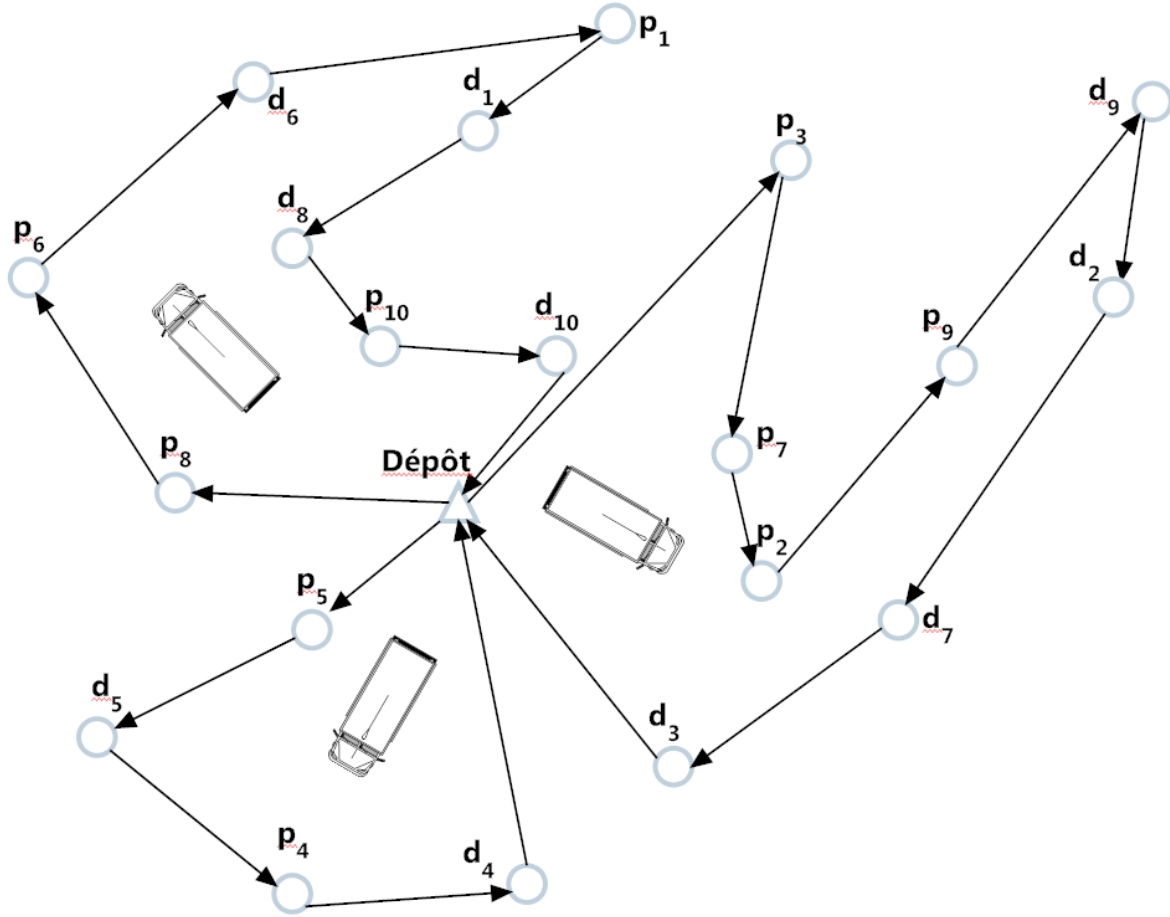


FIGURE 3.1 – Exemple d’une solution d’un problème de collectes et livraisons avec fenêtre de temps et flotte homogène de véhicules composé de 20 demandes et 3 véhicules.

Comme nous l’avons évoqué et illustré au chapitre 2, les études sur le problème de collectes et livraisons avec fenêtre de temps et flotte homogène de véhicules ont donné lieu au développement de méthodes de résolution approchées et exactes. La majorité des travaux ont utilisé le benchmark de Li and Lim [3] (obtenu à partir du [benchmark de Solomon](#)) pour montrer l’efficacité des méthodes de résolution proposées. Ceci a motivé le choix de ce benchmark pour évaluer les méthodes de résolution que nous proposons dans ce chapitre.

Dans les sections suivantes, nous présentons les différentes approches que nous avons développées dans le cadre de notre étude sur le sujet. Nous commençons par aborder les méthodes exactes et ensuite les méthodes approchées. Enfin, nous concluons en comparant nos différentes approches de résolution entre elles mais aussi avec des méthodes de résolution présentes dans la littérature.

3.2 Résolution exacte du problème

Les méthodes dites exactes en recherche opérationnelle sont des algorithmes qui permettent de garantir l'obtention d'une solution optimale. Avant de présenter nos différentes approches de résolution mais aussi les techniques d'accélération développées, nous allons d'abord commencer par proposer une modélisation mathématique du problème.

3.2.1 Modélisation par un programme linéaire en nombres mixtes

Dans cette sous-section, nous proposons une formulation du problème de collecte et livraison avec fenêtre de temps et contraintes de capacité. Elle se base sur la formulation proposée par L. Grandinetti *et al.* [144] pour obtenir une formulation plus compacte.

Considérons une flotte homogène de véhicules K . Tout véhicule $k \in K$ a une capacité Q (flotte homogène), un coût fixe d'utilisation δ (coût de location ou coût d'achat) et un coût v par kilomètre parcouru.

Dans une solution, toute tournée est réalisée par un seul véhicule et quitte du dépôt pour se terminer au dépôt. Notons n le nombre de requêtes. Toute requête est formée d'un couple de demande : une demande de collecte et une demande de livraison associée telle que les deux demandes sont réalisées avec le même véhicule, et la demande de livraison est réalisée après la demande de collecte. Autrement dit, on livre ce que l'on a collecté. Dans le problème que nous étudions, toutes les requêtes doivent être satisfaites, c'est-à-dire que l'on doit avoir un taux de service de 100%.

Soient A l'ensemble des arcs et V l'ensemble des sommets. Le graphe des transports est modélisé par un graphe direct $G = \langle V, A \rangle$, où $V = \{0, 1, \dots, n, n+1, \dots, 2n, 2n+1\}$. Les sommets 0 et $2n+1$ sont des demandes fictives modélisant le départ et le retour au dépôt. Notons $P \subset V$ le sous-ensemble de sommets représentant l'ensemble des demandes de collectes et $D \subset V$ celui des livraisons. Pour toute demande $i \in P$, $q_i > 0$ représente le nombre de chariots à collecter et pour toute demande de collecte $i \in D$ et $q_i < 0$ est le nombre de chariots à livrer pour une demande de livraison. La pondération négative du nombre de chariots pour une demande de livraison est due au fait qu'elle correspond à une diminution du nombre de chariots présents dans le véhicule après avoir effectué la livraison.

À chaque demande $i \in V$, correspond une fenêtre de visite $[a_i, b_i]$. Pour tout $i \in P$, $n+i$ désignera la demande de livraison associée et $|q_i| = |q_{n+i}|$. s_i le temps nécessaire pour effectuer le service i . La valeur du temps de service dépend du nombre de chariots, des caractéristiques du site de chargement (ou déchargement), et éventuellement, du temps de nettoyage du véhicule. d_{ij} (resp. t_{ij}) est la distance (resp. le temps) pour aller du site de la demande i au site de la demande j (ces matrices ne sont pas toujours symétriques).

La réalisation consécutive de deux demandes i et j est réalisable, si en quittant le dépôt et

en arrivant au plus tôt sur le site la demande i , on aurait assez de temps pour arriver au site la demande j avant la date de début au plus tard de la réalisation de cette dernière. Plus formellement :

$$\text{Un arc } (i, j) \in A \text{ si } \max\{t_{0i}, a_i\} + s_i + t_{ij} \leq b_j \text{ et } i \neq j \forall i, j \in V. \quad (3.1)$$

Les variables de décision utilisées pour définir le modèle mathématique sont :

- $x_{i,j}^k$ qui est une variable binaire de valeur 1 si le véhicule k satisfait la demande i et ensuite j , 0 sinon ;
- B_i^k qui est la variable continue représentant la date de début du service de la demande i par le véhicule k ;
- Q_i^k qui est la variable entière représentant le nombre de chariots présents dans le véhicule k lorsque ce dernier quitte le sommet i ;
- y^k est la variable binaire de valeur 1 si le véhicule k est utilisé et 0 sinon.

n	nombre de requêtes
A	ensemble des arcs
V	ensemble de sommets (demandes)
P	ensemble des demandes de collectes
D	ensemble des demandes de livraisons
K	ensemble de véhicules
Q	capacité d'un véhicule
δ	coût fixe d'un véhicule
v	coût au kilomètre d'un véhicule
$ q_i $	nombre de chariots à livrer ou à collecter de i
a_i	date de début au plus tôt de i
b_i	date de début au plus tard de i
s_i	temps de service de i
$t_{i,j}$	temps de parcours du site de la demande i au site de la demande j
$d_{i,j}$	distance de parcours du site de la demande i au site de la demande j
$n+i$	livraison correspondant à la collecte i
M	est un entier "très grand"

TABLE 3.1 – Notations

La formulation mathématique proposée par L. Grandinetti *et al.* [144] est donnée par :

$$(PL0) \quad \text{Min} \sum_{k \in K} \delta \times y^k + \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} v d_{ij} \times x_{ij}^k \quad (3.2)$$

$$\sum_{k \in K} \sum_{j \in V | j \neq i} x_{ij}^k = 1, \quad \forall i \in P, \quad (3.3)$$

$$\sum_{j \in V | j \neq i} x_{ij}^k - \sum_{j \in V | j \neq i} x_{n+i,j}^k = 0, \quad \forall (i, k) \in V \times K, \quad (3.4)$$

$$\sum_{j \in V | j \neq 0} x_{0j}^k = y^k, \quad \forall k \in K \quad (3.5)$$

$$\sum_{i \in V | i \neq 2n+1} x_{i(2n+1)}^k = y^k, \quad \forall k \in K, \quad (3.6)$$

$$\sum_{i \in V} \sum_{j \in V | j \neq i} x_{ij}^k \leq M y^k, \quad \forall k \in K, \quad (3.7)$$

$$\sum_{i \in V} \sum_{j \in V | j \neq i} x_{ij}^k \geq y^k, \quad \forall k \in K, \quad (3.8)$$

$$\sum_{j \in V | j \neq i} x_{ji}^k - \sum_{j \in V | j \neq i} x_{ij}^k = 0, \quad \forall (i, k) \in (P \cup D) \times K, \quad (3.9)$$

$$B_j^k - B_i^k \geq (s_i + t_{ij}) x_{ij}^k - M(1 - x_{ij}^k), \quad \forall (i, j, k) \in V^2 \times K, i \neq j, \quad (3.10)$$

$$B_i^k - B_{n+i}^k + t_{in+i} \sum_{j \in V} x_{ij}^k \leq 0, \quad \forall (i, j, k) \in V^2 \times K, i \neq j, \quad (3.11)$$

$$a_i \sum_{j \in V} x_{ij}^k \leq B_i^k \leq b_i \sum_{j \in V} x_{ij}^k, \quad \forall (i, k) \in V \times K, \quad (3.12)$$

$$Q_j^k - Q_i^k \geq q_j x_{ij}^k - M(1 - x_{ij}^k) \quad \forall (i, j, k) \in V^2 \times K, \quad (3.13)$$

$$\max\{0, q_i\} \sum_{j \in V} x_{ij}^k \leq Q_i^k, \quad \forall (i, k) \in P \times K, \quad (3.14)$$

$$Q_i^k \leq \min\{Q, Q + q_i\} \sum_{j \in V} x_{ij}^k, \quad \forall (i, k) \in V \times K, \quad (3.15)$$

$$y^k \in \{0, 1\}, \quad B_i^k > 0, \quad Q_i^k > 0, \quad x_{ij}^k \in \{0, 1\}, \quad \forall (i, j, k) \in V^2 \times K \quad (3.16)$$

L'objectif (3.2) est de minimiser le coût fixe d'utilisation des véhicules et le coût kilométrique des tournées. Les contraintes (3.3-3.4) imposent que chaque demande de collecte doit être satisfaite et assurent que la demande de livraison associée est satisfaite par le même véhicule. Les contraintes (3.5-3.6) modélisent le fait qu'un véhicule quitte du dépôt (et y retourne) si et seulement si, il est utilisé. Les contraintes (3.8-3.9) imposent que le coût fixe d'utilisation d'un véhicule soit considéré si ce dernier effectue une livraison ou une collecte. Les contraintes (3.10) sont des contraintes de conservation de flux permettant de modéliser les enchaînements entre les différents trajets effectués par un véhicule. Si un véhicule réalise une demande i et ensuite une demande j , alors la contrainte (3.11) permet d'assurer que la différence entre les deux dates de départ soit supérieure ou égale au temps de service de i , additionné au temps de parcours pour aller de i à j . Les contraintes (3.12-3.13) modélisent le respect des fenêtres horaires et la précedence de l'ordre de service entre un point de collecte et le point de livraison correspondant. Les contraintes (3.14-3.16) modélisent le remplissage des véhicules tout au long de la tournée et assurent que la capacité de chaque véhicule soit respectée.

Dans la formulation que nous avons présentée on peut remarquer qu'il y a un nombre de "grande valeur" (voir les contraintes 3.7, 3.10 et 3.13) qui influence la compacité du modèle.

En effet, la présence de M agrandit la surface du polyèdre associé à la relaxation linéaire du modèle (PL0). Afin de supprimer cet effet, nous proposons une formulation plus stricte. Pour ce faire, nous modélisons différemment les contraintes de respect de capacité des véhicules.

Soit les variables de décisions entières : $Q_{i,j}^k$ qui représente le nombre de chariots présents dans le véhicule k sur l'arc (i, j) . En reprenant les notations précédemment introduites, le modèle est donné par :

$$(PL1) \quad \text{Min} \sum_{i \in P} \sum_{k \in K} \delta \times x_{0i}^k + \sum_{(i,j) \in A} \sum_{k \in K} vd_{ij} \times x_{ij}^k \quad (3.17)$$

$$\sum_{k \in K} \sum_{j: (i,j) \in A} x_{ij}^k = 1, \quad \forall i \in P \quad (3.18)$$

$$\sum_{j: (i,j) \in A} x_{ij}^k = \sum_{j: (n+i,j) \in A} x_{n+i,j}^k, \quad \forall (i, k) \in P \times K \quad (3.19)$$

$$\sum_{i \in P} x_{0i}^k \leq 1, \quad \forall k \in K \quad (3.20)$$

$$\sum_{i \in P} x_{0i}^k = \sum_{i \in D} x_{i(2n+1)}^k, \quad \forall k \in K \quad (3.21)$$

$$\sum_{j: (i,j) \in A} x_{ij}^k \leq \sum_{j \in P} x_{0j}^k, \quad \forall (i, k) \in P \cup D \times K, \quad (3.22)$$

$$\sum_{j: (i,j) \in A} x_{ij}^k - \sum_{j: (j,i) \in A} x_{ji}^k = 0, \quad \forall (i, k) \in P \cup D \times K \quad (3.23)$$

$$B_i^k - b_i + (b_i + s_i + t_{ij})x_{ij}^k \leq B_j^k, \quad \forall (i, j) \in A, \forall k \in K \quad (3.24)$$

$$B_i^k - B_{n+i}^k + (s_i + t_{in+i}) \sum_{j \in V} x_{ij}^k \leq 0, \quad \forall (i, k) \in P \times K, \quad (3.25)$$

$$a_i \sum_{j: (i,j) \in A} x_{ij}^k \leq B_i^k \leq b_i \sum_{j: (ij) \in A} x_{ij}^k, \quad \forall (i, k) \in V \times K, \quad (3.26)$$

$$\sum_{i \in P} \sum_{k \in K} Q_{0i}^k + \sum_{i \in D} \sum_{k \in K} Q_{i(2n+1)}^k = 0. \quad (3.27)$$

$$\sum_{k \in K} \sum_{j: (i,j) \in A} Q_{ij}^k - \sum_{k \in K} \sum_{j: (j,i) \in A} Q_{ji}^k = q_i, \quad \forall i \in N, \quad (3.28)$$

$$Q_{ij}^k \leq Q \times x_{ij}^k, \quad \forall (i, j) \in A, \forall k \in K \quad (3.29)$$

$$x_{ij}^k \in \{0, 1\}, \quad Q_{ij}^k > 0, \quad B_i^k > 0 \quad \forall i, j \in N, \forall k \in K \quad (3.30)$$

La fonction objective (3.17) est la minimisation du coût fixe d'utilisation des véhicules et du coût de carburant des tournées. Les contraintes (3.18-3.19) sont respectivement similaires aux contraintes (3.3-3.4). Les contraintes (3.20) imposent aux véhicules de quitter le dépôt au plus une fois, alors que les contraintes (3.21) assurent leur retour au dépôt. Les contraintes (3.22)

contraignent les véhicules à quitter le dépôt pour servir une demande. Les contraintes (3.23-3.26) sont respectivement similaires aux contraintes (3.10-3.13). La contrainte (3.27) assure que tout véhicule quitte et retourne au dépôt à vide. Les contraintes (3.28-3.29) modélisent le remplissage des véhicules tout au long des tournées, et assurent que le nombre de chariots présents dans tout véhicule n'excède pas la capacité de ce véhicule.

Dans la formulation que nous proposons, il n'y a plus de nombre entier de "grande valeur" ce qui montre que la relaxation linéaire de notre modèle est plus compacte. Le modèle (PL0) a $|K||N|^2 + |K||N| + |K|$ variables et $3|K||N|^2 + \frac{9|K||N|}{2} + \frac{N}{2} + 4|K|$ contraintes, et notre modèle a plus de variables ($2|K||N|^2 + |K||N|$ variables) et moins de contraintes ($2|K||N|^2 + \frac{9|K||N|}{2} + \frac{3|N|}{2} + 2|K|$).

Comme nous le verrons plus tard dans les expérimentations numériques, le modèle (PL1) de programmation linéaire proposé ne permet pas de résoudre des instances à 100 demandes en moins de trois heures de calcul. Pour accélérer la résolution du modèle, nous avons travaillé sur des méthodes de réduction de l'arbre des possibilités ainsi qu'au développement de propriétés permettant de supprimer des branches de l'arbre des solutions lors de la résolution.

3.2.2 Bornes sur les fenêtres de temps

Des bornes sur les fenêtres de temps ont été proposées pour la première fois dans l'article [145] pour le problème de voyageur de commerce avec fenêtre de temps (TSPTW). En 2006, J.-F. Cordeau [146] ont renforcé les bornes des variables de temps de début de service B_i d'une demande i , pour le PDPTW (flotte homogène) par les inégalités :

$$B_i \geq a_i + \sum_{j \in N \setminus \{i\}} \max\{0, a_j - a_i + s_j + t_{ij}\} x_{ij} \quad (3.31)$$

$$B_i \leq b_i - \sum_{j \in N \setminus \{i\}} \max\{0, b_i - b_j + s_i + t_{ij}\} x_{ij}. \quad (3.32)$$

3.2.3 Réduction de l'espace de recherche

Dans cette section, nous présentons différents moyens pour réduire l'espace de recherche en supprimant des arcs irréalisables de $G = (V, A)$. Nous pouvons observer que :

- $(0, n + i)$ est irréalisable $\forall i \in P$ (on ne quitte pas du dépôt pour réaliser directement une livraison),
- $(i, 2n + 1)$ est irréalisable $\forall i \in P$ (on ne retourne pas au dépôt après avoir réaliser une collecte).

Y. Dumas *et* M. Desrochers [147] ont proposé d'éliminer pour le problème de collectes et livraisons avec fenêtres de temps (PDPTW) les arcs :

- $(i, n + j)$, si le chemin $(j, i, n + j, n + i)$ est irréalisable.
- $(n + i, j)$, si le chemin $(i, n + i, j, n + j)$ est irréalisable.

- (i, j) , si les chemins $(i, j, n + i, n + j)$ et $(i, j, n + j, n + i)$ sont irréalisables.
- $(n + i, n + j)$, si les chemins $(i, j, n + i, n + j)$ et $(j, i, n + i, n + j)$ sont irréalisables.

Nous proposons de rajouter :

- (i, i) est irréalisable $\forall i \in N$ (on ne peut pas réaliser une même demande deux fois de suite).
- (i, j) et $(j, n + i)$, sont irréalisables, si $a_i + t_{ij} + t_{j, n+i} > b_{n+i} \forall i \in P, j \in N \setminus \{0, i, n + i, 2n + 1\}$.
- (i, j) est irréalisable, si $q_i + q_j > Q \forall i, j \in P$ (la somme des nombres de chariots à collecter successivement pour deux collectes ne doit pas excéder la capacité du véhicule).

Dans la suite, nous supposons supprimé du graphe G , tous les arcs irréalisables décelés par les critères ci-dessus. Toutes ces techniques de réduction de l'espace sont utilisées avant le début de la résolution proprement dit du problème. Dans la suite, nous allons définir et présenter des inégalités valides qui seront utilisées pendant la résolution du problème.

3.2.4 Inégalités valides

Une **inégalité valide** pour un programme linéaire (P) peut être définie comme une inégalité satisfaite par toute solution réalisable de (P) [148].

Proposition 3.2.1 *Considérant les notations précédentes, l'inégalité :*

$$\sum_{k \in K} \sum_{(i,j) \in A} vd_{ij} x_{ij}^k \geq L \quad (3.33)$$

avec $L = \sum_{i \in V} \min_{k \in K} \min_{j \in V | (i,j) \in A} vd_{ij}$, est une inégalité valide pour le PDPTW.

Preuve 3.2.1 Soit $i \in V$, $c_i = \min_{k \in K, j \in V | (i,j) \in A} vd_{ij}$.

$$\sum_{k \in K} \sum_{(i,j) \in A} vd_{ij} x_{ij}^k = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} vd_{ij} x_{ij}^k \quad (3.34)$$

$$\geq \sum_{i \in V} \sum_{k \in K} \sum_{j \in V} \min_{j \in V | (i,j) \in A} (vd_{ij}) x_{ij}^k ; \text{ car } d_{ij} \geq \min_{j \in V | (i,j) \in A} d_{ij}, \quad (3.35)$$

$$\geq \sum_{i \in V} \sum_{k \in K} \min_{j \in V | (i,j) \in A} (vd_{ij}) \sum_{j \in V} x_{ij}^k, \quad (3.36)$$

$$\geq \sum_{i \in V} \min_{k \in K} \min_{j \in V | (i,j) \in A} (vd_{ij}) \sum_{k \in K} \sum_{j \in V} x_{ij}^k \quad (3.37)$$

$$\geq \sum_{i \in V} \min_{k \in K} \min_{j \in V | (i,j) \in A} (vd_{ij}) ; \text{ car } \sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1 \quad (\text{voir 3.18}) \quad (3.38)$$

$$\geq L ; \text{ par définition.} \quad (3.39)$$

L'inégalité 3.33 permet d'obtenir une valeur minimale du coût en carburant nécessaire pour satisfaire toutes les demandes.

Soit $i \in V$, notons m_i le temps nécessaire pour satisfaire la demande i et ensuite aller à son voisin le plus proche. La proposition suivante donne pour un ensemble de requêtes, une condition nécessaire pour qu'un véhicule puisse satisfaire toutes les requêtes de cet ensemble.

Proposition 3.2.2 *Soit $k \in K$ un véhicule, notons $m_i = s_i + \min_{j \in V | (i,j) \in A} t_{ij}$, $\forall i \in V$. En considérant les notations précédentes, l'inégalité :*

$$\sum_{i \in P} \sum_{\substack{j \in V \\ (i,j) \in A}} (m_i + m_{n+i}) x_{ij}^k \leq b_{2n+1} \quad (3.40)$$

est une inégalité valide pour le PDPTW.

Preuve 3.2.2 *Pour démontrer la proposition 3.2.2, il suffit de montrer que pour tout véhicule $k \in K$, toute tournée réalisée par le véhicule k vérifie l'inégalité 3.40. Soit $k \in K$ un véhicule utilisé pour effectuer une tournée $r = (w_0, \dots, w_l)$. Si r est réalisable, alors elle se termine avant la date d'arrivée au plus tard au dépôt. Et on a :*

$$b_{2n+1} \geq \sum_{i=0}^{l-1} (s_{w_i} + t_{w_i w_{i+1}}) \quad (3.41)$$

$$\geq \sum_{i=0}^{l-1} (s_{w_i} + \min_{\substack{e \in V \\ (w_i, e) \in A}} t_{w_i e}) \quad (3.42)$$

$$\geq \sum_{i=0}^{l-1} m_{w_i} \quad (3.43)$$

$$\geq \sum_{\substack{i=0 \\ i \in P}}^{l-1} (m_{w_i} + m_{n+w_i}) \quad (3.44)$$

$$\geq \sum_{\substack{i=0 \\ i \in P}}^{l-1} (m_{w_i} + m_{n+w_i}) \sum_{\substack{j \in V \\ (i,j) \in A}} x_{ij}^k; \quad \text{car} \quad \sum_{\substack{j \in V \\ (i,j) \in A}} x_{ij}^k \leq 1, \quad (\text{d'après 3.20 et 3.22}) \quad (3.45)$$

$$\geq \sum_{\substack{i=0 \\ i \in P}}^{l-1} (m_{w_i} + m_{n+w_i}) \sum_{\substack{j \in V \\ (i,j) \in A}} x_{ij}^k; \quad (3.46)$$

$$\geq \sum_{\substack{i=0 \\ i \in P}}^{l-1} \sum_{\substack{j \in V \\ (i,j) \in A}} (m_{w_i} + m_{n+w_i}) x_{ij}^k; \quad (3.47)$$

D'où le résultat.

Proposition 3.2.3 Soient $i \in P$ et $j \in P$ des demandes de collectes. Si les chemins :

- $(i, j, n + i, n + j)$,
- $(i, j, n + j, n + i)$,
- $(i, n + i, j, n + j)$,
- $(j, i, n + j, n + i)$,
- $(j, i, n + i, n + j)$,
- $(j, n + j, i, n + i)$,

sont tous irréalisables, alors :

$$\sum_{k \in K} \left(\sum_{\substack{l \in V \\ (i, l) \in A}} x_{il}^k + \sum_{\substack{l \in V \\ (j, l) \in A}} x_{jl}^k \right) \geq 2 \quad (3.48)$$

est une inégalité valide.

Preuve 3.2.3 Soient $(i, j) \in P^2$ des demandes de collectes qui satisfont les hypothèses de la proposition 3.2.3. Soit $(i, j) \in S$, par définition de l'ensemble S , il n'existe pas de véhicule pouvant réaliser à la fois les demandes i et j , donc il faut au minimum deux véhicules pour satisfaire les demandes i et j .

Ces coupes (inégalités valides) dans l'espace des solutions ont comme nous le verrons plus tard, un impact considérable dans le temps de résolution des instances ayant des fenêtres de temps "emboîtées".

Proposition 3.2.4 Soient $(d_0, d_1, \dots, d_m)$ un chemin réalisable du point de vue des fenêtres horaires, mais violant la contrainte de respect de la capacité d'un véhicule k . Alors $\exists m_1 \in \{0, \dots, m\}$ de plus grandes valeurs et $\exists m_2 \in \{0, \dots, m\}$ de plus petite valeur tels que : $d_{m_1}, d_{m_2} \in P$ et

$$\sum_{i=m_1}^{m_2-1} x_{i,i+1}^k \leq m_2 - m_1$$

est une inégalité valide.

Preuve 3.2.4 La capacité du véhicule n'étant pas respectée, il existe un indice $m_2 \in \{1, \dots, m\}$ de plus petite valeur telle que la capacité à cette position soit violée. Alors $d_{m_2} \in P$ car la capacité ne pourrait pas être violée à une position où l'on effectue une demande de livraison. À partir de m_2 , reculons d'une position tant que la capacité du véhicule est non nulle, et prenons m_1 la dernière position à laquelle la capacité du véhicule est non nulle. Alors $m_1 \in P$ car la capacité du véhicule est nulle en $m_1 - 1$ et non nulle en m_1 ce qui traduit une croissance de la charge dans le véhicule. Le chemin $(d_{m_1}, \dots, d_{m_2})$ est donc irréalisable et la suppression de ce chemin peut être réalisée en rajoutant la contrainte : $\sum_{i=m_1}^{m_2-1} x_{i,i+1}^k \leq m_2 - m_1$.

Cette inégalité est plus forte que :

$$\sum_{i=1}^{m-1} x_{i,i+1}^k \leq m - 2$$

qui a été proposée par J.-F. Cordeau [146] pour supprimer un chemin irréalisable. D'autres inégalités valides sont présentées dans ces travaux.

3.2.5 Résultats expérimentaux

Les expérimentations ont été réalisées sur deux benchmarks. Le premier est celui de Li and Lim [3] (avec un peu plus de 100 demandes par instance) présent dans la littérature, et le second est formé des instances réelles fournies par le Centre Hospitalier de Troyes (France). La plate-forme logistique de Troyes comprend une flotte de véhicules de capacités différentes. Les demandes de collectes et de livraisons sont périodiques, avec une période de 7 jours. Donc nous disposons de 7 instances à raison d'une instance par jour. Chaque instance est nommée sous le format *Day_*|*V*|.

Définissons (*PL2*) l'algorithme obtenu en ajoutant à (*PL1*) les techniques de réduction de l'espace de recherche. (*PL3*) est l'algorithme obtenu en rajoutant à (*PL2*) les inégalités valides proposées.

Les tests sur les approches proposées dans ce travail ont été réalisés en utilisant ILOG CPLEX (version : 12.6.0) sur Eclipse. Le langage de programmation a été le C++, sur une machine Core i5 ayant 2.6 GHz et 7.7 GB de RAM.

Après 3 heures de calculs, aucune des approches (*PL1*), (*PL2*) et (*PL3*) n'a été en mesure de résoudre les instances de la littérature. En revanche comme le montre le tableau 3.2, nous avons été en mesure de trouver une solution optimale pour chacune des instances au bout de moins de 3 heures de calcul sur le benchmark proposé par le CHT. Un exemple d'instance réelle (celle de dimanche) est donné en annexe dans la figure 7.1.

Instance	(<i>PL1</i>)	(<i>PL2</i>)	(<i>PL3</i>)
<i>Mo_114</i>	–	1011 sec	294 sec
<i>Tu_100</i>	–	–	3 886 sec
<i>We_106</i>	–	168 sec	71 sec
<i>Th_92</i>	–	–	67 sec
<i>Fr_114</i>	–	735 sec	192 sec
<i>Sa_62</i>	24 sec	13 sec	6 sec
<i>Su_52</i>	16 sec	6 sec	2 sec

TABLE 3.2 – Résultats expérimentaux sur les instances réelles.

Nous constatons sur cet échantillon que, sans réduire l'espace de recherche et sans générer des coupes, le programme linéaire proposé résout seulement les instances de petite taille (environ

60 demandes). Les résultats obtenus par (*PL2*) montrent l'impact de la réduction de l'espace de recherche sur le temps de calcul. De plus, désormais l'algorithme résout plus d'instances, et améliore le temps de résolution sur les instances qu'il résolvait préalablement. Le rajout des inégalités valides permet de résoudre toutes les instances en moins de deux heures de calcul.

3.3 Résolution par métaheuristique

Les méthodes exactes pour le moment ont un temps de calcul non exploitable pour un développement d'outil d'aide à la décision (implémentation industrielle). Afin d'avoir rapidement des solutions de bonne qualité, plusieurs heuristiques et métaheuristiques ont été proposées dans la littérature pour résoudre ce problème. En étudiant ces différents algorithmes proposés, nous avons conçu une approche innovante qui permet d'avoir de meilleurs résultats que les algorithmes proposés jusqu'à présent pour résoudre ce problème. La métaheuristique proposée est une hybridation de l'algorithme de colonie de fourmis. Nous avons proposé des recherches locales dédiées aux différents aspects du problème et aussi de nouveaux mécanismes de mise à jour des phéromones et de calcul de la visibilité pour une fourmi donnée. Une taxinomie sur les métaheuristiques a été proposé par Talbi [149]. Dans leurs travaux, ils proposent une terminologie et les mécanismes de classification de ce type de méthode de résolution. Les exemples de réussite de résolution de problèmes réels par des métaheuristiques sont nombreuses, un exemple notoire est celui de Nakib [150] dans le milieu hospitalier sur le problème de segmentation d'images biomédicales.

3.3.1 Algorithme de colonie de fourmis

Principe général

L'heuristique de colonies de fourmis a été initialement proposée par Dorigo and Gambardella [151] pour la recherche de chemins optimaux dans un graphe. Il s'inspire des travaux de Goss et al. [152] qui étudient le comportement des fourmis recherchant un chemin entre leur colonie et une source de nourriture. Cette technique a rencontré de nombreux succès sur des problèmes d'optimisation combinatoire. C'est le cas des travaux de Belmecheri et al. [153] pour résoudre un problème de tournée de véhicules avec fenêtre de temps et une flotte hétérogène de véhicules.

Nous avons proposé une métaheuristique efficace qui est une hybridation de l'algorithme de colonies de fourmis couplé à des algorithmes de recherches locales dédiées. Le choix de l'algorithme de colonies de fourmis a été motivé par le fait qu'il construit des solutions réalisables a contrario de certains algorithmes (comme l'algorithme génétique) qui demandent souvent une reconstruction afin d'avoir une solution réalisable.

Codage d'une solution

Dans l'approche que nous proposons, une fourmi construit une solution composée d'un ensemble de routes. Chaque route est effectuée par un véhicule et les routes sont construites de manière séquentielle. Une nouvelle route est rajoutée si toutes les demandes n'ont pas été satisfaites, et aucune d'elles ne peut être insérée dans la route en construction.

Construction d'une solution

Afin de définir la procédure de choix de la prochaine demande à satisfaire, notons r la dernière route en construction. Notons i la dernière demande satisfaite dans la route r , et $\mathcal{V}_r^i$ l'ensemble des demandes de collectes réalisées mais dont les demandes de livraisons correspondantes n'ont pas été satisfaites après la réalisation de la demande i .

Une demande j est dite *éligible* pour être réalisée, si elle n'a pas encore été réalisée, et que l'une des conditions suivantes est satisfaite :

- c'est une collecte ($0 < j \leq n$) et il existe un chemin pour satisfaire toutes les demandes dans $\mathcal{V}_r^i \cup \{n + j\}$
- c'est une livraison ($n < j \leq 2n \wedge j - n \in r$) et il existe un chemin pour satisfaire toutes les demandes dans $\mathcal{V}_r^i \setminus \{j\}$

Étant donné une solution partielle ρ , se terminant par la réalisation d'une demande notée i ; une demande éligible j est choisie pour être insérée dans la route en construction de façon aléatoire (distribution uniforme) défini par la probabilité :

$$P_{ij}^\rho = \begin{cases} \frac{(\tau_{ij})^\alpha (\eta_{ij}^\rho)^\beta}{\sum_{d \in S_i^\rho} (\tau_{id})^\alpha (\eta_{id}^\rho)^\beta} & \text{si } j \in S_i^\rho \\ 0 & \text{et 0 sinon.} \end{cases} \quad (3.49)$$

Avec τ_{ij} la valeur de phéromone sur l'arc $(i, j) \in A$; S_i^ρ est l'ensemble des demandes éligibles qu'il est possible de réaliser après la dernière demande i présente dans la solution partielle ρ ; α un réel positif. Et η_{ij}^ρ est la valeur utilisée pour exprimer l'attractivité d'une demande j pour une solution partielle ρ se terminant en i .

$$\eta_{ij}^\rho = \frac{\alpha_1 |S_j^{\rho \cup \{j\}}|}{d_{ij}} \quad (3.50)$$

avec $\alpha_1 \in]0, 1]$ un réel donné.

Dans l'équation 3.50, la probabilité de choisir la demande j comme la prochaine demande à réaliser, est d'autant plus grande que le nombre de demandes que l'on pourra satisfaire après la réalisation de j est élevé, et aussi d'autant plus que la distance du site de la demande j au site à la dernière demande réalisée i est faible. La valeur de ce ratio est atténuée par la constante α_1 .

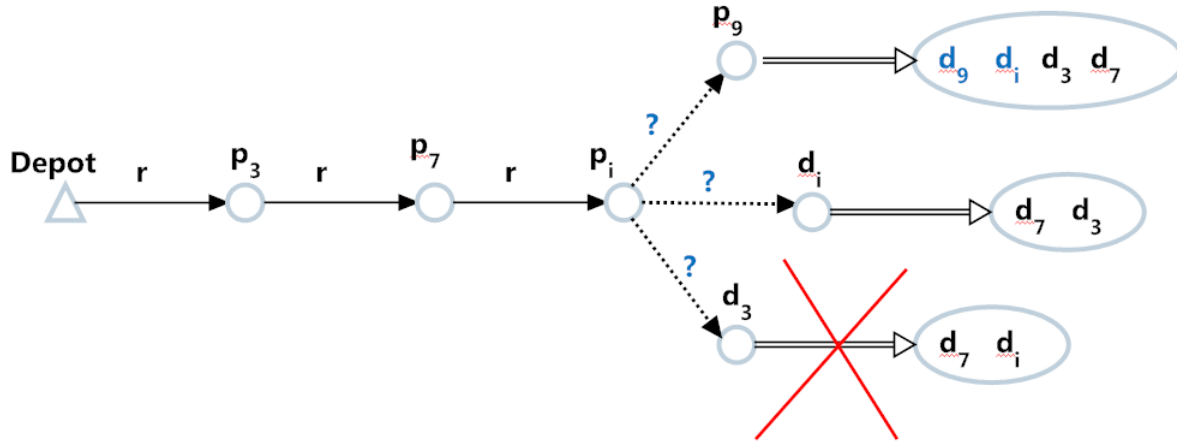


FIGURE 3.2 – Exemple d'éligibilité d'une demande de collecte et d'une demande de livraison

Considérons la figure 3.2, (p_i, d_i) est une requête formée d'une demande de collecte p_i et de la demande de livraison associée d_i . Dans cette illustration, le chemin r se termine en réalisant la demande p_i . Les demandes p_9 et d_i sont éligibles à être satisfaites et la demande d_3 ne l'est pas. En effet, la demande de collecte p_9 est éligible à sa réalisation, car il existe un chemin réalisable pour desservir toutes les demandes de livraisons présentes dans l'ensemble $\{d_9, d_i, d_7, d_3\}$ et de même la demande d_i est éligible à être satisfaite, car il existe un chemin pour réaliser les demandes de livraisons d_7 et d_3 . La non éligibilité de d_3 peut s'expliquer par le fait que la réalisation de celle-ci directement après la demande d_2 entraîne l'impossibilité de réaliser les demandes d_7 et d_i tout en respectant les fenêtres de temps.

Vérification de l'éligibilité

Étant donné une solution partielle ρ , réalisant en dernière position la demande i , la vérification de l'éligibilité d'une demande j (pour insertion) est le problème de recherche d'un chemin hamiltonien. En effet, la réalisation de la demande j implique qu'il doit exister un chemin réalisable contenant toutes les demandes de livraisons dont les demandes de collectes ont été préalablement réalisées. Cette opération de vérification est réalisée à chaque fois qu'une demande est insérée. D'où la nécessité de déterminer des solutions rapides et efficaces. Pour résoudre ce problème, nous proposons l'heuristique d'insertion donnée par le pseudo-code de l'Algorithme 1.

L'objectif de l'algorithme est de déterminer s'il existe un chemin pour livrer les demandes présentes dans un camion après les avoir collectées. Pour ce faire, demande de livraison après demande de livraison, on essaye de construire une solution satisfaisant toutes les demandes de livraisons. Le choix de la demande de livraison à insérer se fait aléatoirement et on l'insère à la position qui minimise le coût induit par l'insertion. Cette opération est répétée tant qu'une solution satisfaisant toutes les demandes n'a pas été trouvée, ou tant que le nombre maximal

Algorithme 1 Heuristique de recherche d'un chemin Hamiltonien

1) Inputs : $\mathcal{V}$ (un ensemble de demandes à satisfaire), $ItMax1$ (un nombre d'itérations), i (la dernière demande satisfaite) et t (le temps de fin de service de la demande i)
tant que le nombre d'itérations est inférieur à $ItMax1$, et qu'il n'existe pas de chemin réalisable satisfaisant les demandes présentes dans $\mathcal{V}$ **faire**
 2) $\mathcal{V}' = \mathcal{V}$
 3) Initialiser une route r commençant au nœud i au temps t
 tant que $\mathcal{V}'$ n'est pas vide **faire**
 4) Choisir aléatoirement une demande de livraison $d \in \mathcal{V}'$, retirer d dans $\mathcal{V}'$, et essayer de l'insérer dans la route r à la position minimisant le coût d'insertion.
 si d n'est pas insérée dans la route r **alors**
 | Retourner à l'étape 3)
 fin
 fin
fin

d'itérations n'a pas été atteint.

Mise à jour des phéromones

Dans la métaheuristique de colonies de fourmis, l'initialisation des phéromones est donnée par l'équation :

$$\tau_{ij} = \tau_0 \text{ si } (i, j) \in A, \text{ et } 0 \text{ sinon.} \quad (3.51)$$

avec τ_0 un scalaire fixé.

Les phéromones sont ensuite mises à jour, comme nous l'avons expliqué à la section 3.3.1.

Dans l'algorithme de colonies de fourmis que nous avons développé, il y a deux mises à jour : une mise à jour locale et une mise à jour globale des phéromones. La mise à jour locale intervient dans la diversification de l'espace de recherche au cours d'une même itération, et la mise à jour globale permet la convergence de l'algorithme. Elle est effectuée à la fin de chaque itération.

— mise à jour locale des arcs visités par une fourmi :

$$\tau_{ij} = \epsilon_1 \tau_{ij} + \tau_0, \quad (3.52)$$

Cette équation traduit la réalisation d'une perturbation locale des phéromones à chaque itération, afin d'apporter de la diversité à la prochaine itération.

— mise à jour globale :

$$\tau_{ij} = \begin{cases} \epsilon_2 \tau_{ij} + (1 - \epsilon_2) \Delta_{ij}^* & \text{si } (i, j) \text{ est présent dans la meilleure solution} \\ \epsilon_2 \tau_{ij} & \text{sinon.} \end{cases} \quad (3.53)$$

avec :

$$\Delta_{ij}^* = \frac{(\text{nombre de demandes dans } r_i^*)^{\alpha_2}}{\text{distance total de } r_i^*}, \quad (3.54)$$

Où r_i^* est la route qui contient la demande i , $\epsilon_1, \epsilon_2 \in]0,1[$, et α_2 sont des réels positifs.

L'équation 3.53 augmente la valeur des phéromones présente sur les arcs de la meilleure solution obtenue. Ce renforcement est fait sur les arcs de la meilleure solution, et par rapport à la route dans laquelle elle est réalisée. Le renforcement et donc la convergence de la solution se traduit par augmentation de la valeur des phéromones, d'une quantité qui est d'autant plus grande que le nombre de demandes présentes dans la route est grand, et que la distance totale est petite.

Dans l'optique d'améliorer les performances de l'algorithme de colonies de fourmis proposé, nous avons élaboré des recherches locales dédiées aux spécificités du problème étudié en fonction des objectifs définis. Ceci soulève la difficulté de la mise en concert de l'algorithme de colonie de fourmis et des heuristiques. Dans la suite, nous proposons un algorithme permettant d'hybrider l'heuristique de colonies de fourmis par les heuristiques de recherches locales.

3.3.2 Algorithmes de recherche locale

Dans cette section, nous proposons trois algorithmes de recherches locales. Ces algorithmes permettent d'améliorer la qualité de toute solution construite par une fourmi. Chaque recherche locale permet d'explorer le voisinage d'une solution au regard d'une des spécificités du problème PDPTW.

Algorithme de recherche locale **H1** : réduction du nombre de routes

Pour réduire le nombre de routes, nous proposons une heuristique **H1** inspirée de l'algorithme à deux phases de Bent and Van Hentenryck [87]. L'heuristique **H1** fonctionne en sélectionnant aléatoirement une route et en retirant cette dernière de la solution et en réinsérant les requêtes présentes dans cette route dans la solution partielle. La réinsertion des requêtes dans la solution partielle se fait dans l'ordre de visite dans la route supprimée. Chaque requête (couple de collecte et livraison) est insérée dans la route aux positions qui minimisent le coût d'insertion de cette dernière. Si toutes les requêtes présentes dans la route supprimée sont réinsérées dans la solution partielle avec un coût meilleur que la dernière solution alors cette dernière est mise à jour.

Un exemple de la réalisation de l'algorithme **H1** est donné par la figure 3.3. Dans cet exemple, une solution est composée de 3 routes r_1, r_2 et r_3 , la tournée r_2 est supprimée de la solution et les demandes présentes dans ce chemin sont insérées dans les autres tournées de la solution pour former une nouvelle solution composée de deux routes (r'_1 et r'_2).

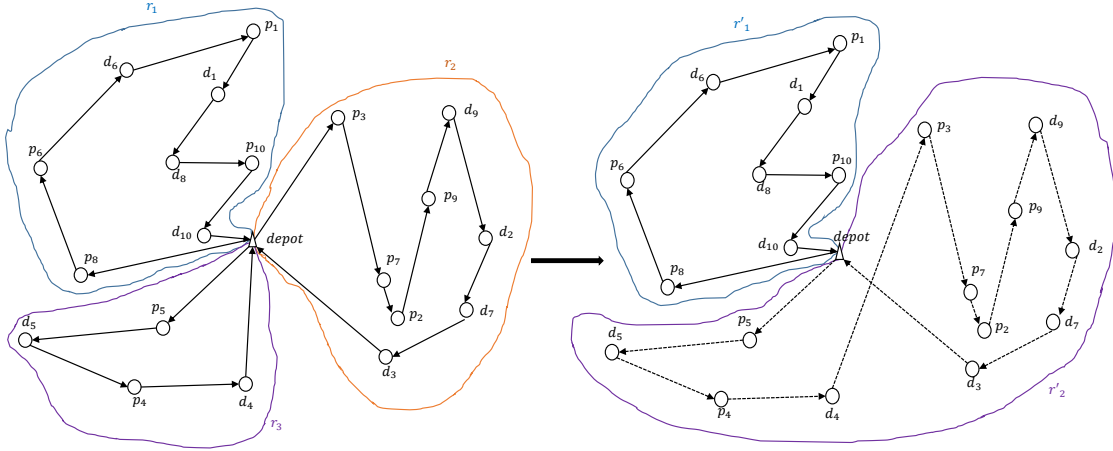


FIGURE 3.3 – Exemple de suppression d'une tournée par l'algorithme H1.

Algorithme de recherche locale H2 : réduction de la distance de la solution dans une tournée

La seconde heuristique $H2$ est proposée pour réduire la distance parcourue dans une tournée. Étant donné une tournée r et un ensemble de demandes réalisées dans cette tournée, on commence par générer aléatoirement un ordre d'insertion (de manière équiprobable) de demandes de collectes présentes dans r . Suivant l'ordre généré, la requête associée à chaque demande de collecte est réinsérée dans une nouvelle route à la position qui minimise l'insertion de cette dernière. Si toutes les requêtes sont réinsérées avec un meilleur coût, alors on remplace la tournée de départ par cette nouvelle tournée. Cette procédure est répétée un certain nombre de fois. Un exemple de la réalisation d'une itération de la recherche locale $H2$ est donné par la figure 3.4.

Algorithme de recherche locale H3 : réduction de la distance de la solution dans une solution

La troisième recherche locale $H3$ est proposée pour minimiser la distance totale dans une solution. Pour ce faire, à chaque itération, une route r est choisie aléatoirement et dans cette route, une demande d est choisie aléatoirement. La requête associée à la demande d est retirée de la route r . Le couple collecte/livraison associée à la demande d est réinséré dans la solution et donc dans la route et aux positions de cette route qui permettent de minimiser le coût d'insertion. Cette recherche locale est inspirée de l'algorithme de recherche à grand voisinage proposé par Ropke and Pisinger [103]. Un exemple d'itération de cet algorithme est donné

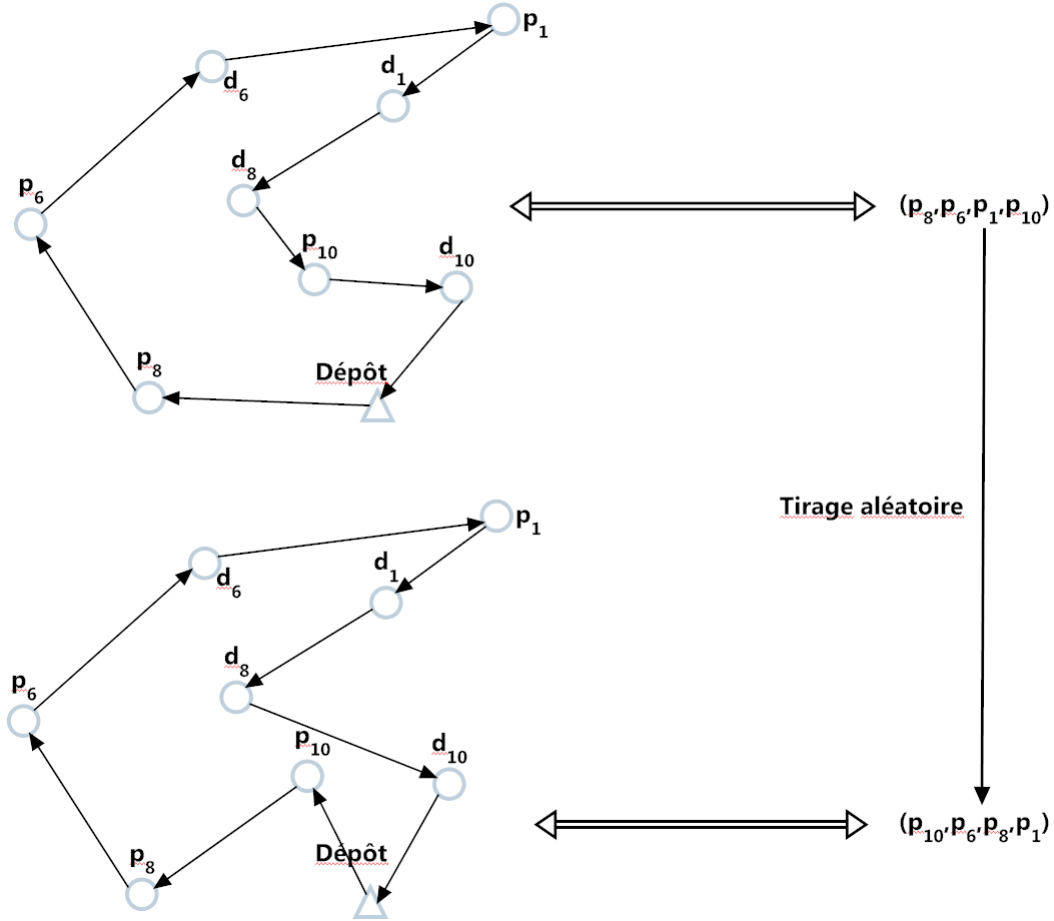


FIGURE 3.4 – Exemple de réduction du coût d'une tournée par l'algorithme H2.

par la Figure 3.5.

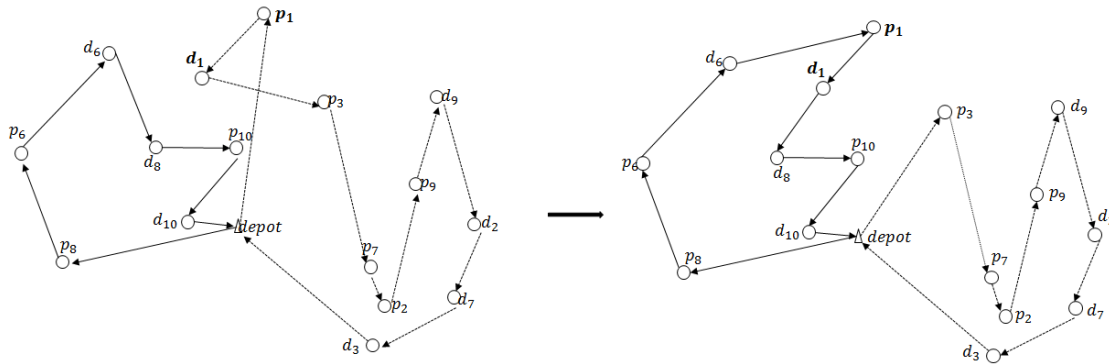


FIGURE 3.5 – Exemple de mouvement réalisé par l'algorithme H3

Le pseudocode de l'algorithme hybride de colonie de fourmi proposé est donné par Algorithme 2. L'ordre d'utilisation des recherches locales a été fixé après plusieurs tests.

Algorithme 2 Pseudo de l'ACO hybride

Initialisation des phéromones

```

tant que La meilleure solution n'est pas améliorée après Itmax2 itérations faire
  pour chaque fourmi de la population faire
    Construire une solution qui satisfait toutes les demandes
    tant que nous pouvons supprimer une route faire
      Appliquer respectivement les algorithmes : H2, H1 et H3
    fin
    si La solution courante a le même nombre de véhicules que la meilleure solution trouvée
      alors
        tant que la distance total décroît faire
          Appliquer respectivement les algorithmes : H2 et H3
        fin
      fin
    Effectuer une mise à jour locale des phéromones
  fin
  Effectuer une mise à jour globale des phéromones
fin

```

3.3.3 Expérimentations

L'algorithme proposé a été implémenté en utilisant l'IDE Eclipse et le langage de programmation C++. L'ordinateur sur lequel les expérimentations ont été menées a un processeur i5 de 1.5 GHz, et d'une mémoire vive de 3.3 GB de RAM.

Pour réaliser les différents tests, nous avons considéré le benchmark de A. Li *et* H. Lim [100], qui est le plus utilisé dans la littérature pour tester les algorithmes sur le problème de collectes et livraisons avec fenêtres de temps et flotte homogène de véhicules.

Analyse de sensibilité

La valeur du paramètre ϵ a été fixé à 0,9 par considération des travaux de Belmecheri et al. [153]. La valeur de chaque paramètre restant a été fixée après une analyse de sensibilité. Pour ce faire, le nombre de fourmis a été pris dans l'ensemble $\{\frac{n}{2}, \frac{2n}{5}, \frac{n}{3}\}$, $\alpha \in \{2; 3; 4\}$, $\beta \in \{1; 2\}$, $\alpha_1 \in \{0, 1; 0, 2; 0, 5; 1\}$, $\alpha_2 \in \{2; 3; 5; 10\}$, $\tau_0 \in \{0, 01; 1; 10\}$, $ItMax1 \in \{5; 10; 20\}$ et $Itmax2 \in \{n; 2n; 3n\}$. Pour les définitions de $ItMax1$ et $ItMax2$, voir respectivement **Algorithme 1** et **Algorithme 2**.

La fixation de ces paramètres a été réalisée en utilisant une stratégie proposée par Ropke and Pisinger [103]. Dans cette dernière, la première phase consiste à utiliser la méthode par essai-erreur pour fixer les valeurs initiales des paramètres. A la seconde phase, les valeurs des paramètres sont ajustées en choisissant et en faisant varier chaque paramètre (pendant

que les autres paramètres gardent une valeur fixe) jusqu'à l'obtention du meilleur ajustement possible de ce paramètre. Ainsi, un à un chaque paramètre est fixé de façon définitive. Pour déterminer la qualité de la valeur d'un paramètre, on résout cinq fois de suite une instance (choisie au préalable au hasard) par l'HACO et on considère la moyenne des valeurs de la fonction objective trouvée.

Nous avons obtenu pour chaque paramètre, la meilleure valeur dans le domaine que nous avons précédemment défini. Le nombre de fourmis est de $\frac{2n}{5}$, $\alpha = 3$, $\beta = 1$, $\alpha_1 = 0,2$, $\alpha_2 = 5$, $\tau_0 = 1$, $ItMax1 = 10$, $Itmax2 = 2n$. L'algorithme s'arrête après 100 itérations non améliorantes.

Résultats expérimentaux

Les résultats expérimentaux (table 3.3, table 3.4) sur les 56 instances à 100 demandes du benchmark de Li and Lim [3] ont montré que la métaheuristique hybride proposée est meilleure que les algorithmes existants jusque-là pour résoudre ce type de problème. En effet, dans 98,2% (55/56) des cas, nous obtenons une solution meilleure ou égale à la meilleure solution connue. Et pour 44,6% (25/56) instances, notre algorithme trouve une solution de meilleure qualité. Ces solutions sont donc de nouvelles meilleures solutions connues pour ces instances.

Meilleure solution connue				Hybrid ACS			
Inst	NV	TD	REF	NV	TD	CPU	
lc101*	10	828,94	[154]	10	828,94	86	
lc102	10	828,94	[3]	10	828,94	103	
lc103	9	1035,35	[87]	9	968,92	96	
lc104	9	860,01	[103]	9	860,01	156	
lc105	10	828,94	[3]	10	828,94	37	
lc106	10	828,94	[3]	10	828,94	43	
lc107	10	828,94	[3]	10	828,94	39	
lc108	10	826,44	[3]	10	826,44	49	
lc109	9	1000,60	[87]	10	827,82	96	
lc201*	3	591,56	[154]	3	591,56	49	
lc202*	3	591,56	[154]	3	591,56	260	
lc203	3	591,17	[103]	3	591,17	190	
lc204	3	590,6	[103]	3	590,39	634	
lc205*	3	588,88	[154]	3	588,88	77	

Meilleure solution connue				Hybrid ACS			
Inst	NV	TD	REF	NV	TD	CPU	
lr112	9	1003,77	[3]	9	1002,38	167	
lr201	4	1253,23	[103]	4	1253,23	207	
lr202	3	1197,67	[3]	3	1197,67	455	
lr203	3	949,4	[3]	3	949,4	556	
lr204	3	849,05	[3]	3	847,83	906	
lr205	3	1054,02	[3]	3	1053,98	134	
lr206	3	931,63	[3]	3	931,63	641	
lr207	2	903,06	[3]	2	902,24	434	
lr208	2	734,85	[3]	2	734,09	1555	
lr209	3	930,59	[103]	3	930,59	234	
lr210	3	964,22	[3]	3	964,22	375	
lr211	2	911,52	[103]	2	903,76	1525	
lrc101	14	1708,8	[3]	14	1708,77	98	
lrc102	12	1558,07	[103]	12	1556,82	49	

Inst est le nom de l'instance, **NV** nombre de véhicules utilisés, **TD** est la distance totale parcourue, **REF** est la référence de l'article qui a obtenu ce résultat et **CPU** est le temps de résolution en *secondes*. Le symbole "*" indique que l'instance a été résolu à l'optimalité et est gras, les valeurs des instances pour lesquelles nous avons trouvé une meilleure solution que celle trouvée jusqu'à présent.

TABLE 3.3 – Résultats des expérimentations de l'HACO sur le Benchmark à 100 demandes de Li and Lim [3].

Meilleure solution connue				Hybrid ACS			
Inst	NV	TD	REF	NV	TD	CPU	
lc206	3	588,49	[3]	3	588,49	144	
lc207	3	588,29	[3]	3	588,29	102	
lc208	3	588,32	[3]	3	588,32	108	
lr101	19	1650,80	[3]	19	1650,8	31	
lr102	17	1487,57	[3]	17	1487,49	54	
lr103	13	1292,68	[3]	13	1284,93	61	
lr104	9	1013,39	[3]	9	999,27	87	
lr105	14	1377,11	[3]	14	1377,11	31	
lr106	12	1252,62	[3]	12	1248,93	37	
lr107	10	1111,31	[3]	10	1101,89	59	
lr108	9	968,97	[3]	9	966,30	59	
lr109	11	1208,96	[103]	11	1208,92	43	
lr110	10	1159,35	[3]	10	1158,27	113	
lr111	10	1108,9	[3]	10	1108,9	74	

Meilleure solution connue				Hybrid ACS			
Inst	NV	TD	REF	NV	TD	CPU	
lrc103	11	1258,74	[3]	11	1256,06	57	
lrc104	10	1128,4	[3]	10	1126,23	98	
lrc105	13	1637,62	[3]	13	1633,56	52	
lrc106	11	1424,73	[103]	11	1424,73	210	
lrc107	11	1230,14	[103]	11	1225,68	58	
lrc108	10	1147,43	[103]	10	1147,96	70	
lrc201	4	1406,94	[103]	4	1396,88	195	
lrc202	3	1374,27	[3]	3	1361,24	153	
lrc203	3	1089,07	[3]	3	1089,07	350	
lrc204	3	818,66	[103]	3	818,66	885	
lrc205	4	1302,2	[3]	4	1302,20	274	
lrc206	3	1159,03	[103]	3	1156,54	135	
lrc207	3	1062,05	[103]	3	1054,24	205	
lrc208	3	852,76	[3]	3	852,76	469	

TABLE 3.4 – Résultats des expérimentations de l'HACO sur le Benchmark à 100 demandes de Li and Lim [3].

3.4 Conclusion

Dans ce chapitre, nous avons traité le problème de collectes et livraisons avec fenêtres de temps et flotte homogène de véhicules. Notre étude a permis d'apporter un certain nombre de contributions à la résolution de cette problématique. Nous avons proposé une formulation mathématique plus compacte du problème. Nous avons aussi proposé des techniques de réduction de l'espace de recherche, et des inégalités valides pour accélérer la résolution du problème. Ces résultats ont permis de résoudre des instances réelles à plus de 100 demandes proposées par le Centre Hospitalier de Troyes. Mais le temps de résolution important (2 heures pour certaines instances) et la difficulté à résoudre des instances de la littérature nous ont amené à développer un algorithme hybride de colonie de fourmis. Pour ce faire, nous avons proposé de nouvelles formules de calcul de la visibilité des fourmis, et aussi de mise à jour des phéromones. Ensuite, des algorithmes de recherches locales ont été proposés pour améliorer les solutions construites par les fourmis. Tout ceci a permis de résoudre les instances à 100 demandes de Li and Lim [3] en obtenant dans 98,2% des cas une solution meilleure ou égale à la meilleure solution connue. Et pour 44,6% instances, notre approche a pu trouver une solution de meilleure qualité. On peut conclure que notre algorithme est meilleur que les algorithmes proposés jusqu'à présent pour résoudre le problème de collectes et livraisons avec fenêtre de temps et flotte homogène de véhicule. Dans le chapitre suivant, nous généralisons les techniques et les méthodes de résolution proposées dans ce chapitre afin de résoudre ce problème dans le cas d'une flotte hétérogène de véhicules.

Chapitre 4

Collectes et livraisons avec fenêtre horaires : flotte hétérogène

4.1 Introduction

Dans le chapitre précédent, nous avons traité du problème de collectes et livraisons avec fenêtre de temps et flotte homogène de véhicules. Dans ce chapitre, nous généralisons les techniques et les méthodes de résolution proposées dans le chapitre 3 pour résoudre le problème de collectes et livraisons dans le cas d'une flotte hétérogène de véhicules. Le caractère hétérogène de la flotte des véhicules est en termes de capacité, de coût fixe d'acquisition (achat ou location) d'un véhicule, et de la consommation de carburant au kilomètre. Nous traitons de l'hétérogénéité de la flotte des véhicules dans deux cas distincts : le cas d'une flotte ayant une taille finie et le cas d'une flotte ayant une taille supposée infinie. Dans le cas d'une flotte de taille finie, on manipule directement les véhicules alors que dans le cas d'une flotte infinie on travaille avec les types de véhicules (chaque type de véhicules est disponible en quantité illimitée). Cette dernière problématique arrive très souvent dans les cas industriels, où les véhicules sont acquis par location. L'objectif visé dans les études que nous avons menées sur ces problématiques est de satisfaire toutes les requêtes avec un coût minimum. Le coût considéré ici est composé : du coût fixe d'acquisition de la flotte de véhicules, et du coût au km multiplié par la distance parcourue nécessaire pour réaliser la solution proposée.

Dans la figure 4.1, un exemple d'une solution du problème de collectes et livraisons avec fenêtres horaires et flotte hétérogène de véhicules composé de 20 demandes (soit 10 requêtes), desservies par 2 véhicules de même type, et d'un autre type véhicule.

Le chapitre est divisé en deux grandes sections. Dans la section 4.2, nous étudions le problème de collectes et livraisons avec fenêtre de temps et flotte hétérogène de véhicules de taille finie. Une modélisation en programme linéaire en variables mixtes est proposée à la sous-section 4.2.1, et une méthode de décomposition de Benders est proposée à la sous-section 4.2.2. Dans

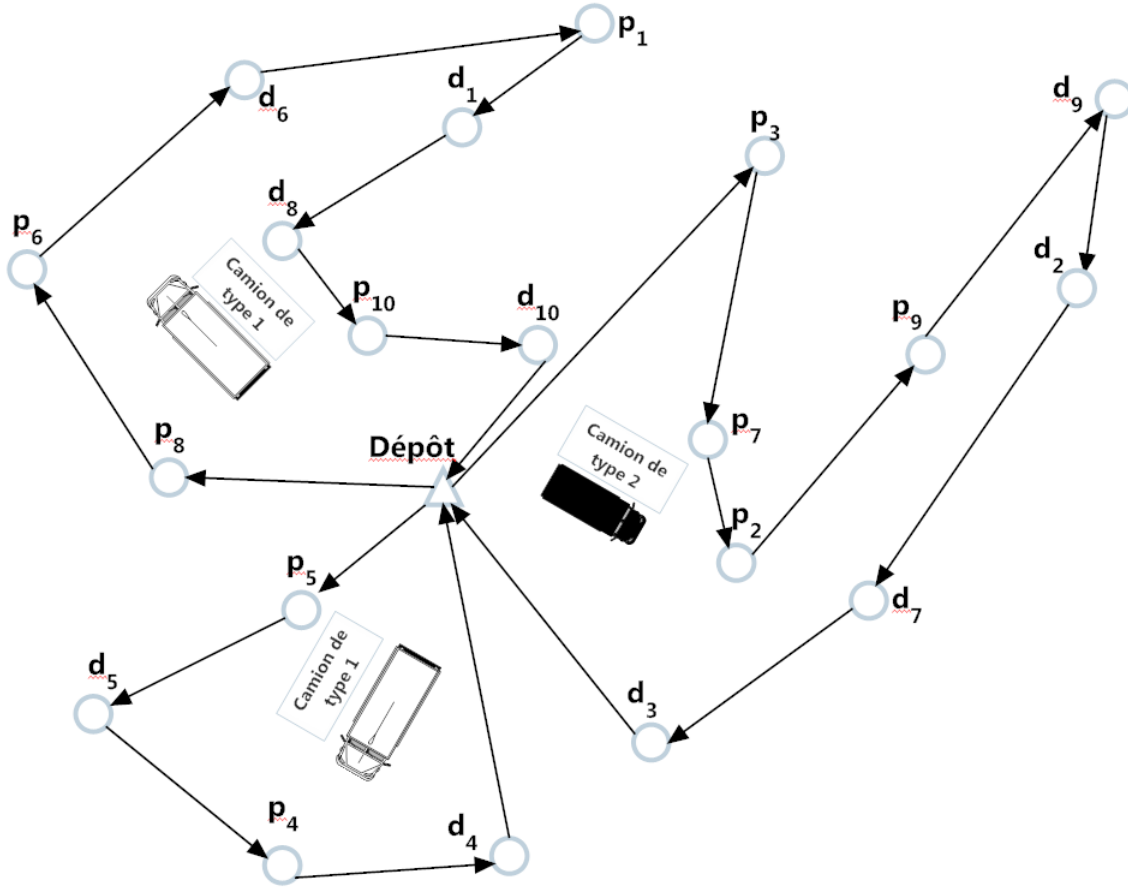


FIGURE 4.1 – Exemple d’une solution de PDPTW avec flotte hétérogène.

la seconde partie, nous traitons du problème de collecte et livraison avec fenêtres horaires et flotte hétérogène de véhicule de taille infinie. Pour ce faire, une modélisation en programme linéaire en nombres entiers est proposée. Nous avons développé un algorithme de génération de colonnes, dont le problème de pricing a été résolu par deux méthodes différentes, et nous avons finalement proposé un algorithme hybride de colonie de fourmis.

4.2 Problème de collectes et livraisons avec fenêtres horaires et flotte hétérogène de véhicules : cas d’une flotte de taille finie

Dans cette section nous modélisons le problème de collectes et livraisons avec fenêtre de temps et flotte hétérogène de véhicules de taille finie. Pour y arriver, nous proposons une modélisation en programme linéaire en variables mixtes. Cette modélisation est une extension du modèle de la section 3.2.1 du chapitre 3. Pour résoudre ce problème, une méthode de décomposition de Benders est proposée.

4.2.1 Programme linéaire en variables mixtes

Afin de présenter ce nouveau modèle pour résoudre le problème de collectes et livraisons avec fenêtres de temps et flotte hétérogène de véhicules de taille finie, nous introduisons de nouvelles notations (Table 4.1) en complément des notations présentes dans la Table 3.1 du chapitre 3.

$\mathcal{K}$	ensemble des véhicules (chaque véhicule ayant ses caractéristiques)
Q^k	capacité d'un véhicule $k \in \mathcal{K}$
δ^k	coût fixe d'un véhicule $k \in \mathcal{K}$
v^k	coût au kilomètre d'un véhicule $k \in \mathcal{K}$

TABLE 4.1 – Notations

Modélisation mathématique

Nous définissons les variables de décision :

- $x_{i,j}^k$: variable binaire de valeur 1 si un véhicule k satisfait la demande i et ensuite j , 0 sinon ;
- B_i^k : variable continue représentant la date de début du service de la demande i par le véhicule k ;
- $Q_{i,j}^k$: nombre de chariots présents dans le véhicule k lorsque ce dernier traverse l'arc (i, j) .

La formulation en programme mathématique linéaire en nombres mixtes est donnée par :

$$(PL4) \quad \text{Min} \sum_{i \in P} \sum_{k \in \mathcal{K}} \delta^k x_{0i}^k + \sum_{(i,j) \in A} \sum_{k \in \mathcal{K}} v^k d_{ij} x_{ij}^k \quad (4.1)$$

$$\sum_{k \in \mathcal{K}} \sum_{(i,j) \in A} x_{ij}^k = 1, \quad \forall i \in P, \quad (4.2)$$

$$\sum_{j|(i,j) \in A} x_{ij}^k = \sum_{j|(n+i,j) \in A} x_{n+i,j}^k, \quad \forall (i, k) \in P \times \mathcal{K}, \quad (4.3)$$

$$\sum_{i \in P} x_{0i}^k \leq 1, \quad \forall k \in \mathcal{K}, \quad (4.4)$$

$$\sum_{i \in P} x_{0i}^k = \sum_{i \in D} x_{i(2n+1)}^k, \quad \forall k \in \mathcal{K}, \quad (4.5)$$

$$\sum_{j|(i,j) \in A} x_{ij}^k \leq \sum_{j \in P} x_{0j}^k, \quad \forall (i, k) \in P \cup D \times \mathcal{K}, \quad (4.6)$$

$$\sum_{j|(i,j) \in A} x_{ij}^k - \sum_{j|(j,i) \in A} x_{ji}^k = 0, \quad \forall (i, k) \in P \cup D \times \mathcal{K} \quad (4.7)$$

$$B_i^k - b_i + (b_i + s_i + t_{ij}) x_{ij}^k \leq B_j^k, \quad \forall (i, j) \in A, \forall k \in \mathcal{K}, \quad (4.8)$$

$$B_i^k - B_{n+i}^k + (s_i + t_{in+i}) \sum_{j \in V} x_{ij}^k \leq 0, \quad \forall (i, k) \in P \times \mathcal{K}, \quad (4.9)$$

$$a_i \sum_{j|(i,j) \in A} x_{ij}^k \leq B_i^k \leq b_i \sum_{j|(ij) \in A} x_{ij}^k, \quad \forall (i, k) \in V \times \mathcal{K}, \quad (4.10)$$

$$\sum_{i \in P} \sum_{k \in \mathcal{K}} Q_{0i}^k + \sum_{i \in D} \sum_{k \in \mathcal{K}} Q_{i(2n+1)}^k = 0, \quad (4.11)$$

$$\sum_{k \in \mathcal{K}} \sum_{j|(i,j) \in A} Q_{ij}^k - \sum_{k \in \mathcal{K}} \sum_{j|(j,i) \in A} Q_{ji}^k = q_i, \quad \forall i \in N, \quad (4.12)$$

$$Q_{ij}^k \leq Q^k x_{ij}^k, \quad \forall (i, j) \in A, \forall k \in \mathcal{K}, \quad (4.13)$$

$$x_{ij}^k \in \{0, 1\}, \quad Q_{ij}^k > 0, \quad \forall (i, j) \in A, \forall k \in \mathcal{K}, \quad (4.14)$$

$$B_i^k > 0, \quad \forall (i, k) \in P \times \mathcal{K}, \quad (4.15)$$

La fonction objective (4.1) est la minimisation du coût fixe d'utilisation des véhicules et du coût kilométrique des tournées. Les contraintes (4.2) et (4.3) modélisent le respect des fenêtres horaires et la précédence de service entre un point de collecte et le point de livraison correspondant. Les contraintes (4.4) imposent aux véhicules de quitter le dépôt au plus une fois, alors que les contraintes (4.5) assurent leur retour au dépôt. Les contraintes (4.6) contraignent les véhicules à quitter le dépôt pour servir une demande. Les contraintes (4.7), (4.8), (4.9) et (4.10) modélisent le remplissage des véhicules tout au long du trajet et assurent que la capacité de chaque véhicule ne soit pas violée. La contrainte (4.11) assure que tout véhicule quitte et retourne au dépôt à vide. Les contraintes (4.12) et (4.13) modélisent le remplissage des véhicules tout au long des tournées et assurent que le nombre de chariots présents dans tout véhicule n'excède pas la capacité du véhicule.

Nous prolongeons le renforcement des bornes des variables, modélisant le temps de début de service (B_i^k) des demandes proposées, par Cordeau [91], au cas d'une flotte hétérogène par :

$$B_i^k \geq a_i + \sum_{j \in N \setminus \{i\}} \max\{0, a_j - a_i + s_j + t_{ij}\} x_{ij}^k, \quad \forall k \in \mathcal{K} \quad (4.16)$$

$$B_i^k \leq b_i - \sum_{j \in N \setminus \{i\}} \max\{0, b_i - b_j + s_i + t_{ij}\} x_{ij}^k, \quad \forall k \in \mathcal{K}. \quad (4.17)$$

Les expérimentations sur cette modélisation, sont présentées à la sous-section 4.2.3.

4.2.2 Décomposition de Benders

Dans cette section, nous proposons de résoudre le problème de collectes et livraisons avec fenêtres de temps et flotte hétérogène de véhicules de taille finie en utilisant la décomposition de Benders.

La décomposition de Benders (Benders [155]) est une méthode de décomposition utilisée en recherche opérationnelle pour résoudre les problèmes d'optimisation linéaire ayant une structure de blocs. Son principe consiste à partir d'un problème restreint issu du problème initial que l'on cherche à résoudre (très souvent par relaxation combinatoire), et de générer et rajouter les contraintes violées par la solution optimale obtenue à chaque itération jusqu'à une condition d'arrêt. Si la condition d'arrêt est que la solution obtenue à une itération ne viole aucune contrainte du problème de départ, alors la solution obtenue est optimale. Procédant par génération de lignes (contraintes), il peut être perçu comme le dual de la méthode de décomposition de Dantzig and Wolfe [156] basée sur la génération de colonnes.

Pour résoudre ce problème, nous réalisons une relaxation combinatoire du problème, en ne considérant pas les contraintes de respect de la capacité des véhicules. On obtient le programme maître (encore appelé programme linéaire restreint) suivant :

$$(Q1) \quad \min \sum_{i \in P} \sum_{k \in \mathcal{K}} \delta^k x_{0i}^k + \sum_{(i,j) \in A} \sum_{k \in \mathcal{K}} v^k d_{ij} x_{ij}^k \quad (4.18)$$

soumis à : (4.1)–(4.10), (4.14) et (4.15).

Dans la suite, nous noterons x^* une solution optimale de (Q).

Algorithme 3 Algorithme de décomposition de Benders

Algorithme :

```

1) Résoudre (Q1) si  $x^*$  est réalisable alors
| 2)  $x^*$  est optimal pour (PL4), stop.
sinon
| tant que il existe une contrainte de capacité violée par  $x^*$  faire
| | 3) Rajouter les inégalités valides de la proposition 3.2.4 correspondantes dans (Q1)
| | 4) Résoudre (Q1)
| fin
|  $x^*$  est une solution optimale de (PL4), stop.
fin

```

L'idée dans Algorithme 3 est de ne pas considérer simultanément le nombre important de contraintes. L'intérêt de cette méthode vient du fait d'avoir capturé une partie de l'information sur les contraintes de précédence dans le graphe grâce aux propositions 3.2.1, 3.2.2, 3.2.3 et 3.2.4 du chapitre 3. De ce fait, générer au fur et à mesure les contraintes de précédence violées apparaît naturel.

4.2.3 Expérimentations

Les tests ont été réalisés en utilisant ILOG CPLEX (version 12.6.0) sur Eclipse. Le langage de programmation a été le C++, sur une machine Core i5 ayant 2,6 GHz et 7,7 GB de RAM.

Les expérimentations ont été réalisées d'une part, sur les instances réelles fournies par le Centre Hospitalier de Troyes (France) et d'autre part sur des instances de la littérature. La plate-forme logistique du centre hospitalier de Troyes comprend une flotte de véhicules de capacités différentes. Les caractéristiques des instances du CHT sont données à la sous-section 3.3.3 du chapitre 3.

Lors de nos expérimentations, nous nous sommes focalisés sur trois aspects :

1. mesurer l'impact des outils que nous avons proposés pour réduire l'espace de recherche,
2. mesurer l'impact des inégalités valides,
3. comparer les performances de la décomposition de Benders au MILP proposé.

Il est important de noter que le temps est donné seulement pour les cas où la solution optimale est obtenue. Dans le cas contraire, la barre “ – ” signifie qu'aucune solution optimale a été trouvée au bout de deux heures de calcul.

Expérimentations des algorithmes (PL4) et (Q1)

Le tableau 4.2 donne les résultats obtenus par les deux méthodes : MILP et décomposition de Benders sur les instances réelles sans réduction de l'espace de recherche et sans inégalités valides.

Instance	PL4(s)	Q1(s)	Gap
<i>Mo</i> _114	–	6286	–
<i>Tu</i> _100	–	–	–
<i>We</i> _106	–	6249	–
<i>Th</i> _92	–	–	–
<i>Fr</i> _114	–	5894	–
<i>Sa</i> _62	24	88	+266 %
<i>Su</i> _52	16	17	+6 %

TABLE 4.2 – Instances réelles sans réduction de l'espace de recherche et sans coupes.

Nous constatons sur cet échantillon que, sans réduire l'espace de recherche et sans générer des coupes, le MILP résout seulement les instances de petite taille (environ 60 demandes). En revanche, la décomposition de Benders résout certaines instances à 100 demandes mais avec un temps de résolution considérable. Nous notons aussi que le MILP est plus rapide que la décomposition de Benders sur les instances de petites tailles. Dans ce cas, les deux méthodes ne résolvent pas à l'optimalité toutes les instances de cet échantillon au bout de deux heures de calcul.

Expérimentations des algorithmes (PL4) et (Q1) avec addition des techniques de réduction de l'espace de recherche

Définissons respectivement PL4_1 et Q1_1 les algorithmes obtenus en ajoutant respectivement à PL4 et au programme maître (Q1) de l'algorithme de Benders les techniques de réduction de l'espace de recherche (sous-section 3.2.3). Les résultats du tableau 4.2 montrent l'impact de la réduction de l'espace de recherche sur le temps de calcul.

Instance	PL4(s)	PL4_1(s)	Q1(s)	Q1_1(s)
<i>Mo</i> _114	–	1011	6286	814
<i>Tu</i> _100	–	–	–	–
<i>We</i> _106	–	168	6249	142
<i>Th</i> _92	–	–	–	3923
<i>Fr</i> _114	–	735	5894	406
<i>Sa</i> _62	24	13	88	13
<i>Su</i> _52	16	6	17	5

TABLE 4.3 – Instances réelles avec réduction de l'espace de recherche et sans coupes.

Les techniques de réduction de l'espace de recherche (voir sous-sections 3.2.2 et 3.2.3) ont permis à (PL4) de résoudre plus d'instances, et d'améliorer le temps de résolution sur les instances qu'il résolvait préalablement. Il en est de même pour la décomposition de Benders. Nous remarquons aussi qu'en réduisant l'espace de recherche, la méthode de décomposition de Benders trouve une solution optimale en moins de temps que le MILP sur toutes les instances résolues par ce dernier.

Expérimentations des algorithmes (PL4) et (Q1) avec addition des techniques de réduction de l'espace de recherche et inégalités valides

Définissons respectivement PL4_2 et Q1_2 les algorithmes obtenus en ajoutant respectivement à PL4_1 et Q1_1 les inégalités valides de la sous-section 3.2.4. Le tableau 4.4 résume les résultats obtenus en réduisant l'espace de recherche et en rajoutant les inégalités valides sur l'échantillon d'instances réelles du CHT. La première observation est le fait que les deux méthodes proposées sont en mesure de trouver la solution optimale pour ces instances en moins de deux heures de calcul. Autre observation, l'algorithme basé sur la décomposition de Benders est en moyenne deux fois plus rapide sur ces instances.

Conclusion

Dans cette première partie, nous nous sommes intéressés au problème de collectes et livraisons avec fenêtre de temps et flotte hétérogène de véhicules de taille finie. Nous avons proposé deux méthodes pour résoudre notre problème : une formulation en modèle linéaire en nombres mixtes, et un algorithme basé sur la décomposition de Benders. Ces approches ont été expéri-

Instance	PL4(s)	PL4_1(s)	PL4_2(s)	Q1(s)	Q1_1(s)	Q1_2(s)
<i>Mo</i> _114	–	1011	458	6286	814	122
<i>Tu</i> _100	–	–	5261	–	–	1659
<i>We</i> _106	–	168	136	6249	142	21
<i>Th</i> _92	–	–	113	–	3923	78
<i>Fr</i> _114	–	735	246	5894	406	269
<i>Sa</i> _62	24	13	8	88	13	5
<i>Su</i> _52	16	6	4	17	5	2

TABLE 4.4 – Instances réelles avec réduction de l’espace de recherche et inégalités valides.

mentées sur les données du Centre Hospitalier de Troyes (environ 100 demandes). Les résultats des expérimentations ont montré que la méthode basée sur la décomposition de Benders est en moyenne deux fois plus rapide que le modèle linéaire proposé. L’étude de l’impact de la réduction de l’espace de recherche, puis de l’ajout des inégalités valides proposées au chapitre 3 montre que ceux-ci permettent de diminuer considérablement le temps de résolution à l’optimalité. Dans la suite, nous traitons du problème de collectes et livraisons avec fenêtre de temps et flotte hétérogène de véhicules de taille infinie.

4.3 Problème de collecte et livraisons avec fenêtre horaires et flotte hétérogène de véhicules : cas d’une flotte de taille infinie

Dans cette section, nous présentons nos travaux sur ce nouveau problème, que nous introduisons, et qui est issu d’un problème rencontré sur le terrain. À notre connaissance ce problème n’a jamais été étudié dans la littérature (voir section 2.3). Pour mener les expérimentations sur ce problème, nous avons proposé un nouvel ensemble d’instances provenant du couplage des instances de Li and Lim [3] (pour les demandes) et de celui de Liu and Shen [157] (pour les différents types de véhicules).

Nous proposons un algorithme de génération de colonnes dans lequel le sous problème de recherche de variable de coût réduit négatif est subdivisé en sous problèmes (un problème pour chaque type de véhicule). Le problème de recherche de variable de coût réduit négatif est un problème de plus court chemin de collecte et livraison avec fenêtre de temps. Des modèles linéaires en nombres entiers sont proposés pour modéliser et résoudre les sous-problèmes. Afin d’accélérer la résolution des sous problèmes, des heuristiques sont développées. Les motivations qui ont abouti à l’utilisation de ces méthodes de résolution seront expliquées dans les prochaines sections.

4.3.1 Modélisation

Dans cette section, nous modélisons le problème de collectes et livraisons avec flotte hétérogène comme un problème de partitionnement. Chaque partition étant une suite de demandes (tournée) pouvant être réalisée par un type de véhicule. Une route débute et se termine au dépôt. La suite des demandes définissant une route est séquencée de sorte à respecter les contraintes de fenêtre de temps, de capacité du véhicule, et de précédence des demandes. Ce dernier type de contrainte est encore appelé contrainte de parité, et exprime le fait que toute demande de collecte est couplée à une demande de livraison telle que les deux demandes sont réalisées dans la même route. Et la demande de collecte doit être satisfaite avant la demande de livraison.

Soit $\mathcal{TK}$ l'ensemble des types de véhicules. Notons $\mathcal{R}_k$ l'ensemble de toutes les routes réalisables pour un type de véhicule $k \in \mathcal{TK}$ et posons $\mathcal{R} = \bigcup_{k \in \mathcal{TK}} \mathcal{R}_k$.

Pour chaque route $r \in \mathcal{R}_k$, le coût de la tournée dû à la distance parcourue est noté c_r^k . Soit $\mathcal{R}_{k_r}$ le sous ensemble des demandes satisfaites par la route $r \in \mathcal{R}_k$.

Soit λ_s une variable binaire telle que :

$$\lambda_s = \begin{cases} 1 & \text{si la route } s \in \mathcal{R} \text{ est intégrée à la solution;} \\ 0 & \text{sinon.} \end{cases} \quad (4.19)$$

Notons que pour tout $s \in \mathcal{R}$, il existe $k_s \in \mathcal{TK}$ et $r_s \in \mathcal{R}_{k_s}$ tel que : $s = r_s$. Un tableau récapitulatif des notations est donné par la table 4.5.

n	nombre de requêtes
$\mathcal{TK}$	ensemble des types de véhicules
M	le nombre de type de véhicules
$\mathcal{R}_k$	ensemble des routes que l'on peut effectuer avec un véhicule de type $k \in \mathcal{TK}$
Q^k	capacité d'un véhicule de type $k \in \mathcal{TK}$
δ^k	coût fixe d'un véhicule de type $k \in \mathcal{TK}$
v^k	coût au kilomètre d'un véhicule de type $k \in \mathcal{TK}$
a_{is}	est une constante de valeur 1 si la demande $i \in P$ est réalisée dans la route $s \in \mathcal{R}$; et 0 sinon
b_{ks}	est une constante de valeur 1 si la route $s \in \mathcal{R}$ est réalisée avec un véhicule de type $k \in \mathcal{TK}$; et 0 sinon
v^k	coût au kilomètre d'un véhicule de type $k \in \mathcal{TK}$
c_s^k	coût dû à la distance parcourue nécessaire par un véhicule de type $k \in \mathcal{TK}$ pour réaliser la route $s \in \mathcal{R}_k$

TABLE 4.5 – Notations

Nous modélisons le problème par :

$$(PL5) \quad \text{Min} \quad \sum_{s \in \mathcal{R}} (\delta^{k_s} + c_{r_s}^{k_s}) \lambda_s \quad (4.20)$$

$$\sum_{s \in \mathcal{R}} a_{is} \lambda_s = 1, \quad \forall i \in P; \quad (4.21)$$

$$\lambda_s \in \{0, 1\}, \quad \forall s \in \mathcal{R}. \quad (4.22)$$

Dans cette formulation, la fonction objective (4.20) minimise la somme totale des coûts fixes d'obtention de la flotte des véhicules et le coût au km multiplié par la distance parcourue. Les contraintes (4.21) imposent que chaque demande de collecte doit être réalisée, et ce dans une seule route présente dans la solution. Cette condition est suffisante pour modéliser le fait que toutes les demandes doivent être satisfaites. En effet, si une demande de collecte est présente dans une route, alors la demande de dépose/livraison correspondante est réalisée dans la même route. Les contraintes (4.22) définissent le domaine de définition des variables de décision. Le programme linéaire (PL5) comporte un nombre exponentiel de variables ($|\mathcal{R}| = |\mathcal{TK}|2^{2n+1}$) et $n = |P|$ contraintes.

Dans la suite, $\forall i \in P$, π_i est la variable duale associée à la contrainte $\sum_{s \in \mathcal{R}} a_{is} \lambda_s = 1$.

Après cette modélisation du problème, dans les prochaines sections, nous allons proposer des méthodes de résolution. Nous proposons une méthode de génération de colonnes, et ensuite un algorithme hybride de colonie de fourmis.

4.3.2 Algorithme de génération de colonnes

L'algorithme de génération de colonnes peut être défini comme un algorithme à plusieurs étapes où à l'étape initial un problème restreint (en terme de variables) est considéré, et à chaque itération des variables qui sont susceptibles d'améliorer la solution courante sont générées et rajoutées au problème restreint. Pour mettre en application l'algorithme de génération de colonnes. Il est donc nécessaire d'avoir une décomposition du problème qui permet de le résoudre itérativement. La technique de décomposition couramment utilisée à cet escient est la décomposition de B. Dantzig et P. Wolfe [158].

La décomposition de B. Dantzig et P. Wolfe [158] permet d'exprimer un programme mathématique comme un ensemble de sous-problèmes indépendants que l'on sait résoudre. On décrit l'espace réalisable de chaque sous problème par l'ensemble des points qui le compose au lieu des équations qui le décrivent. Ces sous problèmes sont reliés entre eux par une contrainte linéaire dite *contrainte couplant linéaire*. Résoudre le problème de départ revient alors à résoudre un problème linéaire qui contient un très grand nombre de variables.

La décomposition de Dantzig-Wolfe permet d'obtenir pour notre problème, une décomposition de la forme :

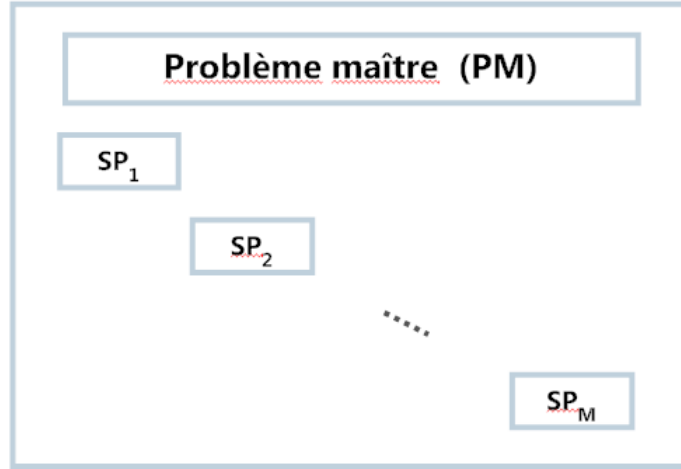


FIGURE 4.2 – Décomposition de Dantzig-Wolfe

Où SP_k (voir le programme linéaire 4.24-4.37) est une modélisation du problème de génération de routes "intéressantes" pour un véhicule de type $k \in \mathcal{TK}$. Autrement dit ce modèle résout le problème de recherche d'une route de coût réduit négatif. Ces sous-problèmes sont appelés problèmes de pricing.

Le processus de détermination du problème de pricing peut être décomposé en trois phases :

1. Faire une relaxation linéaire du programme linéaire ;
2. Déterminer le dual du programme linéaire relaxé ;
3. Extraire des contraintes, l'expression de la fonction objective du problème de pricing.

Un exemple du déroulement du procédé de détermination du problème de pricing appliqué à notre cas d'étude est donné par la figure 4.3.

Notons ($PL5L$) la relaxation linéaire du programme maître. Et si la solution optimale obtenue est entière alors elle est aussi une solution optimale du programme maître d'après Edmonds and Giles [159]. Pour des instances de grande taille (en terme de nombre de demandes), le nombre de variables croît exponentiellement, et il devient impossible d'expliciter toutes les variables. Par exemple dans notre cas d'étude, le nombre de variables est égal au nombre de type de véhicules multiplié par le nombre de routes. Or le nombre de route est de l'ordre du nombre de sous-ensembles (2^{2n}) de l'ensemble. Notons ($PL5L$) la relaxation linéaire du programme maître. Et si la solution optimale obtenue est entière alors elle est aussi une solution optimale du programme maître d'après Edmonds and Giles [159]. Pour des instances de grande taille (en terme de nombre de demandes), le nombre de variables croît exponentiellement, et il devient impossible d'expliciter toutes les variables. Par exemple dans notre cas d'étude, le nombre de variable est égal au nombre de type de véhicules multiplié par le nombre de routes. Or le nombre de route est de l'ordre du nombre de sous-ensembles (2^{2n}) de l'ensemble des demandes. Et cette dernière est exponentielle. La génération de colonnes permet de travailler

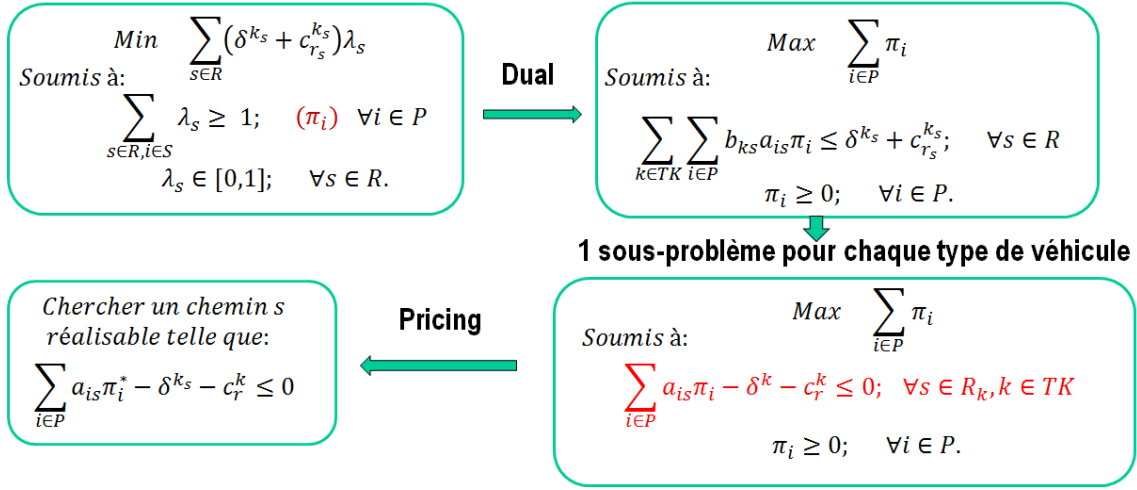


FIGURE 4.3 – Processus de détermination du problème de pricing.

à la résolution du problème sans une énumération totale des variables, d'où l'intérêt de cette méthode de résolution. des demandes. Et cette dernière est exponentielle. La génération de colonnes permet de travailler à la résolution du problème sans une énumération totale des variables, d'où l'intérêt de cette méthode de résolution.

Heuristique

Il arrive très souvent que la résolution du problème de pricing à l'exact prenne du temps. L'objectif du problème de pricing est de trouver une colonne de coût réduit négatif, lorsqu'elle existe. Il est donc plus intéressant par rapport au temps de calcul, de chercher des heuristiques rapides pour déterminer une colonne de coût réduit négatif.

Nous proposons pour résoudre le problème de pricing, d'utiliser un algorithme de type glouton inspiré des heuristiques proposées par Ropke and Cordeau [97]. L'algorithme proposé insère à chaque étape, une requête (couple de collecte et de livraison) qui minimise la différence de coût entre le coût avant et après insertion de cette dernière. L'algorithme s'arrête lorsqu'il n'y a plus de requête à insérer dans la tournée qui permettrait de réduire le coût de la solution après insertion.

Soit $\Delta_{i\rho}$ la différence de coût après insertion de la demande de collecte $i \in P$ et de la demande de livraison associée $n + i$ dans la solution partielle ρ . Si la requête associée à la demande i ne peut être insérée dans ρ alors $\Delta_{i\rho}$ est de valeur infinie. À chaque itération, le choix de la demande de collecte p à insérer (avec la livraison associée) est déterminé par la formule :

$$p = \arg \min_{i \in U} \Delta_{i\rho}, \quad (4.23)$$

où U est l'ensemble des demandes de collectes non réalisées.

Formulation d'un problème de pricing pour un type de véhicule

Dans cette sous-section, nous explicitons le programme linéaire de pricing pour un véhicule de type $k_0 \in \mathcal{TK}$. Afin de modéliser le problème de pricing, un arc $(i, j) \in A$ sera dit réalisable si :

1. $e_i + s_i + t_{ij} \leq l_j$;
2. $\exists k \in \mathcal{TK} / |q_i| \leq Q^k$;
3. $\exists k \in \mathcal{TK} / |q_j| \leq Q^k$;
4. $\exists k \in \mathcal{TK} / q_i + q_j \leq Q^k$;

En effet, un véhicule de type k peut réaliser successivement les demandes i et j si :

- (i) le chemin $(0, i, j)$ est réalisable en terme de fenêtre de temps,
- (ii) un véhicule de type k a assez de capacité pour satisfaire la demande i ,
- (iii) un véhicule de type k a assez de capacité pour satisfaire la demande j ,
- (iv) un véhicule de type k a assez de capacité pour satisfaire successivement les demandes i et j .

Rappellerons que $\forall i \in P$, π_i est la variable duale associée à la contrainte $\sum_{s \in \mathcal{R}} a_{is} \lambda_s = 1$. Les variables de décision nécessaires pour définir le problème de pricing associé au type de véhicule $k_0 \in \mathcal{TK}$ sont données par :

- x_{ij} est une variable binaire de valeur 1 si un véhicule de type k_0 réalise successivement les demandes i et j ; et 0 sinon.
- T_i est le temps auquel débute le service de la demande i ;
- $Q_{i,j}$ est le nombre de chariots présents dans le véhicule lorsqu'il se déplace sur l'arc (i, j) , et 0 s'il ne s'y déplace pas.

$$(SP_{k_0}) \quad \text{Minimiser :} \quad \delta^{k_0} + \sum_{(i,j) \in A} v^{k_0} d_{ij} x_{ij} - \sum_{i \in P, j \in N} \pi_i^* x_{ij} \quad (4.24)$$

Soumis à :

$$\sum_{j: (i,j) \in A} x_{ij} \leq 1, \quad \forall i \in P, \quad (4.25)$$

$$\sum_{j: (i,j) \in A} x_{ij} = \sum_{j: (n+i,j) \in A} x_{n+i,j}, \quad \forall i \in P, \quad (4.26)$$

$$\sum_{i \in P} x_{0i} \leq 1, \quad (4.27)$$

$$\sum_{i \in P} x_{0i} = \sum_{i \in D} x_{i0}, \quad (4.28)$$

$$\sum_{j:(i,j) \in A} x_{ij} = \sum_{j:(j,i) \in A} x_{ji}, \quad \forall i \in P \cup D, \quad (4.29)$$

$$\sum_{j:(i,j) \in A} x_{ij} = \sum_{i \in P} x_{0i}, \quad \forall i \in P, \quad (4.30)$$

$$T_i + (b_i + s_i + t_{ij})x_{ij} \leq T_j + b_j, \quad \forall (i, j) \in A, \quad (4.31)$$

$$\sum_{j:(i,j) \in A} Q_{ij} - \sum_{j:(i,j) \in A} Q_{ji} = q_i \sum_{j:(i,j) \in A} x_{ij}, \quad \forall i \in P \cup D, \quad (4.32)$$

$$Q_{ij} \leq Q^{k_0} x_{ij}, \quad \forall (i, j) \in A, \quad (4.33)$$

$$\sum_{i \in P} Q_{0i} = \sum_{i \in D} Q_{i0}, \quad (4.34)$$

$$T_i + (s_i + t_{i,n+i}) \sum_{j:(i,j) \in A} x_{ij} \leq T_{n+i}, \quad \forall i \in P, \quad (4.35)$$

$$\sum_{j:(i,j) \in A} x_{ij} = 0, \quad \forall i \in P / m_i v^{k_0} \geq \pi_i^*, \quad (4.36)$$

$$x_{ij} \in \{0, 1\}, T_i \in [a_i, b_i], Q_{ij} \in [0, Q_{k_0}] \quad \forall i, j \in N. \quad (4.37)$$

L'objectif (4.24) est de trouver un chemin de coût réduit négatif (voir figure 4.3). Les contraintes (4.25) et (4.26) assurent que chaque demande de collecte est satisfaite au plus une fois et dans le cas où cette dernière est réalisée, la demande de livraison correspondante est satisfaite aussi. Les contraintes (4.27)-(4.30) garantissent que la route débute et se termine au dépôt. Les contraintes (4.31)-(4.34) imposent le respect des fenêtres de temps et la contrainte de capacité du véhicule. Les contraintes (4.35) imposent que toute demande de livraison dans le cas où elle est satisfaite, ne doit l'être qu'après que la demande de collecte correspondante le soit, mais pas forcément à la suite.

Théorème 4.3.1 Soient $i_0 \in P$, $k \in \mathcal{TK}$ et $\pi_{i_0}^*$ la valeur de la variable duale associée à la contrainte de satisfaction de la demande de collecte i_0 (4.21).

$$\text{Si } v^k m_{i_0} > \pi_{i_0}, \quad \text{alors } \sum_{j:(i_0,j) \in A} x_{i_0j} = 0; \quad \text{avec :}$$

$$m_{i_0} = \min_{i,j \in N: (i, i_0, j) \text{ est réalisable}} (d_{i, i_0} + d_{i_0, j} - d_{i, j}) + \min_{i,j \in N: (i, n+i_0, j) \text{ est réalisable}} (d_{i, n+i_0} + d_{n+i_0, j} - d_{i, j}).$$

Preuve 4.3.1 Soit un chemin $r = (w_1, \cdot, w_p, w_{i_0}, w_{p+2}, \cdot, w'_p, w_{n+i_0}, w_{p'+2}, \dots, w_l)$ une tournée réalisable contenant les demandes i_0 et $n+i_0$ respectivement aux positions $p+1$ et $p'+1$. Supposons que $m_{i_0} v^k > \pi_{i_0}^*$, et montrons que le chemin $r_1 = (w_1, \cdot, w_p, w_{p+2}, \cdot, w'_p, w_{p'+2}, \dots, w_l)$ est un chemin réalisable et de meilleur coût. La suppression des demandes dans une tournée

a pour conséquence des arrivées au plus tôt sur les sites de collectes ou de livraisons, et donc n'entraîne pas de violation de fenêtres de temps. De même, la quantité maximum d'items transportés dans le véhicule décroît ou reste constante après suppression d'une requête dans une tournée, donc la contrainte de capacité est respectée. La route r_1 est donc une tournée réalisable. Il reste à montrer que la route r_1 a un coût au moins aussi meilleur que r .

Le coût de la tournée r est :

$$c_r^k = \delta^k + \sum_{i=1}^{l-1} v^k d_{w_i w_{i+1}} - \sum_{\substack{i=1 \\ w_i \in P}}^{l-1} \pi_{w_i}^* \quad (4.38)$$

$$\begin{aligned} &= \delta^k + v^k \left(\sum_{i=1}^{p-1} d_{w_i w_{i+1}} + \sum_{i=p+2}^{p'-1} d_{w_i w_{i+1}} + \sum_{i=p'+2}^{l-1} d_{w_i w_{i+1}} \right) - \sum_{\substack{i=1 \\ w_i \in P}}^{l-1} \pi_{w_i}^* \\ &\quad + v^k (d_{w_p w_{i_0}} + d_{w_{i_0} w_{p+2}} + d_{w_{p'} w_{n+i_0}} + d_{w_{n+i_0} w_{p'+2}}) \end{aligned} \quad (4.39)$$

$$\begin{aligned} &\geq \delta^k + v^k \left(\sum_{i=1}^{p-1} d_{w_i w_{i+1}} + \sum_{i=p+2}^{p'-1} d_{w_i w_{i+1}} + \sum_{i=p'+2}^{l-1} d_{w_i w_{i+1}} \right) - \sum_{\substack{i=1 \\ w_i \in P}}^{l-1} \pi_{w_i}^* + v^k m_{i_0} \\ &\quad (\text{ par définition de } m_{i_0}) \end{aligned} \quad (4.40)$$

$$\begin{aligned} &\geq \delta^k + v^k \left(\sum_{i=1}^{p-1} d_{w_i w_{i+1}} + \sum_{i=p+2}^{p'-1} d_{w_i w_{i+1}} + \sum_{i=p'+2}^{l-1} d_{w_i w_{i+1}} \right) - \sum_{\substack{i=1 \\ w_i \in P, i \neq i_0}}^{l-1} \pi_{w_i}^* - \pi_{w_{i_0}}^* \\ &\quad + v^k m_{i_0} \end{aligned} \quad (4.41)$$

$$\begin{aligned} &> \delta^k + v^k \left(\sum_{i=1}^{p-1} d_{w_i w_{i+1}} + \sum_{i=p+2}^{p'-1} d_{w_i w_{i+1}} + \sum_{i=p'+2}^{l-1} d_{w_i w_{i+1}} \right) - \sum_{\substack{i=1 \\ w_i \in P, i \neq i_0}}^{l-1} \pi_{w_i}^* \\ &\quad (\text{ car } v^k m_{i_0} > \pi_{i_0}^* \text{ par hypothèse }) \end{aligned} \quad (4.42)$$

$$= c_{r'}^k \quad (\text{ par définition de }) r' \quad (4.43)$$

Donc le coût de la route r est strictement supérieur à celui de la route r' . Ainsi, toute solution optimale du problème de pricing à cette étape ne contient pas la demande i_0 . D'où $\sum_{j:(i_0,j) \in A} x_{i_0 j} = 0$.

Ce résultat donne un critère pour ne pas considérer les requêtes dont la valeur de la variable duale n'est pas intéressante comparée au coût minimum nécessaire pour insérer cette requête

dans une tournée. Ce fait permet de réduire les variables de décisions du problème de pricing et donc d'accélérer la résolution de cette dernière. Autre fait intéressant, ce résultat est valable pour toute méthode de résolution employée pour résoudre le problème de pricing.

Comme nous le verrons dans les expérimentations, la résolution du problème de pricing en utilisant le Solveur ILOG CPLEX n'est pas en mesure de résoudre plus de 50% des instances que nous avons proposées au bout de 3 heures de calcul.

Dans la suite, nous avons proposé un algorithme d'étiquetage pour résoudre le problème de pricing pour chaque type de véhicule. Ceci dans l'objectif de construire un algorithme de meilleure qualité en terme de qualité de solutions obtenues et/ou temps de résolution.

Le schéma de l'algorithme de génération de colonnes dont les problèmes de pricing sont résolus avec le modèle linéaire en nombres mixtes est donné par la figure 4.4.

Algorithme à labels

Dans cette partie, un nouvel algorithme à labels (ou étiquettes) est proposé pour résoudre les problèmes de pricing. Nous rappelons que dans notre cas d'étude, le problème de pricing est la recherche d'un chemin de coût réduit négatif. L'algorithme à labels est un algorithme d'énumération totale, et donc une méthode de résolution exacte. L'idée de cet algorithme est d'évaluer intelligemment tous les chemins possibles du graphe des demandes. Pour accélérer les processus de génération et d'évaluation des chemins, un critère de dominance est utilisé. L'algorithme prend en entrée un graphe orienté pondéré par des réels et un sommet source. Il fonctionne en construisant un sous-graphe de manière à ce que le coût entre un sommet du sous graphe et le sommet source soit connu et soit minimum dans le graphe. Initialement, le sous graphe ne contient que le sommet source et donc un label. A chaque étape, un label est choisi, étendu (parcours en largeur) lorsque ce dernier a des voisins, pour créer de nouveaux labels. Ce label est ensuite supprimé de l'ensemble des labels. C'est ce processus que l'on appellera dans la suite traitement d'un label. L'algorithme se termine lorsque tous les labels ont été traités.

Chaque label L est constitué de :

- η_L le dernier nœud dans la tournée partielle,
- t_L le temps de fin de service du dernier nœud de la tournée en construction,
- qt_L le remplissage du véhicule après le service du dernier nœud de la tournée en construction,
- c_L le coût de la tournée après le service du dernier nœud de la tournée en construction,
- $\mathcal{D}_L$ l'ensemble des demandes de livraisons non encore satisfaites, mais dont les demandes de livraisons ont déjà été réalisées dans la tournée en construction,

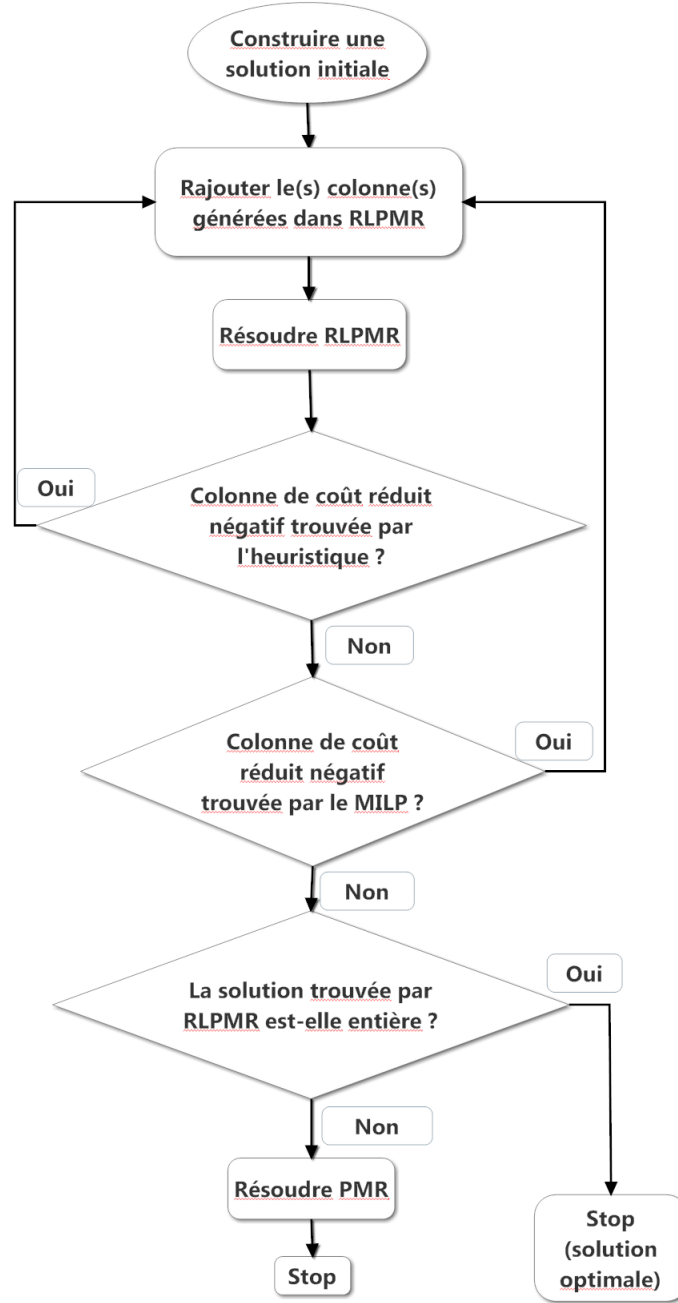


FIGURE 4.4 – Algorithme de génération de colonnes résolut avec le modèle linéaire en nombres mixtes.

- $\mathcal{P}_L$ l'ensemble des demandes de collectes satisfaites, et dont les livraisons correspondantes ont été réalisées dans la tournée en construction.

Un label L peut être étendu par rapport à l'arc (η_L, p_j) (où p_j est une demande de collecte) si les conditions suivantes sont satisfaites :

1. $(\eta_L, p_j) \in A$ et $p_j \notin \mathcal{P}_L$,

2. $qt_L + q_{p_j} \leq Q^k$ (pour le problème de pricing associé au type de véhicule $k \in \mathcal{TK}$).

Un label L peut être étendu par rapport à l'arc (η_L, d_j) (où d_j est une demande de livraison) si les conditions suivantes sont satisfaites :

$$(\eta_L, d_j) \in A \text{ et } d_j \in \mathcal{D}_L. \quad (4.44)$$

Afin d'accélérer la résolution du problème, nous pouvons supprimer des labels non pertinents en utilisant des propriétés de dominance.

Proposition 4.3.1 *Un label L domine un label L' si :*

1. $\eta_L = \eta_{L'}$,
2. $T_L \leq T_{L'}$,
3. $c_L \leq c_{L'}$,
4. $\mathcal{P}_L \subseteq \mathcal{P}_{L'}$,
5. $\mathcal{D}_L \subseteq \mathcal{D}_{L'}$.

Cet algorithme permet de résoudre à l'optimalité le problème de pricing pour chaque type de véhicule. Comme nous le verrons dans les expérimentations, il est plus performant en temps de calcul que le modèle linéaire en nombres mixtes.

Afin de générer plus rapidement des colonnes ayant des coûts réduits négatifs à chaque itération de l'algorithme de génération de colonnes, nous proposons dans la suite, une heuristique inspirée de l'algorithme glouton (voir [160]).

Le schéma de l'algorithme de génération de colonnes dont les problèmes de pricing sont résolus avec l'algorithme à labels est donné par la figure 4.5.

Expérimentations

Les expérimentations menées sur les méthodes de résolution ont été réalisées sur l'environnement de développement Eclipse. Le langage de programmation utilisé a été Java, sur un ordinateur ayant un processeur avec une fréquence d'horloge de : 2,6 GHz et 8 Go de RAM.

Afin de mesurer la performance de la méthode de génération de colonnes proposée, nous avons élaboré un nouveau benchmark, obtenu en couplant le benchmark de Li and Lim [3] (benchmark de Solomon [161] adapté) pour les demandes, avec celui de Liu and Shen [157] pour les véhicules. En effet, Liu and Shen [157] ont proposé des types de véhicules compatibles aux instances de tournées de véhicules de Solomon ([161]). Dans la suite de ce chapitre, nous noterons ce nouveau benchmark *FLT100*.

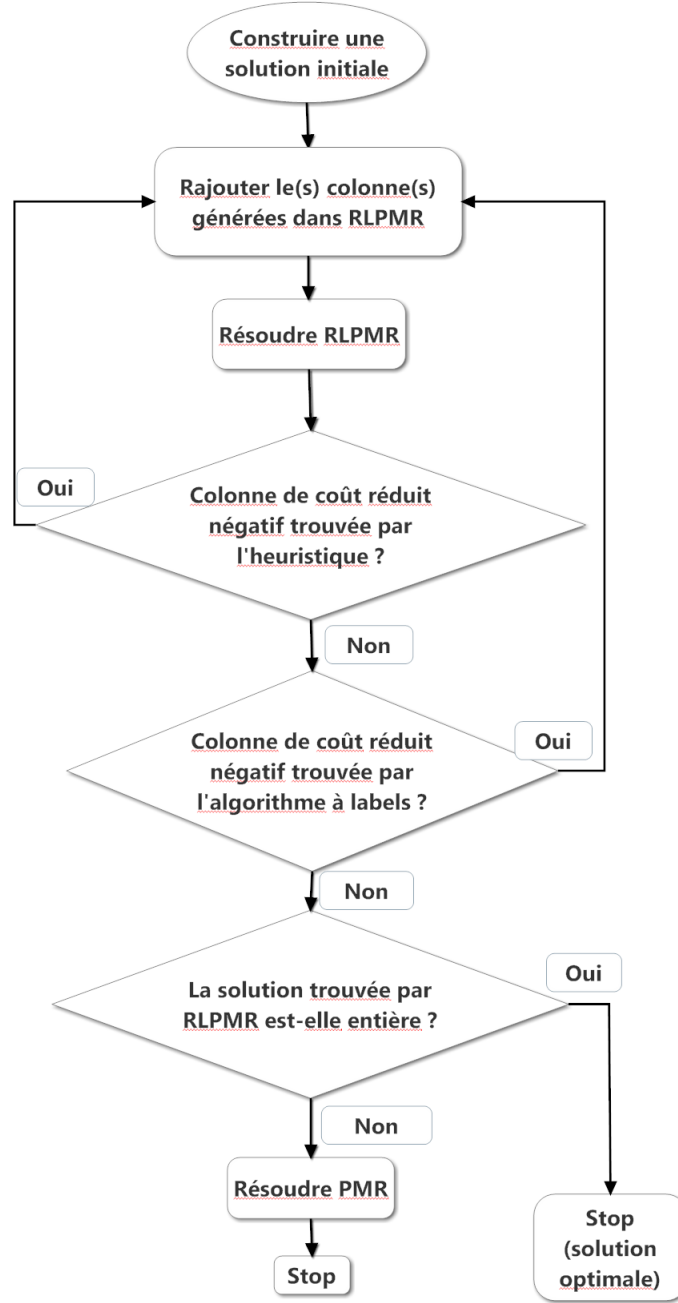


FIGURE 4.5 – Algorithme de génération de colonnes résolu avec l'algorithme à labels.

Les résultats des expérimentations sont donnés dans les tables 4.6-4.8. Dans ces tables, **Inst** est le nom de l'instance, **Lab** est le résultat obtenu en utilisant l'algorithme à labels pour résoudre le problème de pricing, **LB** est la meilleure borne inférieure obtenue pendant la résolution de l'instance par la méthode *Lab*. **SP** est le résultat obtenu en utilisant le programme linéaire (défini par 4.24-4.37) pour résoudre le problème de pricing. Les instances dont le nom est en **gras** sont les instances qui ont pu être résolues à l'optimalité. La meilleure solution entre

l'approche par génération de colonnes résolue avec le programme linéaire et celle résolue avec l'algorithmes à labels est indiqué en italique.

G est le gap entre la solution obtenue et la borne inférieure, calculé en utilisant la formule :

$$G = 100 * \frac{LAB - LB}{LB} \quad (4.45)$$

D'après les expérimentations (tables 4.6-4.8), l'algorithme *LAB* est meilleur en terme de temps de calcul que l'algorithme *SP*. De plus, l'algorithme *LAB* permet d'obtenir une borne inférieure pour 98% des instances. En moyenne, le gap entre la meilleure solution trouvée et la borne inférieure obtenue par l'algorithme *LAB* est de 12% (pour les instances dont on a une borne inférieure).

Inst	Coût		LB	Temps(min)		G%
	Lab	SP		Lab	SP	
lc101	<i>2057,26</i>	2064,95	2014	1	5	2
lc102	<i>2241,81</i>	2296,28	2187	35	57	5
lc103	<i>1800,61</i>	1813,87	1754	62	105	3
lc104	<i>1642,48</i>	1642,48	1497	95	120	9
lc105	<i>2108,24</i>	2108,24	2106	17	35	1
lc106	<i>2100,34</i>	2106,23	2061	56	81	2
lc107	<i>2247,07</i>	2247,07	2168	53	120	4
lc108	<i>2078,23</i>	2087,66	2074	75	120	1
lc109	<i>1702,41</i>	1702,41	1684	72	108	1

Inst	Coût		LB	Temps(min)		G%
	Lab	SP		Lab	SP	
lc201	<i>1950,49</i>	1967,06	1762	63	120	10
lc202	<i>2076,02</i>	2575,95	1908	96	120	8
lc203	<i>1972,22</i>	2390,52	1639	133	120	17
lc204	<i>1800,37</i>	2210,03	1449	60	120	20
lc205	<i>1451,86</i>	1792,03	1390	61	120	4
lc206	<i>2083,75</i>	2144,84	1746	92	120	16
lc207	<i>1531,84</i>	1752,52	1415	60	120	8
lc208	<i>1509,06</i>	1817,73	1417	90	120	6

TABLE 4.6 – Résultats obtenus sur les instances avec une distribution géographique groupée des demandes.

Inst	Coût		LB	Temps(min)		G%
	Lab	SP		Lab	SP	
lr101	3050,76	3050,76	3050	0,1	0,1	0
lr102	2713,17	2713,17	2713	22	42	0
lr103	2343,15	2223,84	2223	31	84	0
lr104	<i>1733,56</i>	1733,56	1730	86	120	0,2
lr105	<i>2649,53</i>	2651,06	2613	0,1	1	1,4
lr106	<i>2343,66</i>	2343,66	2335	5	25	0,3
lr107	1837,26	1837,26	1837	52	68	0
lr108	1775,42	1775,42	1775	60	78	0
lr109	<i>2227,15</i>	2256,99	2181	4	18	3,3
lr110	<i>2217,79</i>	2217,79	1995	23	52	10
lr111	<i>1936,11</i>	1936,11	1867	38	87	3,5
lr112	<i>1881,31</i>	1881,31	-	120	120	-

Inst	Coût		LB	Temps(min)		G%
	Lab	SP		Lab	SP	
lr201	<i>2340,53</i>	2340,53	2139	61	120	9
lr202	<i>2823,71</i>	2823,71	2053	64	120	27
lr203	3181,84	<i>3037,15</i>	1895	80	120	40
lr204	<i>2848,01</i>	3163,52	1704	60	120	40
lr205	<i>2635,37</i>	3042,47	1999	60	120	24
lr206	3339,36	<i>3182,30</i>	2100	77	120	37
lr207	<i>2692,2</i>	3250,98	1601	106	120	41
lr208	<i>2600,08</i>	3016,7	1601	60	120	38
lr209	<i>3105,56</i>	3133	1921	82	120	38
lr210	<i>2692,60</i>	3133,29	1915	84	120	29
lr211	<i>3098,91</i>	3197	1624	60	120	48

TABLE 4.7 – Résultats obtenus sur les instances avec une distribution géographique uniforme des demandes.

Inst	Coût		LB	Temps(min)		G%
	Lab	SP		Lab	SP	
lrc101	2841,29	2841,29	2841	1	0,1	0
lrc102	2546,89	2546,89	2537	6	19	0,4
lrc103	2172,36	2172,36	2114	31	99	2,6
lrc104	1920,75	1920,75	1890	91	109	1,6
lrc105	2774,13	2774,13	2748	1	2	0,9
lrc106	2619,42	2619,42	2522	1	3	3,7
lrc107	2219,31	2219,31	2183	6	19	1,6
lrc108	2038,63	2038,63	2014	31	100	1,2

Inst	Coût		LB	Temps(min)		G%
	Lab	SP		Lab	SP	
lrc201	2850,87	2850,87	2614	61	120	9
lrc202	3173,58	3025,69	2505	60	120	21
lrc203	3511,22	3187,89	2134	73	120	39
lrc204	2954,55	2729,7	1719	60	120	42
lrc205	2903,55	3120,42	2426	63	120	16
lrc206	2578,4	2990,31	2261	62	120	12
lrc207	2874,87	2874,87	2265	74	120	26
lrc208	3581,08	3794,06	2121	71	120	41

TABLE 4.8 – Résultats obtenus sur les instances avec une distribution géographique semi-groupée des demandes.

4.3.3 Algorithme hybride de colonie de fourmis

Dans cette section, nous proposons une métaheuristique basée sur une hybridation de l'algorithme de colonie des fourmis avec des algorithmes de recherches locales ciblés. Cette approche fait suite au succès de cette dernière sur le problème de collectes et livraisons dans le cas d'une flotte homogène comme nous l'avons vu au chapitre 3.

La métaheuristique proposée dans cette section étant une extension du cas d'une flotte hétérogène de véhicules, nous ne reviendrons pas en détail sur un certain nombre d'éléments précédemment énoncés à la section 3.3.1.

Dans toute cette section, nous considérerons les notations de la section 3.3. Dans la suite nous décrirons brièvement : le processus de construction d'une solution et les algorithmes de recherches locales permettant d'améliorer les solutions construites par les fourmis. Les résultats des expérimentations que nous avons effectuées seront ensuite proposés. Nous verrons dans les résultats de ces expérimentations, l'impact des recherches locales sur la qualité des solutions retournées.

Construction d'une solution

Chaque fourmi construit une solution entière et réalisable. Une solution est un ensemble de tournées, chaque tournée étant réalisée par un véhicule d'un type précis. Les tournées formant une solution sont construites de manière séquentielle par une fourmi. La fourmi choisit initialement le véhicule ayant la plus grande capacité, et suivant la valeur des phéromones et la visibilité (voir la section 3.3.1) acquise, elle choisit une demande de livraison non encore satisfait, et insère le couple collecte/livraison correspondant. À chaque itération, la fourmi va choisir la prochaine demande à satisfaire, tout en vérifiant que toutes les contraintes sont satisfaites, et va ainsi construire une tournée.

Lorsque la fourmi n'arrive plus à trouver une demande de livraison à effectuer et qu'il y a des demandes de collectes non réalisées, alors la fourmi crée une nouvelle tournée avec un

nouveau véhicule. L'algorithme s'arrête lorsque toutes les demandes ont été satisfaites. Dans la suite, cette métaheuristique sera notée *HACO*.

Soit ρ une solution partiellement construite par une fourmi. Soit r_ρ la dernière route construite (ou en construction) dans la solution partielle ρ . Notons k_{r_ρ} le type de véhicule utilisé pour réaliser la route r_ρ , i_{r_ρ} la dernière demande satisfaite dans la route r_ρ et q_{r_ρ} le nombre d'items présents dans le véhicule après la réalisation de la demande i_{r_ρ} dans la solution partielle ρ . Soit $\mathcal{V}_\rho$ l'ensemble des demandes de livraisons non encore réalisées, mais dont les demandes de collectes correspondantes l'ont déjà été.

Une demande j sera éligible à sa réalisation par la fourmi après la demande i_{r_ρ} si :

- (i) $j = p_l \in P$ et il existe un chemin réalisable pour satisfaire toutes les demandes présentes dans l'ensemble $\mathcal{V}_\rho \cup \{d_l\}$ avec un véhicule de type k_{r_ρ} ,
- (ii) $j = d_l \in D \wedge p_l \in r$ et il existe un chemin réalisable pour satisfaire toutes les demandes présentes dans l'ensemble $\mathcal{V}_\rho \setminus \{j\}$ avec un véhicule de type k_{r_ρ} .

Les formules de calcul de probabilité de choix des demandes, de visibilité, d'initialisation des phéromones et de leur mise à jour sont les mêmes que celles de la section 3.3.1. Dans cet algorithme, nous reprendrons aussi les techniques de réduction de l'espace de recherche, d'insertion des requêtes et les algorithmes de recherche locale décrit au chapitre 3.

Le pseudo-code de l'algorithme hybride des colonies de fourmis est donné à l'Algorithme 4. Dans ce pseudo-code, les algorithmes sont intégrés tels que, dans la première phase on fait décroître le coût de la solution dû à la flotte de véhicules utilisés, et dans une seconde phase, on fait décroître la valeur de la somme des distances des tournées de la solution. Le choix de l'ordre d'utilisation des algorithmes de recherches locales a été fixé après de nombreuses expérimentations.

À titre de rappel, l'heuristique **H1** est utilisée pour réduire le nombre de routes dans une solution (voir section 3.3.2). L'heuristique **H2** a été proposée pour réduire la distance totale dans une tournée (voir section 3.3.2). Et l'heuristique **H3** est utilisée pour faire décroître la valeur de la somme des distances des tournées de la solution (voir section 3.3.2). Les phases de mise à jour locale et globale des phéromones sont explicitées à la section 3.3.1.

Expérimentations

Les expérimentations sur l'HACO ont été menées en deux phases :

1. comparaison avec l'algorithme de décomposition de Benders,
2. mesure de l'impact des heuristiques de recherche locale sur la qualité de résolution.

Les expérimentations ont été menées dans les mêmes conditions que les algorithmes de générations de colonnes proposés à la section 4.3.2.

Algorithme 4 Pseudo code de l'algorithme HACO

```

tant que la meilleure solution n'est pas améliorée après un nombre d'itérations donné faire
    pour chaque fourmi de la population faire
        Construire une solution satisfaisant toutes les demandes
        tant que le coût de la solution courante décroît faire
            Appliquer respectivement les algorithmes H2 et H3
            Pour chaque route de la solution, utiliser le véhicule de coût fixe minimum pouvant
            satisfaire toutes les demandes de cette route
            Appliquer l'algorithme H1
        fin
        Effectuer une mise à jour locale des phéromones
    fin
    Effectuer une mise à jour globale des phéromones
fin

```

Afin de mesurer la qualité de l'algorithme *HACO*, nous l'avons comparé avec la méthode exacte de décomposition de Benders proposée à la sous-section 4.2.2. Pour ce faire, nous sommes partis du Benchmark proposé à la section 4.3.2, et avons conçu un Benchmark *T20* formé des instances à 20 demandes. Chaque instance de ce Benchmark est obtenu en prenant les 10 premières requêtes dans l'ordre d'apparition de ces dernières dans le Benchmark *T20*.

Nous noterons pour la suite :

- *T20C1* les instances du benchmark *T20* ayant une distribution géographique groupée des demandes, et des fenêtres horaires de faible amplitude.
- *T20C2* les instances du benchmark *T20* ayant une distribution géographique groupée des demandes, et des fenêtres horaires de grande amplitude.
- *T20R1* les instances du benchmark *T20* ayant une distribution géographique uniforme des demandes, et des fenêtres horaires de faible amplitude.
- *T20R2* les instances du benchmark *T20* ayant une distribution géographique uniforme des demandes, et des fenêtres horaires de grande amplitude.
- *T20RC1* les instances du benchmark *T20* ayant une distribution géographique semi-groupée des demandes, et des fenêtres horaires de faible amplitude.
- *T20RC2* les instances du benchmark *T20* ayant une distribution géographique semi-groupée des demandes, et des fenêtres horaires de grande amplitude.

Pour déterminer une valeur robuste de chaque paramètre, nous avons utilisé la même stratégie (inspirée de [103]) que celle présentée à la section 3.3.3 du chapitre 3. Les valeurs des paramètres obtenus pour l'algorithme *HACO* sont spécifiées dans la table 4.9.

L'algorithme de décomposition de Benders a été proposé pour résoudre le PDPTW dans le cas d'une et flotte hétérogène de véhicule de taille finie. Afin de pouvoir l'adapter pour la

Paramètres	α_1	α_2	α_3	α_4	ϵ_1	ϵ_2	ϵ_3	π_0	τ_0
Valeur	3	3	0,1	5	0,9	0,9	0,9	$\frac{1}{\max_{k \in \mathcal{TK}} f^k}$	1

TABLE 4.9 – Paramètres de l'HACO

résolution d'une flotte de taille infinie, il est nécessaire pour chaque instance d'estimer une borne supérieure sur le nombre de véhicules (de chaque type de véhicules) pouvant être utilisé. Ce qui garantit l'optimalité de la solution trouvée.

Étant donné une instance I (ensemble de demandes) et un type de véhicule k , une procédure pour déterminer une borne supérieure sur le nombre de véhicules utilisables est donnée par :

1. Retirer de l'instance I les demandes qui ne peuvent pas être satisfaite par un véhicule de type k .
2. Pour l'instance obtenu I' , en utilisant l'HACO, la résoudre en considérant uniquement une flotte de véhicules de type k .

Cette procédure permet d'obtenir une borne supérieure sur le nombre de véhicules de chaque type de véhicule.

Pour les expérimentations, nous avons fixé le temps de résolution de l'algorithme de décomposition de Benders à 600 secondes, vu la tailles des instances (20 demandes). Comme le montre la table 4.10, l'algorithme de décomposition de Benders dans la limite de temps fixé, trouve une solution optimale dans 76% des instances ayant une distribution géographique groupée. Pour toutes les instances pour lesquelles l'algorithme de décomposition de Benders trouve une solution, l'HACO trouve une solution de même coût, donc trouve une solution optimale. Il est aussi important de noter que toujours d'après les résultats obtenus, l'algorithme HACO surpasse l'algorithme de décomposition de Benders en temps de résolution.

Pour les instances des benchmarks $T20R1$ et $T20R2$, les résultats de la table 4.12 montrent que l'algorithme de décomposition de Benders trouve dans 83% des instances de $T20R1$, et dans 18% des instances de $T20R2$ une solution optimale (dans la limite de temps fixé). Pour ces instances dont on connaît la valeur d'une solution optimale, l'HACO trouve dans 100% des cas une solution optimale. On remarque que dans la configuration d'une distribution uniforme, les instances ayant de larges fenêtres de temps prennent plus de temps de résolution.

Les résultats des expérimentions sur les instances des benchmarks $T20RC1$ et $T20RC2$ sont données dans la table 4.11. L'algorithme de décomposition de Benders trouve une solution optimale dans 100% des cas dans la limite de temps fixé. L'HACO trouve aussi une solution optimale pour 100% des instances des benchmarks $T20RC1$ et $T20RC2$. On remarque aussi que l'HACO met plus de temps pour résoudre les instances ayant de petites fenêtres de temps ($T20RC1$), que celles ayant de grandes fenêtre de temps ($T20RC2$).

Instance	Décomposition de Benders		HACO		
	Coût	Temps(s)	Coût	Gap(%)	Temps(s)
T20C1_1	375,502	0	375,502	0	1
T20C1_2	-	600	425,191	-	6
T20C1_3	347,589	271	347,589	0	4
T20C1_4	277,147	296	277,147	0	6
T20C1_5	375,502	0	375,502	0	2
T20C1_6	435,502	2	435,502	0	2
T20C1_7	435,502	9	435,502	0	2
T20C1_8	-	600	357,602	-	2
T20C1_9	-	600	363,755	-	4
T20C2_1	464,809	0	464,809	0	3
T20C2_2	553,401	5	553,401	0	4
T20C2_3	400,569	178	400,569	0	16
T20C2_4	-	600	460,165	-	13
T20C2_5	443,262	64	443,262	0	5
T20C2_6	445,277	1	445,277	0	5
T20C2_7	424,293	39	424,293	0	8
T20C2_8	402,698	14	402,698	0	8

TABLE 4.10 – Résultats sur le benchmark de $T20C1$ et $T20C2$

Instance	Décomposition de Benders		HACO		
	Coût	Temps(s)	Coût	Gap(%)	Temps(s)
T20RC1_1	720,017	1	720,017	0	2
T20RC1_2	672,053	18	672,053	0	4
T20RC1_3	514,163	133	514,163	0	3
T20RC1_4	460,109	99	460,109	0	3
T20RC1_5	654,544	15	654,544	0	4
T20RC1_6	644,314	53	644,314	0	4
T20RC1_7	367,902	578	367,902	0	3
T20RC1_8	362,032	284	362,032	0	6
T20RC2_1	-	600	548,903	-	5
T20RC2_2	-	600	631,881	-	7
T20RC2_3	-	600	669,825	-	16
T20RC2_4	-	600	459,842	-	28
T20RC2_5	618,999	7	618,999	0	5
T20RC2_6	667,514	34	667,514	0	4
T20RC2_7	-	600	567,792	-	13
T20RC2_8	-	600	587,277	-	10

TABLE 4.11 – Résultats sur les benchmarks $T20RC1$ et $T20RC2$.

Instance	Décomposition de Benders		HACO		
	Coût	Temps(s)	Coût	Gap(%)	Temps(s)
T20R1_1	983,048	0	983,048	0	3
T20R1_2	602,738	4	602,738	0	3
T20R1_3	488,182	194	488,182	0	4
T20R1_4	-	600	556,369	-	4
T20R1_5	626,519	0	626,519	0	2
T20R1_6	579,668	5	579,668	0	3
T20R1_7	495,435	364	495,435	0	4
T20R1_8	-	600	465,521	-	4
T20R1_9	493,609	2	493,609	0	3
T20R1_10	553,631	185	553,631	0	3
T20R1_11	586,264	104	586,264	0	2
T20R1_12	475,934	409	475,934	0	4
T20R2_1	526,206	3	526,206	0	4
T20R2_2	508,161	312	508,161	0	7
T20R2_3	-	600	474,038	-	19
T20R2_4	-	600	506,428	-	20
T20R2_5	-	600	527,247	-	7
T20R2_6	-	600	448,97	-	12
T20R2_7	-	600	437,7	-	19
T20R2_8	-	600	437,12	-	31
T20R2_9	-	600	466,668	-	14
T20R2_10	-	600	547,688	-	6
T20R2_11	-	600	415,882	-	19

TABLE 4.12 – Résultats sur les benchmarks *T20R1* et *T20R2*.

Les résultats sur le benchmark *T20* montrent que dans tous les cas où on a eu une preuve de l’optimalité de la solution fournie par l’algorithme de décomposition de Benders, l’algorithme HACO a trouvé une solution de même coût (et donc optimale).

Impact des algorithmes de recherche locale

Afin de montrer l’impact de l’ajout des recherches locales sur l’algorithme HACO, nous avons réalisé les expérimentations sur les versions de l’HACO suivantes :

- l’algorithme de colonie de fourmis couplé à la recherche locale *H1*,
- l’algorithme de colonie de fourmis couplé aux recherches locales *H1* et *H2*,
- l’algorithme de colonie de fourmis couplé aux recherches locales *H1*, *H2* et *H3* (HACO).

Les tables 4.16-4.19 présentent les résultats des expérimentations menées. Et la table 4.13 résume les résultats obtenus.

Instances	ACO +H1		ACO+H1+H2		ACO+H1+H2+H3	
	CM	TM (s)	CM	TM (s)	CM	TM (s)
FLT100C1	5176,2	103,6	2431,1	129,5	2089,6	233,3
FLT100C2	5369,9	621,0	2116,7	813,8	1641,4	907,7
FLT100R1	4476,1	131,0	2536,0	165,1	2257,0	260,7
FLT100R2	3649,2	1464,9	2280,3	1717,8	1998,3	2264,9
FLT100RC1	4981,1	106,8	2720,4	126,2	2500,5	229,3
FLT100RC2	4684,0	836,1	2765,7	687,1	2300,2	1370,7

CM := Coût moyen ; TM := Temps moyen.

TABLE 4.13 – Résumé des expérimentations sur l'étude de l'impact des recherches locales sur la qualité de la solution retournée par l'algorithme HACO.

Comme on peut le constater sur la figure 4.6, l'ajout successif des algorithmes de recherche locale permet d'améliorer significativement la qualité de la solution retournée, et les temps de calculs s'accroissent avec les rajouts successifs des heuristiques de recherche locale, seul pour les instances ayant une distribution géographique semi-groupée.

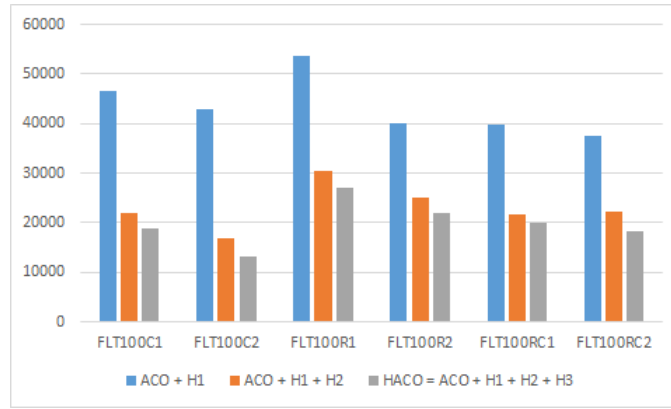


FIGURE 4.6 – Impact des algorithmes de recherche locale sur le coût de la solution.

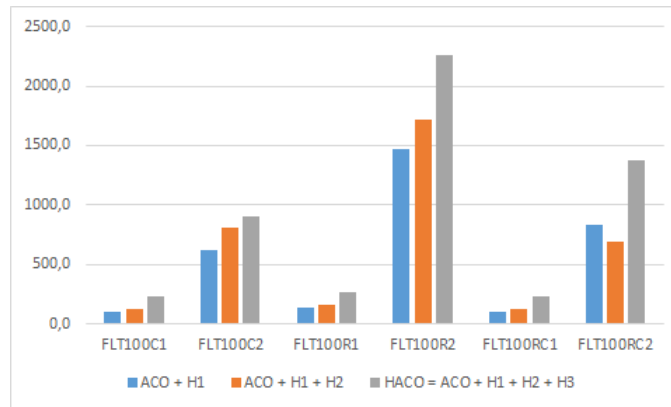


FIGURE 4.7 – Impact des algorithmes de recherche locale sur le temps de résolution.

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules uti- lisés	Temps (s)
FLT100R1_1	5620	D^1E^{48}	22	FLT100R2_1	5066	D^3E^{36}	88
FLT100R1_2	5333,2	D^4E^{40}	28	FLT100R2_2	4712,6	$C^3D^1E^{23}$	459
FLT100R1_3	4648,5	D^4E^{33}	59	FLT100R2_3	3572,8	$B^1C^2D^2E^{10}$	2379
FLT100R1_4	3493,2	D^4E^{22}	153	FLT100R2_4	3686,1	$A^1B^1D^1E^9$	3634
FLT100R1_5	5604,3	D^2E^{47}	32	FLT100R2_5	3937,8	$C^1D^5E^{14}$	219
FLT100R1_6	5395,9	D^4E^{42}	40	FLT100R2_6	3923,9	$C^1D^4E^{16}$	404
FLT100R1_7	4822,5	D^1E^{38}	68	FLT100R2_7	3300,9	$B^1C^1D^3E^8$	2387
FLT100R1_8	3302,5	$C^1D^2E^{21}$	575	FLT100R2_8	2810,5	$A^1B^1C^1D^1E^1$	3630
FLT100R1_9	4991,4	D^3E^{38}	64	FLT100R2_9	3149,2	$C^1D^3E^{12}$	456
FLT100R1_10	3657,7	D^2E^{26}	130	FLT100R2_10	3593,7	$C^1D^4E^{13}$	515
FLT100R1_11	3989,4	D^2E^{29}	160	FLT100R2_11	2387,9	C^4E^1	1937
FLT100R1_12	2853,9	D^6E^{11}	235				
Total	53712,8		1572	Total	40141,5		16114

TABLE 4.14 – Expérimentations ACO + H1 sur le benchmark FLT100R.

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules utilisés	Temps (s)
FLT100RC_1	6199,9	D^4E^{42}	28	FLT100RC_1	6504,2	D^4E^{36}	33
FLT100RC_2	5884,6	D^7E^{36}	29	FLT100RC_2	5370,5	$C^1D^4E^{23}$	171
FLT100RC_3	4962,3	D^4E^{31}	52	FLT100RC_3	4164,3	$C^2D^1E^{17}$	2318
FLT100RC_4	3638	$C^2D^3E^{17}$	375	FLT100RC_4	3367,2	$B^1C^2D^1E^5$	1851
FLT100RC_5	5831,1	D^5E^{36}	47	FLT100RC_5	5692,8	$C^1D^3E^{27}$	209
FLT100RC_6	5423,2	D^3E^{27}	80	FLT100RC_6	5058,5	$C^1D^7E^{27}$	427
FLT100RC_7	4510,2	$C^1D^5E^{24}$	55	FLT100RC_7	4227,3	$C^3D^3E^9$	483
FLT100RC_8	3399	D^6E^{15}	185,4	FLT100RC_8	3087,137	$B^1C^4D^2$	1194
Total	39848,5		854	Total	37472		6688

TABLE 4.15 – Expérimentations ACO + H1 sur le benchmark FLT100RC.

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules utilisés	Temps (s)
FLT100C1_1	6183,2	D^6E^{47}	27	FLT100C2_1	6522,2	D^6E^{43}	38
FLT100C1_2	5710,3	D^6E^{42}	54	FLT100C2_2	6376	D^5E^{41}	60
FLT100C1_3	4928,2	D^5E^{34}	83	FLT100C2_3	5381,2	D^6E^{31}	940
FLT100C1_4	3666,5	D^3E^{22}	407	FLT100C2_4	4728,7	$A^1C^1D^4E^{17}$	3318
FLT100C1_5	5705,7	D^8E^{40}	37	FLT100C2_5	5698,5	D^3E^{38}	63
FLT100C1_6	382,9	D^5E^{41}	33	FLT100C2_6	5547,6	D^4E^{34}	110
FLT100C1_7	5746	D^6E^{42}	39	FLT100C2_7	4650,9	D^3E^{28}	149
FLT100C1_8	5003,2	D^8E^{33}	117	FLT100C2_8	4053,7	D^4E^{22}	289
FLT100C1_9	4260	D^5E^{28}	135				
Total	46586,1		933	Total	42958,8		4968

TABLE 4.16 – Expérimentations ACO + H1 sur le benchmark FLT100C.

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules utilisés	Temps (s)
FLT100RC_1	3292,6	$D^4 E^{16}$	59	FLT100RC_1	3241	$D^3 E^7$	142
FLT100RC_2	2906,5	$D^4 E^{13}$	92	FLT100RC_2	2778,7	$C^2 D^1 E^4$	390
FLT100RC_3	2293,6	$D^4 E^8$	153	FLT100RC_3	2420,6	$C^1 D^2 E^2$	1169
FLT100RC_4	2337,8	$D^3 E^8$	252	FLT100RC_4	2412	$B^1 C^2 E^1$	1488
FLT100RC_5	2959,7	$D^4 E^{13}$	51	FLT100RC_5	2988,9	$C^1 D^3 E^3$	205
FLT100RC_6	2967,1	$D^4 E^{13}$	73	FLT100RC_6	2783	$C^1 D^2 E^3$	167
FLT100RC_7	2678,3	$D^3 E^{12}$	93	FLT100RC_7	2792,6	$B^1 C^1 D^2 E^1$	286
FLT100RC_8	2327,8	$D^6 E^7$	232	FLT100RC_8	2708,6	$B^2 C^1 D^2$	1645
Total	21763,5		1009	Total	22125,4		5497

TABLE 4.17 – Expérimentations ACO + H1 et H2, sur le benchmark FLT100RC.

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules utilisés	Temps (s)
FLT100C1_1	2234,7	$D^5 E^{10}$	58	FLT100C2_1	2526,1	$D^4 E^9$	69
FLT100C1_2	2517,4	$D^6 E^{10}$	98	FLT100C2_2	2350,4	$C^1 D^3 E^5$	206
FLT100C1_3	2102,3	$D^4 E^8$	163	FLT100C2_3	2004,7	$B^1 D^3$	1902
FLT100C1_4	2135,6	$D^1 E^9$	246	FLT100C2_4	2172,7	$C^1 D^2 E^2$	3641
FLT100C1_5	2743,5	$D^6 E^{11}$	71	FLT100C2_5	1904,5	$D^4 E^4$	104
FLT100C1_6	2610,6	$D^4 E^{13}$	102	FLT100C2_6	2210,5	$C^1 D^3 E^6$	188
FLT100C1_7	2734,8	$D^5 E^{11}$	133	FLT100C2_7	1845,7	$D^3 E^4$	196
FLT100C1_8	2699,5	$D^6 E^9$	137	FLT100C2_8	1919,5	$C^1 D^2 E^3$	200
FLT100C1_9	2101,6	$D^3 E^8$	153				
Total	21880		1165	Total	16933,8		6510

TABLE 4.18 – Expérimentations ACO + H1 et H2, sur le benchmark FLT100C.

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules utilisés	Temps (s)
FLT100RC1_1	2970,1	$C^1 D^{11} E^4$	77	FLT100RC2_1	2601,4	$C^4 D^1 E^2$	186
FLT100RC1_2	2610	$C^2 D^6 E^5$	105	FLT100RC2_2	2537,3	$C^3 D^3$	342
FLT100RC1_3	2210,1	$C^1 D^4 E^7$	157	FLT100RC2_3	2207,3	$C^2 D^1 E^2$	3486
FLT100RC1_4	2023,3	$D^6 E^4$	503	FLT100RC2_4	2062,6	$B^1 C^1 D^2$	2061
FLT100RC1_5	2836,5	$C^1 D^6 E^8$	259	FLT100RC2_5	2509,4	$C^4 D^2$	211
FLT100RC1_6	2734,7	$C^2 D^4 E^8$	122	FLT100RC2_6	2322,6	$B^2 C^2$	276
FLT100RC1_7	2432,7	$C^3 D^6 E^3$	241	FLT100RC2_7	2317,1	$C^4 E^1$	758
FLT100RC1_8	2186,2	$C^1 D^4 E^7$	367	FLT100RC2_8	2243,9	$C^4 D^1$	3641
Total	20003,7		1834	Total	18401,738		10965

TABLE 4.19 – Expérimentations ACO + H1, H2 et H3 sur le benchmark FLT100RC.

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules utilisés	Temps (s)
FLT100C1_1	2078,7	$C^3D^6E^3$	157	FLT100C2_1	1898	$B^2C^2D^1E^1$	118
FLT100C1_2	2385,1	$B^1C^3D^3E^6$	146	FLT100C2_2	1923	$B^2C^2D^1$	407
FLT100C1_3	1837	$B^1D^8E^1$	219	FLT100C2_3	1690,9	$C^3D^1E^1$	1743
FLT100C1_4	1732	$C^1D^4E^4$	534	FLT100C2_4	1691,4	$C^2D^1E^1$	3644
FLT100C1_5	2189,4	$B^1C^3D^7$	304	FLT100C2_5	1472,5	$C^3D^1E^1$	265
FLT100C1_6	2176,6	$C^2D^5E^7$	130	FLT100C2_6	1791,5	$B^1C^3D^1$	256
FLT100C1_7	2296,1	$C^5D^4E^4$	125	FLT100C2_7	1421,5	C^3D^1	520
FLT100C1_8	2273,9	$C^5D^6E^1$	292	FLT100C2_8	1442,2	$B^1C^2D^1$	305
FLT100C1_9	1837,9	$C^2D^4E^5$	189				
Total	18806,7		2099	Total	13131		7261

TABLE 4.20 – Expérimentations ACO + H1, H2 et H3, sur le benchmark FLT100C.

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules utilisés	Temps (s)
FLT100R1_1	3050,7	$C^1D^{12}E^6$	127	FLT100R2_1	2191	C^5D^1	124
FLT100R1_2	2730,3	$C^1D^7E^9$	126	FLT100R2_2	2060,7	C^3D^1	452
FLT100R1_3	2242,4	$C^1D^5E^7$	374	FLT100R2_3	1924,2	C^4	3637
FLT100R1_4	1751,1	D^6E^4	444	FLT100R2_4	2063,4	B^1C^2	3637
FLT100R1_5	2729,5	$C^4D^5E^6$	132	FLT100R2_5	2017,1	C^3D^2	546
FLT100R1_6	2350,2	$C^2D^6E^5$	223	FLT100R2_6	2176,3	$B^1C^2E^3$	1098
FLT100R1_7	1882,3	$C^2D^2E^6$	393	FLT100R2_7	1927,1	$B^1C^2E^1$	3623
FLT100R1_8	1910,9	$C^1D^6E^3$	295	FLT100R2_8	2135,5	B^2C^1	3625
FLT100R1_9	2358,7	$C^2D^3E^9$	109	FLT100R2_9	2012,8	C^3D^1	2421
FLT100R1_10	2213,2	$C^1D^6E^6$	116	FLT100R2_10	1992	$B^1C^1D^2$	2114
FLT100R1_11	1944,3	$C^1D^3E^7$	292	FLT100R2_11	1780,8	C^3	3632
FLT100R1_12	1920,2	$C^1D^3E^7$	490				
Total	27083,9		3128	Total	21980,8		24913

TABLE 4.21 – Expérimentations ACO + H1, H2 et H3, sur le benchmark FLT100R.

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules utilisés	Temps (s)
FLT100R1_1	3568,6	D^2E^{25}	86	FLT100R2_1	2502	D^3E^7	68
FLT100R1_2	2973,6	D^3E^{17}	106	FLT100R2_2	2393,7	$C^2D^1E^3$	513
FLT100R1_3	2611,3	D^3E^{13}	108	FLT100R2_3	2176,4	C^1D^3	2240
FLT100R1_4	1964,8	D^4E^7	471	FLT100R2_4	2511,5	$B^1C^2D^1$	3606
FLT100R1_5	2988,3	D^3E^{17}	75	FLT100R2_5	2262,5	$C^2D^2E^1$	209
FLT100R1_6	2682	D^3E^{14}	60	FLT100R2_6	2346,8	C^2E^4	597
FLT100R1_7	2153	D^1E^{11}	152	FLT100R2_7	2259	$B^1C^1E^2$	2345
FLT100R1_8	2212	D^3E^9	176	FLT100R2_8	2366,9	$B^1C^1D^2$	3640
FLT100R1_9	2519,3	D^3E^{13}	107	FLT100R2_9	2006,1	D^4E^1	1762
FLT100R1_10	2407	$C^1D^2E^{11}$	82	FLT100R2_10	2138	$C^1D^3E^1$	492
FLT100R1_11	2258	D^3E^{11}	120	FLT100R2_11	2120,6	C^2D^2	3419
FLT100R1_12	2093,3	D^6E^6	432				
Total	30432		1980	Total	25083,9		18896

TABLE 4.22 – Expérimentations ACO + H1 et H2, sur le benchmark FLT100R.

4.4 Conclusion

Dans ce chapitre, nous avons traité du problème de collectes et livraisons avec fenêtres de temps et flotte hétérogène de véhicules. Nous avons traité ce problème dans le cas où la flotte des véhicules est de taille finie, et dans le cas où elle est de taille infinie. L'objectif étant de déterminer les solutions minimisant le coût d'acquisition des véhicules, additionné au coût de carburant.

Pour le problème de collectes et livraisons avec fenêtres de temps et flotte hétérogène de véhicules de taille finie, nous avons proposé un modèle de programmation linéaire en nombres mixtes. Pour le résoudre, nous avons, après avoir remarqué que le problème avait une structure en blocs, proposé un algorithme basé sur la décomposition de Benders. Afin d'accélérer la résolution des instances, nous avons proposé des inégalités valides et des techniques de réduction de l'espace des solutions. Ces méthodes exactes sont les premières méthodes de résolution de ce nouveau problème. Malgré le fait qu'elles n'arrivent pas à résoudre des instances au-delà de 20 demandes et 4 types de véhicules dans des temps raisonnables, elles peuvent être utilisées comme des critères d'évaluation des méthodes approchées sur le problème de collectes et livraisons avec fenêtres de temps et flotte hétérogène de véhicules de taille finie.

Ensuite, nous avons proposé une méthode de génération de colonnes. Du fait que le temps moyen de résolution est important, nous avons développé un algorithme hybride de colonie de fourmis permettant d'obtenir des solutions de qualité avec un temps de calcul plus faible.

Concernant l'algorithme de génération de colonnes, nous avons résolu les sous-problèmes de pricing avec un algorithme à labels. Nous avons aussi proposé dans le cadre de cette méthode de résolution, un résultat central qui permet à chaque itération de la génération de colonnes de savoir si une requête est intéressante ou pas à considérer. Ceci permet de diminuer considérablement le temps d'exécution de l'algorithme.

Dans l'algorithme hybride de colonie de fourmis, l'hybridation s'est faite par l'intégration des algorithmes de recherche locale qui ont permis d'améliorer considérablement la qualité de la solution trouvée par l'algorithme de colonie de fourmis. Tout en retournant des solutions de qualité, cet algorithme trouve des solutions avec des temps de calcul beaucoup plus faibles que celle de la génération de colonnes. L'efficacité de cet algorithme a été montrée en comparant les résultats qu'il retourne sur des instances de petite taille. Sur ces instances, dans tous les cas où l'algorithme de décomposition de Benders trouve une solution optimale, il trouve une solution optimale.

Dans le prochain chapitre, nous allons traiter du problème de PDPTW avec flotte hétérogène de véhicules couplé à l'affectation des chauffeurs. Nous exploiterons les méthodes de résolution présentées dans ce chapitre afin de proposer une solution à ce problème.

Chapitre 5

Optimisation des tournées et planification des chauffeurs

5.1 Introduction

Dans les précédents chapitres, nous avons traité des problématiques de collectes et livraisons avec fenêtres horaires. Dans ce chapitre, nous allons traiter d'un problème de collecte et livraison plus général. En effet, dans de nombreux cas réels, les tournées de véhicules sont abordées conjointement avec le problème d'affectation des ressources humaines (chauffeurs). Les tournées de véhicules couplées au problème d'affectation des ressources humaines (chauffeurs) jouent un rôle central dans la logistique.

La plupart des études menées sur les problèmes de tournées de véhicules se focalisent sur l'utilisation au mieux du parc automobile afin de satisfaire des demandes de collecte et/ou de livraisons. Cela se traduit par la détermination pour chaque demande, du véhicule qui la réalisera, et de l'heure à laquelle débutera le service de cette dernière, ceci dans le but de minimiser le coût total de l'activité. De nos jours, de plus en plus d'études essayent d'introduire des contraintes réelles dans les problèmes de tournées de véhicules étudiés afin d'être le plus proche possible des cas réels de problèmes complexes de tournées de véhicules.

Dans cette partie de notre travail, nous traitons d'un problème de tournées de véhicules couplé à l'affectation des ressources humaine présent dans le milieu hospitalier. Dans ce problème, nous intégrons plusieurs aspects que l'on retrouve dans des cas pratiques, mais qui n'ont pas été intégrés conjointement dans les études scientifiques menées jusqu'à présent. Dans ce problème, nous avons une équipe de chauffeurs, un parc hétérogène de véhicules et l'objectif est de satisfaire quotidiennement, un ensemble de requêtes à moindre coût. C'est une problématique rencontrée au sein du réseau logistique du Centre Hospitalier de Troyes.

Dans la figure 5.1, un exemple d'une solution du problème étudié composé de 10 demandes

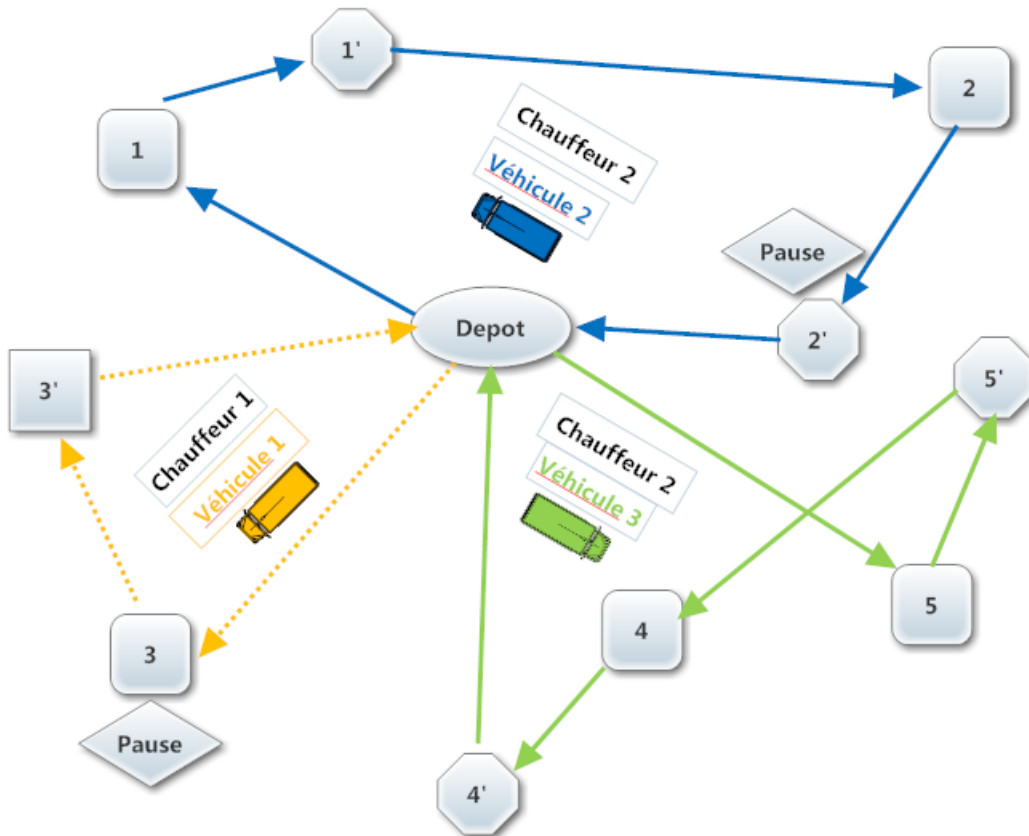


FIGURE 5.1 – Exemple de solution du problème étudié.

(soit 5 requêtes) desservies par 2 chauffeurs, en utilisant 3 véhicules de différents types. On peut remarquer que le chauffeur 2 retourne au dépôt pour réaliser un changement de véhicule et débiter une nouvelle tournée.

Dans la suite de ce chapitre, nous présenterons dans la prochaine section une description du problème, une modélisation sous forme de programme linéaire en nombres entiers (section 5.3), ensuite nous proposons une méthode de résolution à la section 5.4, et un nouveau benchmark. Des expérimentations pour montrer l'efficacité de la méthode sont menées à la section 5.2.

5.2 Description du problème

Une requête est constituée d'une demande de collecte d'un nombre de chariots d'un type de produit, et d'une demande de livraison (associée) des chariots préalablement collectés. Chaque couple de demandes constituant une requête est réalisé par le même chauffeur en utilisant le même véhicule. Du fait que l'on livre ce que l'on collecte, une demande de livraison est toujours réalisée après la demande de collecte correspondante. Chaque demande est située sur un point géographique, et est constituée d'un nombre fini de chariots d'un type de produit, d'un temps

de service, et d'une fenêtre de temps dans laquelle doit débiter le service. Les types de produits et les types de véhicules sont soumis à des contraintes d'incompatibilité. En effet, certains produits ne peuvent pas être transportés au même moment dans le même véhicule, et certains type de véhicules ne peuvent pas transporter certains types de produits. Par exemple, un véhicule non frigorifique ne peut transporter que des chariots de repas. Certains véhicules ne pouvant pas accéder à certains sites il y a des contraintes d'incompatibilité supplémentaires entre les types de véhicules et les sites de collectes et/ou de livraisons. Le changement de véhicule par un chauffeur est autorisé, mais doit se faire au dépôt. En effet, il possible que le véhicule à la disposition du chauffeur ne soit pas adapté pour satisfaire les demandes à venir. Le planning des chauffeurs est soumis à des règles qui régissent les temps de travail, les temps de conduite, les temps de service et les temps de pause.

L'objectif dans notre étude est de satisfaire toutes les requêtes tout en minimisant le coût qui est composé :

- du coût dû au nombre de chauffeurs nécessaires dans la solution,
- du coût dû à l'acquisition (location ou achat) du parc de véhicules,
- du coût de carburant (fonction du type de véhicule et de la distance parcourue).

Le problème étudié est un couplage du problème de collectes et livraisons avec fenêtres de temps, multiple-flux, multiple-véhicules, avec un problème de planification du personnel. La section 2.4 du chapitre 2, montre que ce problème est un problème difficile et qu'il n'a pas été étudié dans la littérature. Dans la prochaine section, nous donnons une description formelle du problème étudié.

5.3 Modélisation mathématique

Dans cette section, nous proposons une description mathématique du problème étudié. Soient $\mathcal{N}$ l'ensemble des requêtes et $\mathcal{H}$ l'ensemble des chauffeurs. Chaque requête $i \in \mathcal{N}$ est constituée d'une demande de collecte p_i et d'une demande de livraison d_i . Le couple de demandes (collecte et livraison) constitue une requête qui doit être réalisée par le même chauffeur, en utilisant le même véhicule. Si le chauffeur arrive avant la date de début de service au plus tôt d'une demande, alors il devra attendre le début du service de la demande avant de réaliser cette dernière. Chaque requête $i \in \mathcal{N}$, est composée :

- d'une quantité de chariots d'un type spécifique de produit à collecter ou à livrer,
- d'un temps de service (temps de chargement ou temps de déchargement),
- d'un site géographique,
- d'une fenêtre horaire de début de service.

Dans la suite, d'une part nous considérons les notations de la table 5.1.

$2N$	Nombre de demandes
V'	Ensemble des demandes (incluant le dépôt)
V	Ensemble des demandes (sans le dépôt)
$\mathcal{A}$	Ensemble d'arcs
P	Ensembles des demandes de collectes
D	Ensemble des demandes de livraisons
K	Ensemble des types de véhicules
$\mathcal{F}$	Ensemble des types de produits
$a_{f_1 f_2}$	est une constante de valeur 1 si les types de produits f_1 et f_2 sont compatibles; 0 sinon
Q_k	Capacité d'un véhicule de type k
v^k	Coût par unité de distance d'un véhicule de type k
δ	Coût d'un chauffeur
c^k	Coût fixe d'un véhicule de type k
d_{ij}	Distance du site de la demande i au site de la demande j
t_{ij}	Temps de parcours du site de la demande i au site de la demande j
s_i	Temps de service nécessaire pour charger (ou décharger) les chariots de la demande i
q_i	Nombre de chariots à collecter ou à livrer pour la demande i
e_i	Date de début au plus tôt du service de la demande i
l_i	Date de début au plus tard du service de la demande i
f_i	le type de produit associé à la demande i
tch	Temps de changement de véhicule (au dépôt)

TABLE 5.1 – Notations du chapitre 5.

L'objectif du problème étudié est de proposer un ensemble de plannings des chauffeurs qui minimise le coût de mise en application de cette solution. Les coûts retenus dans notre étude sont : le coût dû au nombre de chauffeurs, le coût dû à l'acquisition de la flotte des véhicules, le coût de carburant nécessaire pour satisfaire toutes les tournées.

Un planning est une suite de tournées réalisées par un chauffeur. Et une tournée est une suite de demandes qui débute au dépôt des véhicules et se termine au dépôt. Elle est réalisée par un chauffeur, en utilisant un véhicule. Dans une tournée, doivent être respectées : les contraintes de capacité des véhicules, et les contraintes d'incompatibilité entre véhicules et produits, entre produits, et entre véhicules et sites de collectes et/ou livraison. Si entre deux tournées un chauffeur change de véhicule, alors un temps de changement de véhicule est considéré. Et le changement de véhicule s'effectue toujours au dépôt.

Dans la suite, nous proposons un nombre de paramètres qui permettent de définir la plupart des règles journalières auxquelles le planning d'un chauffeur peut être soumis. Pour ce faire, nous avons intégré d'une part le règlement européen sur les temps de conduite et de repos des chauffeurs de véhicules de plus de 3,5T, et d'autre part, les règles sur le temps de travail dans la fonction publique hospitalière en France.

5.3.1 Règles sur le planning journalier des chauffeurs

Dans cette section, on propose un ensemble d'éléments pour pouvoir modéliser tout règlement auquel on voudrait soumettre le planning journalier d'un poste de chauffeur. Pendant sa journée, dans notre cas d'étude un chauffeur est amené :

1. soit à conduire un véhicule entre deux points,
2. soit à charger ou décharger les chariots pour satisfaire une demande,
3. soit à effectuer le nettoyage du véhicule,
4. soit en attente pour un prochain départ pour satisfaire une demande,
5. soit en pause.

Nous regroupons ces cinq moments en trois états :

1. en conduite,
2. en service (chargement, déchargement, ou nettoyage du véhicule),
3. en pause.

Un exemple de ces différents états dans un planning de chauffeur est donné par la figure 5.2. Dans cet exemple, la durée de la pause est d'une heure et vingt-deux minutes, et le temps total de travail est de 7 heures et 30 minutes.

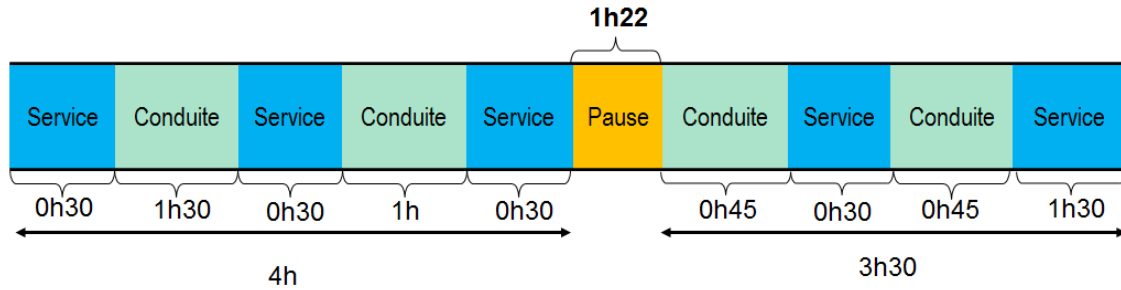


FIGURE 5.2 – Exemple de différents états de travail d'un chauffeur dans un planning.

Un chauffeur peut changer de véhicule, et un véhicule peut être utilisé par plusieurs chauffeurs dans la même journée. Une illustration de ce cas est donné par la figure 5.3.

Dans la table 5.2, nous avons proposé un ensemble de paramètres pour modéliser un planning. Ils permettent, d'intégrer sous forme linéaire, les différentes règles d'un planning.

5.3.2 Contraintes d'incompatibilité

Dans les réseaux logistiques, il peut arriver que l'on ne puisse pas transporter deux types de produits dans le même véhicule. Par exemple les chariots de repas ne peuvent pas être transportés en même temps que des chariots de pharmacie. En effet, afin de conserver la

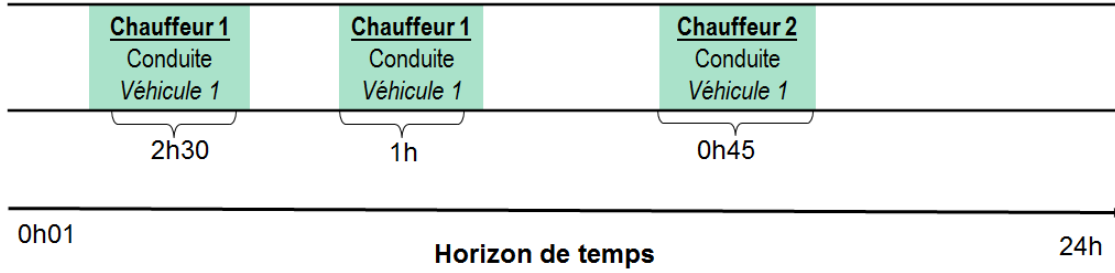


FIGURE 5.3 – Exemple d'utilisation d'un véhicule dans deux plannings de chauffeur.

$\underline{WT}$	durée minimum de travail
$\overline{WT}$	durée maximum de travail
$\overline{WTWB}$	durée de travail maximum sans pause
$\underline{WTAB}$	durée de travail minimum après la pause
$\underline{WTBB}$	durée de travail minimum avant la pause
$\underline{BD}$	durée minimum de pause
$\overline{BD}$	durée maximum de pause
DT	durée maximum de conduite sans interruption

TABLE 5.2 – Paramètres pour modéliser les règles d'un planning de chauffeur.

chaîne de froid, les chariots de repas sont transportés dans des véhicules frigorifiques qui maintiennent une température constante; ce qui peut être préjudiciable pour certains types de médicaments.

Un autre type d'incompatibilité est le fait qu'on ne peut pas transporter un type de produit avec un type de véhicule. Par exemple, toujours à cause de la conservation de la chaîne de froid, on ne peut pas transporter des chariots de repas dans un véhicule non frigorifique.

Les deux types de contraintes d'incompatibilité mentionnées précédemment, ont été prises en compte par Xu et al. [83].

Nous avons considéré un troisième type d'incompatibilité : celui entre le type de véhicule et le site d'une demande. En effet, souvent pour des raisons d'accès à la plate-forme de collecte ou de livraison d'un produit, il est nécessaire d'avoir un véhicule ayant les spécificités adéquates. Il y a aussi les limitations sur les routes donnant accès à certaines villes qui font en sorte que certains véhicules ne puissent pas desservir les demandes de cette ville. C'est le cas de la ville d'Arcis-sur-Aube (France), pour laquelle présentement, aucun véhicule de plus de 10 tonnes ne peut y accéder.

Pour résumer, nous considérons dans ce travail trois types d'incompatibilité :

1. incompatibilité entre types de produits,
2. incompatibilité entre type de véhicule et type de produit,
3. incompatibilité entre type de véhicule et site d'une demande.

L'intégration de la planification des chauffeurs et des contraintes d'incompatibilité marque la différence entre ce chapitre et le précédent.

5.3.3 Modèle mathématique

Dans cette section, nous proposons un modèle linéaire en nombres mixtes pour modéliser le problème étudié. Pour chaque demande $i \in \mathcal{N}$, et chaque type de véhicule $k \in \mathcal{K}$, nous notons $\mathcal{N}_i$ l'ensemble des requêtes incompatibles avec la requête i , et $\mathcal{N}^k$ l'ensemble des requêtes qui ne peuvent pas être transportées par un véhicule de type k .

Afin de pouvoir évaluer à chaque instant de la journée le nombre de véhicules utilisés, nous avons opté pour une discrétisation du temps à l'unité de temps. Ceci permet de calculer pour chaque type de véhicule, le nombre de véhicules présent dans une solution donnée. Dans le cas industriel l'unité a été la minute.

Les variables de décision sont :

- $x_{itkjt'k'}^h$ variable binaire de valeur 1 si le chauffeur h réalise successivement la demande i avec un véhicule de type k au temps t , et la demande j avec un véhicule de type k' au temps t' ; et de valeur 0 sinon;
- $\mu_{itkjt'k'}^h$ variable binaire de valeur 1 si le chauffeur h réalise successivement la demande i avec le véhicule de type k au temps t , et la demande j avec un véhicule de type k' au temps t' avant le début de la pause; et de valeur 0 sinon;
- w_{kt}^h est une variable binaire de valeur 1 si le chauffeur h utilise un véhicule de type k au temps t ; et de valeur 0 sinon;
- z^h est une variable binaire de valeur 1 si le chauffeur h est présent dans la solution; et de valeur 0 sinon;
- y_k est le nombre de véhicules de type k présent dans la solution;
- Φ_{ij} est le coût de carburant utilisé pour satisfaire successivement les demandes i et j ;
- bb^h est le temps de début de la pause pour le chauffeur h ;
- eb^h est le temps de fin de pause du conducteur h ;
- bs^h est le temps de début de service du chauffeur h ;
- es^h est le temps de fin de service du chauffeur h ;
- Q_{ijf}^k est le nombre de chariots pour le type de produit f présents dans le véhicule de type k , lorsque ce dernier passe sur l'arc (i, j) .

La présence de deux indexations sur le temps dans la variable $x_{itkj t'k}^h$ permet de modéliser le changement de véhicule entre deux demandes. En effet, une durée de changement de véhicule est considérée lorsque c'est le cas.

Le programme linéaire en nombres mixtes est donné par :

$$\text{Minimiser : } \sum_{h \in \mathcal{H}} \delta z^h + \sum_{k \in \mathcal{K}} c^k y_k + \sum_{(i,j) \in \mathcal{A}} \Phi_{ij} \quad (5.1)$$

$$\Phi_{ij} = \sum_{h \in \mathcal{H}} \sum_{k \in \mathcal{K}} \sum_{t, t' \in \mathcal{T}} \left(v^k d_{ij} x_{itkj t'k}^h + \sum_{k' \in \mathcal{K}/k' \neq k} \left(v^k d_{i0} + v^{k'} d_{0j} \right) x_{itkj t'k'}^h \right), \quad \forall (i, j) \in \mathcal{A}; \quad (5.2)$$

$$\sum_{h \in \mathcal{H}} \sum_{k \in \mathcal{K}} \sum_{t, t' \in \mathcal{T}} \sum_{j: (i,j) \in \mathcal{A}} x_{itkj t'k'}^h = 1, \quad \forall i \in P; \quad (5.3)$$

$$\sum_{t, t' \in \mathcal{T}} \left(\sum_{j \in V'} x_{itkj t'k}^h - \sum_{j \in V'} x_{(i+N)t_1kj t'k}^h \right) = 0, \quad \forall i \in P, h \in \mathcal{H}, k \in \mathcal{K}, \quad (5.4)$$

$$\sum_{k \in \mathcal{K}} \sum_{i \in P} \sum_{t, t' \in \mathcal{T}} x_{0tkit'k}^h = \sum_{k \in \mathcal{K}} \sum_{i \in D} \sum_{t, t' \in \mathcal{T}} x_{itk0t'k}^h, \quad \forall h \in \mathcal{H} \quad (5.5)$$

$$\sum_{k' \in \mathcal{K}} \sum_{t, t' \in \mathcal{T}} \sum_{j: (j,i) \in \mathcal{A}} x_{jt'k'itk}^h = \sum_{k' \in \mathcal{K}} \sum_{t, t' \in \mathcal{T}} \sum_{j: (i,j) \in \mathcal{A}} x_{itkj t'k'}^h, \quad \forall i \in P \cup D, k \in \mathcal{K}, h \in \mathcal{H}; \quad (5.6)$$

$$\sum_{k \in \mathcal{K}} \sum_{t, t' \in \mathcal{T}} \sum_{j: (i,j) \in \mathcal{A}} \sum_{k' \in \mathcal{K}} x_{itkj t'k'}^h = \sum_{k \in \mathcal{K}} \sum_{t, t' \in \mathcal{T}} \sum_{j \in P} \sum_{k' \in \mathcal{K}} x_{0tkj t'k'}^h, \quad \forall i \in P \cup D, h \in \mathcal{H}; \quad (5.7)$$

$$\sum_{h \in \mathcal{H}} \sum_{k \in \mathcal{K}} x_{itkj t'k}^h = 0, \quad \forall (i, j) \in \mathcal{A}, t, t' \in \mathcal{T} \text{ tel que : } t + s_i + t_{ij} > t' \text{ ou } t \notin [e_i, l_i] \quad (5.8)$$

ou $t' \notin [e_j, l_j]$ ou $t_{ij} > DT$

$$(t' - t - s_i - t_{i0} - t_{ch} - t_{0j}) \sum_{h \in \mathcal{H}} x_{itkj t'k'}^h \geq 0, \quad \forall (i, j) \in \mathcal{A}; \forall t, t' \in \mathcal{T}; \forall k, k' \in \mathcal{K} \quad (5.9)$$

tel que : $k \neq k'$

$$(t' - t - s_i - t_{ij}) \sum_{h \in \mathcal{H}} x_{itkj t'k}^h \geq 0, \quad \forall (i, j) \in \mathcal{A}; \forall t, t' \in \mathcal{T}; \forall k \in \mathcal{K} \quad (5.10)$$

$$\sum_{h \in \mathcal{H}} \sum_{j: (i,j) \in \mathcal{A}} \sum_{k \in \mathcal{K}} \sum_{t, t' \in \mathcal{T}} t * x_{itkj t'k'}^h \geq \sum_{h \in \mathcal{H}} \sum_{j: (j,i) \in \mathcal{A}} \sum_{k \in \mathcal{K}} \sum_{t, t' \in \mathcal{T}} (t + s_i) * x_{jt'k'itk}^h \quad \forall i \in P \cup D; \quad (5.11)$$

$$x_{itkj t'k'}^h \leq z^h, \quad \forall i, j \in N; \forall t, t' \in \mathcal{T}; \forall k, k' \in \mathcal{K} \quad (5.12)$$

$$w_{kt_1}^h \geq x_{itkjt'k'}^h, \quad \forall i, j \in N \forall k, k' \in \mathcal{K}; \forall t_1, t, t' \in \mathcal{T}/t_1 \in [t, t']; \quad (5.13)$$

$$\sum_{h \in \mathcal{H}} w_{kt}^h \leq y_k, \quad \forall k \in \mathcal{K}; \forall t \in \mathcal{T}; \quad (5.14)$$

$$t' x_{itkjt'k'}^h \leq bb^h + l_j \mu_{itkjt'k'}, \quad \forall k, k' \in \mathcal{K}; \forall t, t' \in \mathcal{T}; \forall (i, j) \in \mathcal{A}; \quad (5.15)$$

$$t x_{itkjt'k'}^h \geq eb^h - l_i (1 - \mu_{itkjt'k'}), \quad \forall k, k' \in \mathcal{K}; \forall t, t' \in \mathcal{T}; \forall (i, j) \in \mathcal{A}; \quad (5.16)$$

$$bs^h \leq t x_{itkjt'k'}^h, \quad \forall k, k' \in \mathcal{K}; \forall t, t' \in \mathcal{T}; \forall (i, j) \in \mathcal{A}; \quad (5.17)$$

$$es^h \geq t' x_{itkjt'k'}^h, \quad \forall k, k' \in \mathcal{K}; \forall t, t' \in \mathcal{T}; \forall (i, j) \in \mathcal{A}; \quad (5.18)$$

$$\underline{BD} z^h \geq eb^h - bb^h, \quad \forall h \in \mathcal{H}; \quad (5.19)$$

$$\overline{BD} z^h \leq eb^h - bb^h, \quad \forall h \in \mathcal{H}; \quad (5.20)$$

$$\underline{WTBB} z^h \leq bb^h - bs^h, \quad \forall h \in \mathcal{H}; \quad (5.21)$$

$$\overline{WTWB} z^h \geq bb^h - bs^h, \quad \forall h \in \mathcal{H}; \quad (5.22)$$

$$\underline{WTAB} z^h \leq es^h - eb^h, \quad \forall h \in \mathcal{H}; \quad (5.23)$$

$$\underline{WT} z^h \leq es^h - bs^h, \quad \forall h \in \mathcal{H}; \quad (5.24)$$

$$\overline{WT} z^h \geq es^h - bs^h, \quad \forall h \in \mathcal{H}; \quad (5.25)$$

$$\sum_{i \in P} \sum_{f \in \mathcal{F}} Q_{0if}^k = \sum_{i \in D} \sum_{f \in \mathcal{F}} Q_{i0f}^k; \quad \forall k \in \mathcal{K}; \quad (5.26)$$

$$\sum_{i \in P} \sum_{f \in \mathcal{F}} Q_{0if}^k = 0; \quad \forall k \in \mathcal{K}; \quad (5.27)$$

$$\sum_{j: (i,j) \in \mathcal{A}} a_{f_i f_j} Q_{ijf_i}^k - \sum_{j: (i,j) \in \mathcal{A}} a_{f_j f_i} Q_{ji f_i}^k = q_i \sum_{h \in \mathcal{H}} \sum_{t, t' \in \mathcal{T}} \sum_{k' \in \mathcal{K}} \sum_{j: (i,j) \in \mathcal{A}} x_{itkjt'k'}^h, \quad \forall k \in \mathcal{K}; \forall i \in P \cup D; \quad (5.28)$$

$$\sum_{f \in \mathcal{F}} Q_{ijf}^k \leq Q_k; \quad \forall k \in \mathcal{K}; \forall (i, j) \in \mathcal{A}; \quad (5.29)$$

$$\sum_{h \in \mathcal{H}} \sum_{j: (i,j) \in \mathcal{A}} \sum_{k' \in \mathcal{K}} \sum_{t, t' \in \mathcal{T}} x_{itkjt'k'}^h = 0; \quad \forall i \in P \cup D, \forall k \in \mathcal{K}, \text{ tels que le site ou le type}$$

de produit de la demande i n'est pas compatible au type de véhicule k . (5.30)

$$x_{itkjt'k'}^h, \mu_{itkjt'k'}^h, w_{kt}^h, z^h, y_k \in \{0, 1\}; \quad \forall k, k' \in \mathcal{K}; \forall t, t' \in \mathcal{T}; \forall (i, j) \in \mathcal{A}. \quad (5.31)$$

$$\Phi_{ij}, bb^h, eb^h, bs^h, es^h, Q_{ijf}^k \in \mathbb{N}^+ \quad \forall k, k' \in \mathcal{K}; \forall t, t' \in \mathcal{T}; \forall (i, j) \in \mathcal{A}. \quad (5.32)$$

La fonction objective (5.1) représente la minimisation du coût composé des coûts dûs au nombre de chauffeurs, du coût fixe d'utilisation des véhicules, et du coût dûs à la distance

parcourue. La contrainte (5.2) modélise la variable de calcul de coût de carburant entre deux demandes satisfaites successivement dans la solution. Elle est de valeur nulle, si les demandes en question ne sont pas réalisées successivement. La contrainte (5.3) assure que chaque demande de collecte soit réalisée au moins une fois, et la contrainte (5.4) impose que la collecte soit réalisée dans le même véhicule et par le même chauffeur. Les contraintes (5.5-5.7) garantissent que toute tournée d'un planning débute et se termine au dépôt. Le respect de la fenêtre de temps pour débiter le service de la demande, et la contrainte de respect du temps maximum de conduite sans interruption sont modélisés par les contraintes (5.8) et (5.10). La contrainte (5.9) modélise le changement de véhicule par un chauffeur. Les équations (5.12-5.14) sont des contraintes de liaison entre les variables. Les variables désignant : le temps de début de service, le temps de début de pause, le temps de fin de pause, et le temps de fin de service sont modélisées par les contraintes (5.15-5.18). La contrainte sur la durée minimale de la pause est modélisée par la contrainte (5.19). Le respect de la durée maximale de la pause est modélisée par la contrainte (5.20). La contrainte sur la durée minimale de travail avant le début de la pause est modélisée par la contrainte (5.21). Le respect de la durée maximale de travail avant le début de la pause est modélisée par la contrainte (5.22). La contrainte sur la durée minimale de travail après la fin de la pause est modélisée par la contrainte (5.23). Le respect de la durée minimale de travail est modélisée par la contrainte (5.24). La contrainte sur la durée maximale de travail est modélisée par la contrainte (5.25). Le respect de la capacité des véhicules est modélisé par les contraintes (5.26-5.29). Les contraintes d'incompatibilité produit/véhicule et véhicule/site sont modélisés par l'inéquation 5.30. Les équations (5.31-5.32) définissent le domaine des variables de décision.

Le programme mathématique proposé pour résoudre le problème étudié est en mesure de résoudre les instances jusqu'à 20 demandes au bout de 3 heures de calcul. Pour résoudre les instances de tailles raisonnables (100 demandes), nous avons proposé une métaheuristique de type colonie de fourmis qui reprend les concepts abordés dans les problèmes étudiés aux chapitres 3 et 4.

5.4 Méthode de résolution : Algorithme hybride de colonie de fourmis

Dans cette section, nous proposons une adaptation de l'algorithme hybride de colonie de fourmis présenté au chapitre 4. Tout d'abord, nous allons expliciter la procédure de construction d'une solution par une fourmi en utilisant les phéromones présentes sur les arcs du graphe des demandes. Les formules d'initialisation des phéromones et de mise à jour sont ensuite présentées. Et enfin, les algorithmes de recherches locales dédiés à l'amélioration de la solution construite par une fourmi sont explicités.

5.4.1 Construction d'une solution

Une solution est construite par une fourmi, et est composée d'un ensemble de planning pour chaque chauffeur. Un planning est une suite de tournées telles que l'on peut y insérer une pause pour respecter les règles de temps de travail, de service et de repos. Un planning est réalisé par un chauffeur en utilisant un ou plusieurs véhicules. Une tournée est dite réalisable lorsqu'elle débute et se termine au dépôt tout en respectant les contraintes de fenêtres horaires, de capacité du véhicule et d'incompatibilité. Chaque fourmi utilise les phéromones présentes sur les arcs du graphe des demandes pour construire les plannings, de manière séquentielle. En effet, un nouveau planning est créé lorsque dans le planning courant il n'est plus possible de satisfaire une requête, et qu'il y a encore des requêtes non réalisées.

L'initialisation d'un nouveau planning se fait en deux étapes :

1. choisir une demande de collecte $p_i \in P$ en utilisant les phéromones,
2. choisir, le véhicule qui a la plus grande capacité, et qui peut satisfaire la requête précédemment choisie.

Un exemple de solution à deux plannings est donné dans la figure 5.4. Le planning réalisé par le chauffeur 1 est constitué d'une tournée (3, Pause, 3'), et le planning du chauffeur 2 est constituée de la tournée (1, 1', Pause, 2, 2'), et le planning du chauffeur 3 est constituée de la tournée (5, 5', 4, Pause, 4').

Étant donné un planning s , soit r la dernière tournée construite dans ce planning. Notons p_i la dernière demande de collecte satisfaite dans r . La prochaine requête j à insérer est choisie de manière aléatoire avec une probabilité :

$$P_{ij}^r = \begin{cases} \frac{\tau_{ij}^{\alpha_3} \eta_{ij}}{\sum_{l \in S_i^r} \tau_{il}^{\alpha_3} \eta_{il}} & \text{if } j \in S_i^r \\ 0 & \text{sinon} \end{cases} \quad (5.33)$$

τ_{ij} représente la quantité de phéromones sur l'arc (p_i, p_j) . S_i^r est l'ensemble des requêtes pouvant être satisfaites après la demande p_i . Les paramètres α_3 (réel positif) et η_{ij} contrôlent successivement l'importance entre la visibilité et la quantité de phéromone par rapport à la prochaine requête à satisfaire, ceci dans le but de guider plus efficacement chaque fourmi.

$$\eta_{ij} = \frac{1}{l_j + \alpha_4 d_{p_i p_j}} \quad (5.34)$$

avec $\alpha_4 > 0$ un réel positif fixé.

À chaque insertion d'une requête dans un planning, on vérifie d'une part, que les contraintes de fenêtres de temps, de capacité, et d'incompatibilités soient respectées et d'autre part, que les règles sur le temps de conduite, de pause et de travail soient satisfaites.

L'insertion d'une pause dans un planning se fait en deux étapes :

1. Tentative d'une pause entre deux tournées,

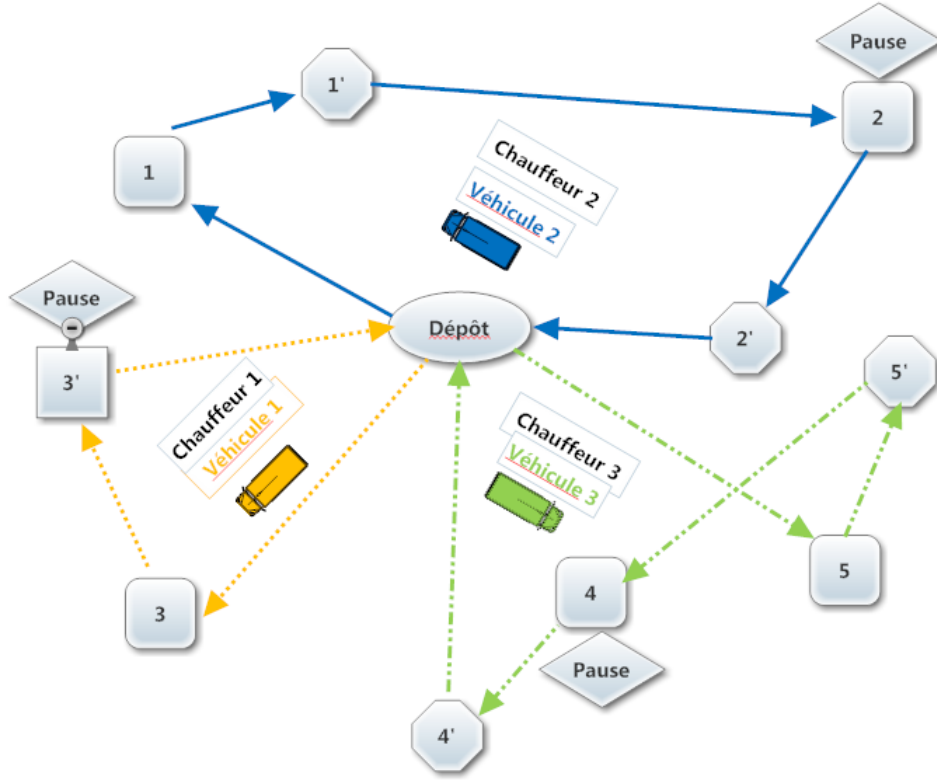


FIGURE 5.4 – Exemple d'une solution construite par une fourmi.

2. Sinon, tentative d'insertion d'une pause entre les demandes de chaque tournée.

Après avoir décrit le processus de construction d'une solution par une fourmi à partir des phéromones, dans la suite nous présentons les formules d'initialisation et de mise à jour des phéromones.

5.4.2 Initialisation et mise à jour des phéromones

Dans cette section, nous présentons les formules pour initialiser et mettre à jour les phéromones sur les arcs du graphe des demandes. Ceci permet la convergence de l'algorithme en un nombre fini d'itérations.

Au départ de la résolution du problème par l'algorithme hybride de colonie de fourmis, les phéromones sont initialisées en utilisant la formule :

$$\tau_{ij} = \begin{cases} \tau_0 & \text{si } (p_i, p_j) \in \mathcal{A} \\ 0 & \text{sinon} \end{cases} \quad (5.35)$$

avec τ_0 un réel positif fixé.

Cette formule d'initialisation est celle couramment utilisée dans les algorithmes de colonies de fourmis.

La mise à jour des phéromones est réalisée après que toutes les fourmis de la population aient construit chacune une solution. Elle est réalisée par la formule :

$$\tau_{ij} = \begin{cases} \rho\tau_{ij} + (1 - \rho)\Delta_{ij}^* & \text{si } p_i \text{ précède } p_j \text{ dans la meilleure fourmi} \\ \rho\tau_{ij} & \text{sinon} \end{cases} \quad (5.36)$$

Notons s_i^* le planning contenant la requête i dans la meilleure solution obtenue. Alors, Δ_{ij}^* est défini par :

$$\Delta_{ij}^* = \frac{(\text{nombre de demandes dans la solution } s_i^*)^{\alpha_5}}{\text{coût total de la meilleure solution trouvée}} \quad (5.37)$$

Avec, $\rho \in [0,1]$ fixé et α_5 un réel positif.

L'idée dans la formulation de Δ_{ij}^* est de renforcer la construction des plannings "bien remplis". En effet, des plannings ayant un nombre élevé de demandes, et un coût faible de carburant sont intéressants dans l'objectif de minimisation des coûts que nous nous sommes fixés.

Ce processus d'augmentation progressive des phéromones sur les arcs présents dans la meilleure solution garantit la convergence de l'algorithme. Afin d'améliorer la qualité des solutions construites par chaque fourmi et accélérer la convergence des solutions, nous proposons dans la prochaine section des méthodes de recherche locale.

5.4.3 Heuristiques de recherche locale

Les recherches locales présentées dans cette section sont pour la plupart une généralisation des heuristiques de recherche locale de l'algorithme hybride de colonie de fourmis de la section 4.3.3. Ces heuristiques ont pour but d'améliorer les solutions construites par chaque fourmi. Nous avons proposé quatre algorithmes de recherches locales pour améliorer une solution. Chacune, a pour but, soit de décroître le coût dû au nombre de chauffeurs (heuristique $H1$), soit le coût dû à l'acquisition de la flotte de véhicules (heuristique $H2$), soit le coût dû à la distance parcourue (heuristiques $H3$, $H4$ et $H5$).

Procédure d'insertion d'une requête dans un planning

Les algorithmes de recherche locale sont basés sur la procédure d'insertion d'une requête (couple de demande de collecte et de demande de livraison) aux positions qui minimisent le coût d'insertion dans la solution. Dans la suite, nous présentons une procédure permettant d'accélérer la détermination des positions d'insertion d'une requête à moindre coût dans une solution partielle.

Proposition 5.4.1 *Soit une tournée $r = (< 0, t_0 >, < i_1, t_1 >, \dots, < i_m, t_m >, < 0, t_{m+1} >)$ avec i_u une demande et t_{i_u} le temps d'arrivée au site de la demande $i_u \forall u \in \{1, \dots, m+1\}$. Étant donné une requête formée par un couple de collecte p et une demande de livraison d à insérer dans r , nous avons :*

1. si $\exists u \in \{1, \dots, m\}$ tel que : $t_{i_u} + s_{i_u} + t_{i_u p} > l_p$, alors l'insertion de la demande p et de la demande d à la position $v \in \{u + 1, \dots, m + 1\}$ est irréalisable,
2. si $\exists u \in \{1, \dots, m\}$ tel que $t_{i_u} + s_{i_u} + t_{i_u d} > l_p$, alors l'insertion de la demande d à la position $v \in \{u + 1, \dots, m + 1\}$ est irréalisable.

Ce résultat permet d'accélérer la vérification de la faisabilité de l'insertion d'une requête dans une solution partiellement construite. Nous avons aussi intégré les techniques d'accélération d'insertion des requêtes proposées par Campbell and Savelsbergh [162].

Heuristique de recherche locale **H1** : diminution du nombre de chauffeur

Une solution est composée d'un ensemble de plannings et chaque planning est réalisé par un unique chauffeur. Afin d'améliorer le coût une solution, la diminution du nombre de plannings est une éventualité. L'heuristique de recherche locale **H1** est dédiée à cette tâche.

À chaque itération, l'heuristique **H1** sélectionne un planning (du dernier au premier) dans la solution, le retire de la solution, et essaie de réinsérer dans la solution partielle, les requêtes présentes dans le planning supprimé. Les requêtes sont réinsérées dans l'ordre dans lequel leur demande de collecte étaient réalisées dans le planning supprimé. Si toutes les requêtes ont pu être réinsérées dans la solution partielle, une nouvelle solution est obtenue, et son coût est comparé à la solution précédente. La meilleure des deux est conservée. Un Pseudo-code de l'exécution de l'heuristique **H1** est donné par Algorithme 5.

Algorithme 5 : Pseudo-code de l'heuristique **H1**

```

1) Entrée : une solution  $\mathcal{S}$ 
2)  $nbS$  = nombre de chauffeurs (ou nombre de planning) dans la solution  $\mathcal{S}$ 
pour tout entier  $s$  de  $nbS$  à 1 faire
    3) Supprimer la  $s^{ième}$  planning de  $\mathcal{S}$ 
    4) Essayer de réinsérer les requêtes dans le  $s^{ième}$  planning
    si toutes les requêtes sont insérées alors
        si la solution obtenue a un meilleur coût que  $\mathcal{S}$  alors
            5) Mettre à jour la solution  $\mathcal{S}$ 
        fin
    fin
fin

```

Un exemple du déroulement d'une itération de l'algorithme **H1** est donné par la figure 5.5. Cet exemple présente une solution composée de trois plannings. Dans cet exemple, l'algorithme **H1** supprime le planning du chauffeur 3, et réinsère les requêtes qui y étaient présentes, dans le planning du chauffeur 2.

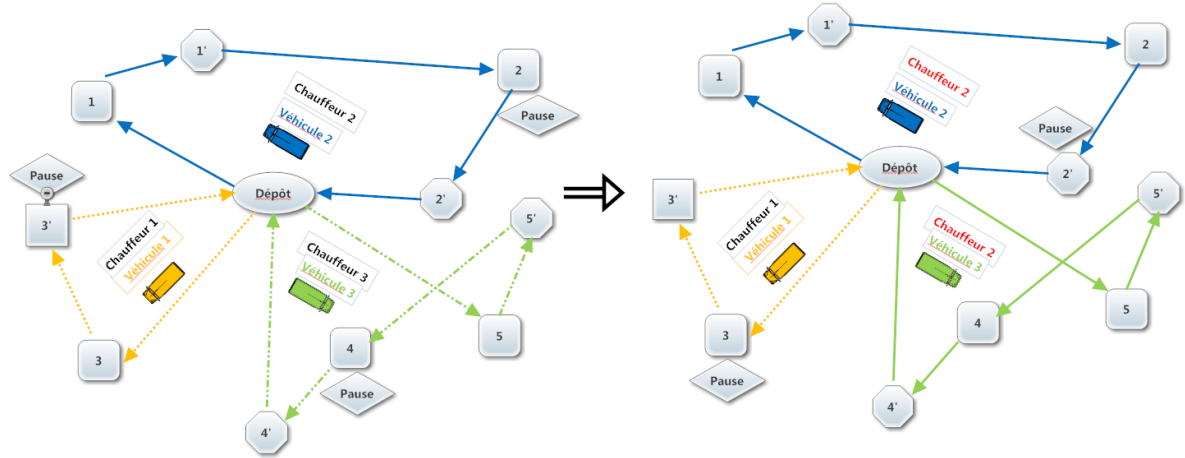


FIGURE 5.5 – Suppression d'un planning dans une solution

Généralement, le coût d'une solution obtenu après l'heuristique de recherche locale *H1* est meilleur. Toutefois, il peut arriver que la solution obtenue ait un coût plus important, ceci peut s'expliquer par la flotte de véhicules nécessaire pour réaliser cette solution.

Heuristique de recherche locale *H2* : diminution du coût d'acquisition de la flotte

L'heuristique de recherche locale *H2* a pour objectif de déterminer une solution de meilleur coût dans le voisinage de notre solution. Elle s'appuie pour y parvenir sur la réduction du coût dû à la flotte de véhicules utilisée.

À chaque itération de l'algorithme *H2*, le type de véhicule réalisant chaque tournée de chaque planning est modifié. Le type de véhicule retenu pour réaliser une tournée est celui qui minimise le coût total de la solution. Un pseudo-code de l'heuristique *H2* est donné par Algorithme 6.

Algorithme 6 : Pseudo-code de l'algorithme *H2*

```

1) Entrée : Une solution  $\mathcal{S}$ 
2)  $nbS1$  = nombre de chauffeur (ou nombre de planning) dans la solution  $\mathcal{S}$ 
pour tout entier  $s$  allant de  $nbS1$  à 1 faire
    3)  $nbTrips$  = nombre de tournées dans le  $s^{ième}$  planning de la solution  $\mathcal{S}$ 
    pour tout entier  $t$  allant de  $nbTrips$  à 1 faire
        4) Remplacer le type de véhicule utilisé pour réaliser la  $t^{ième}$  tournée par le type de
           véhicule qui permet de décroître le coût total de la solution
    fin
fin

```

Un exemple de déroulement d'une itération de l'algorithme *H2* est donné par la figure 5.6. Dans cet exemple, le type de véhicule utilisé par le *chauffeur 3* pour réaliser sa tournée est modifié, et une nouvelle solution est obtenue en utilisant le même véhicule utilisé par le

chauffeur 1. Cet algorithme a été inspiré par celui de Belmecheri et al. [153] pour décroître le coût d'une solution en modifiant les types de véhicules.

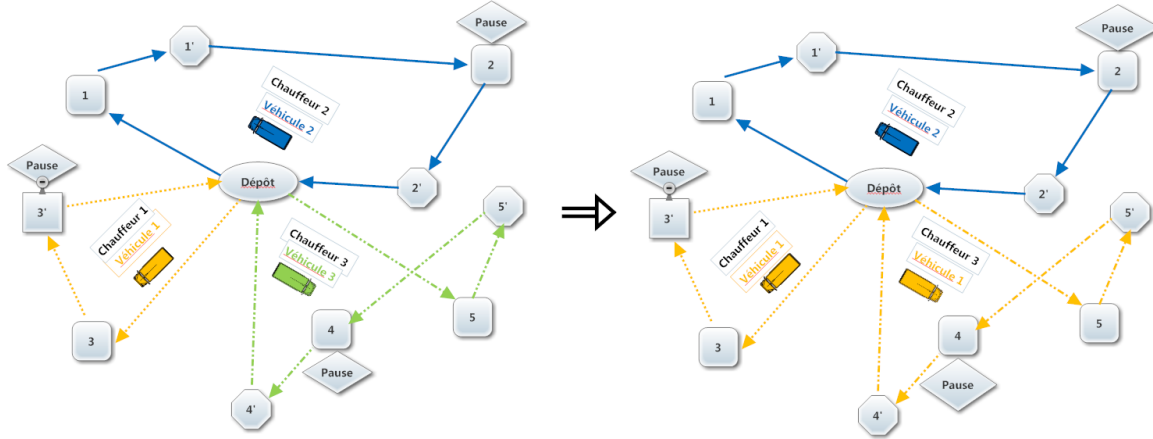


FIGURE 5.6 – Modification du type d'un véhicule dans une solution.

Heuristiques de recherche locale H3, H4 et H5 : diminution du coût dû à la distance parcourue

L'objectif de cette partie est de présenter des heuristiques de recherche locale pour faire décroître le coût (dû à la distance parcourue) d'une solution.

L'algorithme H3 est utilisé pour itérativement supprimer et réinsérer une requête de manière efficiente.

L'heuristique $H4$ quant à elle, supprime une tournée dans un planning ayant plus d'une tournée, et réinsère les requêtes présentes dans la tournée supprimée dans la solution (partielle) de manière à minimiser le coût de réinsertion.

L'algorithme $H5$ procède différemment des deux autres algorithmes. en effet, à chaque itération de cet algorithme une tournée est optimisée en modifiant l'ordre de réalisation des demandes présentes dans cette tournée.

À chaque itération de l'heuristique $H3$, une requête est choisie aléatoirement, et supprimée de la solution. Ensuite, la requête est réinsérée dans le planning qui permet de décroître le coût de la solution après réinsertion. Le processus est répété un certain nombre d'itérations donné. Cet algorithme est inspiré par l'algorithme de recherche à grand voisinage développé par Ropke and Pisinger [103].

Le pseudo-code de cet algorithme est donné par Algorithme 7.

Une illustration du fonctionnement d'une itération de l'algorithme H3 est donnée par la figure 5.7. Dans cette figure, la requête (5, 5') est retirée du planning du *chauffeur 3*, et réinsérée

Algorithme 7 : Pseudo-code de l'algorithme **H3**

```

1) Entrées : une solution  $\mathcal{S}$ , un nombre d'itérations  $nbIter$ 
2)  $cpt = 0$ 
tant que  $cpt < nbIter$  faire
    3) Choisir aléatoirement une requête  $i$  dans la solution  $\mathcal{S}$ 
    4) Retirer  $i$  et la réinsérer aux positions (collecte et livraison) qui minimise le coût d'insertion
    5)  $cpt = cpt + 1$ 
fin

```

dans la tournée du *chauffeur 2*. Si cette opération permet de diminuer le coût d'une solution, alors la précédente solution est remplacée par cette dernière.

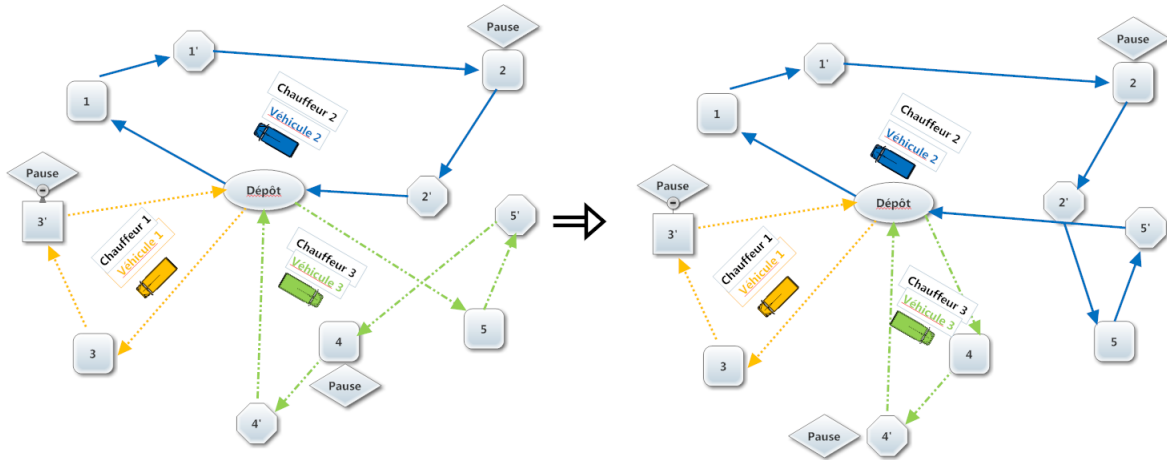


FIGURE 5.7 – Suppression et réinsertion d'une requête dans une solution.

L'algorithme H4 est une généralisation de l'algorithme H3. En effet, là où l'algorithme H3 à chaque itération retire et réinsère une requête, l'algorithme H4 retire une tournée, et réinsère les requêtes présentes dans cette tournée. Si toutes les requêtes sont insérées telles que le coût de la solution obtenue est meilleur, alors la précédente solution est remplacée par cette dernière. Le pseudo-code de cet algorithme est donné par Algorithme 8.

Dans l'illustration du fonctionnement de l'algorithme H4 donnée par la figure 5.8, la tournée du *chauffeur 3* est retirée de la solution. Les requêtes présentes dans la tournée supprimée sont réinsérées dans les plannings du *chauffeur 1* et du *chauffeur 2*.

L'algorithme H5 a pour but de diminuer le coût d'une solution en réordonnant à chaque itération les demandes présentes dans une tournée pour minimiser le coût de carburant nécessaire pour réaliser cette dernière. Afin de réordonner les demandes de la tournée, un ordre d'insertion est généré de façon aléatoire, et les requêtes sont insérées dans cet ordre. Le nombre de générations d'ordre d'insertion des requêtes est donné en paramètre. Un pseudo-code de l'algorithme H5 est donné par Algorithme 9.

Algorithme 8 : Pseudo-code de l'algorithme **H4**

```

1) Entrée : une solution  $\mathcal{S}$ 
2)  $nbS2$  = nombre de chauffeurs (ou de plannings) présents dans  $\mathcal{S}$ 
pour tout entier  $s$  allant de  $nbS2$  à 1 faire
    3)  $nbTrips$  = nombre de tournées dans le  $s^{ième}$  planning de  $\mathcal{S}$ 
    pour tout entier  $t$  allant de  $nbTrips$  à 1 faire
        4) Générer aléatoirement un ordre d'insertion  $\mathcal{O}$  des requêtes présentes dans la  $t^{ième}$ 
           du  $s^{ième}$  planning
        5) Créer une nouvelle tournée et y insérer les requêtes dans l'ordre d'insertion préalable-
           ment généré
        si toutes les requêtes ont pu être insérées dans la nouvelle tournée alors
            si le coût de la solution contenant la nouvelle tournée est inférieur à la précédente
            solution alors
                6) Remplacer la  $t^{ième}$  tournée du  $s^{ième}$  planning par la nouvelle tournée gé-
                nérée
            fin
        fin
    fin
fin

```

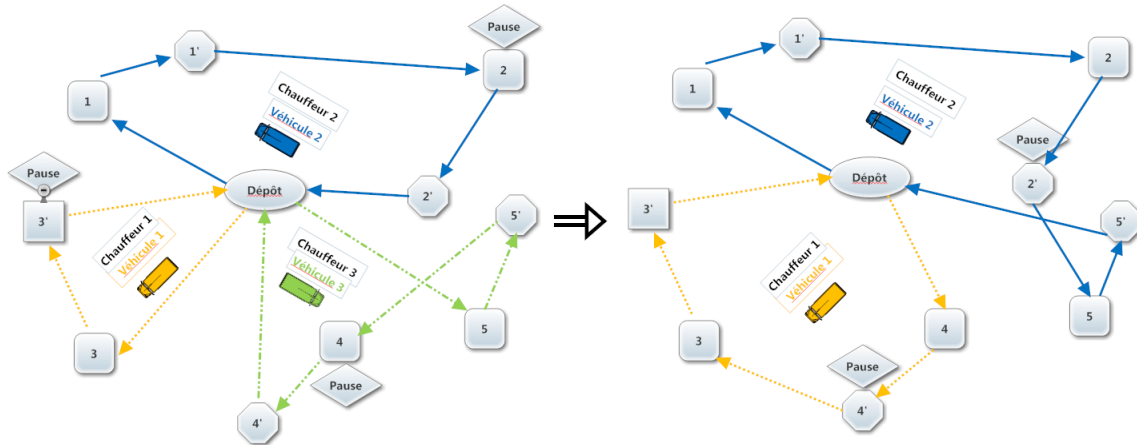


FIGURE 5.8 – Suppression d'une tournée et réinsertion des requêtes dans une solution.

Un exemple de résultat obtenu après l'utilisation de l'algorithme H5 est donné par la figure 5.9. Dans cet exemple, la tournée du *chauffeur 3* est telle que l'ordre de réalisation des demandes de collecte est (5, 4). En générant un nouvel ordre d'insertion (4, 5), la solution obtenue peut être meilleure que la précédente.

Un nombre important d'expérimentations a été réalisé afin de positionner de façon efficiente les heuristiques de recherche locale dans l'algorithme de colonie de fourmis. Le Pseudo-code final de l'algorithme hybride de colonie de fourmi est donné par Algorithme 10.

Algorithme 9 : Pseudo-code de l'algorithme **H5**

```

1) Entrée : une solution  $\mathfrak{S}$ 
2)  $nbS3$  = nombre de chauffeurs (ou de plannings) présents dans  $\mathfrak{S}$ 
pour tout entier  $s$  allant de  $nbS3$  à 1 faire
    3)  $nbTrips$  = nombre de tournées dans le  $s^{ième}$  planning de  $\mathfrak{S}$ 
    pour tout entier  $t$  allant de  $nbTrips$  à 1 faire
        4) Générer aléatoirement un ordre d'insertion  $\mathcal{O}$  des requêtes présentes dans la  $t^{ième}$ 
           du  $s^{ième}$  planning
        5) Créer une nouvelle tournée et y insérer les requêtes dans l'ordre d'insertion préala-
           blement généré
        si toutes les requêtes ont pu être insérées dans la nouvelle tournée alors
            si le coût de la solution contenant la nouvelle tournée est inférieur à la précédente
            solution alors
                6) Remplacer la  $t^{ième}$  tournée du  $s^{ième}$  planning par la nouvelle tournée gé-
                née
            fin
        fin
    fin
fin

```

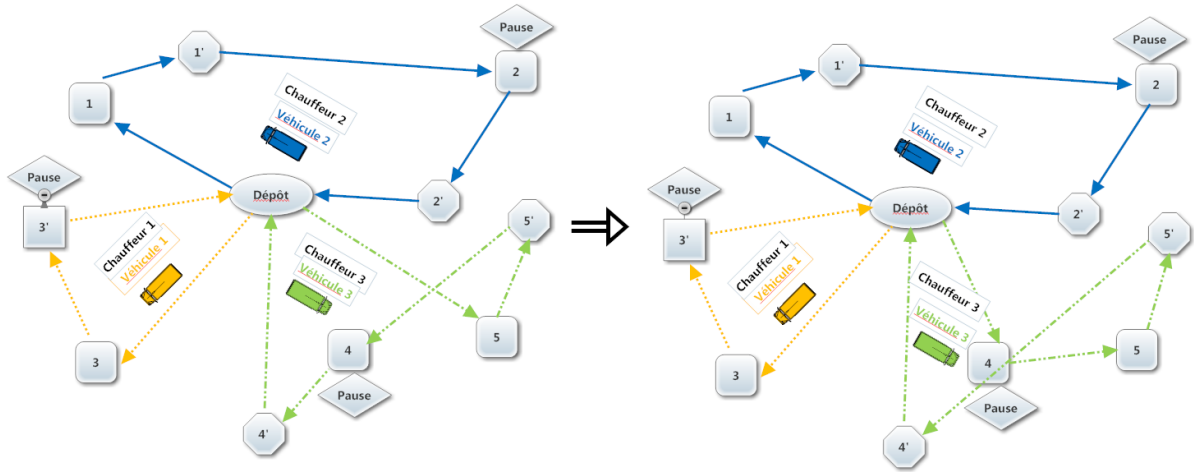


FIGURE 5.9 – Réorganisation de l'ordre de réalisation des demandes d'une tournée.

5.5 Expérimentations

Les algorithmes ont été implémentés sur Eclipse et le langage de programmation a été le Java. L'ordinateur sur lequel les expérimentations ont été menés a un processeur i5 de 2,6 GHz et une mémoire vive de 8 Go.

Dans cette section, nous présentons les différentes expérimentations qui ont été menées afin de conclure sur l'efficacité de notre algorithme. Pour ce faire, nous avons :

- proposé une procédure de génération d'un benchmark,

Algorithme 10 : Pseudo-code de l'algorithme hybride de colonie de fourmi

```

1) Initialisation des paramètres
tant que la meilleure solution n'est pas améliorée durant un nombre d'itérations donné faire
    pour chaque fourmi de la population faire
        2) Construire une solution satisfaisant toutes les demandes
        tant que la diminution du nombre de chauffeur décroît le coût de la solution faire
            3) Appliquer respectivement les algorithmes de recherche locale : H1, H3, H4 and H5
        fin
        4) Appliquer l'algorithme H2
        tant que le coût de la solution obtenue est meilleure que la précédente faire
            5) Appliquer respectivement les algorithmes de recherche locale : H1, H3, H4, H5 and H2
        fin
    fin
    6) Appliquer la mise jour des phéromones
fin

```

- utilisé une démarche bien connue pour fixer les paramètres de l'algorithme hybride de colonie de fourmis,
- mesuré l'impact des heuristiques de recherche locale.

5.5.1 Procédure de génération d'un benchmark

La construction du benchmark s'est faite à partir des benchmarks connus tels que : Li and Lim [3], Solomon [161], et Liu and Shen [157].

Les types de véhicules utilisés (table 5.3) ont été proposé par Liu and Shen [157] pour les instances de VRPTW de Solomon [161].

TABLE 5.3 – Types de Véhicule

Type de véhicule	Coût fixe	coût de carburant/km	Capacité	Type de véhicule
A	500	1	200	frigorifique
B	250	0,6	120	non frigorifique
C	140	0,3	80	frigorifique
D	80	0,21	50	non frigorifique
E	50	0,12	30	non frigorifique

Un benchmark $L100T$ avec une taille raisonnable (un peu plus de 100 demandes) est obtenu par extension des demandes des instances de Li and Lim [3]. Ce benchmark est composé de six types d'instances :

- $LC100T$ pour une distribution géographique groupée des demandes ayant une "petite"

- fenêtre horaire,
- *LC200T* pour une distribution géographique groupée des demandes ayant une "grande" fenêtre horaire,
 - *LR100T* pour une distribution géographique uniforme des demandes ayant une "petite" fenêtre horaire,
 - *LR200T* pour une distribution géographique uniforme des demandes ayant une "grande" fenêtre horaire,
 - *LRC100T* pour une distribution géographique semi-groupée des demandes ayant une "petite" fenêtre horaire,
 - *LRC200T* pour une distribution géographique semi-groupée des demandes ayant une "grande" fenêtre horaire.

Pour obtenir le benchmark final, chaque instance de Li and Lim [3] a été modifiée telle que l'intervalle de temps dans lequel évoluent les demandes soit de longueur **1440** (nombre de minutes dans une plage horaire de 24 heures). Pour ce faire, les temps de début de service au plus tôt et au plus tard ont été multipliés par un réel tel que leurs valeurs soient comprises dans un horizon de 0 à 1440 unité de temps. La vitesse de parcours d'une unité de distance a été fixée à **45** unités de distances pour 60 unités de temps. Toutes les durées de service de demandes supérieures à 30 unités de temps dans le benchmark de Li and Lim [3], sont fixées à 30 unités de temps. Et le temps de changement de véhicule est fixé à **5** unités de temps.

En nous basant sur la réglementation européenne sur le temps de repos, de travail et de conduite d'un chauffeur de véhicule de plus de 3,5 T, et aussi sur le statut d'agent de la fonction publique hospitalière, nous proposons deux types de journées (voir table 5.4).

Journée	$\underline{WT}$	$\overline{WT}$	$\overline{WTWB}$	$\underline{WTBB}$	$\underline{WTAB}$	$\underline{BD}$	$\overline{BD}$	$\underline{DT}$
1	7,5 h	7,5 h	6 h	3 h	3 h	0,5 h	3 h	4,5 h
2	7,5 h	7,5 h	6 h	6 h	6 h	0,4 h	0,4 h	4,5 h

TABLE 5.4 – Différents types de journées

Dans le benchmark proposé, nous avons **8** types de produits différents. Ce chiffre correspond au nombre de types de produits présents dans le cas industriel. L'affectation des types de produits aux requêtes a été faite de façon aléatoire. Il est est de même pour les matrices d'incompatibilité. Le benchmark ainsi obtenu est composé de 56 instances, et peut-être obtenu via ce [lien](#).

5.5.2 Expérimentations sur l'algorithme hybride de colonie de fourmis

Dans cette section, nous allons dans un premier temps présenté la procédure utilisée pour déterminer une valeur robuste pour chaque variable, et dans un second temps, nous montrons l'impact des algorithmes de recherche locale sur la qualité des solutions.

Procédure de fixation des valeurs des paramètres

Dans cet algorithme hybride de colonie de fourmis, nous avons un nombre important de paramètres. Mais les expérimentations ont montré qu'il y a six paramètres qui ont un impact considérable sur la solution obtenue. Ces paramètres et les domaines dans lesquels ils évoluent sont donnés dans la table 5.5.

paramètres	Code	Domaine				
		1	2	3	4	5
ρ	A	0,7	0,8	0,85	0,9	0,95
α_3	B	0	0,5	1	3	4
α_4	C	0	1	10	50	100
α_5	D	0	1	2	3	4
τ_0	E	1	10	50	100	1000
Nombre de fourmis	F	2N	N	$\frac{N}{2}$	$\frac{N}{3}$	$\frac{N}{4}$

TABLE 5.5 – Paramètres et domaine des valeurs.

Pour déterminer la valeur la plus robuste de chaque paramètre, nous avons utilisé la méthode statistique : méthode Taguchi (voir Rosa et al. [163]). Cette méthode permet d'obtenir les valeurs des paramètres les plus robustes possible, c'est-à-dire celles qui sont le moins sensible aux perturbations externes. Les valeurs des paramètres obtenues par cette méthode sont données en gras dans la table 5.5. L'étude a été réalisée en utilisant le logiciel [Minitab ANOVA](#).

Impact des recherches locales et résultats des expérimentations

Les algorithmes de recherche locale ont un impact sur la qualité de la solution retournée par l'algorithme hybride de colonie de fourmis. Dans cette partie, nous mettons en évidence cet impact, et pour ce faire, chaque instance est résolue cinq fois pour chacune des configurations suivantes :

- l'algorithme de colonie de fourmis est couplé avec l'algorithme de recherche locale H1 ;
- l'algorithme de colonie de fourmis est couplé avec l'algorithme de recherche locale H2 ;
- l'algorithme de colonie de fourmis est couplé avec l'algorithme de recherche locale H3 ;
- l'algorithme de colonie de fourmis est couplé avec l'algorithme de recherche locale H4 ;
- l'algorithme de colonie de fourmis est couplé avec l'algorithme de recherche locale H5 ;

Les tables 5.7-5.12 donnent les résultats des expérimentations menées. On constate que pour chaque recherche locale et pour chaque distribution géographique (groupé, uniforme et semi-groupé) l'algorithme de recherche locale permet d'améliorer considérablement la qualité de la solution retournée.

Instance	Gap	
	HACO vs UB	HACO vs LB
T20C1	-0,02	34,19
T20C2	-4,19	27,89
T20R1	-0,15	34,69
T20R2	-3,25	42,47
T20RC1	-6,40	28,44
T20RC2	-10,38	36,21
Moyenne	-4,06	33,98

TABLE 5.6 – Expérimentations de l'HACO et du MILP sur des instances à 20 demandes.

Comparaison de l'HACO avec le modèle linéaire

Pour déterminer l'efficacité de notre algorithme, nous avons mené des expérimentations sur les benchmarks de 20 demandes. En effet, même si le modèle linéaire n'est pas en mesure de résoudre à l'optimalité les instances à 20 demandes, il est toutefois possible par CPLEX d'obtenir une borne supérieure et une borne inférieure sur le coût.

Les résultats des expérimentations sont donnés par les tables 7.1, 7.2, 7.3, 7.4, 7.5, et 7.6.

Un résumé des résultats obtenus est donné par la table 5.6. Dans cette table les gaps sont calculés par les formules :

$$\text{HACO vs UB} = 100 * \frac{UB - HACO}{UB} \quad (5.38)$$

$$\text{HACO vs LB} = 100 * \frac{HACO - LB}{LB} \quad (5.39)$$

Les expérimentations montrent que les solutions obtenues par l'HACO sont en moyennes plus intéressantes que celles trouvées par CPLEX au bout de 3 heures de calcul. En revanche le gap important entre la valeur de la solution réalisable obtenue par le MILP utilisant CPLEX, et la borne inférieure de cette dernière est très important ce qui ne permet pas de conclure sur l'optimalité des solutions tournées par l'HACO.

Les résultats des expérimentations sur le benchmark de taille réelle T100 sont donnés par les tables 5.13-5.15. Ces solutions donnent une base de tests pour de futures approches.

5.5.3 Application sur un cas industrielle : Centre Hospitalier de Troyes

Le réseau logistique du centre hospitalier de Troyes est composé de :

- d'un dépôt de véhicule sur le site de l'hôpital Simone Veil,
- d'une pharmacie,
- d'une blanchisserie,
- d'une cuisine centrale,

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules utilisés	Temps (s)
T100C1_1	6183,2	$D^6 E^{47}$	27	T100C2_1	6522,2	$D^6 E^{43}$	38
T100C1_2	5710,3	$D^6 E^{42}$	54	T100C2_2	6376	$D^5 E^{41}$	60
T100C1_3	4928,2	$D^5 E^{34}$	83	T100C2_3	5381,2	$D^6 E^{31}$	940
T100C1_4	3666,5	$D^3 E^{22}$	407	T100C2_4	4728,7	$A^1 C^1 D^4 E^{17}$	3318
T100C1_5	5705,7	$D^8 E^{40}$	37	T100C2_5	5698,5	$D^3 E^{38}$	63
T100C1_6	382,9	$D^5 E^{41}$	33	T100C2_6	5547,6	$D^4 E^{34}$	110
T100C1_7	5746	$D^6 E^{42}$	39	T100C2_7	4650,9	$D^3 E^{28}$	149
T100C1_8	5003,2	$D^8 E^{33}$	117	T100C2_8	4053,7	$D^4 E^{22}$	289
T100C1_9	4260	$D^5 E^{28}$	135				
Total	46586,1		933	Total	42958,8		4968

TABLE 5.7 – Expérimentations ACO + H1 sur le benchmark T100C.

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules utilisés	Temps (s)
T100R1_1	5620	$D^1 E^{48}$	22	T100R2_1	5066	$D^3 E^{36}$	88
T100R1_2	5333,2	$D^4 E^{40}$	28	T100R2_2	4712,6	$C^3 D^1 E^{23}$	459
T100R1_3	4648,5	$D^4 E^{33}$	59	T100R2_3	3572,8	$B^1 C^2 D^2 E^{10}$	2379
T100R1_4	3493,2	$D^4 E^{22}$	153	T100R2_4	3686,1	$A^1 B^1 D^1 E^9$	3634
T100R1_5	5604,3	$D^2 E^{47}$	32	T100R2_5	3937,8	$C^1 D^5 E^{14}$	219
T100R1_6	5395,9	$D^4 E^{42}$	40	T100R2_6	3923,9	$C^1 D^4 E^{16}$	404
T100R1_7	4822,5	$D^1 E^{38}$	68	T100R2_7	3300,9	$B^1 C^1 D^3 E^8$	2387
T100R1_8	3302,5	$C^1 D^2 E^{21}$	575	T100R2_8	2810,5	$A^1 B^1 C^1 D^1 E^1$	3630
T100R1_9	4991,4	$D^3 E^{38}$	64	T100R2_9	3149,2	$C^1 D^3 E^{12}$	456
T100R1_10	3657,7	$D^2 E^{26}$	130	T100R2_10	3593,7	$C^1 D^4 E^{13}$	515
T100R1_11	3989,4	$D^2 E^{29}$	160	T100R2_11	2387,9	$C^4 E^1$	1937
T100R1_12	2853,9	$D^6 E^{11}$	235				
Total	53712,8		1572	Total	40141,5		16114

TABLE 5.8 – Expérimentations ACO + H1 sur le benchmark T100R.

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules utilisés	Temps (s)
T100RC_1	6199,9	$D^4 E^{42}$	28	T100RC_1	6504,2	$D^4 E^{36}$	33
T100RC_2	5884,6	$D^7 E^{36}$	29	T100RC_2	5370,5	$C^1 D^4 E^{23}$	171
T100RC_3	4962,3	$D^4 E^{31}$	52	T100RC_3	4164,3	$C^2 D^1 E^{17}$	2318
T100RC_4	3638	$C^2 D^3 E^{17}$	375	T100RC_4	3367,2	$B^1 C^2 D^1 E^5$	1851
T100RC_5	5831,1	$D^5 E^{36}$	47	T100RC_5	5692,8	$C^1 D^3 E^{27}$	209
T100RC_6	5423,2	$D^3 E^{27}$	80	T100RC_6	5058,5	$C^1 D^7 E^{27}$	427
T100RC_7	4510,2	$C^1 D^5 E^{24}$	55	T100RC_7	4227,3	$C^3 D^3 E^9$	483
T100RC_8	3399	$D^6 E^{15}$	185,4	T100RC_8	3087,137	$B^1 C^4 D^2$	1194
Total	39848,5		854	Total	37472		6688

TABLE 5.9 – Expérimentations ACO + H1 H1 sur le benchmark T100RC.

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules utilisés	Temps (s)
T100C1_1	2234,7	$D^5 E^{10}$	58	T100C2_1	2526,1	$D^4 E^9$	69
T100C1_2	2517,4	$D^6 E^{10}$	98	T100C2_2	2350,4	$C^1 D^3 E^5$	206
T100C1_3	2102,3	$D^4 E^8$	163	T100C2_3	2004,7	$B^1 D^3$	1902
T100C1_4	2135,6	$D^1 E^9$	246	T100C2_4	2172,7	$C^1 D^2 E^2$	3641
T100C1_5	2743,5	$D^6 E^{11}$	71	T100C2_5	1904,5	$D^4 E^4$	104
T100C1_6	2610,6	$D^4 E^{13}$	102	T100C2_6	2210,5	$C^1 D^3 E^6$	188
T100C1_7	2734,8	$D^5 E^{11}$	133	T100C2_7	1845,7	$D^3 E^4$	196
T100C1_8	2699,5	$D^6 E^9$	137	T100C2_8	1919,5	$C^1 D^2 E^3$	200
T100C1_9	2101,6	$D^3 E^8$	153				
Total	21880		1165	Total	16933,8		6510

TABLE 5.10 – Expérimentations ACO + H1 et H2, sur le benchmark T100C.

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules utilisés	Temps (s)
T100R1_1	3568,6	$D^2 E^{25}$	86	T100R2_1	2502	$D^3 E^7$	68
T100R1_2	2973,6	$D^3 E^{17}$	106	T100R2_2	2393,7	$C^2 D^1 E^3$	513
T100R1_3	2611,3	$D^3 E^{13}$	108	T100R2_3	2176,4	$C^1 D^3$	2240
T100R1_4	1964,8	$D^4 E^7$	471	T100R2_4	2511,5	$B^1 C^2 D^1$	3606
T100R1_5	2988,3	$D^3 E^{17}$	75	T100R2_5	2262,5	$C^2 D^2 E^1$	209
T100R1_6	2682	$D^3 E^{14}$	60	T100R2_6	2346,8	$C^2 E^4$	597
T100R1_7	2153	$D^1 E^{11}$	152	T100R2_7	2259	$B^1 C^1 E^2$	2345
T100R1_8	2212	$D^3 E^9$	176	T100R2_8	2366,9	$B^1 C^1 D^2$	3640
T100R1_9	2519,3	$D^3 E^{13}$	107	T100R2_9	2006,1	$D^4 E^1$	1762
T100R1_10	2407	$C^1 D^2 E^{11}$	82	T100R2_10	2138	$C^1 D^3 E^1$	492
T100R1_11	2258	$D^3 E^{11}$	120	T100R2_11	2120,6	$C^2 D^2$	3419
T100R1_12	2093,3	$D^6 E^6$	432				
Total	30432		1980	Total	25083,9		18896

TABLE 5.11 – Expérimentations ACO + H1 et H2, sur le benchmark T100R.

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules utilisés	Temps (s)
T100RC_1	3292,6	$D^4 E^{16}$	59	T100RC_1	3241	$D^3 E^7$	142
T100RC_2	2906,5	$D^4 E^{13}$	92	T100RC_2	2778,7	$C^2 D^1 E^4$	390
T100RC_3	2293,6	$D^4 E^8$	153	T100RC_3	2420,6	$C^1 D^2 E^2$	1169
T100RC_4	2337,8	$D^3 E^8$	252	T100RC_4	2412	$B^1 C^2 E^1$	1488
T100RC_5	2959,7	$D^4 E^{13}$	51	T100RC_5	2988,9	$C^1 D^3 E^3$	205
T100RC_6	2967,1	$D^4 E^{13}$	73	T100RC_6	2783	$C^1 D^2 E^3$	167
T100RC_7	2678,3	$D^3 E^{12}$	93	T100RC_7	2792,6	$B^1 C^1 D^2 E^1$	286
T100RC_8	2327,8	$D^6 E^7$	232	T100RC_8	2708,6	$B^2 C^1 D^2$	1645
Total	21763,5		1009	Total	22125,4		5497

TABLE 5.12 – Expérimentations ACO + H1 et H2, sur le benchmark T100RC.

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules utilisés	Temps (s)
T100C1_1	2078,7	$C^3D^6E^3$	157	T100C2_1	1898	$B^2C^2D^1E^1$	118
T100C1_2	2385,1	$B^1C^3D^3E^1$	146	T100C2_2	1823	$B^2C^2D^1$	407
T100C1_3	1837	$B^1D^8E^1$	219	T100C2_3	1690,9	$C^3D^1E^1$	1743
T100C1_4	1732	$C^1D^4E^4$	534	T100C2_4	1691,4	$C^2D^1E^1$	3644
T100C1_5	2189,4	$B^1C^3D^7$	304	T100C2_5	1472,5	$C^3D^1E^1$	265
T100C1_6	2176,6	$C^2D^5E^7$	130	T100C2_6	1691,5	$B^1C^3D^1$	256
T100C1_7	2296,1	$C^5D^4E^4$	125	T100C2_7	1421,5	C^3D^1	520
T100C1_8	2273,9	$C^5D^6E^1$	292	T100C2_8	1442,2	$B^1C^2D^1$	305
T100C1_9	1837,9	$C^2D^4E^5$	189				
Total	18806,7		2099	Total	13131		7261

TABLE 5.13 – Expérimentations ACO + H1, H2 et H3, sur le benchmark T100C.

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules utilisés	Temps (s)
T100R1_1	3050,7	$C^1D^{12}E^6$	127	T100R2_1	2191	C^5D^1	124
T100R1_2	2730,3	$C^1D^7E^9$	126	T100R2_2	2060,7	C^3D^1	452
T100R1_3	2242,4	$C^1D^5E^7$	374	T100R2_3	1924,2	C^4	3637
T100R1_4	1751,1	D^6E^4	444	T100R2_4	2063,4	B^1C^2	3637
T100R1_5	2729,5	$C^4D^5E^6$	132	T100R2_5	2017,1	C^3D^2	546
T100R1_6	2350,2	$C^2D^6E^5$	223	T100R2_6	2176,3	$B^1C^2E^3$	1098
T100R1_7	1882,3	$C^2D^2E^6$	393	T100R2_7	1927,1	$B^1C^2E^1$	3623
T100R1_8	1910,9	$C^1D^6E^3$	295	T100R2_8	2135,5	B^2C^1	3625
T100R1_9	2358,7	$C^2D^3E^9$	109	T100R2_9	1812,8	C^3D^1	2421
T100R1_10	2213,2	$C^1D^6E^6$	116	T100R2_10	1892	$B^1C^1D^2$	2114
T100R1_11	1944,3	$C^1D^3E^7$	292	T100R2_11	1780,8	C^3	3632
T100R1_12	1920,2	$C^1D^3E^7$	490				
Total	27083,9		3128	Total	21980,8		24913

TABLE 5.14 – Expérimentations ACO + H1, H2 et H3, sur le benchmark T100R.

Instance	Coût	Véhicules utilisés	Temps (s)	Instance	Coût	Véhicules utilisés	Temps (s)
T100RC1_1	2970,1	$C^1D^{11}E^4$	77	T100RC2_1	2501,4	$C^4D^1E^2$	186
T100RC1_2	2610	$C^2D^6E^5$	105	T100RC2_2	2437,3	C^3D^3	342
T100RC1_3	2210,1	$C^1D^4E^7$	157	T100RC2_3	2107,3	$C^2D^1E^2$	3486
T100RC1_4	2023,3	D^6E^4	503	T100RC2_4	2062,6	$B^1C^1D^2$	2061
T100RC1_5	2836,5	$C^1D^6E^8$	259	T100RC2_5	2409,4	C^4D^2	211
T100RC1_6	2734,7	$C^2D^4E^8$	122	T100RC2_6	2322,6	B^2C^2	276
T100RC1_7	2432,7	$C^3D^6E^3$	241	T100RC2_7	2317,1	C^4E^1	758
T100RC1_8	2186,2	$C^1D^4E^7$	367	T100RC2_8	2243,9	C^4D^1	3641
Total	20003,7		1834	Total	18401,738		10965

TABLE 5.15 – Expérimentations ACO + H1, H2 et H3, sur le benchmark T100RC.

- d'un magasin central,
- d'un ensemble d'acteurs à approvisionner :
 - des centres hospitaliers,
 - des maisons de retraite,
 - des cliniques psychiatriques.

Chaque acteur a des besoins journaliers et très souvent périodiques. Comme nous l'avons déjà précisé au chapitre 1, le problème de tournée de véhicule sous-jacent est un problème de collecte et livraison avec fenêtre de temps.

Pour réaliser l'ensemble de ces demandes, le CHT possède une équipe de chauffeurs. La planification de ces chauffeurs est soumise d'une part à réglementation européenne, et d'autre par à la réglementation du secteur publique hospitalière, sur les temps de conduite, temps de travail, et temps de repos.

La flotte des véhicules du CHT est composée des véhicules loués (sur une période de 5 ans). ces véhicules sont de différents types pour pouvoir transporter différents produits (repas, linge, médicaments, ...). La typologie d'un véhicule est fonction de sa capacité, et du fait qu'il a ou pas une caisse frigorifique. Une table des types de véhicules utilisés au CHT est donné par la table 5.16.

TABLE 5.16 – Types de véhicules

Type de véhicule	Coût fixe	véhicule frigorifique	coût moyen au km	longueur	largeur
AI	450,2€	Oui	0,5€	5,998 m	2,454 m
BI	412,2€	Oui	0,25€	4,498 m	2,454 m
CI	422,2€	Non	0,3€	8,283 m	2,466 m
DI	343,2€	Non	0,21€	6,583 m	2,466 m
EI	296,1€	Non	0,12€	4,562 m	2,238 m

Le problème soumis à notre étude a été résolu en utilisant l'algorithme hybride de colonie de fourmis proposé. Pour des raisons confidentielles nous ne détaillons pas les données du CHT.

Le benchmark des instances du CHT est composé de 7 instances, en raison d'une par jour de la semaine. Pour les nommer nous utiliserons le format "*jour_nombreDemandes*". Les instances de lundi à vendredi sont constitués d'une moyenne de plus de 100 demandes par jour, et pour les journées de samedi et dimanche on a une cinquantaine de demandes.

Les valeurs de la solution trouvée par la métaheuristique hybride de colonie de fourmis, et de la solution en place au début de l'étude sont donnés par la table 5.17.

D'après les expérimentations, en moyenne, l'algorithme proposé améliore de 26.8% la qualité de la solution pour chaque jour de la semaine. De plus le temps de calcul est au plus de

deux minutes, alors qu'une telle étude prendrait des semaines (voire des mois) d'après le questionnaire de la logistique du CHT.

TABLE 5.17 – Solution Industrielle VS Solution de l'HACO

Instance	Coût total			Véhicules utilisés		Temps (s)
	Avant l'étude	Avec HACO	Gap(%)	Avant l'étude	Avec HACO	
Lundi_114	7835.9	6053.6	-29.1	$AI^1BI^1CI^1DI^1EI^1$	$CI^2DI^1EI^1$	253
Mardi_100	7811.3	6030.8	-29.5	$AI^1BI^1CI^1DI^1EI^1$	$CI^2DI^1EI^1$	116
Mercredi_106	7826.6	6034	-29.6	$AI^1BI^1CI^1DI^1EI^1$	$CI^2DI^1EI^1$	141
Jeudi_92	7809.6	6053.4	-29.3	$AI^1BI^1CI^1DI^1EI^1$	$CI^2DI^1EI^1$	249
Vendredi_114	7828.2	6050.9	-29.3	$AI^1BI^1CI^1DI^1EI^1$	$CI^2DI^1EI^1$	132
Samedi_62	3549.6	3125.4	-13	$AI^1BI^1CI^1$	CI^1DI^1	29
Dimanche_52	3532.2	3115.9	-13	AI^1BI^1	CI^1DI^1	24
Total	46193.3	36763.8				

5.6 Conclusion

Dans ce chapitre, nous avons présenté un nouveau problème de collecte et livraison qui intègre plusieurs aspects et contraintes que l'on retrouve dans des problèmes industriels. En effet, elle intègre : les fenêtres horaires, l'aspect hétérogène de la flotte des véhicules, des règles sur les temps de repos, de travail et de service des chauffeurs et des contraintes d'incompatibilité entre différents produits, entre produit et véhicule et entre véhicule et site de collecte et site de livraison. Ces aspects se retrouvent dans de nombreux problèmes d'optimisation des réseaux logistiques rencontré sur le terrain, mais peu de travaux s'y attardent. L'une des explications à notre avis se trouve dans la difficulté de la prise en compte simultanée des nombreuses facettes qu'a ce problème.

Pour résoudre ce problème, nous avons d'une part proposé une modélisation en programme mathématique linéaire en nombres mixtes, et d'autre part proposer un algorithme hybride de colonie de fourmis. Le programme linéaire exécuté sous CPLEX ne permet pas de résoudre à l'optimalité des instances à 20 demandes au bout de 3 heures de calcul. L'hybridation de la métaheuristique de colonie de fourmis s'est faite avec cinq algorithmes de recherche locale. Chaque algorithme étant dédié à l'optimisation précise d'un composant de la fonction objective. Cette dernière est composée du coût dû au nombre de chauffeur utilisé, du coût dû aux véhicules utilisés, et du coût de carburant nécessaire pour réaliser ces tournées.

Nous avons proposé pour réaliser les expérimentations un benchmark qui est une extension du benchmark de Solomon dédié au problème de tournées de véhicules. Les tests menés ont montré l'impact considérable qu'ont les recherches locales sur l'obtention d'une solution de meilleure qualité. Les résultats obtenus sur des instances de petite taille (20 demandes), ont

permis de montrer que la métaheuristique proposée était meilleure de que modèle linéaire. Les résultats que nous avons trouvés sur ce nouveau problème forment une base pour de futures expérimentations pour de nouveaux algorithmes sur ce problème. Dans la suite de ces travaux, il serait intéressant d'une part de proposer d'autre approche de résolution par exemple une méthode de génération de colonnes, et d'autre part, d'étendre encore plus ce problème au cas de multiple-dépôts et de demandes avec plusieurs fenêtres horaires, gestion de stock, et prendre en compte les contraintes liées aux lignes de production.

Chapitre 6

Outils d'aide à la décision

6.1 Introduction

L'objectif des travaux de recherche menés dans le cadre de l'optimisation de la logistique au sein du CHT est de proposer des solutions efficaces afin de dimensionner l'équipe de chauffeurs et la flotte de véhicules. L'horizon de planification retenue est la semaine, car les besoins sont périodiques de période une semaine. Les coûts retenus dans notre étude sont :

- le coût induit par le nombre de chauffeur,
- le coût fixe induit par l'acquisition de la flotte des véhicules,
- le coût de carburant utilisé pour satisfaire l'ensemble des besoins.

Le dimensionnement des équipes de chauffeurs est soumis à la réglementation sur les temps de conduite et de pauses des chauffeurs de véhicules de plus de 3T5 et à la réglementation de temps de travail d'un agent de la fonction publique hospitalière en France. Dans la construction du planning d'un chauffeur, il peut arriver que ce dernier change de véhicules et dans ce cas, un temps de changement de véhicule est considéré.

Nous avons proposé un outil d'aide à la décision au CHT pour :

- évalué une organisation des transports, c'est-à-dire un le coût nécessaire pour la réaliser,
- déterminé une organisation efficiente des transports qui respecte l'ensemble des contraintes, et assure la réalisation de toutes les activités.

Pour la construction de cet outil, nous avons procédé par étape :

- Recueil des données : il s'agit de recueillir l'ensemble des éléments, paramètres et contraintes auxquels est soumis l'outil à concevoir ;
- Hiérarchisation des informations : il s'agit de trier, vérifier et valider les informations collectées sur le terrain.
- Conception de la boîte noire : cette boîte est le cœur de l'outil qui permettra de donner des solutions aux responsables.

- Conception des interfaces qui permettront de piloter la boîte noire, de visualiser et d'extraire les solutions retournées.

Au début de nos travaux de thèse, nous avons d'abord entrepris la collecte des données. Cette phase nous a permis de rencontrer l'ensemble des acteurs de la chaîne concernée par les transports. Nous avons travaillé en collaboration avec la direction de l'accueil et de la logistique, la direction des achats, le responsable de la blanchisserie, le responsable de la restauration, et des unités de soins. De nombreuses réunions ont été nécessaires pour recueillir l'ensemble des informations utiles. Outre les réunions, nous avons suivi l'activité des chauffeurs au quotidien, en enregistrant les horaires de passage à chaque site, et les activités réalisées par les chauffeurs. Les informations recueillies sur le terrain ont été envoyées aux différents acteurs afin d'être validées. Nous avons ainsi construit une première base de données sur laquelle nous avons bâti l'outil.

L'outil que nous avons construit a pour objectif de répondre aux questions :

- quelle est le nombre de postes de chauffeurs ?
- qu'elle est le type de parc de véhicules dont on a besoin ?
- dans quel ordre réaliser les tâches ?
- pour chaque tâche, quel est le chauffeur qui la réalise et avec quel véhicule ?

Dans la suite de ce chapitre, nous présenterons les différents modules qui composent l'outil, et les différentes interfaces mises à la disposition du gestionnaire de la chaîne logistique.

6.2 Présentation de l'application : OPTA Logistics®

L'ensemble des étapes de la construction de l'outil d'aide à la décision est passé par des phases de réunions, de rencontres et de validation. Un cahier des charges a ainsi été construit.

Les principales fonctionnalités de l'outil sont :

- Le dimensionnement de l'équipe des chauffeurs, de la flotte de véhicules, et la construction de leurs plannings en fonction des activités de transport à réaliser.
- La mise en forme du résultat obtenu à l'aide des gants et de tableaux de bord dédiés à chaque ressource utilisée : véhicule, chauffeur.
- L'extraction sous forme de fichier Excel des solutions.

L'outil est conçu de manière à laisser le plus possible la main aux responsables, sauf sur la méthode de résolution et les sorties de l'outil. Les responsables ont accès aux données d'entrée, et à la possibilité de les modifier. Les données d'entrée peuvent être :

- ensemble de besoins,
- les types de véhicules,
- matrice de distance entre les sites des demandes,

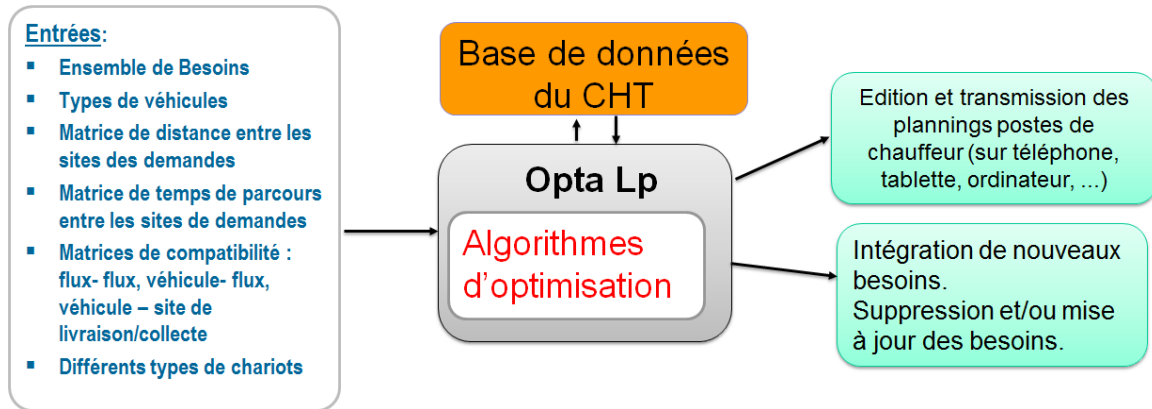


FIGURE 6.1 – Architecture d'OPTA Logistics.

- matrice de temps de parcours entre les sites de demandes,
- matrices de compatibilité : flux-flux, véhicule-flux, véhicule–site de livraison/collecte,
- types de chariots.

L'architecture de l'outil est donné par la figure 6.1. Afin de pouvoir manipuler (ajout, suppression, visualisation) les données d'entrée et les solutions de sorties, l'outil est composé d'une partie :

- "Accueil" résumant l'organisation en place.
- "Administration" pour gérer :
 - les utilisateurs,
 - les requêtes,
 - les sites des demandes,
 - les types de flux,
 - les types de camions,
 - les types de chariots,
 - les types de journées.
- "Mises à jour" des matrices de temps et de distances entre les différents sites de collecte/livraison.
- "Lancer une optimisation".
- "Planning" pour visualiser en détail les plannings.
- "Gant" pour visualiser synthétiquement un planning.
- "Extraction générale" pour extraire les solutions sous format Excel.

Dans la suite, nous abordons plus en détail les différentes parties de l'outil.

6.2.1 Accueil

Lorsque le gestionnaire de flotte se connecte à l'outil, il a une page d'accueil sur lequel on retrouve :

- l'évolution sur la semaine du nombre de véhicules utilisé,
- l'évolution sur la semaine du nombre de km parcourus,
- l'évolution sur la semaine du taux d'utilisation moyen des camions,
- la distance parcourue chaque jour par chaque camion.

Une image de l'écran d'accueil est donné par la figure 6.2.

6.2.2 Administration

Dans le module "Administration", on peut modifier et supprimer les données du problème étudié. En particulier pour les différents types de journées nous avons donné en plus des deux types de journées (*Journée continue* et *Journée standard*), une journée personnalisable (*Journée personnalisée*) par l'utilisateur. Une image de la page de modification des types de journées est donné par la figure 6.3.

6.2.3 Mises à jour des matrices de distances et de temps de parcours entre les sites

Le module de mise à jour des matrices de distances et de temps de parcours entre les sites est possible par une interface sur l'outil. Une image de cette interface est donné par l'image 6.4.

6.2.4 Lancer une optimisation

Pour lancer une optimisation, on a le choix de remise en question de la flotte de véhicules utilisés. L'on peut aussi autoriser pour un chauffeur le changement ou non de véhicule dans un planning. Une troisième possibilité est de choisir pour chaque jour de la semaine, un ordre de préférence du type de journée à utiliser. Une image de cette page est donnée par la figure 6.5.

6.2.5 Planning

L'outil donne la possibilité de visualiser le planning journalier d'un camion, et de faire une extraction Excel de ce dernier. Une image de cette page est donnée par la figure 6.6.

Le logiciel donne aussi la possibilité de visualiser sur une carte donne le planning d'un chauffeur. La carte est décomposée en deux parties : une visualisation du planning le matin et une autre l'après-midi. Une image d'une elle page est donné par la figure 6.7.

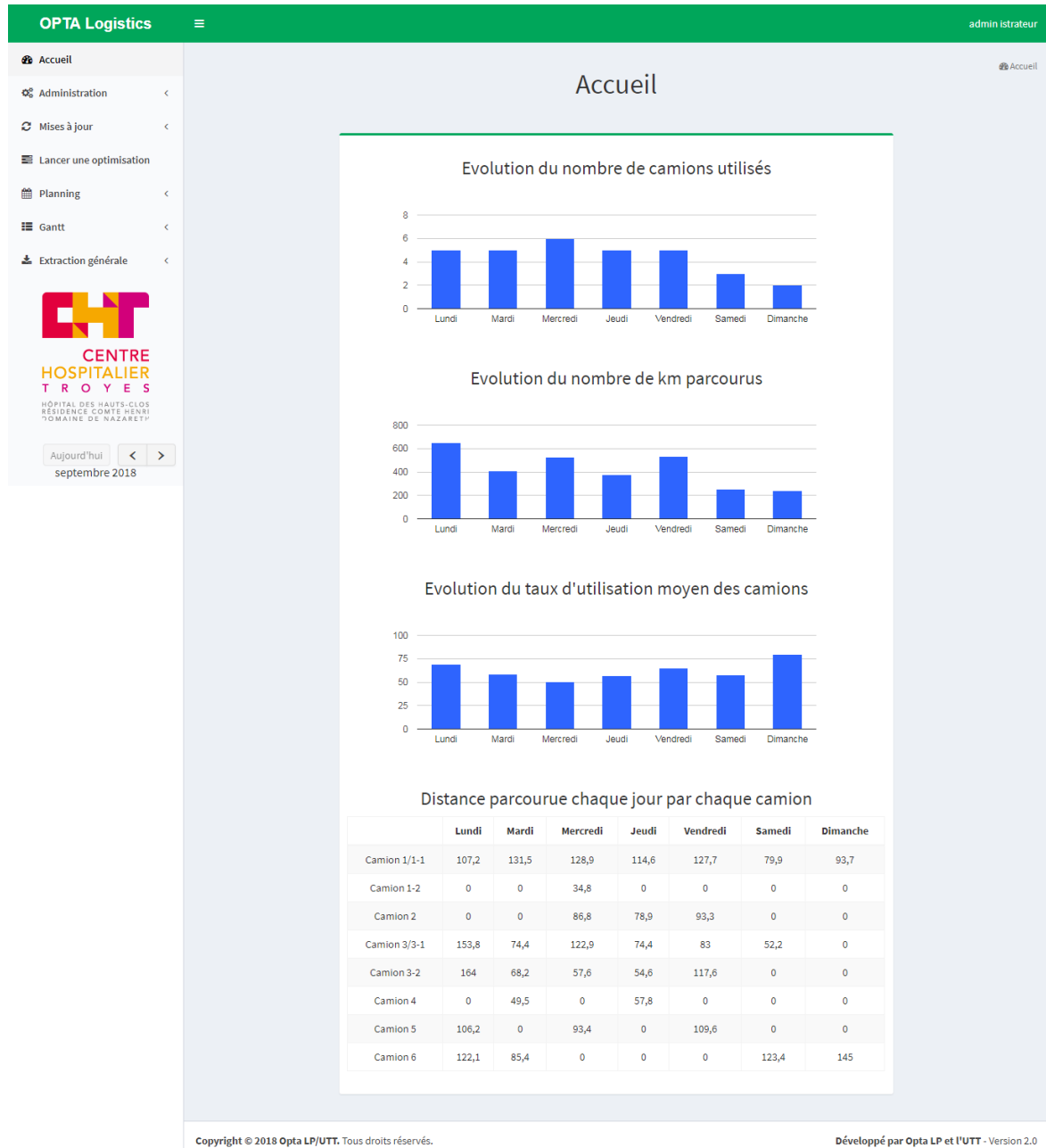


FIGURE 6.2 – Une image de l'écran d'accueil.

OPTA Logistics

admin istrateur

Accueil

Administration

Gestion des utilisateurs

Gestion des commandes

Gestion des sites

Gestion des flux

Gestion des types de camion

Gestion des camions

Gestion des chariots

Gestion des types de journée

Mises à jour

Lancer une optimisation

Planning

Gantt

Extraction générale

Accueil

Gestion des types de journée

Gestion des types de journée

Journée standard

Durée maximum de conduite

h

min

4

30

Durée maximum de travail

h

min

10

30

Durée minimum de la pause

h

min

2

30

Durée maximum de la pause

h

min

2

30

Temps de travail minimum avant la pause

h

min

3

0

Temps de travail minimum après la pause

h

min

3

0

Temps de travail maximum sans pause

h

min

6

0

Possibilité de faire la pause sur le lieu actuel

Non

Compter la pause dans le temps de travail

Non

Journée continue

Durée maximum de conduite

h

min

4

30

Durée maximum de travail

h

min

7

30

Durée minimum de la pause

h

min

0

20

Durée maximum de la pause

h

min

0

20

Temps de travail minimum avant la pause

h

min

0

0

Temps de travail minimum après la pause

h

min

0

0

Temps de travail maximum sans pause

h

min

7

30

Possibilité de faire la pause sur le lieu actuel

Oui

Compter la pause dans le temps de travail

Oui

Journée personnalisée

Durée maximum de conduite

h

min

4

30

Durée maximum de travail

h

min

10

30

Durée minimum de la pause

h

min

0

30

Durée maximum de la pause

h

min

3

0

Temps de travail minimum avant la pause

h

min

3

0

Temps de travail minimum après la pause

h

min

3

0

Temps de travail maximum sans pause

h

min

6

0

Possibilité de faire la pause sur le lieu actuel

Oui

Compter la pause dans le temps de travail

Oui

Valider

Copyright © 2018 Opta LP/UTT. Tous droits réservés.

Développé par Opta LP et l'UTT - Version 2.0

FIGURE 6.3 – Une image de la page des différents types de journée.

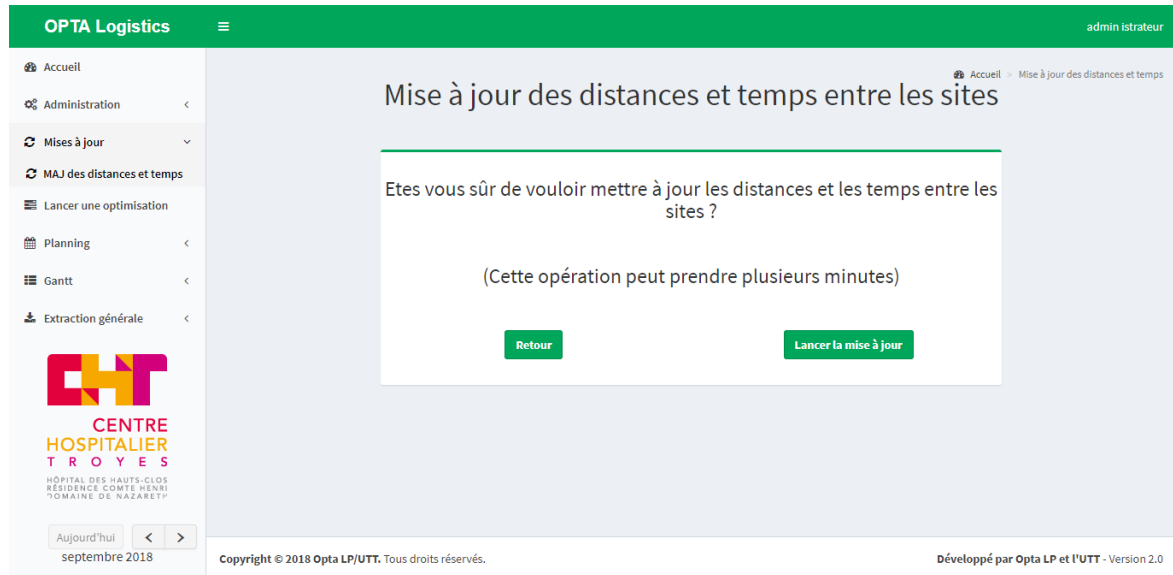


FIGURE 6.4 – Image de la page de mise à jour des des distances et des temps de parcours.

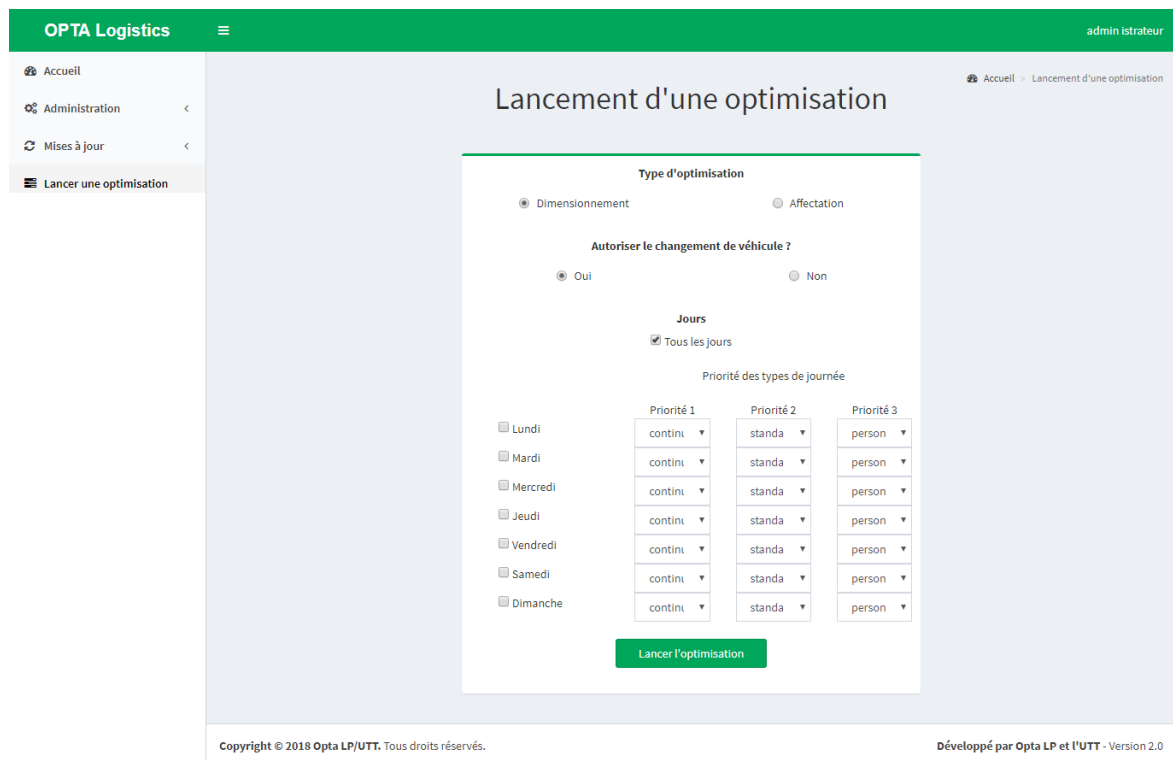


FIGURE 6.5 – Image de la page de lancement d'une optimisation.

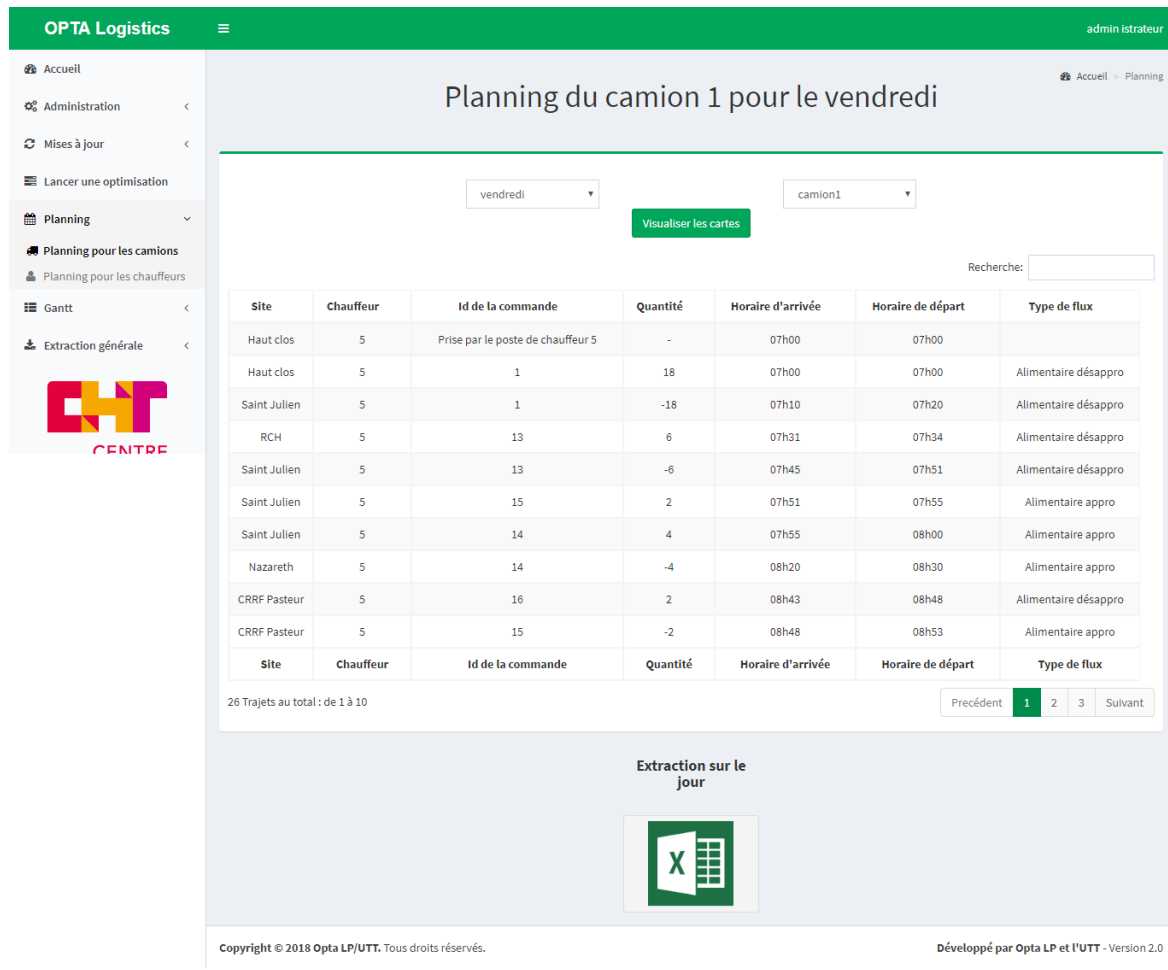


FIGURE 6.6 – Image d'un planning de camion pour une journée donnée.

6.2.6 Gant

L'outil permet une visualisation de l'ensemble des plannings des chauffeurs pour une journée. On y peut voir les différents moments (en service, en conduite, en pause) d'un chauffeur. Et donne aussi la possibilité d'une extraction sous format Excel. Et l'ensemble des plannings des camions peut être visualisé sur l'outil par un gant, avec la capacité d'une extraction Excel. Les images des différents gants sont données par les figures 6.8 et 6.9.

6.3 Conclusion

Dans ce chapitre nous avons présenté l'outil d'optimisation mis en place au sein du centre hospitalier de Troyes. Cet outil fait suite à la collaboration entre le laboratoire d'optimisation des systèmes industriels de l'université de technologie de Troyes, du centre hospitalier de Troyes et de la start-up Opta-Lp. Par nos travaux présentés dans les précédents chapitres, nous avons proposé et implémenté en Java des méthodes d'optimisation pour résoudre le problème

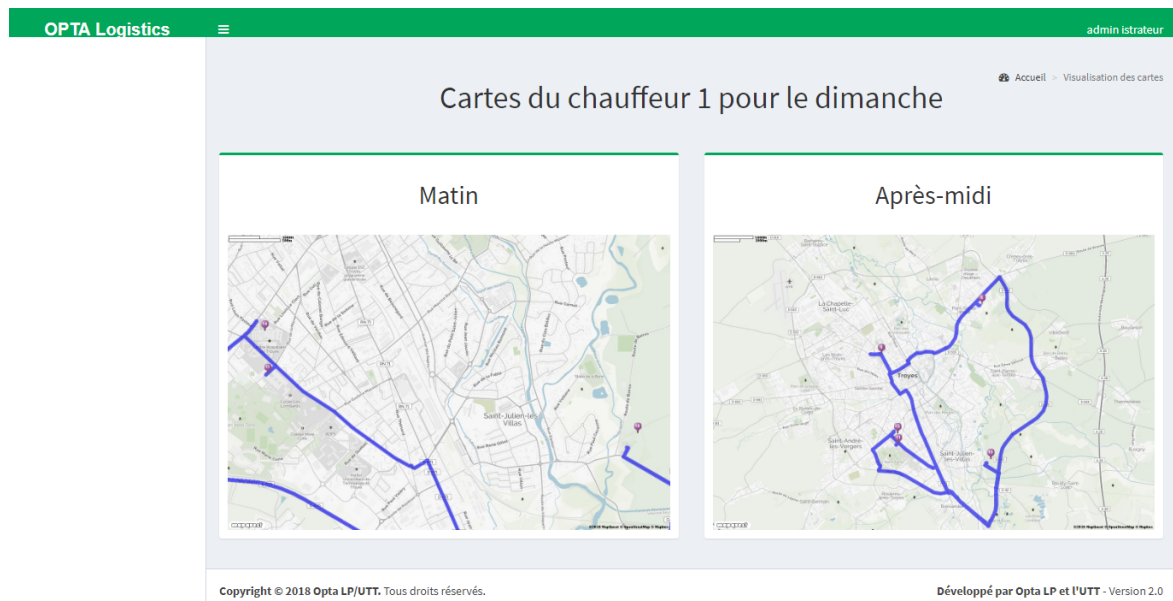


FIGURE 6.7 – Image des cartes d’un chauffeur pour une journée.

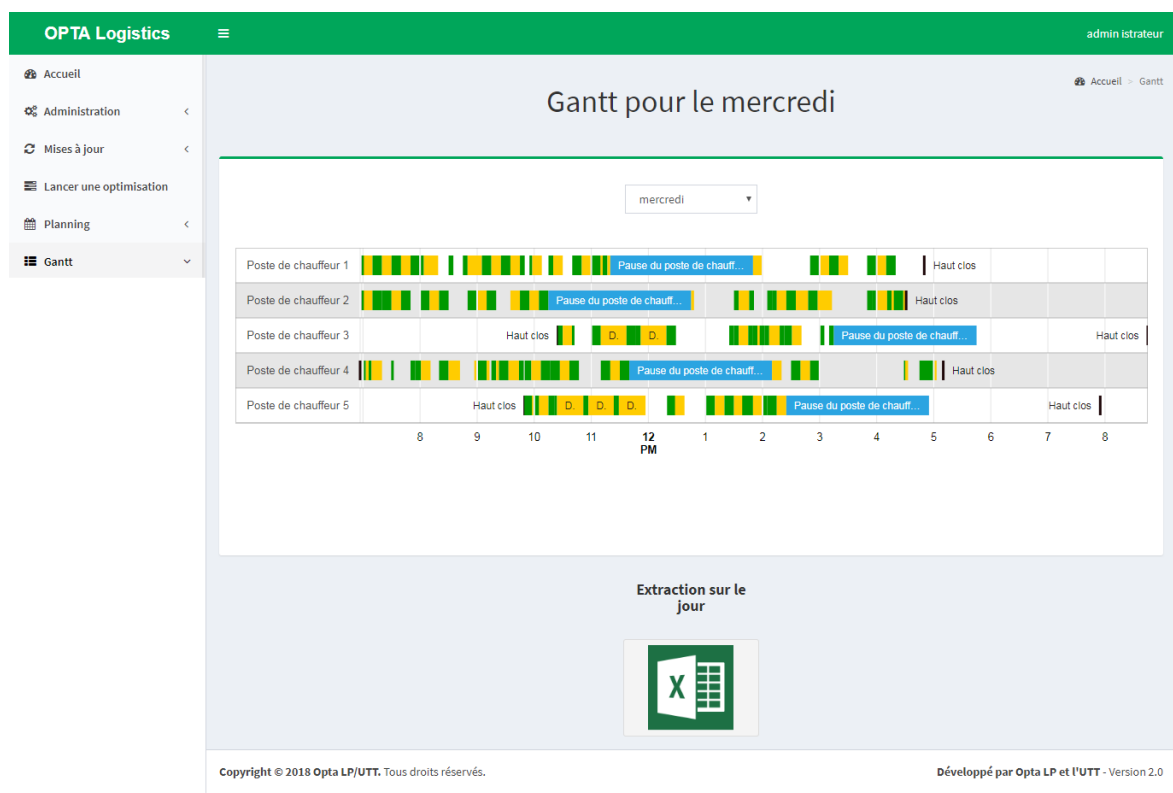


FIGURE 6.8 – Image du planning des chauffeurs pour une journée donnée.

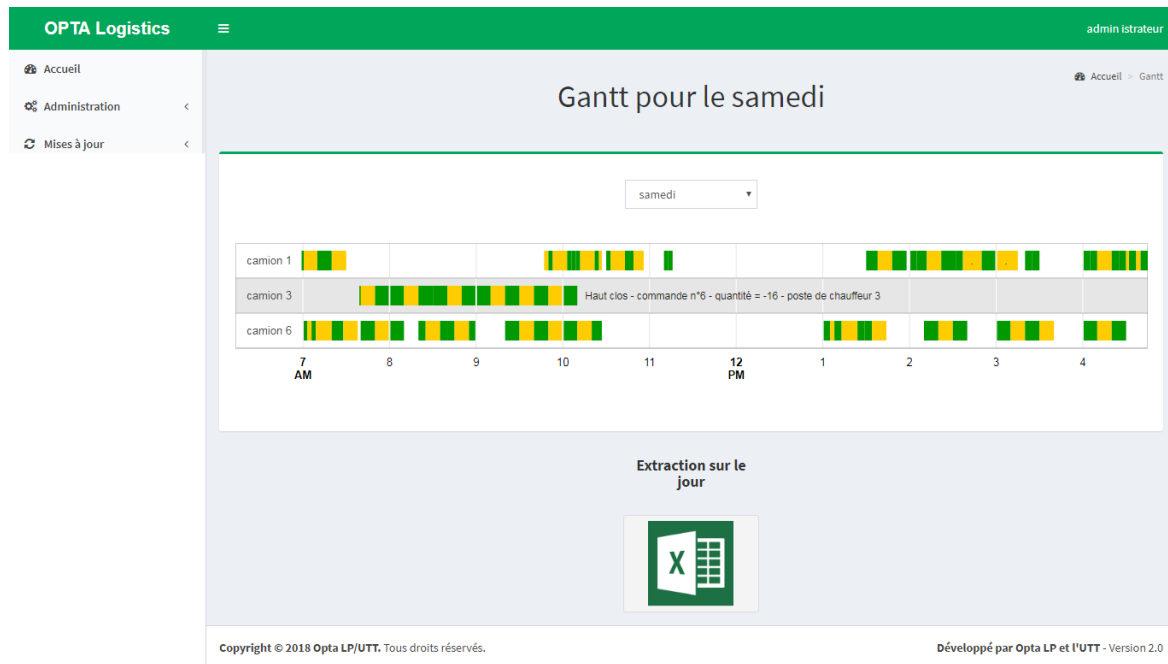


FIGURE 6.9 – Image du planning des camions pour une journée donné.

industriel. Et la start-up Opta-Lp a réalisé par les interfaces graphiques, et l'implémentation de l'outil dans le système d'information du centre hospitalier de Troyes. Cet outil a été mis à la disposition du personnel administratif en 2017 pour la première version.

Conclusion générale

Dans le cadre de cette thèse, nous avons étudié un problème conjoint de collecte et livraison couplé à la planification du personnel. L'étude de cette problématique découle des travaux d'optimisation de la chaîne logistique du Centre Hospitalier de Troyes. L'objectif de cette thèse a été de :

- réaliser une analyse de l'existant et une synthèse bibliographique sur les différentes problématiques de la logistique hospitalière,
- développer des techniques efficaces et rapides d'optimisation,
- mettre en place des outils d'aide à la décision, et d'accompagner le changement de l'organisation.

La problématique industrielle est l'optimisation de la réorganisation des activités de collecte et livraison des produits (repas, linge, médicaments, produits magasiniers). Les acteurs dans ce réseau logistique sont : des centres hospitaliers, des maisons de retraite, des cliniques psychiatriques, une cuisine, une blanchisserie, un magasin, une pharmacie, et un dépôt. L'horizon de planification retenu est à la semaine, car les besoins sont périodiques de période une semaine. Les coûts retenus dans notre étude sont : le coût induit par le nombre de chauffeur, le coût fixe induit par l'acquisition de la flotte des véhicules, le coût de carburant utilisé pour satisfaire l'ensemble des besoins. Le dimensionnement des équipes de chauffeurs est soumis à la réglementation sur les temps de conduite et de pauses des chauffeurs de véhicules de plus de 3T5 et à la réglementation de temps de travail d'un agent de la fonction publique hospitalière en France.

La problématique scientifique identifiée est celle d'un problème conjoint de collecte et livraison avec fenêtre de temps et la planification du personnel. Afin d'apporter des solutions efficaces à ce problème, nous avons adopté la démarche suivante :

- Dans un premier temps, nous avons traité le problème de collectes et livraisons avec fenêtres de temps et flotte homogène de véhicules. Nous avons proposé une formulation mathématique compacte du problème, et proposé des techniques de réduction de l'espace de recherche, et des inégalités valides pour accélérer la résolution du problème. Nous avons aussi proposé un algorithme hybride de colonie de fourmis. L'hybridation est réalisée par des heuristiques de recherche locale.

- Dans un second temps, nous avons traité des problèmes de collectes et livraisons avec fenêtres de temps et flotte hétérogène de véhicules. Nous avons distingué deux cas : le cas d'une flotte des véhicules de taille finie et cas d'une flotte des véhicules de taille infinie. Nous avons proposé un modèle de programmation linéaire en nombres mixtes pour modéliser le problème. Pour le résoudre, nous avons proposé un algorithme basé sur la décomposition de Benders, une méthode de génération de colonnes, et un algorithme hybride de colonie de fourmis permettant d'obtenir des solutions de qualité avec un temps de calcul plus faible.
- Au final, nous avons intégré au précédent problème les aspects et contraintes que l'on retrouve dans des problèmes industriels (les règles sur les temps de repos, de travail et de service des chauffeurs et des contraintes d'incompatibilité entre type de produit, type de véhicule, et site de collecte ou livraison). Nous avons d'une part proposé une modélisation en programme mathématique linéaire en nombres mixtes, et d'autre part, un algorithme hybride de colonie de fourmis. L'hybridation de la métaheuristique de colonie de fourmis s'est faite avec cinq algorithmes de recherche locale, chaque algorithme étant dédié à l'optimisation précise d'un composant de la fonction objective.

Au démarrage de cette étude, le service intérieur du CH de Troyes assurait les livraisons avec une flotte de 5 véhicules, ainsi que 5 postes de chauffeurs du lundi au vendredi, 3 chauffeurs le samedi et 2 chauffeurs le dimanche. Les véhicules concernés font l'objet de contrats de location pour un coût annuel qui était en 2015 de 89 215 euros par an. Les postes de chauffeurs étaient assurés par une équipe de 12 chauffeurs. Notre travail a permis en 2016 de mettre en place une nouvelle organisation permettant d'assurer les mêmes demandes avec un chauffeur de moins du lundi au samedi. Cette nouvelle organisation a représenté un gain d'un poste de chauffeur sur l'équipe, soit un gain de plus de 10%. Un autre effet de cette réorganisation a été la réduction du temps annuel d'attente des chauffeurs entre deux trajets de 1 524 heures à 432 heures. La réussite de ces actions d'optimisation au CH de Troyes illustre l'ampleur du potentiel d'amélioration des fonctions logistiques existant dans les établissements des Hôpitaux Champagne Sud.

Les résultats obtenus suite à nos travaux ont permis de mettre en place une nouvelle organisation à chaque modification du réseau logistique (ajout, modification, ou suppression d'une requête).

En ce qui concerne les méthodes de résolution du problème étudié, plusieurs perspectives peuvent être envisagées :

- l'amélioration des étapes (construction d'une solution, mise à jour des phéromones, recherche locale) de l'algorithme hybride de colonie de fourmis.
- une approche basée sur un algorithme de génération de colonnes, où une colonne serait un planning est une piste intéressante.

- une approche basée sur la génération des contraintes est aussi une piste intéressante, car elle permet de prendre en compte qu'une partie des contraintes.

Les perspectives sur le problème étudié sont nombreuses :

- la prise en compte des conditions de travail des chauffeurs (quels indicateurs de performance ?),
- l'introduction des différents aléas (temps de conduite, retards dans les unités de fabrication, ...),
- la prise en compte en amont de la chaîne de la problématique de gestion stock,
- la prise en aval de l'acheminement des produits depuis les quais des sites jusque dans les unités de soins.

Chapitre 7

Annexe

Instance	HACO		MILP		Gap (%)	
	Meilleure coût	Temps	UB	LB	HACO vs UB	HACO vs LB
<i>FLT20C1_1</i>	40968,95	4	40968,95	32775,16	0	25
<i>FLT20C1_2</i>	30953,92	8	30953,92	24763,13	0	25
<i>FLT20C1_3</i>	30949,35	14	30972,52	21680,76	-0,07	42,75
<i>FLT20C1_4</i>	20926,56	13	20927,65	18834,89	-0,01	11,11
<i>FLT20C1_5</i>	40961,56	8	40982,60	20491,30	-0,05	99,90
<i>FLT20C1_6</i>	41426,33	5	41426,33	28998,43	0	42,86
<i>FLT20C1_7</i>	40959,17	9	40959,17	32767,33	0	25
<i>FLT20C1_8</i>	31016,82	11	31019,55	27917,59	-0,01	11,10
<i>FLT20C1_9</i>	20933,84	18	20936,98	16749,58	-0,02	24,98
Moyenne :					-0,02	34,19

TABLE 7.1 – Expérimentations HACO et MILP sur le benchmark à 20 demandes FLT20C1.

Instance	HACO		MILP		Gap (%)	
	Meilleure coût	Temps	UB	LB	HACO vs UB	HACO vs LB
<i>FLT20C2_1</i>	61424,15	5	61424,15	55281,74	0	11,11
<i>FLT20C2_2</i>	61536,41	6	61536,41	43075,49	0	42,86
<i>FLT20C2_3</i>	40961,73	8	40961,84	36865,65	0	11,11
<i>FLT20C2_4</i>	20600,41	17	30978,25	18586,95	-33,50	10,83
<i>FLT20C2_5</i>	31032,71	7	31032,71	27929,44	0	11,11
<i>FLT20C2_6</i>	31013,01	7	31013,46	15506,73	0	100
<i>FLT20C2_7</i>	50996,42	6	50996,42	40797,14	0	25
<i>FLT20C2_8</i>	41011,19	7	41011,19	36910,07	0	11,11
Moyenne :					-4,19	27,89

TABLE 7.2 – Expérimentations HACO et MILP sur le benchmark à 20 demandes FLT20C2.

id demande	id site	quantité	horaire au plus tôt	horaire au plus tard	temps de service	id collecte/livraison	id collecte/livraison	type de flux
0	6	0	0	1440	0	0	0	0
1	6	18	450	460	0	0	2	1
2	12	-18	460	470	10	1	0	1
3	6	9	480	500	10	0	4	1
4	12	-9	490	510	10	3	0	1
5	12	11	540	560	10	0	6	0
6	6	-11	550	570	10	5	0	0
7	12	13	590	610	10	0	8	0
8	6	-13	600	620	10	7	0	0
9	6	15	810	830	10	0	10	1
10	12	-15	820	840	10	9	0	1
11	6	12	840	860	10	0	12	1
12	12	-12	850	870	10	11	0	1
13	12	7	900	920	10	0	14	0
14	6	-7	910	930	10	13	0	0
15	12	14	960	980	10	0	16	0
16	6	-14	970	990	10	15	0	0
17	6	2	420	480	3	0	18	1
18	12	-2	430	480	2	17	0	1
19	10	6	420	480	3	0	20	1
20	12	-6	430	480	6	19	0	1
21	6	9	460	480	10	0	22	1
22	12	-9	470	480	10	21	0	1
23	12	4	500	520	5	0	24	0
24	9	-4	510	530	5	23	0	0
25	9	4	510	530	5	0	26	1
26	12	-4	530	550	5	25	0	1
27	12	11	560	580	10	0	28	0
28	6	-11	570	590	10	27	0	0
29	12	2	580	610	3	0	30	0
30	3	-2	590	630	3	29	0	0
31	3	2	590	630	3	0	32	1
32	6	-2	670	690	3	31	0	1
33	12	6	580	610	3	0	34	0
34	10	-6	630	650	3	33	0	1
35	12	2	580	610	3	0	36	0
36	4	-2	630	650	3	35	0	0
37	4	2	630	650	3	0	38	1
38	6	-2	670	690	3	37	0	1
39	6	4	780	790	5	0	40	1
40	12	-4	790	810	5	39	0	1
41	10	6	780	790	5	0	42	1
42	12	-6	790	810	5	41	0	1
43	12	8	840	860	7	0	44	0
44	9	-8	850	870	10	43	0	0
45	9	8	850	890	5	0	46	1
46	12	-8	900	940	5	45	0	1
47	12	2	840	860	5	0	48	0
48	4	-2	890	910	5	47	0	0
49	4	2	890	910	5	0	50	1
50	12	-2	920	940	5	49	0	1
51	12	2	960	970	5	0	52	0
52	3	-2	970	990	5	51	0	0
53	3	2	970	990	5	0	54	1
54	6	-2	980	1010	5	53	0	1
55	12	6	960	970	5	0	56	0
56	10	-6	980	1010	5	55	0	0

FIGURE 7.1 – Exemple d'une instance réelle (celle de dimanche).

Instance	HACO		MILP		Gap (%)	
	Meilleure coût	Temps	UB	LB	HACO vs UB	HACO vs LB
<i>FLT20R1_1</i>	31474,53	5	31496,09	22047,26	-0,07	42,76
<i>FLT20R1_2</i>	41003,62	8	41008,27	36907,44	-0,01	11,10
<i>FLT20R1_3</i>	20654,89	10	20662,06	16529,64	-0,03	24,96
<i>FLT20R1_4</i>	30997,51	11	30997,51	27897,76	0	11,11
<i>FLT20R1_5</i>	30996,78	8	31015,29	21710,70	-0,06	42,77
<i>FLT20R1_6</i>	21025,47	9	21047,60	16838,08	-0,11	24,87
<i>FLT20R1_7</i>	20663,23	25	20670,73	16536,59	-0,04	24,95
<i>FLT20R1_8</i>	20623,33	28	20624,78	14437,35	-0,01	42,85
<i>FLT20R1_9</i>	20695,11	20	20998,23	16798,59	-1,44	23,20
<i>FLT20R1_10</i>	20966,97	21	20979,56	14685,69	-0,06	42,77
<i>FLT20R1_11</i>	20984,68	12	20984,68	16787,75	0	25
<i>FLT20R1_12</i>	20983,88	48	20989,32	10494,66	-0,03	99,95
Moyenne :					-0,15	34,69

TABLE 7.3 – Expérimentations HACO et MILP sur le benchmark à 20 demandes FLT20R1.

Instance	HACO		MILP		Gap (%)	
	Meilleure coût	Temps	UB	LB	HACO vs UB	HACO vs LB
<i>FLT20R2_1</i>	41490,36	5	41500,60	37350,54	-0,02	11,08
<i>FLT20R2_2</i>	41125,34	9	41158,63	24695,18	-0,08	66,53
<i>FLT20R2_3</i>	21109,09	10	21111,34	12666,80	-0,01	66,65
<i>FLT20R2_4</i>	20688,85	18	20995,84	18896,26	-1,46	9,49
<i>FLT20R2_5</i>	21035,48	10	31072,04	18643,22	-32,30	12,83
<i>FLT20R2_6</i>	20677,78	15	21015,73	18914,15	-1,61	9,32
<i>FLT20R2_7</i>	20618,73	23	20627,84	12376,71	-0,04	66,59
<i>FLT20R2_8</i>	20624,38	22	20656,87	16525,50	-0,16	24,80
<i>FLT20R2_9</i>	20641,87	12	20647,21	12388,33	-0,03	66,62
<i>FLT20R2_10</i>	30987,21	12	30990,78	18594,47	-0,01	66,65
<i>FLT20R2_11</i>	20631,53	30	20634,47	12380,68	-0,01	66,64
Moyenne :					-3,25	42,47

TABLE 7.4 – Expérimentations HACO et MILP sur le benchmark à 20 demandes FLT20R2.

Instance	HACO		MILP		Gap (%)	
	Meilleure coût	Temps	UB	LB	HACO vs UB	HACO vs LB
<i>FLT20RC1_1</i>	21024,59	7	21024,59	14717,22	0	42,86
<i>FLT20RC1_2</i>	21007,69	8	21007,69	14705,38	0	42,86
<i>FLT20RC1_3</i>	20617,18	15	20652,14	12391,28	-0,17	66,38
<i>FLT20RC1_4</i>	20668,95	23	20676,93	18609,24	-0,04	11,07
<i>FLT20RC1_5</i>	31056,52	9	31056,87	21739,81	0	42,86
<i>FLT20RC1_6</i>	20971,30	23	20971,30	18874,17	0	11,11
<i>FLT20RC1_7</i>	20690,22	29	20989,54	18890,59	-1,43	9,53
<i>FLT20RC1_8</i>	10574,69	26	20964,34	10482,17	-49,56	0,88
Moyenne :					-6,40	28,44

TABLE 7.5 – Expérimentations HACO et MILP sur le benchmark à 20 demandes FLT20RC1.

Instance	HACO		MILP		Gap (%)	
	Meilleure coût	Temps	UB	LB	HACO vs UB	HACO vs LB
<i>FLT20RC2_1</i>	31072,40	10	31072,40	18610,65	0	66,96
<i>FLT20RC2_2</i>	20629,03	8	30991,91	18643,44	-33,44	10,65
<i>FLT20RC2_3</i>	10573,84	13	20964,48	15495,96	-49,56	-31,76
<i>FLT20RC2_4</i>	20670,47	33	20670,47	16771,58	0	23,25
<i>FLT20RC2_5</i>	20639,24	7	20639,24	10335,24	0	99,70
<i>FLT20RC2_6</i>	20952,57	24	20959,97	12383,54	-0,04	69,20
<i>FLT20RC2_7</i>	20630,62	19	20630,62	14671,98	0	40,61
<i>FLT20RC2_8</i>	20630,62	37	20630,62	18567,56	0	11,11
	20724,85			Moyenne :	-10,38	36,21

TABLE 7.6 – Expérimentations HACO et MILP sur le benchmark à 20 demandes FLT20RC2.

Bibliographie

- [1] V. André, “Problème de livraison-collecte dans un environnement hospitalier : méthodes d’optimisation, modèle de simulation et couplages,” Ph.D. dissertation, Université Blaise Pascal-Clermont-Ferrand II, 2011.
- [2] J. Volland, A. Fügner, J. Schoenfelder, and J. O. Brunner, “Material logistics in hospitals : a literature review,” *Omega*, vol. 69, pp. 82–101, 2017.
- [3] H. Li and A. Lim, “A metaheuristic for the pickup and delivery problem with time windows,” *International Journal on Artificial Intelligence Tools*, vol. 12, no. 02, pp. 173–186, 2003.
- [4] G. Chow, T. D. Heaven, and L. E. Henriksson, “Logistics performance : definition and measurement,” *International journal of physical distribution & logistics management*, vol. 24, no. 1, pp. 17–28, 1994.
- [5] J. Ferebee, “Controlling fixed-route operations,” *Industrial Engineering*, vol. 6, no. 10, pp. 28–31, 1974.
- [6] M. Ball, L. Bodin, and R. Dial, “A matching based heuristic for scheduling mass transit crews and vehicles,” *Transportation Science*, vol. 17, no. 1, pp. 4–31, 1983.
- [7] W. Henning, “The financial impact of materials management.” *Hospital financial management*, vol. 34, no. 2, pp. 36–42, 1980.
- [8] J. C. Kowalski, “Materials management crucial to overall efficiency.” *Healthcare financial management : journal of the Healthcare Financial Management Association*, vol. 45, no. 1, pp. 40–42, 1991.
- [9] B. Bourgeon, A. Constantin, G. Karolszyk, J.-F. Marquot, S. Pedrini, S. Landry, A. Diaz, and D. Estampe, “Evaluation des coûts logistiques hospitaliers en france et aux pays-bas,” *Logistique & Management*, vol. 9, no. 1, pp. 81–87, 2001.
- [10] R. N. Davis, “No more chances for supply-chain savings ? look again ! many hospitals assume they have exhausted opportunities to reduce supply-chain costs. but a well-

- coordinated, interdisciplinary effort may still yield substantial savings,” *Healthcare Financial Management*, vol. 58, no. 1, pp. 68–76, 2004.
- [11] N. Sampieri-Teissier, “Valoriser la logistique hospitalière au service des patients,” *Supply Chain Magazine*, no. 102, pp. 144–145, 2016.
- [12] Y. Kergosien, C. Lenté, D. Piton, and J.-C. Billaut, “A tabu search heuristic for a dynamic transportation problem of patients between care units,” 2010.
- [13] B. E. Fries, “Bibliography of operations research in health-care systems,” *Operations Research*, vol. 24, no. 5, pp. 801–814, 1976.
- [14] J. Rosenhead, “Operational research in health services planning,” *European Journal of Operational Research*, vol. 2, no. 2, pp. 75–85, 1978.
- [15] D. P. Boldy and P. C. O’Kane, “Health operational research, a selective overview,” *European Journal of Operational Research*, vol. 10, no. 1, pp. 1–9, 1982.
- [16] D. Boldy, “The relationship between decision support systems and operational research : health care examples,” *European Journal of Operational Research*, vol. 29, no. 2, pp. 128–134, 1987.
- [17] V. C. Wiers, “A review of the applicability of or and ai scheduling techniques in practice,” *Omega*, vol. 25, no. 2, pp. 145–153, 1997.
- [18] Y. Shi, T. Boudouh, and O. Grunder, “A hybrid genetic algorithm for a home health care routing problem with time window and fuzzy demand,” *Expert Systems with Applications*, vol. 72, pp. 160–176, 2017.
- [19] B. Cardoen, E. Demeulemeester, and J. Beliën, “Operating room planning and scheduling : A literature review,” *European journal of operational research*, vol. 201, no. 3, pp. 921–932, 2010.
- [20] E. K. Burke, P. De Causmaecker, G. V. Berghe, and H. Van Landeghem, “The state of the art of nurse rostering,” *Journal of scheduling*, vol. 7, no. 6, pp. 441–499, 2004.
- [21] L. Li and W. Benton, “Performance measurement criteria in health care organizations : review and future research directions,” *European Journal of Operational Research*, vol. 93, no. 3, pp. 449–468, 1996.
- [22] M. J. Liberatore and R. L. Nydick, “The analytic hierarchy process in medical and health care decision making : A literature review,” *European Journal of Operational Research*, vol. 189, no. 1, pp. 194–207, 2008.

-
- [23] Y. Kergosien, “Algorithmes de tournées de véhicules pour l’optimisation des flux de produits et de patients dans un complexe hospitalier,” Ph.D. dissertation, Université François Rabelais-Tours, 2010.
 - [24] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte, “Static pickup and delivery problems : a classification scheme and survey,” *Top*, vol. 15, no. 1, pp. 1–31, 2007.
 - [25] S. Anily and R. Hassin, “The swapping problem,” *Networks*, vol. 22, no. 4, pp. 419–433, 1992.
 - [26] H. Hernández-Pérez and J.-J. Salazar-González, “A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery,” *Discrete Applied Mathematics*, vol. 145, no. 1, pp. 126–139, 2004.
 - [27] H. Hernández-Pérez and J.-J. Salazar-González, “Heuristics for the one-commodity pickup-and-delivery traveling salesman problem,” *Transportation Science*, vol. 38, no. 2, pp. 245–255, 2004.
 - [28] P. Chalasani and R. Motwani, “Approximating capacitated routing and delivery problems,” *SIAM Journal on Computing*, vol. 28, no. 6, pp. 2133–2149, 1999.
 - [29] S. Anily and J. Bramel, “Approximation algorithms for the capacitated traveling salesman problem with pickups and deliveries,” *Naval Research Logistics (NRL)*, vol. 46, no. 6, pp. 654–670, 1999.
 - [30] F. Wang, A. Lim, and Z. Xu, “The one-commodity pickup and delivery travelling salesman problem on a path or a tree,” *Networks : An International Journal*, vol. 48, no. 1, pp. 24–35, 2006.
 - [31] I. Gribkovskaia, Ø. Halskau sr, G. Laporte, and M. Vlček, “General solutions to the single vehicle routing problem with pickups and deliveries,” *European Journal of Operational Research*, vol. 180, no. 2, pp. 568–584, 2007.
 - [32] M. Dror and P. Trudeau, “Savings by split delivery routing,” *Transportation Science*, vol. 23, no. 2, pp. 141–145, 1989.
 - [33] N. Katoh and T. Yano, “An approximation algorithm for the pickup and delivery vehicle routing problem on trees,” *Discrete Applied Mathematics*, vol. 154, no. 16, pp. 2335–2349, 2006.
 - [34] A. Hoff and A. Løkketangen, “Creating lasso-solutions for the traveling salesman problem with pickup and delivery by tabu search,” *Central European Journal of Operations Research*, vol. 14, no. 2, pp. 125–140, 2006.

- [35] J. A. Chisman, "The clustered traveling salesman problem," *Computers & Operations Research*, vol. 2, no. 2, pp. 115–119, 1975.
- [36] M. Gendreau, G. Laporte, and A. Hertz, "An approximation algorithm for the traveling salesman problem with backhauls," *Operations Research*, vol. 45, no. 4, pp. 639–641, 1997.
- [37] H. Ghaziri and I. H. Osman, "A neural network algorithm for the traveling salesman problem with backhauls," *Computers & Industrial Engineering*, vol. 44, no. 2, pp. 267–281, 2003.
- [38] G. Mosheiov, "The travelling salesman problem with pick-up and delivery," *European Journal of Operational Research*, vol. 79, no. 2, pp. 299–310, 1994.
- [39] R. Baldacci, E. Hadjiconstantinou, and A. Mingozzi, "An exact algorithm for the traveling salesman problem with deliveries and collections," *Networks : An International Journal*, vol. 42, no. 1, pp. 26–41, 2003.
- [40] H. Süral and J. H. Bookbinder, "The single-vehicle routing problem with unrestricted backhauls," *Networks : An International Journal*, vol. 41, no. 3, pp. 127–136, 2003.
- [41] A. Alshamrani, K. Mathur, and R. H. Ballou, "Reverse logistics : simultaneous design of delivery routes and returns strategies," *Computers & Operations Research*, vol. 34, no. 2, pp. 595–619, 2007.
- [42] H. Min, "The multiple vehicle routing problem with simultaneous delivery and pick-up points," *Transportation Research Part A : General*, vol. 23, no. 5, pp. 377–386, 1989.
- [43] G. Nagy and S. Salhi, "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries," *European journal of operational research*, vol. 162, no. 1, pp. 126–141, 2005.
- [44] J.-F. Chen and T.-H. Wu, "Vehicle routing problem with simultaneous deliveries and pickups," *Journal of the Operational Research Society*, vol. 57, no. 5, pp. 579–587, 2006.
- [45] N. Bianchessi and G. Righini, "Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery," *Computers & Operations Research*, vol. 34, no. 2, pp. 578–594, 2007.
- [46] F. A. T. Montané and R. D. Galvao, "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service," *Computers & Operations Research*, vol. 33, no. 3, pp. 595–619, 2006.
- [47] J. Dethloff, "Vehicle routing and reverse logistics : the vehicle routing problem with simultaneous delivery and pick-up," *OR-Spektrum*, vol. 23, no. 1, pp. 79–96, 2001.

-
- [48] M. Dell Amico, G. Righini, and M. Salani, "A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection," *Transportation science*, vol. 40, no. 2, pp. 235–247, 2006.
- [49] E. Angelelli and R. Mansini, "The vehicle routing problem with time windows and simultaneous pick-up and delivery," in *Quantitative approaches to distribution logistics and supply chain management*. Springer, 2002, pp. 249–267.
- [50] I. Deif and L. Bodin, "Extension of the clarke and wright algorithm for solving the vehicle routing problem with backhauling," in *Proceedings of the Babson conference on software uses in transportation and logistics management*. Babson Park, MA, 1984, pp. 75–96.
- [51] M. Goetschalckx and C. Jacobs-Blecha, "The vehicle routing problem with backhauls," *European Journal of Operational Research*, vol. 42, no. 1, pp. 39–51, 1989.
- [52] P. Toth and D. Vigo, "A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls," *European Journal of Operational Research*, vol. 113, no. 3, pp. 528–543, 1999.
- [53] I. H. Osman and N. A. Wassan, "A reactive tabu search meta-heuristic for the vehicle routing problem with back-hauls," *Journal of Scheduling*, vol. 5, no. 4, pp. 263–285, 2002.
- [54] J. Brandao, "A new tabu search algorithm for the vehicle routing problem with backhauls," *European Journal of Operational Research*, vol. 173, no. 2, pp. 540–555, 2006.
- [55] C. A. Yano, T. J. Chan, L. K. Richter, T. Cutler, K. G. Murty, and D. McGettigan, "Vehicle routing at quality stores," *Interfaces*, vol. 17, no. 2, pp. 52–63, 1987.
- [56] P. Toth and D. Vigo, "An exact algorithm for the vehicle routing problem with backhauls," *Transportation science*, vol. 31, no. 4, pp. 372–385, 1997.
- [57] A. Mingozzi, S. Giorgi, and R. Baldacci, "An exact method for the vehicle routing problem with backhauls," *Transportation Science*, vol. 33, no. 3, pp. 315–329, 1999.
- [58] S. Gélinas, M. Desrochers, J. Desrosiers, and M. M. Solomon, "A new branching strategy for time constrained routing problems with application to backhauling," *Annals of Operations Research*, vol. 61, no. 1, pp. 91–109, 1995.
- [59] J.-Y. Potvin and S. Bengio, "The vehicle routing problem with time windows part ii : genetic search," *INFORMS journal on Computing*, vol. 8, no. 2, pp. 165–172, 1996.
- [60] C. Duhamel, J.-Y. Potvin, and J.-M. Rousseau, "A tabu search heuristic for the vehicle routing problem with backhauls and time windows," *Transportation science*, vol. 31, no. 1, pp. 49–59, 1997.

- [61] S. R. Thangiah, J.-Y. Potvin, and T. Sun, "Heuristic approaches to vehicle routing with backhauls and time windows," *Computers and Operations Research*, vol. 23, no. 11, pp. 1043–1058, 1996.
- [62] S. Ropke and D. Pisinger, "A unified heuristic for a large class of vehicle routing problems with backhauls," *European Journal of Operational Research*, vol. 171, no. 3, pp. 750–775, 2006.
- [63] G. Kontoravdis and J. F. Bard, "A grasp for the vehicle routing problem with time windows," *ORSA journal on Computing*, vol. 7, no. 1, pp. 10–23, 1995.
- [64] Y. Zhong and M. H. Cole, "A vehicle routing problem with backhauls and time windows : a guided local search solution," *Transportation Research Part E : Logistics and Transportation Review*, vol. 41, no. 2, pp. 131–144, 2005.
- [65] A. Wade and S. Salhi, "An investigation into a new class of vehicle routing problem with backhauls," *Omega*, vol. 30, no. 6, pp. 479–487, 2002.
- [66] G. N. Frederickson and D. Guan, "Preemptive ensemble motion planning on a tree," *SIAM Journal on Computing*, vol. 21, no. 6, pp. 1130–1152, 1992.
- [67] M. J. Atallah and S. R. Kosaraju, "Efficient solutions to some transportation problems with applications to minimizing robot arm travel," *SIAM Journal on Computing*, vol. 17, no. 5, pp. 849–869, 1988.
- [68] T. R. Sexton and L. D. Bodin, "Optimizing single vehicle many-to-many operations with desired delivery times : I. scheduling," *Transportation Science*, vol. 19, no. 4, pp. 378–410, 1985.
- [69] G. Brønmo, M. Christiansen, K. Fagerholt, and B. Nygreen, "A multi-start local search heuristic for ship scheduling - a computational study," *Computers & Operations Research*, vol. 34, no. 3, pp. 900–917, 2007.
- [70] S. Mitrović-Minić and G. Laporte, "Waiting strategies for the dynamic pickup and delivery problem with time windows," *Transportation Research Part B : Methodological*, vol. 38, no. 7, pp. 635–655, 2004.
- [71] T. R. Sexton and Y.-M. Choi, "Pickup and delivery of partial loads with "soft" time windows," *American Journal of Mathematical and Management Sciences*, vol. 6, no. 3-4, pp. 369–398, 1986.
- [72] J. Beardwood, J. H. Halton, and J. M. Hammersley, "The shortest path through many points," in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 55, no. 4. Cambridge University Press, 1959, pp. 299–327.

-
- [73] H. N. Psaraftis, "A multi-commodity, capacitated pickup and delivery problem : The single and two-vehicle cases," *European Journal of Operational Research*, vol. 215, no. 3, pp. 572–580, 2011.
- [74] M. Kubo and H. Kasugai, "Heuristic algorithms for the single vehicle dial-a-ride problem," *Journal of the Operations Research Society of Japan*, vol. 33, no. 4, pp. 354–365, 1990.
- [75] L. Van der Bruggen, J. K. Lenstra, and P. Schuur, "Variable-depth search for the single-vehicle pickup and delivery problem with time windows," *Transportation Science*, vol. 27, no. 3, pp. 298–311, 1993.
- [76] B. Kalantari, A. V. Hill, and S. R. Arora, "An algorithm for the traveling salesman problem with pickup and delivery customers," *European Journal of Operational Research*, vol. 22, no. 3, pp. 377–386, 1985.
- [77] K. Ruland and E. Rodin, "The pickup and delivery problem : Faces and branch-and-cut algorithm," *Computers & mathematics with applications*, vol. 33, no. 12, pp. 1–13, 1997.
- [78] J. Pacheco, "Metaheuristic based on a simulated annealing process for one vehicle pickup and delivery problem in lifo unloading systems," in *Proceedings of the Tenth Meeting of the European Chapter of Combinatorial Optimization (ECCO X). Tenerife, Spain*, 1997.
- [79] F. Carrabs, J.-F. Cordeau, and G. Laporte, "Variable neighborhood search for the pickup and delivery traveling salesman problem with lifo loading," *INFORMS Journal on Computing*, vol. 19, no. 4, pp. 618–632, 2007.
- [80] H. N. Psaraftis, "A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem," *Transportation Science*, vol. 14, no. 2, pp. 130–154, 1980.
- [81] Y. Dumas, J. Desrosiers, and F. Soumis, "The pickup and delivery problem with time windows," *European journal of operational research*, vol. 54, no. 1, pp. 7–22, 1991.
- [82] M. Savelsbergh and M. Sol, "Drive : Dynamic routing of independent vehicles," *Operations Research*, vol. 46, no. 4, pp. 474–490, 1998.
- [83] H. Xu, Z.-L. Chen, S. Rajagopal, and S. Arunapuram, "Solving a practical pickup and delivery problem," *Transportation science*, vol. 37, no. 3, pp. 347–364, 2003.
- [84] S. Ropke, J.-F. Cordeau, and G. Laporte, "Models and branch-and-cut algorithms for pickup and delivery problems with time windows," *Networks : An International Journal*, vol. 49, no. 4, pp. 258–272, 2007.

- [85] W. P. Nanry and J. W. Barnes, "Solving the pickup and delivery problem with time windows using reactive tabu search," *Transportation Research Part B : Methodological*, vol. 34, no. 2, pp. 107–121, 2000.
- [86] H. C. Lau and Z. Liang, "Pickup and delivery with time windows : Algorithms and test case generation," *International Journal on Artificial Intelligence Tools*, vol. 11, no. 03, pp. 455–472, 2002.
- [87] R. Bent and P. Van Hentenryck, "A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows," *Computers & Operations Research*, vol. 33, no. 4, pp. 875–893, 2006.
- [88] J.-J. Jaw, A. R. Odoni, H. N. Psaraftis, and N. H. Wilson, "A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows," *Transportation Research Part B : Methodological*, vol. 20, no. 3, pp. 243–257, 1986.
- [89] F. H. Cullen, J. J. Jarvis, and H. D. Ratliff, "Set partitioning based heuristics for interactive routing," *Networks*, vol. 11, no. 2, pp. 125–143, 1981.
- [90] L. D. Bodin and T. Sexton, "The multi-vehicle subscriber dial-a-ride problem," *TIMS studies in Management Science*, vol. 26, pp. 73–86, 1986.
- [91] J.-F. Cordeau, "A branch-and-cut algorithm for the dial-a-ride problem," *Operations Research*, vol. 54, no. 3, pp. 573–586, 2006.
- [92] J.-F. Cordeau and G. Laporte, "A tabu search heuristic for the static multi-vehicle dial-a-ride problem," *Transportation Research Part B : Methodological*, vol. 37, no. 6, pp. 579–594, 2003.
- [93] C. E. Cortés, M. Matamala, and C. Contardo, "The pickup and delivery problem with transfers : Formulation and a branch-and-cut solution method," *European Journal of Operational Research*, vol. 200, no. 3, pp. 711–724, 2010.
- [94] S. Mitrović-Minić and G. Laporte, "The pickup and delivery problem with time windows and transshipment," *INFOR : Information Systems and Operational Research*, vol. 44, no. 3, pp. 217–227, 2006.
- [95] J. Desrosiers, Y. Dumas, and F. Soumis, "A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows," *American Journal of Mathematical and Management Sciences*, vol. 6, no. 3-4, pp. 301–325, 1986.
- [96] Q. Lu and M. Dessouky, "An exact algorithm for the multiple vehicle pickup and delivery problem," *Transportation Science*, vol. 38, no. 4, pp. 503–514, 2004.

-
- [97] S. Ropke and J.-F. Cordeau, "Branch and cut and price for the pickup and delivery problem with time windows," *Transportation Science*, vol. 43, no. 3, pp. 267–286, 2009.
 - [98] M. Sigurd, D. Pisinger, and M. Sig, "Scheduling transportation of live animals to avoid the spread of diseases," *Transportation Science*, vol. 38, no. 2, pp. 197–209, 2004.
 - [99] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations research*, vol. 35, no. 2, pp. 254–265, 1987.
 - [100] . A. Li *et* H. Lim, "A metaheuristic for the pickup and delivery problem with time windows," vol. 65, pp. 23–35, 2003.
 - [101] J.-C. Créput, A. Koukam, J. Kozlak, and J. Lukasik, "An evolutionary approach to pickup and delivery problem with time windows," in *International Conference on Computational Science*. Springer, 2004, pp. 1102–1108.
 - [102] G. Pankratz, "A grouping genetic algorithm for the pickup and delivery problem with time windows," *Or Spectrum*, vol. 27, no. 1, pp. 21–41, 2005.
 - [103] S. Ropke and D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows," *Transportation science*, vol. 40, no. 4, pp. 455–472, 2006.
 - [104] Y. Nagata and S. Kobayashi, "Guided ejection search for the pickup and delivery problem with time windows," in *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer, 2010, pp. 202–213.
 - [105] M. Blocho and J. Nalepa, "A parallel algorithm for minimizing the fleet size in the pickup and delivery problem with time windows," in *Proceedings of the 22nd European MPI Users' Group Meeting*. ACM, 2015, p. 15.
 - [106] J. Nalepa and M. Blocho, "Enhanced guided ejection search for the pickup and delivery problem with time windows," in *Asian Conference on Intelligent Information and Database Systems*. Springer, 2016, pp. 388–398.
 - [107] J. Nalepa and M. Blocho, "A parallel memetic algorithm for the pickup and delivery problem with time windows," in *Parallel, Distributed and Network-based Processing (PDP), 2017 25th Euromicro International Conference on*. IEEE, 2017, pp. 1–8.
 - [108] H. Lim, A. Lim, and B. Rodrigues, "Solving the pickup and delivery problem with time windows using" squeaky wheel" optimization with local search." AIS, 2002.
 - [109] Y. Qu and J. F. Bard, "The heterogeneous pickup and delivery problem with configurable vehicle capacity," *Transportation Research Part C : Emerging Technologies*, vol. 32, pp. 1–20, 2013.

- [110] Y. Qu and J. F. Bard, "A branch-and-price-and-cut algorithm for heterogeneous pickup and delivery problems with configurable vehicle capacity," *Transportation Science*, vol. 49, no. 2, pp. 254–270, 2014.
- [111] A. Bettinelli, A. Ceselli, and G. Righini, "A branch-and-price algorithm for the multi-depot heterogeneous-fleet pickup and delivery problem with soft time windows," *Mathematical Programming Computation*, vol. 6, no. 2, pp. 171–197, 2014.
- [112] S. Irnich, "A multi-depot pickup and delivery problem with a single hub and heterogeneous vehicles," *European Journal of Operational Research*, vol. 122, no. 2, pp. 310–328, 2000.
- [113] L. C. Edie, "Traffic delays at toll booths," *Journal of the operations research society of America*, vol. 2, no. 2, pp. 107–138, 1954.
- [114] G. B. Dantzig, "Letter to the editor - a comment on edie's "traffic delays at toll booths",", *Journal of the Operations Research Society of America*, vol. 2, no. 3, pp. 339–341, 1954.
- [115] T. Aykin, "A comparative evaluation of modeling approaches to the labor shift scheduling problem," *European Journal of Operational Research*, vol. 125, no. 2, pp. 381–397, 2000.
- [116] S. E. Bechtold and L. W. Jacobs, "Implicit modeling of flexible break assignments in optimal shift scheduling," *Management Science*, vol. 36, no. 11, pp. 1339–1351, 1990.
- [117] G. M. Thompson, "Improved implicit optimal modeling of the labor shift scheduling problem," *Management Science*, vol. 41, no. 4, pp. 595–607, 1995.
- [118] O. Battaïa, X. Delorme, A. Dolgui, J. Hagemann, A. Horlemann, S. Kovalev, and S. Malyutin, "Workforce minimization for a mixed-model assembly line in the automotive industry," *International Journal of Production Economics*, vol. 170, pp. 489–500, 2015.
- [119] J. Gärtner, N. Musliu, and W. Slany, "A heuristic based system for generation of shifts with breaks," in *Applications and Innovations in Intelligent Systems XII*. Springer, 2005, pp. 95–106.
- [120] L. Di Gaspero, J. Gärtner, N. Musliu, A. Schaerf, W. Schafhauser, and W. Slany, "A hybrid ls-cp solver for the shifts and breaks design problem," in *International Workshop on Hybrid Metaheuristics*. Springer, 2010, pp. 46–61.
- [121] N. Musliu, A. Schaerf, and W. Slany, "Local search for shift design," *European journal of operational research*, vol. 153, no. 1, pp. 51–64, 2004.
- [122] K. R. Baker, "Workforce allocation in cyclical scheduling problems : A survey," *Journal of the Operational Research Society*, vol. 27, no. 1, pp. 155–167, 1976.

-
- [123] H. E. Miller, W. P. Pierskalla, and G. J. Rath, "Nurse scheduling using mathematical programming," *Operations Research*, vol. 24, no. 5, pp. 857–870, 1976.
- [124] R. T. Golembiewski and C. W. Proehl Jr, "A survey of the empirical literature on flexible workhours : Character and consequences of a major innovation," *Academy of Management Review*, vol. 3, no. 4, pp. 837–853, 1978.
- [125] B. Cheang, H. Li, A. Lim, and B. Rodrigues, "Nurse rostering problems—a bibliographic survey," *European Journal of Operational Research*, vol. 151, no. 3, pp. 447–460, 2003.
- [126] H. K. Alfares, "Survey, categorization, and comparison of recent tour scheduling literature," *Annals of Operations Research*, vol. 127, no. 1-4, pp. 145–175, 2004.
- [127] A. T. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier, "Staff scheduling and rostering : A review of applications, methods and models," *European journal of operational research*, vol. 153, no. 1, pp. 3–27, 2004.
- [128] J. Brandao and A. Mercer, "A tabu search algorithm for the multi-trip vehicle routing and scheduling problem," *European journal of operational research*, vol. 100, no. 1, pp. 180–191, 1997.
- [129] D. T. Eliiyi, A. Ornek, and S. S. Karakütük, "A vehicle scheduling problem with fixed trips and time limitations," *International Journal of Production Economics*, vol. 117, no. 1, pp. 150–161, 2009.
- [130] S. Bertels and T. Fahle, "A hybrid setup for a hybrid scenario : combining heuristics for the home health care problem," *Computers & Operations Research*, vol. 33, no. 10, pp. 2866–2890, 2006.
- [131] S. V. Begur, D. M. Miller, and J. R. Weaver, "An integrated spatial dss for scheduling and routing home-health-care nurses," *Interfaces*, vol. 27, no. 4, pp. 35–48, 1997.
- [132] E. Cheng and J. L. Rich, "A home health care routing and scheduling problem," Tech. Rep., 1998.
- [133] C. Akjiratikarl, P. Yenradee, and P. R. Drake, "Pso-based algorithm for home care worker scheduling in the uk," *Computers & Industrial Engineering*, vol. 53, no. 4, pp. 559–583, 2007.
- [134] P. Eveborn, M. Rönnqvist, H. Einarsdóttir, M. Eklund, K. Lidén, and M. Almroth, "Operations research improves quality and efficiency in home care," *Interfaces*, vol. 39, no. 1, pp. 18–34, 2009.
- [135] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance, "Branch-and-price : Column generation for solving huge integer programs," *Operations research*, vol. 46, no. 3, pp. 316–329, 1998.

- [136] D. Bredstrom and M. Rönnqvist, “A branch and price algorithm for the combined vehicle routing and scheduling problem with synchronization constraints,” 2007.
- [137] Y. Kergosien, C. Lenté, and J.-C. Billaut, “Home health care problem : An extended multiple traveling salesman problem,” in *Proceedings of the 4th Multidisciplinary International Scheduling Conference : Theory and Applications (MISTA 2009)*, 2009, pp. 85–92.
- [138] L.-H. Shih and H.-C. Chang, “A routing and scheduling system for infectious waste collection,” *Environmental Modeling & Assessment*, vol. 6, no. 4, pp. 261–269, 2001.
- [139] J. M. Swaminathan, “Decision support for allocating scarce drugs,” *Interfaces*, vol. 33, no. 2, pp. 1–11, 2003.
- [140] A. L. Medaglia, J. G. Villegas, and D. M. Rodríguez-Coca, “Hybrid biobjective evolutionary algorithms for the design of a hospital waste management network,” *Journal of Heuristics*, vol. 15, no. 2, p. 153, 2009.
- [141] G. Bailey, T. Cherrett, B. Waterson, and R. Long, “Can locker box logistics enable more human-centric medical supply chains ?” *International Journal of Logistics Research and Applications*, vol. 16, no. 6, pp. 447–460, 2013.
- [142] Y. Kergosien, C. Lenté, J.-C. Billaut, and S. Perrin, “Metaheuristic algorithms for solving two interconnected vehicle routing problems in a hospital complex,” *Computers & Operations Research*, vol. 40, no. 10, pp. 2508–2518, 2013.
- [143] S. N. Parragh, K. F. Doerner, and R. F. Hartl, “A survey on pickup and delivery problems,” *Journal für Betriebswirtschaft*, vol. 58, no. 1, pp. 21–51, 2008.
- [144] . L. Grandinetti *et al.*, “The multi-objective multi-vehicle pickup et delivery problem with time windows,” vol. 111, pp. 203–212, 2014.
- [145] . M. Desrochers *et* G. Laporte, “Improvements and extensions to the miller-tucker-zemlin subtour elimination constraints,” 1991.
- [146] . J.-F. Cordeau, “A branch-and-cut algorithm for the dial-a-ride problem,” 2006.
- [147] . Y. Dumas *et* M. Desrochers, “The pickup and delivery problem with time windows,” 1991.
- [148] . G. Cornuéjols, “Valid inequalities for mixed integer linear programs,” pp. 3–44, 2006.
- [149] E.-G. Talbi, “A taxonomy of hybrid metaheuristics,” *Journal of heuristics*, vol. 8, no. 5, pp. 541–564, 2002.

-
- [150] A. Nakib, “Conception de métaheuristiques d’optimisation pour la segmentation d’images. application aux images biomédicales,” Ph.D. dissertation, Université Paris XII Val de Marne, 2007.
 - [151] M. Dorigo and L. M. Gambardella, “Ant colony system : a cooperative learning approach to the traveling salesman problem,” *IEEE Transactions on evolutionary computation*, vol. 1, no. 1, pp. 53–66, 1997.
 - [152] S. Goss, S. Aron, J.-L. Deneubourg, and J. M. Pasteels, “Self-organized shortcuts in the argentine ant,” *Naturwissenschaften*, vol. 76, no. 12, pp. 579–581, 1989.
 - [153] F. Belmecheri, C. Prins, F. Yalaoui, and L. Amodeo, “An ant colony optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows,” *IFAC Proceedings Volumes*, vol. 42, no. 4, pp. 1550–1555, 2009.
 - [154] M. N. Tchoupo, A. Yalaoui, L. Amodeo, F. Yalaoui, and F. Lutz, “Ant colony optimization algorithm for pickup and delivery problem with time windows,” in *International Conference on Optimization and Decision Science*. Springer, 2017, pp. 181–191.
 - [155] J. F. Benders, “Partitioning procedures for solving mixed-variables programming problems,” *Numerische Mathematik*, vol. 4, no. 1, pp. 238–252, Dec 1962.
 - [156] G. B. Dantzig and P. Wolfe, “Decomposition principle for linear programs,” *Operations research*, vol. 8, no. 1, pp. 101–111, 1960.
 - [157] F.-H. Liu and S.-Y. Shen, “The fleet size and mix vehicle routing problem with time windows,” *Journal of the Operational Research society*, vol. 50, no. 7, pp. 721–732, 1999.
 - [158] . B. Dantzig et P. Wolfe, “Decomposition principle for linear programs,” 1960.
 - [159] J. Edmonds and R. Giles, “A min-max relation for submodular functions on graphs,” in *Annals of Discrete Mathematics*. Elsevier, 1977, vol. 1, pp. 185–204.
 - [160] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, “Introduction à l’algorithmique,” 1994.
 - [161] M. M. Solomon, “Vrptw benchmark problems,” <http://web.cba.neu.edu/~msolomon/problems.htm>, 2003.
 - [162] A. M. Campbell and M. Savelsbergh, “Efficient insertion heuristics for vehicle routing and scheduling problems,” *Transportation science*, vol. 38, no. 3, pp. 369–378, 2004.
 - [163] J. L. Rosa, A. Robin, M. Silva, C. A. Baldan, and M. P. Peres, “Electrodeposition of copper on titanium wires : Taguchi experimental design approach,” *Journal of Materials Processing Technology*, vol. 209, no. 3, pp. 1181–1188, 2009.

Moïse Aimé NOUMBISSI TCHOUP

Doctorat : Optimisation et Sûreté des Systèmes

Année 2019

Optimisation de la logistique hospitalière : problème conjoint de tournées de véhicules et de planification des chauffeurs

Nos travaux de recherche ont été motivés par une problématique réelle d'optimisation du transport logistique du Centre Hospitalier de Troyes (CHT). Le CHT s'inscrit actuellement dans la revue et l'amélioration des processus et l'implémentation des techniques de recherche opérationnelle afin d'apporter des solutions aux problèmes conjoints de transports de produits et planification des chauffeurs.

L'objectif de cette étude est de proposer des solutions efficaces afin d'assurer au quotidien le transport des chariots de repas, de linge, de produits pharmaceutiques et de produits magasiniers. Après la formalisation du problème, nous avons proposé une démarche de résolution composée de trois phases : la première traite du problème connu de collectes et livraisons avec fenêtre de temps et flotte homogène de véhicules, la seconde considère une flotte hétérogène de véhicules, et la troisième phase intègre l'affectation et le dimensionnement de l'équipe des chauffeurs. Nous avons abouti au développement et à la mise en place d'un outil d'aide à la décision. Nous avons suite à nos travaux, développé et mis en place en partenariat avec la *start-up* OPTA-LP un logiciel au CHT. Tout au long de nos travaux, nous avons développé des méthodes de résolution exactes et approchées, et aussi élaborer des techniques pour diminuer le temps de résolution dans une approche de résolution exacte, ou encore pour améliorer une solution donnée dans le cas d'une méthode approchée.

Mots clés : recherche opérationnelle – problèmes de transport (programmation) – problèmes d'affectation (programmation) – optimisation combinatoire.

Optimization of Hospital Logistics: Combination of Vehicle Routing and Personal Scheduling Problems

Our research work was motivated by a real problem of optimizing logistics transport at the Troyes Hospital Centre (CHT). The CHT is currently involved in the review and improvement of processes and the implementation of operational research techniques in order to provide solutions to the joint problems of product transport and driver scheduling.

The objective of this study is to propose effective solutions to ensure the daily transport of food trolleys, linens, pharmaceuticals and warehouse products. After the problem was formalized, we proposed a three-phase resolution approach: the first deals with the known problem of collections and deliveries with a time window and a homogeneous fleet of vehicles, the second considers a heterogeneous fleet of vehicles, and the third phase integrates the assignment and sizing of the driver team. We have resulted in the development and implementation of a decision support tool. Following our work, we have developed and implemented in partnership with the start-up OPTA-LP, a software at the CHT. Throughout our work, we have developed accurate and approximate resolution methods, and also some techniques to reduce resolution time in an accurate resolution approach, or to improve a given solution in the case of an approximate method.

Keywords: operations research – transportation problems (programming) – assignment problems (programming) – combinatorial optimization.

Thèse réalisée en partenariat entre :

